Numerical Pricing of Bermudan Options with Shannon Wavelet Expansions.

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

STEFANUS CORNELIS MAREE

Delft, the Netherlands
August 2015

Copyright © 2015 by Stef Maree. All rights reserved.
MSc thesis APPLIED MATHEMATICS

“Numerical Pricing of Bermudan Options using Shannon Wavelet Expansions”

Stefanus Cornelis Maree

Delft University of Technology

Responsible Professor
Prof.dr.ir. C.W. Oosterlee

Other members of the thesis committee
Dr.ir. M.B. van Gijzen
Dr. R.J. Fokkink
Dr. L. Ortiz-Gracia

August, 2015

Delft
Acknowledgements

This thesis has been submitted for the degree Master of Science in Applied Mathematics at Delft University of Technology. The supervisor of this thesis was Kees Oosterlee, professor at the Numerical Analysis group of Delft Institute of Applied Mathematics. Research for this thesis took partly place under his supervision at Centrum Wiskunde & Informatica. The main part of the research for this thesis was conducted in the Financial Mathematics and Risk Control group at Centre de Recerca Matemàtica (CRM), Barcelona under supervision of Luis Ortiz-Gracia.

I would like to thank my supervisors Kees Oosterlee and Luis Ortiz-Gracia for their advise and support throughout this thesis. Same thanks go out to Gemma Coldeforms-Papiol for the useful conversations we had.
Abstract

This thesis is about pricing Bermudan options with the SWIFT method (Shannon Wavelets Inverse Fourier Technique). We reformulate the SWIFT pricing formula for European options to improve robustness, which allows us to heuristically select - and test the goodness - of all of the parameters \textit{a priori}.

Furthermore, we propose a simplified version of the SWIFT method, based on the Whittaker-Shannon sampling theory, which is an easy to implement method that possesses algebraic convergence in the pricing of European and Bermudan options.

The main contribution of this thesis is a new pricing method for Bermudan options by the SWIFT method, for exponential Lévy processes using the Fast Fourier Transform. We compare the results of the SWIFT method to those of the COS method.

\textbf{Keywords:} Option pricing, Bermudan options, exponential Lévy processes, wavelet series approximations, Shannon wavelets, Shannon-Whittaker sampling theory, Fourier transform inversion.
Contents

1 Introduction 4
   1.1 Structure of this thesis ........................................... 5

2 Wavelet Approximation Theory 6
   2.1 Fourier Transform ................................................. 6
      2.1.1 Hankel Matrix Multiplication ............................. 8
      2.1.2 Toeplitz Matrix Multiplication .......................... 8
   2.2 Mother Wavelets .................................................. 9
   2.3 Multi Resolution Analysis ....................................... 10
   2.4 Shannon Wavelets ................................................ 12
      2.4.1 Real wavelet scale \( m \) .................................. 15

3 European option pricing methods 17
   3.1 Black-Scholes option pricing framework .......................... 17
      3.1.1 Modeling Information ........................................ 19
      3.1.2 Risk neutral rate ............................................. 20
      3.1.3 No-Arbitrage .................................................. 20
      3.1.4 Replicating Portfolio ....................................... 21
   3.2 Connection with partial differential equations .................. 22
   3.3 Exponential \( \text{Lévy} \) Asset Dynamics .......................... 23
      3.3.1 Risk-Neutral European option valuation formula .......... 25
   3.4 COS method ....................................................... 25
      3.4.1 Density Coefficients ........................................ 26
      3.4.2 Plain Vanilla Payoff Coefficients ......................... 27
      3.4.3 Pricing multiple strikes ..................................... 28
   3.5 \( \text{WA}^{[a,b]} \) method .............................................. 28
      3.5.1 Density Coefficients ........................................ 29
      3.5.2 Plain Vanilla Payoff Coefficients ......................... 30
   3.6 SWIFT-method ..................................................... 31
      3.6.1 Density Coefficients ........................................ 31
      3.6.2 Payoff coefficients .......................................... 33
      3.6.3 Pricing multiple strikes ..................................... 34
   3.7 Numerical Results ................................................ 34

4 Analysis of the SWIFT Method 38
   4.1 Payoff Coefficients ................................................ 39
      4.1.1 Cash-or-Nothing option ...................................... 39
      4.1.2 European put .................................................. 40
      4.1.3 European call ................................................ 40
      4.1.4 Unbounded Payoff ............................................. 41
   4.2 Local error analysis ............................................... 41
      4.2.1 Wavelet scale \( m \) ........................................... 42
      4.2.2 Truncation of \( k \) ............................................. 43
   4.3 Spatial Boundary Error ............................................. 44
Chapter 1

Introduction

The focus of this research is on the robust pricing of Bermudan options. An option is a type of derivative. In finance, a derivative is a contract that derives its value from the performance of an underlying entity. This underlying entity can be an asset, index or interest rate and is referred to as the underlying. We model the price process of the underlying by a stochastic process $S_t$.

Commonly traded derivatives are forwards, futures, swaps and options. In finance, an option is a contract which gives the buyer the right, but not the obligation, to buy or sell the underlying at a specified strike price on or before a specified date, depending on the form of the option.

A common option is a (European) call option, which gives the buyer the right, but not the obligation, to purchase the underlying stock at a certain expiration time $T$ for a certain strike price $K$. A put option, on the other hand, gives the buyer the right to sell the underlying stock for a prefixed strike price $K$.

Hence, the payoff value of a call and a put option at expiration are respectively given by,

$$\text{Call}(S_T, T) = \max(S_T - K, 0) \quad \text{and} \quad \text{Put}(S_T, T) = \max(K - S_T, 0),$$

where $S_T$ is the price of the underlying at time $T$. Because exercising the option is not required, their value is always non-negative.

A Bermudan option, in contrast to a European option, can be exercised at a predetermined set of moments prior to expiry. Whereas the payoff of a European option depends only on the value of the underlying at time $T$. Bermudan options are path-dependent; their value depends on the path of the underlying. Since Bermudan options can be exercised at different moments prior to maturity, the buyer of the option needs an exercise strategy to determine when to exercise the option.

Both the path-dependency and the exercise strategy make the pricing of Bermudan options more complicated than the pricing of European options.

In 1973, Black and Scholes [BS73] introduced the famous Black-Scholes model for European derivatives pricing. They derived a pricing PDE that defines the evolution of the option price over time. We are interested in the solution of this PDE, i.e., the option value at time $t \leq T$. This solution can also be found by means of the Feynman-Kac formula [OGF14] as the discounted expected payoff of the option.

A new direction in efficient numerical pricing based on the Fast Fourier Transform (FFT) was popularized by Carr and Madan [CM98]. Inspired by the Carr-Madan method, extensive research on pricing in the Fourier domain has emerged.

Fourier methods depend on the availability of an expression for the characteristic function of the underlying. When this characteristic function is available, Fourier methods are generally the fastest methods among finite difference and Monte Carlo methods [vSea15].

A state-of-the-art general purpose Fourier pricing method is the cosine series expansion method (COS) of [FO08]. Except for extreme cases such as deeply out-of-the-money options, the COS method is a fast and elegant method. The COS method has been extended to path-dependent options in [FO09], and higher dimensions in [RO12].

The WA$_{[0,1]}$ method [OGO13] was the first method to use wavelet expansions in the field of option pricing, where Haar wavelet and B-spline approximations were used to recover the probability density function of the underlying. A different approach using the same basis can be found in [Kir15], where the Fast Fourier Transform has been applied to increase efficiency.
In the SWIFT method [OGO15], the Shannon wavelet basis is used to recover the underlying density function in the pricing of European options. Shannon wavelets are regular smooth wavelets, which approximate the density functions occurring in finance well, resulting in an exponentially converging method.

In this thesis, we extend the applications of the SWIFT method. We do an extensive error analysis for European options, especially focusing on domain truncation errors. We propose a novel approach to the simultaneous pricing of options with different strike values and generalize the SWIFT method to the pricing of Bermudan options when the underlying is driven by an exponential Lévy process. The structure of this thesis is laid out in the next section.

1.1 Structure of this thesis

We start out in Chapter 2 with the preliminary theory of Wavelet approximations, which we will use later to approximate the probability density function of the underlying. We introduce wavelets from a point of Fourier analysis and present mother wavelet series expansions as an alternative to these Fourier series. Then, using the Multi Resolution Analysis framework, we move on to scaling functions or father wavelets, which we will use throughout the rest of the thesis. In Section 2.4 we look in detail to Shannon wavelets and discuss the important Whittaker-Shannon interpolation polynomial.

In Chapter 3, we discuss three European option pricing methods, the COS method [FO08], the WA \[n,a,b] method [OGO13] and the SWIFT method [OGO15]. We show that these methods follow the same pricing approach and demonstrate the differences.

We further analyze the SWIFT method in Chapter 4. We reformulate the pricing formula, we derive heuristics to choose the values for all of the parameters and we present tests to a priori test the goodness of these choices. We benefit from the FFT algorithm to approximate extra coefficients to reduce wiggles occurring due a series truncation. We also present a novel approach to multiple strikes pricing when the strikes are located on a grid and finally, we present the SWIFT-Whittaker method, which is a simplified version of the SWIFT method, with slower convergence, but which is computationally cheaper than the SWIFT method and it can be widely applied. The SWIFT-Whittaker method has algebraic convergence, and is thus relatively slow compared to the COS method when high accuracy is required.

Finally, the main result of this thesis is presented in Chapter 5, the novel SWIFT method for Bermudan options. We present two pricing approaches for Bermudan options, the SWIFT and SWIFT-Whittaker methods. We show their computational efficiency and robustness with respect to the boundary. We price American options using Richardson extrapolation on Bermudan option prices and price discretely monitored barrier options.
Chapter 2

Wavelet Approximation Theory

In this chapter, we discuss the necessary theory of wavelets required in the later chapters. This section is based on the first of the “Ten lectures on wavelets” by I. Daubechies [Dau92], “A Wavelet Tour of Signal Processing” by S. Mallat [Mal09] and “An Introduction to Wavelets” by C.K. Chui [Chu92]. We aim to give an easy to read introduction of how wavelets can be used to approximate functions, as a competitor to the Fourier expansions. This will be the basis of the methods we discuss in the rest of this thesis, where we approximate the density function of the underlying method using wavelet expansions.

The Fourier cosine expansion of a function $f \in L^2[a, b]$ is given by,

$$f(x) = \sum_{k=0}^{\infty} D_k \cos \left( k\pi \frac{x - a}{b - a} \right), \quad x \in [a, b],$$

where the Fourier coefficients $D_k$ are defined as,

$$D_k = \frac{2}{b - a} \int_a^b f(x) \cos \left( k\pi \frac{x - a}{b - a} \right) \, dx.$$

Fourier series periodically replicate functions with period $[a, b]$. However, the density functions we work with in finance are generally defined on the whole real line, and thus some artificial truncation needs to be performed. This may lead to undesirable behavior around the boundaries.

Another property of Fourier series is that they form what we call a ‘global basis’. Each of the coefficients $D_k$ captures the behavior of the function on the whole domain $[a, b]$. When, for example, the function $f$ is highly peaked, this influences all the coefficients, as well as the rate of convergence of the series.

Wavelet series expansions seem a solution to the above issues, since they form a local basis to the function space $L^2(\mathbb{R})$, where each of the coefficients captures only local behavior of the function.

2.1 Fourier Transform

The following function spaces are of importance throughout the MSc thesis.

Definition 2.1. A function $f : \mathbb{R} \mapsto \mathbb{C}$ is said to be an (Lebesgue-)integrable function, if,

$$\int_{\mathbb{R}} |f(x)| \, dx < \infty,$$

and we define the space of all integrable functions as $L^1(\mathbb{R})$, equipped with the norm,

$$\|f\|_1 := \int_{\mathbb{R}} |f(x)| \, dx.$$

Definition 2.2. A function $f$ is in $L^2(\mathbb{R})$ if,

$$\int_{\mathbb{R}} |f(x)|^2 \, dx < \infty,$$
where the space of functions in $L^2(\mathbb{R})$ is equipped with a norm and inner product,

$$
\|f\|_2 := \left( \int_{\mathbb{R}} |f(x)|^2 \, dx \right)^{\frac{1}{2}}, \quad \text{and} \quad \langle f, g \rangle := \int_{\mathbb{R}} f(x) \overline{g(x)} \, dx,
$$

for $f, g \in L^2(\mathbb{R})$, and $\overline{g}$ denotes the complex conjugate of $g$.

**Definition 2.3** (Fourier Transform). Given a function $f$, its **Fourier transform**, $\hat{f}$, is defined as,

$$
(\mathcal{F}f)(\omega) := \hat{f}(\omega) := \int_{\mathbb{R}} f(x) e^{-i\omega x} \, dx, \quad \omega \in \mathbb{C},
$$

(2.1)

where $i$ denotes the complex unit. Inversion of the Fourier transform is given by,

$$
f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} (\mathcal{F}f)(\omega) e^{i\omega x} \, d\omega.
$$

For functions $f$ in $L^1(\mathbb{R})$, we use that for real $x$ the equality $|f(x) e^{-i\omega x}| = |f(x)|$ holds, implying the existence of the right-hand side integral in (2.1). For $f \in L^2(\mathbb{R})$ we should define the Fourier transform $(\mathcal{F}f)(\omega)$ via a limiting process. We conveniently abuse notation and use (2.1), even when a limiting process is understood.

A special function of interest is the Dirac $\delta$-function, $\delta(f) = f(0)$ for a function $f$. The Dirac $\delta$-function should not to be confused with the Kronecker delta, $\delta_{n,m}$, defined for $n, m \in \mathbb{N}$, where $\delta_{n,m} = 1$ only if $n = m$ and zero otherwise.

The Fourier transform of the constant function $1$ is the Dirac $\delta$-function, $(\mathcal{F}1)(\omega) = 2\pi \delta(\omega)$. Using the Fourier transform of $1$, it is easy to show that for two functions $f, g \in L^2(\mathbb{R})$, the inner product satisfies,

$$
\langle f, g \rangle = \frac{1}{2\pi} \langle \hat{f}, \hat{g} \rangle,
$$

which implies that $\|f\|_2 = \frac{1}{\sqrt{2\pi}} \|\hat{f}\|_2$. This is Parseval’s Identity.

**Definition 2.4.** Let $f$ and $g$ be two functions defined on a subset of $\mathbb{R}$. We write $f(x) = O(g(x))$ as $x \to \infty$ if and only if there is a positive constant $M$ and real number $x_0$ such that,

$$
|f(x)| \leq M |g(x)|, \quad \text{for all} \ x \geq x_0,
$$

and we say that $f$ is of order $g$.

Let $X$ be a uni-variate continuous random variable with probability density function $f_X$. Then, it follows from the properties of density functions that, $f_X$ is a function in $L^1(\mathbb{R})$, the space of integrable functions.

**Definition 2.5.** Let $X$ be a uni-variate continuous random variable with probability density function $f_X$. Then, the characteristic function $\hat{f}_X$ is defined as,

$$
\hat{f}_X(\omega) := f_X(-\omega) = E \left[ e^{i\omega X} \right] = \int_{\mathbb{R}} f_X(x) e^{i\omega x} \, dx.
$$

(2.2)

**Remark 2.1.** Different definitions of the Fourier transform are used in literature. In probability theory, the Fourier transform is often defined equivalent to the characteristic function.

We however follow the definition common in wavelet theory. When $f$ is a probability density function, then we use ‘check-f’, i.e., $f(\omega)$ to denote the characteristic function, which is equal to the Fourier transform of the probability density function in $-\omega$, i.e., $\hat{f}(\omega) = \hat{f}(-\omega)$.

One of the major advantages of the Fourier methods we consider is that we can speed up the computational time from $O(N^2)$ to $O(N \log_2 N)$ using the so-called Fast Fourier Transform (FFT) algorithm, which we will gratefully use in the upcoming chapters.

The Discrete Fourier Transform (DFT) is defined analogously to the continuous Fourier transform, but only on a set of discrete frequencies $\omega$. 

7
Definition 2.6. Let \( z \in \mathbb{C}^N \) with entries \( z = \{ z_j \}_{j=1}^N \). Then the Discrete Fourier Transform is defined as

\[
D_k(z) = \sum_{j=1}^{N} z_j e^{-\frac{2\pi i}{N}(k-1)(j-1)}, \quad \text{where} \quad k = 1 - N/2, \ldots, N/2.
\]

(2.3)

Remark 2.2. There are different definitions regarding the DFT, but we use the one as implemented in Matlab to reduce the differences between notation and implementation. This is equivalent to \( Z = \text{fftshift} \left( \text{fft} \left( z \right) \right) \) in Matlab-code.

We discuss two types of matrices that have a special structure, which allows us to compute matrix-vector products with a complexity of \( \mathcal{O}(N \log_2 N) \) instead of \( \mathcal{O}(N^2) \). This is in detail described in [FO09].

### 2.1.1 Hankel Matrix Multiplication

A **Hankel matrix** \( \mathcal{M} \) is an \( N \times N \) matrix with constant anti-diagonals, i.e.,

\[
\mathcal{M} := 
\begin{bmatrix}
  m_0 & m_1 & m_2 & \cdots & m_{N-1} \\
  m_1 & m_2 & \cdots & m_{N-1} & m_N \\
  \vdots & \ddots & \ddots & \ddots & \vdots \\
  m_{N-2} & m_{N-1} & \cdots & \cdots & m_{2N-3} \\
  m_{N-1} & \cdots & \cdots & m_{2N-3} & m_{2N-3}
\end{bmatrix}
\]

For a vector \( x \in \mathbb{R}^N \), the matrix-vector product \( \mathcal{M} x \) is equal to the first \( N \) elements of the circular convolution \( m_h \odot x_h \), with the \( 2N \)-vectors,

\[
m_h := [m_0, m_{-1}, m_{-2}, \cdots, m_{1-N}, 0, m_{N-1}, m_{N-2}, \cdots, m_1]^T, \\
x_h := [x_0, x_1, x_2, \cdots, x_N, 0, \cdots, 0]^T.
\]

(2.4)

A circular convolution of two vectors is equal to the inverse discrete Fourier transform \( (\mathcal{D}^{-1}) \) of the product of the forward DFTs, \( \mathcal{D} \), i.e.,

\[
x \odot y = (\mathcal{D}(x) \cdot \mathcal{D}(y)).
\]

Thus, in total 3 times the FFT algorithm has to be applied on a vector of length \( 2N \).

### 2.1.2 Toeplitz Matrix Multiplication

Similar to the previous section, we describe how to efficiently compute the matrix-vector multiplication \( \mathcal{M} x \), when \( \mathcal{M} \) is an \( N \times N \) **Toeplitz matrix**, i.e.,

\[
\mathcal{M} := 
\begin{bmatrix}
  m_0 & m_1 & m_2 & \cdots & m_{N-2} & m_{N-1} \\
  m_{-1} & m_0 & m_1 & \cdots & m_{N-2} & \vdots \\
  \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
  m_{2-N} & \cdots & m_{-1} & m_0 & m_1 \\
  m_{1-N} & m_{2-N} & \cdots & m_{-1} & m_0
\end{bmatrix}
\]

For a vector \( x \in \mathbb{R}^N \), the matrix-vector product \( \mathcal{M} x \) is equal to the first \( N \) elements, in reversed order, of the circular convolution \( m_t \odot x_t \) of the \( 2N \)-vectors \( m_t \) and \( x_t \),

\[
m_t := [m_{2N-1}, m_{2N-2}, \cdots, m_1, m_0]^T, \\
x_t := [0, 0, \cdots, 0, x_0, x_1, x_2, \cdots, x_N]^T.
\]

(2.5)

Thus, in total 3 times the FFT algorithm has to be applied on a vector of length \( 2N \).
2.2 Mother Wavelets

In this section, we define what we mean when we use the term ‘wavelet’, and we start off by motivating the use of so-called mother wavelets. Wavelets are basically a set of functions which form a basis of $L^2(\mathbb{R})$, with some special properties. We require this basis to be a Riesz basis, which is a relaxation of the well known orthogonal basis.

**Definition 2.7.** A countable set $\{f_n\}$ of a Hilbert space is a **Riesz basis** if every $f$ in the space can be written uniquely as $f = \sum_n c_n f_n$, and there exists positive constants $A$ and $B$ such that,

$$A\|f\|^2 \leq \sum_n |c_n|^2 \leq B\|f\|^2.$$  

Let $\psi$ be a function in $L^2(\mathbb{R})$ and let the family generated by $\psi$ be the set $\{\psi_{m,k}\}_{m,k \in \mathbb{Z}}$ where,

$$\psi_{m,k}(x) := 2^{m/2}\psi(2^m x - k), \quad \text{for } m, k \in \mathbb{Z}. \quad (2.6)$$

We refer to $k$ as the translation parameter, as it translates $\psi$ along the $x$-axis, and $m$ as the scale parameter, as it compresses or stretches instances of $\psi$. The factor $2^{m/2}$ ensures that $\|\psi_{m,k}\|_2 = \|\psi\|_2$.

If $\psi$ generates the family $\{\psi_{m,k}\}_{m,k \in \mathbb{Z}}$, and this family forms a Riesz basis of $L^2(\mathbb{R})$, then the generating function $\psi$ is called an **$R$-function**.

**Theorem 2.1.** Let $\psi \in L^2(\mathbb{R})$ be an $R$-function, and let its generated family be $\{\psi_{m,k}\}_{m,k \in \mathbb{Z}}$, defined as $\psi_{m,k}(x) := 2^{m/2}\psi(2^m x - k)(x)$. Then there exists a unique Riesz basis $\{\psi_{n,l}\}_{n,l \in \mathbb{Z}}$ which is dual to the family $\{\psi_{m,k}\}_{m,k \in \mathbb{Z}}$ in the sense that $\langle \psi_{m,k}, \psi_{n,l} \rangle = \delta_{m,n}\delta_{k,l}$, for all $m, n, k, l \in \mathbb{Z}$.

**Proof.** The proof can be found in Chapter 1 of [Chn92].

By Theorem 2.1, for every $R$-function, there always exists some unique dual basis. This dual basis however, is certainly not always a family generated by a single function as in Equation (2.6). This motivates the definition of wavelet.

**Definition 2.8.** A function $\psi \in L^2(\mathbb{R})$ is a **mother wavelet** if there exists a function $\tilde{\psi} \in L^2(\mathbb{R})$, such that the families $\{\psi_{m,k}\}$ and $\{\tilde{\psi}_{m,k}\}$, defined as,

$$\psi_{m,k}(x) := 2^{m/2}\psi(2^m x - k), \quad \text{and} \quad \tilde{\psi}_{m,k}(x) := 2^{m/2}\tilde{\psi}(2^m x - k),$$

are dual bases of $L^2(\mathbb{R})$. If $\psi$ is a mother wavelet, then $\tilde{\psi}$ is called a dual mother wavelet corresponding to $\psi$, the functions $\psi_{m,k}$ are called **mother wavelet functions** and the wavelet family $\{\psi_{m,k}\}_{m,k \in \mathbb{Z}}$ forms a **mother wavelet basis** of $L^2(\mathbb{R})$.

We use the mother wavelet basis to formulate a series expansion for functions $f \in L^2(\mathbb{R})$.

**Theorem 2.2.** Let $f$ be a function in $L^2(\mathbb{R})$ and $\psi$ a mother wavelet with wavelet family $\{\psi_{m,k}\}_{m,k \in \mathbb{Z}}$, defined as in (2.6). Since the wavelet family forms a Riesz basis of $L^2(\mathbb{R})$ we can uniquely write the function $f$ as,

$$f(x) = \sum_{m \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} c_{m,k} \psi_{m,k}(x), \quad (2.7)$$

where the mother wavelets coefficients are given by $c_{m,k} := \langle f, \tilde{\psi}_{m,k} \rangle$.

**Proof.** Since $\psi$ is a mother wavelet, the wavelet family $\{\psi_{m,k}\}_{m,k \in \mathbb{Z}}$ forms a Riesz basis of $L^2(\mathbb{R})$, thus justifying the series expansion. To find the coefficients, we take the inner product of both sides of Equation (2.7) with the complex conjugate of a wavelet function from the dual basis, $\tilde{\psi}_{n,l}$ for some $n, l \in \mathbb{Z}$, this gives,

$$\langle f, \tilde{\psi}_{n,l} \rangle = \sum_{m,k \in \mathbb{Z}} c_{m,k} \psi_{m,k} \cdot \tilde{\psi}_{n,l} = \sum_{m,k \in \mathbb{Z}} c_{k,m} \langle \psi_{m,k}, \tilde{\psi}_{n,l} \rangle = c_{n,l}.$$
2. WAVELET APPROXIMATION THEORY

Note that the mother wavelet pair \((\psi, \tilde{\psi})\) is symmetric in the sense that the dual mother wavelet \(\tilde{\psi}\) is a mother wavelet itself, and thus \(\psi\) is also the dual of \(\tilde{\psi}\). We can therefore rewrite the mother wavelet expansion of a function \(f \in L^2(\mathbb{R})\) in Theorem 2.2 as,

\[
f(x) = \sum_{m,k \in \mathbb{Z}} (f, \psi_{m,k}) \tilde{\psi}_{m,k}(x) = \sum_{m,k \in \mathbb{Z}} (f, \tilde{\psi}_{m,k}) \psi_{m,k}(x).
\]

We highlight one special type of wavelets, which are the orthogonal wavelets, before we dig deeper into the wavelet series expansion.

**Definition 2.9.** A wavelet \(\psi\) is called an **orthogonal wavelet** if the family \(\{\psi_{m,k}\}_{m,k \in \mathbb{Z}}\) forms an orthonormal basis of \(L^2(\mathbb{R})\), that is, \((\psi_{m,k}, \psi_{n,l}) = \delta_{m,n} \delta_{k,l}\) for all \(m,n,k,l \in \mathbb{Z}\).

It follows directly from the definition of orthogonal wavelets that orthogonal mother wavelets are self-dual, i.e., \(\tilde{\psi} \equiv \psi\).

In the next section, we explain Multi Resolution Analysis, which is a useful framework we use to reformulate the series expansion in Theorem 2.2 from a double summation to a single one. We also use this framework to prove convergence of the wavelet series.

### 2.3 Multi Resolution Analysis

A multi resolution analysis (MRA) consists of a sequence of successive approximation spaces \(V_j\) in \(L^2(\mathbb{R})\), being closed subspaces that satisfy,

\[
\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots,
\]

with,

\[
\bigcup_{j \in \mathbb{Z}} V_j = L^2(\mathbb{R}), \quad \text{and} \quad \bigcap_{j \in \mathbb{Z}} V_j = \{0\}.
\]

There are many subspaces that satisfy the two properties above that have nothing to do with multi resolution. Multi resolution is a consequence of an additional requirement,

\[
f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1},
\]

or equivalently, \(f(x) \in V_0 \Leftrightarrow f(2^j x) \in V_j\), that is, all the spaces \(V_j\) are scaled versions of the central space \(V_0\). A well known function that satisfies these properties is the block function, as shown in the next example.

**Example 1** (Haar basis). *An example of spaces \(V_j\) satisfying Equations (2.8)-(2.10) is,*

\[
V_m = \{ f \in L^2(\mathbb{R}) : \forall k \in \mathbb{Z} : f|_{[2^{-m}k, 2^{-m}(k+1))] = \text{constant} \},
\]

which is the space of all functions that are piecewise constant on \([2^{-m}k, 2^{-m}(k+1))\), and the projections \(\mathcal{P}_m : L^2(\mathbb{R}) \mapsto V_m\) for \(m = 2, 3, 5\) are shown in Figure 2.1.

The previous example also demonstrates another feature that we require from an MRA: invariance of \(V_0\) under integer translations,

\[
f(x) \in V_0 \Rightarrow f(x - k) \in V_0, \quad \text{for all} \ k \in \mathbb{Z}.
\]

Because of Requirement (2.10), this implies a similar translation for the spaces \(V_j\), i.e., if \(f(x) \in V_j \Rightarrow f(x - 2^j k) \in V_j\) for all \(k \in \mathbb{Z}\). We are now ready to define MRA.

**Definition 2.10.** Let \(\phi \in L^2(\mathbb{R})\) be the generator of a family \(\{\phi_{m,k}\}_{m,k \in \mathbb{Z}}\) with \(\phi_{m,k}(x) := 2^{m/2} \phi(2^m x - k)\), and define the space \(V_m\) as,

\[
V_m := \text{closure}_{L^2(\mathbb{R})} \langle \{\phi_{m,k}\}_{k \in \mathbb{Z}} \rangle, \quad m \in \mathbb{Z}.
\]

If \(V_m\) satisfies the properties (2.8)-(2.11), and \(\{\phi_{0,k}\}\) forms a Riesz basis of \(V_0\), then we say that \(\phi\) generates an MRA, and \(\phi\) is called a **scaling function**, or **father wavelet**.
2.3. MULTI RESOLUTION ANALYSIS

A function $f$ and its projections

Figure 2.1: The approximation of a function $f$ onto the subspaces $V_2, V_3$ and $V_5$. Note that each increment in $m$ divides the intervals on which the projection is constant by two.

In words, Definition 2.10 states that an MRA is a special structure of nested spaces generated from a single function, called the scaling function. Let us define $P_m f$ as the orthogonal projection of a function $f \in L^2(\mathbb{R})$ on the space $V_m$, which is by construction given by,

$$P_m f(x) = \sum_{k \in \mathbb{Z}} \langle f, \phi_{m,k} \rangle \phi_{m,k}(x), \quad (2.13)$$

Then, the main result of MRA is summarized in the following theorem, and relates the father wavelet to the mother wavelet.

**Theorem 2.3.** Let $\phi$ be a scaling function (father wavelet). Then there exists a mother wavelet $\psi$ with a corresponding Riesz basis $\{\psi_{m,k}\}_{m,k \in \mathbb{Z}}$, where $\psi_{m,k}(x) := 2^{m/2}\psi(2^m x - k)$, such that for all $f \in L^2(\mathbb{R})$, the orthogonal projection onto $V_{m+1}$ can be decomposed as,

$$P_{m+1} f(x) = P_m f(x) + \sum_{k \in \mathbb{Z}} \langle f, \psi_{m,k} \rangle \psi_{m,k}(x). \quad (2.14)$$

The mother wavelet $\psi$ can, moreover, be constructed explicitly.

**Proof.** The reader is referred to [Mal09] for a proof. \(\square\)

Let us have a second look at (2.14). We can see each increment in $m$ as an additional layer of information added to the projection, and by repeating this procedure, we find,

$$P_m f(x) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} \langle f, \psi_{m,k} \rangle \psi_{j,k}(x).$$

This is a truncated wavelet series, and if $m$ were allowed to go to infinity, we would have a full wavelet expansion as in Theorem 2.2, i.e. $f(x) = \lim_{m \to +\infty} P_m f(x)$.

When $f \in L^1(\mathbb{R})$, a proof of convergence can also be given in the $L^1(\mathbb{R})$-norm.

**Lemma 2.1.** If $f \in L^1(\mathbb{R})$, then $P_m f \to f$ in the $L^1(\mathbb{R})$-norm as $m \to \infty$.

**Proof.** A proof is given in [Mal09]. \(\square\)

We now give some examples of common wavelets that have been used before in option pricing methods. In our application, we use wavelets to approximate functions by a series expansion. One of the simplest approaches to approximate a function is to use a piece-wise constant wavelet, which is described in the next example.

**Example 2** (Haar wavelets). Let $\chi_{[a,b]}(x)$ be the indicator function on the interval $[a,b]$. Then, the Haar scaling function is given by,

$$\phi_{Ha}(x) = \chi_{[0,1]}(x) = \begin{cases} 1, & \text{if } x \in [0,1) \\ 0, & \text{otherwise.} \end{cases} \quad (2.15)$$
2. WAVELET APPROXIMATION THEORY

The space \( V_m \), generated by the Haar scaling function, is the space of functions that are piece-wise constant on the intervals \([2^m j, 2^m (j + 1)]\).

The Haar mother wavelet is given by,

\[
\psi_{Ha}(x) = \begin{cases} 
1, & \text{if } x \in [0, \frac{1}{2}) \\
-1, & \text{if } x \in \left[\frac{1}{2}, 1\right) \\
0, & \text{otherwise} 
\end{cases}
\]  

(2.16)

We note that Haar wavelets form an orthogonal basis of \( L^2(\mathbb{R}) \), thus the Haar wavelet is self-dual. It is obvious that Haar wavelets have compact support, but they have slow decay in the frequency domain, as the Fourier transform of the scaling function is given by,

\[
\hat{\phi}_{Ha}(\omega) = \int_{\mathbb{R}} \chi_{[0,1)}(x) e^{-i\omega x} \, dx = \frac{1 - e^{-i\omega}}{i\omega}.
\]

We have seen the Haar basis in the previous example, which projects functions onto the space of piecewise constant functions. An obvious next step is to project functions onto a space with piecewise polynomial functions.

**Example 3 (B-Splines).** Typical examples of scaling functions \( \phi \) are the \( j \)th order cardinal B-splines, \( N_j(x) \), defined recursively by \( N_0(x) = \chi_{[0,1)}(x) \) and,

\[
N_j(x) = \frac{x}{j} N_{j-1}(x) + \frac{j+1-x}{j} N_{j-1}(x-1), \quad j \geq 1.
\]

Cardinal B-Splines are compactly supported with support \([0, j+1]\), shown in Figure 2.2, and their Fourier transform is given by,

\[
\hat{N}_j(\omega) = \left(1 - e^{-i\omega}\right)^{j+1},
\]

where we note that the higher the order \( j \), the faster the decay in frequency. Only the cardinal B-splines of order \( j = 0 \), the Haar wavelets, form an orthogonal basis of \( L^2(\mathbb{R}) \).

![Figure 2.2: Cardinal B-Splines of orders \( j = 0, 1, 2, 3 \). For \( j = 0 \), we have the scaling function of the Haar wavelet system.](image)

2.4 Shannon Wavelets

In the previous section we discussed the Haar wavelet, which lacks smoothness in time but has good localization in time, while the reverse is true in the frequency domain. We use these wavelets to approximate density functions, which are very regular functions. Therefore, it makes sense to choose a wavelet that is smooth in the time-domain.
Shannon wavelets are extensively described by C. Cattani in [Cat08]. The basic scaling function, or father wavelet, is the sinc function, 
\[ \phi_{Sh}(x) = \text{sinc}(x) := \begin{cases} \frac{\sin(\pi x)}{\pi x}, & \text{if } x \neq 0, \\ 1, & \text{if } x = 0 \end{cases} \] 
(2.17)
where the extension to \( x = 0 \) is derived from its limit. The Shannon mother wavelet is given by,
\[ \psi_{Sh}(x) = 2^{j/2} \sin(\pi(x - 1/2)) - \sin(2\pi(x - 1/2)) / \pi(x - 1/2). \]
Both the father and mother wavelet are plotted in Figure 2.3.

The Fourier transform of the Shannon father wavelet is given by \( \hat{\phi}(\omega) = \text{rect}(\omega/\pi) \), where \( \text{rect}() \) is the rectangle function, defined as,
\[ \text{rect}(x) = \begin{cases} 1, & \text{if } |x| < 1/2, \\ 1/2, & \text{if } |x| = 1/2, \\ 0, & \text{if } |x| > 1/2. \end{cases} \]
The Fourier transform of the wavelet instances \( \phi_{m,k} \) is given by,
\[ \hat{\phi}_{m,k}(\omega) = 2^{-m} \text{rect}(\omega/2^{m+1}\pi) e^{-i\omega \frac{m}{2^m}} = 2^{-\frac{m}{2}} \hat{\phi}(\frac{\omega}{2^m}) e^{-i\omega \frac{m}{2^m}}. \]

Shannon wavelets form the basis of the Shannon-Whittaker sampling theorem. This theorem is based around the notion of band-limited functions. We discuss the relevant properties of the sampling theorem.

**Definition 2.11.** A function \( f \) is called **band-limited** if there exists a positive constant \( B < \infty \), such that,
\[ \hat{f}(\omega) = \int_{-B\pi}^{B\pi} f(x) e^{-i\omega x} dx, \]
i.e., its Fourier transform is identically zero on \( |\omega| > B\pi \). The parameter \( B \) is referred to as the bandwidth of \( f \).

Band-limited functions have an interesting relationship with Shannon wavelets. Let \( f \) be band-limited with band-limit \( B = 2^m \). Then,
\[ \hat{f}(\omega) = \hat{f}(\omega) \text{rect}(\frac{\omega}{2^{m+1}\pi}) = \hat{f}(\omega) \hat{\phi}(\frac{\omega}{2^m}). \]

Band-limited functions are “smooth”. Any deviation from “smooth” would result in high frequency components which in turn invalidate the required property of being band-limited. The smoothness of a function between two points prohibits arbitrary variation of the function in that interval. Therefore, band-limited functions can be exactly recovered by discrete sampling. This statement is made precise in the following theorem.
Theorem 2.4 (Whittaker-Shannon Interpolation polynomial). Let \( f \) be a band-limited function with bandwidth \( B \), then \( f \) can be recovered by,

\[
f(x) = \sum_{k \in \mathbb{Z}} f\left(\frac{k}{B}\right) \text{sinc}(Bx - k),
\]

where the ‘cardinal series’ converge uniformly if \( f \in L^2(\mathbb{R}) \) or \( f \in L^1(\mathbb{R}) \).

Proof. The key insight of the proof is to note that since \( f \) is band-limited, its Fourier transform \( \hat{f}(\omega) \) is identically zero for \( |\omega| > B\pi \), and thus we can periodically extend \( \hat{f}(\omega) \) to the real line with period \( 2B\pi \).

The Fourier series of this extended function then equals \( \hat{f}(\omega) \) on \( |\omega| < B\pi \), given by,

\[
\hat{f}(\omega) = \sum_{k \in \mathbb{Z}} c_k e^{-i\omega k/B} \text{rect}\left(\frac{\omega}{2B\pi}\right),
\]

where the Fourier coefficients \( c_k \) can be readily computed,

\[
c_k = \frac{1}{2B\pi} \int_{-B\pi}^{B\pi} \hat{f}(\omega) e^{i\omega k/B} d\omega = \frac{1}{B} f\left(\frac{k}{B}\right),
\]

and substituting these coefficients in (2.19) gives,

\[
\hat{f}(\omega) = \sum_{k \in \mathbb{Z}} \frac{1}{B} f\left(\frac{k}{B}\right) e^{-i\omega k/B} \text{rect}\left(\frac{\omega}{2B\pi}\right).
\]

Now, using the fact that the right hand side of this equality is the Fourier transform of the translated and stretched sinc function,

\[
\int_{\mathbb{R}} \text{sinc}(Bx - k)e^{-i\omega x} dx = \frac{1}{B} e^{-i\omega k/B} \text{rect}\left(\frac{\omega}{2B\pi}\right),
\]

results in,

\[
\hat{f}(\omega) = \sum_{k \in \mathbb{Z}} f\left(\frac{k}{B}\right) \int_{\mathbb{R}} \text{sinc}(Bx - k)e^{-i\omega x} dx,
\]

and the inverse Fourier transform of \( \hat{f} \) results directly into (2.18).

\[ \square \]

A result for a band-limited function \( f \) with bandwidth \( B \) is that its integral can be written directly in terms of its samples. By the application of the Fourier series expansion in (2.21) evaluated at \( \omega = 0 \), we find

\[
\int_{\mathbb{R}} f(x) dx = \hat{f}(0) = \sum_{k \in \mathbb{Z}} f\left(\frac{k}{B}\right).
\]

Thus, Trapezoidal Integration of a band-limited function contains no error due to the piecewise linear approximation of the function if more than \( 2B\pi \) discretization points are used.

Even though we can completely determine a band-limited function by discrete samples, we still require an infinite number of samples. Since band-limited functions are analytic, they cannot be identically zero over any finite subinterval (except for \( f \equiv 0 \)). However, if the function is in \( L^2(\mathbb{R}) \) or \( L^1(\mathbb{R}) \), it vanishes at \( x \to \pm \infty \), and therefore there is always an interval \([a,b]\) outside of which the samples are negligibly small.

We can relate the Whittaker-Shannon interpolation polynomial to the Multi Resolution Analysis framework, which we state in the following lemma.

Lemma 2.2. Following the notation of Multi Resolution Analysis, let \( \phi \) be the Shannon scaling function and let \( \phi_{m,k}(x) = 2^{m/2}\phi(2^m x - k) \) be the wavelets instances for \( k \in \mathbb{Z} \) and let the subspace \( V_m \subset L^2(\mathbb{R}) \) be the space generated by the wavelets of scale \( m \), i.e,

\[
V_m := \text{closure}\left\{ \{\phi_{m,k}\}_{k \in \mathbb{Z}} \right\}.
\]

Let \( f \in L^2(\mathbb{R}) \) be a band-limited function with bandwidth \( B < 2^m \), then \( f \in V_m \).
In the framework of MRA, the wavelet scale

### 2.4.1 Real wavelet scale \( m \)

Following the Whittaker-Shannon Interpolation polynomial, since \( B < 2^m \), we can recover \( f \) by

\[
f(x) = \sum_{k \in \mathbb{Z}} f \left( \frac{k}{2^m} \right) \text{sinc}(2^m x - k).
\]

(2.24)

Since \( f \) can be written as a linear combination of \( \phi_{m,k} \) for \( k \in \mathbb{Z} \), it follows directly that \( f \in V_m \). \( \square \)

From the above lemma, it follows that a band limited function \( f \) with bandwidth \( B < 2^m \), can be recovered exactly by its projection \( P_m f \) onto the space \( V_m \). It also allows us to characterize the space \( V_m \) generated by the Shannon MRA. Namely, \( V_m \) is the space of all functions with bandwidth \( 2^m \).

### 2.4. SHANNON WAVELETS

In the framework of MRA, the wavelet scale \( m \) is chosen to be in \( \mathbb{Z} \). Then, starting from a scaling function \( \phi \in L^2(\mathbb{R}) \), the wavelet instances of the Shannon family are defined as,

\[
\phi_{m,k}(x) := 2^{m/2} \phi(2^m x - k) = 2^{m/2} \text{sinc}(2^m x - k).
\]

However, looking at the Shannon-Whittaker interpolation formula in Theorem 2.4, we can approximate a bandlimited function by wavelet instances of the form \( \text{sinc}(Bx - k) \), where \( B \) is any positive real constant. This suggests that we can take a similar approach and recover the density using a series expansion in terms of \( \{\phi_{m,k}\}_{k \in \mathbb{Z}} \), for a fixed \( m \in \mathbb{R} \).

The idea to allow \( m \) to take values on the real line not fit easily into the framework of MRA. In [Dau92], MRA is extended for dilation factors \( a \), by defining wavelets as \( a^{m/2} \text{sinc}(a^m x - k) \). Then, the relationship between two spaces \( V_m \) and \( V_{m+1} \), (2.10), becomes \( f(x) \in V_m \Rightarrow f(ax) \in V_{m+1} \). However, the whole structure of nested spaces \( V_m \) is not trivially extended for \( m \in \mathbb{R} \). Ignoring the MRA framework for now, we can still prove convergence based on the result in the following lemma.

**Lemma 2.3.** Let \( f \) be a function in \( L^1(\mathbb{R}) \cap L^2(\mathbb{R}) \) with Fourier transform \( \hat{f} \). Then, define \( \hat{f}(\omega; m) := \hat{f}(\omega) \text{rect} \left( \frac{\omega}{2^m \cdot 2\pi} \right) \). Since \( \hat{f}(\omega; m) \to \hat{f}(\omega) \) in \( L^2(\mathbb{R}) \)-sense as \( m \to \infty \), we have \( f(x; m) \to f(x) \), where \( f(x; m) \) is the inverse Fourier transform of \( \hat{f}(\omega; m) \).

**Proof.** Convergence in the Fourier domain, \( \hat{f}(\omega; m) \to \hat{f}(\omega) \) in the \( L^2(\mathbb{R}) \)-sense, follows from the construction of \( \hat{f}(\omega; m) \). Then, since \( |\hat{f}(\omega; m)| \leq |\hat{f}(\omega)| \) we can apply the Dominated Convergence Theorem and find,

\[
f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\omega)e^{ix\omega} \, d\omega
= \frac{1}{2\pi} \int_{\mathbb{R}} \lim_{m \to \infty} \hat{f}(\omega; m)e^{ix\omega} \, d\omega
= \lim_{m \to \infty} \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\omega; m)e^{ix\omega} \, d\omega
= \lim_{m \to \infty} f(x; m).
\]

Now, using the notation from Lemma 2.3, the function \( f(x; m) \) is bandlimited, and by application of the Shannon-Whittaker interpolation polynomial we find,

\[
f(x; m) = \sum_{k \in \mathbb{Z}} 2^{-\frac{m}{2}} f \left( \frac{k}{2^m}; m \right) \phi_{m,k}(x),
\]

but \( f(x; m) \) itself is unknown. Therefore, we use the intermediate result in the proof of Theorem 2.4, stating that,

\[
2^{-\frac{m}{2}} f \left( \frac{k}{2^m}; m \right) = \langle f, \phi_{m,k} \rangle = D_{m,k}.
\]
Combining results we find our convergence result,

\[ f(x) = \lim_{m \to \infty} f(x; m) = \lim_{m \to \infty} \sum_{k \in \mathbb{Z}} D_{m,k} \phi_{m,k}(x). \]

Remark 2.3. Note that when we let \( m \in \mathbb{Z} \), the result in Lemma 2.3 overlaps with the theory of MRA, since,

\[ \mathcal{P}_m f(x) = \sum_{k \in \mathbb{Z}} D_{m,k} \phi_{m,k}(x) = \sum_{k \in \mathbb{Z}} 2^{-\frac{m}{2}} f \left( \frac{k}{2^m}; m \right) \phi_{m,k}(x) = f(x; m). \]

From now on, we extend notation and define \( \mathcal{P}_m f(x) := f(x; m) \) when \( m \in \mathbb{R} \setminus \mathbb{Z} \).
Chapter 3

European option pricing methods

A (European) call option is a contract that gives the buyer the right, but not the obligation, to buy the underlying asset for a fixed strike price $K$ at a fixed time of maturity $T$. A put option, on the other hand, gives the buyer the right to sell the underlying asset for a fixed price. Thus, at final time $T$, the payoff of these option is given by,

$$\text{Call}(S_T, T) = \max(S_T - K, 0) \quad \text{and} \quad \text{Put}(S_T, T) = \max(K - S_T, 0),$$

where $S_T$ is the price of the underlying asset at time $T$. Since the buyer of the call (put) option is never obliged to buy (sell) the underlying asset, the payoff is always non-negative.

The important question however, is not what the value of the option is at time of maturity $T$, but "What is the value of the option at any time prior to maturity?"

The first step in answering this question is to define a so-called market model, which makes assumptions on how the market and its assets behave.

One of the first, but definitely the most successful attempt in option pricing theory is the work by Black and Scholes [BS73], where they established the Black-Scholes option pricing formula, i.e., a closed form solution for the price of put and call options.

In the next section, we describe the pricing of options on a market driven by a single factor of uncertainty using the risk-neutral asset pricing theory, which was implicit in the classical papers of Black-Scholes [BS73] and Merton [Mer73].

In Section 3.3, we extend the results of the Black-Scholes framework to a wide class of asset models called Exponential Lévy processes and we present the risk-neutral option pricing formula.

In Sections 3.4, 3.5 and 3.6, we discuss how to solve risk-neutral option pricing formula with the COS, WA$^{a,b}$ and SWIFT methods respectively.

Finally, we compare the three methods in Section 3.7.

3.1 Black-Scholes option pricing framework

We talk the reader through the theorems and definitions required to derive the risk-neutral option pricing theory.

Details about measure theory and stochastic calculus are omitted whenever possible. For a rigorous derivation of the Black-Scholes option pricing framework and stochastic calculus for finance, we refer the reader to the book 'Stochastic calculus for finance II: Continuous time models', by S. Schreve [Shr04], on which the majority of this section is based. All proofs and details left out in this section can be found in the book.

To determine the value of an option$^1$, it is important to know how the underlying asset behaves. Let $S_t$ denote the price of the underlying asset as time $t$. Then, we define the return of the asset as,

$$\text{Return} = \frac{\text{Stock tomorrow - Stock today}}{\text{Stock today}} = \frac{S_{t+\delta t} - S_t}{S_t}.$$

$^1$For illustration, we use the term option here, but generally, this approach applies to any security derivative. 
Consider the stock Google, where the daily closing prices of the stock and the daily returns are shown in Figure 3.1. We note that the asset price looks very unpredictable, but the daily returns look like noise, of which we shown the histogram and the Empirical Distribution function in the bottom two figures.

Since the returns look like noise, we model them as noise. In probability theory, a real-world process consisting of states that occur randomly is modeled by a probability space. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be such a probability space, consisting of,

- A sample space, $\Omega$, which is the set of all possible outcomes $\omega$.
- A set of events $\mathcal{F}$, where each event is a set containing zero or more outcomes.
- The assignment of probabilities to the events; that is, a function $\mathbb{P}$ from events to probabilities.

On the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, we model the underlying asset by a stochastic process.

**Definition 3.1 (Stochastic Process).** A stochastic process $X_t$ is a collection of random variables,

$$\{X_t\}_{t \in [0,T]} = \{X_t(\omega, t) : t \in [0, T], \omega \in \Omega\}.$$

We fix a maturity time $T > 0$ and let the stochastic process $\{S_t\}_{t \in [0, T]}$ represent the price process of the underlying asset. This asset price consists of two parts. First, there is a long-term growth (or decay) known as the drift. On top of this long-term drift, there is noise. This noise is modeled using Brownian motion.

**Definition 3.2 (Brownian Motion).** Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. For each $\omega \in \Omega$, suppose there is a continuous function $W_t$ of $t \geq 0$ that satisfies $W_0 = 0$ and that depends on $\omega$. Then, $W_t$, $t \geq 0$, is a Brownian motion if for all $0 = t_0 < t_1 < \cdots < t_M$ the increments,

$$W_{t_1} - W_{t_0}, \quad W_{t_2} - W_{t_1}, \quad \ldots, \quad W_{t_M} - W_{t_{M-1}},$$

are independent and each of these increments is normally distributed with,

$$\mathbb{E}[W_{t_{i+1}} - W_{t_i}] = 0,$$
$$\text{Var}(W_{t_{i+1}} - W_{t_i}) = t_{i+1} - t_i. \quad (3.2)$$

One way of modeling the stock is by Geometric Brownian Motion, which leads to the well-known Black-Scholes model [BS73].

**Example 4 (Geometric Brownian Motion).** The most popular stochastic process for generating prices is the Geometric Brownian Motion (GBM) process:

$$dS_t = \mu S_t \, dt + \sigma S_t \, dW_t,$$
where \( \mu \) is the drift parameter and \( \sigma \) the volatility, both constants, and \( W_t \) a Brownian motion. Under GBM, the returns are given by,

\[
\frac{S_{t+\Delta t} - S_t}{S_t} = \mu \Delta t + \sigma (W_{t+\Delta t} - W_t),
\]

and we can distinguish the long-term drift \( \mu \Delta t \) and the noise term, which is, by Definition 3.2, a normal distributed random variable with mean zero and variance \( \sigma^2 \).

The assumption that stock prices follows a Geometric Brownian Motion fits real time data very well for equities and indices. However, real data appears to have a greater probability of large rises and falls than the model predicts. Due to these discrepancies, and other reasons, different asset models have been proposed that can be calibrated more accurately to different markets.

### 3.1.1 Modeling Information

Due to the stochastic factor, the future value of the risky asset cannot be determined using all information known today. This notion of information is modeled in measure-theoretic probability using \( \sigma \)-algebras.

**Definition 3.3** (\( \sigma \)-algebra). Let \( \Omega \) be a nonempty set, and let \( \mathcal{F} \) be a collection of subsets of \( \Omega \). We say that \( \mathcal{F} \) is a \( \sigma \)-algebra (or \( \sigma \)-field) provided that:

(i) the empty set \( \emptyset \) belongs to \( \mathcal{F} \),

(ii) whenever a set \( A \) belongs to \( \mathcal{F} \), its complement \( A^c \) also belongs to \( \mathcal{F} \), and

(iii) whenever a sequence of sets \( A_1, A_2, \ldots \) belongs to \( \mathcal{F} \), their union \( \bigcup_{n=1}^{\infty} A_n \) also belongs to \( \mathcal{F} \).

The information associated with the \( \sigma \)-algebra \( \mathcal{F} \) can be thought of as follows. A random experiment is performed and an outcome \( \omega \) is determined, but the value of \( \omega \) is not revealed. Instead, for each set in the \( \sigma \)-algebra \( \mathcal{F} \), we are told whether \( \omega \) is in the set. The more sets there are on \( \mathcal{F} \), the more information this provides. If \( \mathcal{F} \) is the trivial \( \sigma \)-algebra, containing only \( \emptyset \) and \( \Omega \), this provides no information.

The accumulation of information over time is described by a filtration.

**Definition 3.4** (Filtration for the Brownian motion). Let \( (\Omega, \mathcal{F}, \mathbb{P}) \) be a probability space on with a Brownian motion \( \{W_t\}_{t \geq 0} \). A filtration for the Brownian motion is a collection of \( \sigma \)-algebras \( \{\mathcal{F}_t\}_{t \geq 0} \) satisfying,

(i) **(Information accumulates)** There is at least as much information available at the later time \( \mathcal{F}_t \) as there is at the earlier time \( \mathcal{F}_s \).

(ii) **(Adaptivity)** The information available at time \( t \) is sufficient to evaluate the Brownian motion \( W_t \) at that time.

(iii) **(Independence of future increments)** Any increment of the Brownian motion after time \( t \) is independent of the information available at time \( t \).

Let \( \Delta_t \) be a stochastic process. We say that \( \Delta_t \) is adapted to the filtration \( \mathcal{F}_t \) if for each \( t \geq 0 \), the random variable \( \Delta_t \) is \( \mathcal{F}_t \)-measurable.\(^2\)

A filtration tells us the information we will have at future times. More precisely, when we get to time \( t \), we will know for each set in \( \mathcal{F}_t \) whether the true \( \omega \) lies in that set. The properties (i) and (ii) in the definition above guarantee that the information available at each time \( t \) is at least as much as one would learn from observing the Brownian motion up to time \( t \). Property (iii) says that this information is of no use in predicting future movements of the Brownian motion.

In Example 4, we defined the asset price process through a *stochastic differential equation* (SDE). To find the solution of this SDE, and others, we apply Ito’s Lemma.

\(^2\)A random variable \( X \) is \( \mathcal{F}_t \)-measurable if and only if the information in \( \mathcal{F}_t \) is sufficient to determine the value of \( X \).
Definition 3.5 (Ito Process). Let \( W_t \) be a Brownian motion and let \( \mathcal{F}_t \) be an associated filtration. An Ito process is a stochastic process of the form,
\[
    dX_t = \Theta_t \, dt + \Delta_t \, dW_t,
\]
where \( X(0) \) is nonrandom and \( \Theta_t \) and \( \Delta_t \) are two adapted processes.

Theorem 3.1 (Ito’s Lemma). Let \( X_t \) be an Ito process, and let \( g(t, x) \) be a function for which the partial derivatives \( g_t(t, x) \), \( g_x(t, x) \) and \( g_{xx}(t, x) \) are defined and continuous. Then, the differential of the stochastic process \( g(t, X_t) \) is given by,
\[
    dg_t(t, X_t) = g_t(t, X_t) \, dt + g_x(t, X_t) \, dX_t + \frac{1}{2} g_{xx}(t, X_t)(dX_t)^2.
\]

Ito’s lemma is a powerful tool in stochastic calculus. Basically, stochastic calculus consists of the repeated application of Ito’s lemma.

3.1.2 Risk neutral rate

Before we can talk about option pricing, we introduce a second asset on the market, which is a risk-free asset. Let \( \{Z_t\}_{t \geq 0} \) be the process representing a bank account with \( Z_0 = 1 \) and whose differential is modeled by the diffusion process,
\[
    dZ_t = r \, dt, \tag{3.3}
\]
where \( r \geq 0 \) is the ‘spot’ interest rate, which we assume is constant. One should see \( Z_t \) as a ‘savings account’, representing the return at time \( t \) when one monetary unit was invested at time 0.

Taking the logarithm of Equation (3.3) and applying Ito’s Lemma allows us to solve the stochastic differential equation for \( Z_t \) and we find,
\[
    Z_t = e^{rt}. \tag{3.4}
\]

Note that \( Z_t \) is deterministic. The value of the bank account at any future time is known. We therefore say that \( r \) is the risk-free rate; the rate at which our money grows, without taking any risk.

Intuitively, when somebody invests in the stock market, he takes a risk. Nobody, at least no clever person, is willing to take this risk for free. Therefore, the long-term return of any risky investment should be higher than the risk-free rate. On the other hand, no investment opportunities exist paying more than the risk-free rate, without taking risk. This is a result of the no-arbitrage assumption of the next subsection.

3.1.3 No-Arbitrage

The pricing of assets can be best described using the no-arbitrage principle. Consider an agent on the market with a portfolio consisting of currency, assets and options. Then, when the agent starts with zero money, and, as some later time \( T \), is sure to not have lost money, and a positive probability of having made money.

This is made precise in the following definition.

Definition 3.6 (Arbitrage). An arbitrage is a portfolio value process \( \Pi_t \) satisfying \( \Pi_0 = 0 \) and also satisfying for some \( T > 0 \),
\[
    P(\Pi_T \geq 0) = 1, \quad P(\Pi_T > 0) > 0.
\]

We make the assumption that no arbitrage possibilities exist in the market. Then, under the no-arbitrage assumption, two assets with the same payoff should have the same price. This forms the basis of option pricing theory.

A type of process that plays an important role in no-arbitrage pricing are the martingales.

Definition 3.7 (Martingale). Let \( (\Omega, \mathcal{F}, P) \) be a probability space with fixed maturity \( T \) and let \( \mathcal{F}_t \) be a filtration. Consider an adapted process \( M_t, 0 \leq t \leq T \). If
\[
    \mathbb{E}[M_t \mid \mathcal{F}_s] = M_s \text{ for all } 0 \leq s \leq t \leq T,
\]
we say the process \( M_t \) is a martingale.
In words, when a stochastic process is a martingale, we say it has no tendency to rise or fall. A stochastic process is a martingale if the drift-term is zero.

Fix a time horizon $0 \leq t \leq T$ and consider a Brownian motion $W_t$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and let $\mathcal{F}_t$ be a filtration for this Brownian motion. Let $S_t$ be a stock price process follow a generalized geometric Brownian motion whose differential is given by,

$$dS_t = \alpha_t S_t dt + \sigma_t S_t dW_t,$$

where $\sigma_t$ and $\alpha_t$ are allowed to be adapted processes. We define the discounted asset price process as $S_t / Z_t$, and its differential can be found by application of Ito's Lemma, and is given by,

$$d(S_t / Z_t) = \sigma_t (S_t / Z_t)[\Theta_t dt + dW_t],$$

where we defined the market price of risk as,

$$\Theta_t = \frac{\alpha_t - r}{\sigma_t}.$$

We note that the discounted asset price process is not a martingale, due to the occurrence of a nonzero drift term under the real-world measure $\mathbb{P}$. However, we use Girsanov’s theorem to introduce a new measure $\tilde{\mathbb{P}}$, which will render the discounted price process in a martingale.

**Theorem 3.2** (Girsanov). Fix a time horizon $0 \leq t \leq T$ and consider a Brownian motion $W_t$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with a filtration $\mathcal{F}_t$ for this Brownian motion and let $\Theta_t$ be an adapted process. Define the processes $\tilde{W}_t$ and $Z_t$ by their SDE,

$$dZ_t = -\Theta_t Z_t dW_t,$$

$$d\tilde{W}_t = dW_t + \Theta_t dt,$$

and assume that,

$$\mathbb{E} \int_0^T \Theta_u^2 Z_u^2 du < \infty.$$

Set $Z = Z_T$. Then $\mathbb{E} Z = 1$ and under the probability measure $\tilde{\mathbb{P}}$, defined by,

$$\tilde{\mathbb{P}} = \int_A Z_dP(\omega) \text{ for all } A \in \mathcal{F},$$

the process $\tilde{W}$ is a Brownian motion.

We introduce the probability measure $\tilde{\mathbb{P}}$ defined in Girsanov’s Theorem, which uses the marked price of risk $\Theta_t$. Then, under this measure $\tilde{\mathbb{P}}$, the discounted stock price is a martingale,

$$d(S_t / Z_t) = \sigma_t (S_t / Z_t)d\tilde{W}_t.$$

We refer to this measure $\tilde{\mathbb{P}}$ as the risk-neutral measure.

### 3.1.4 Replicating Portfolio

We use a so-called replicating portfolio to price options. Consider an option on a stock $S_t$ with payoff $V_T$ at time $T$.

Next, consider an agent who begins with initial capital $\Pi_0$ and at each time $t, \ 0 \leq t \leq T$, holds $\Delta_t$ shares of stock, investing or borrowing at the interest rate $r$ as necessary to finance this, such that at time $T$, it satisfies $\Pi_T = V_T$. (We show in the next section that this trading strategy exists.)

At any time $t$, the agent holds $\Delta_t$ shares of asset. The remainder of the portfolio value $\Pi_t - \Delta_t S_t$ is invested in the money market bank account, and thus grows with the risk-neutral interest rate $r$. The dynamics $d\Pi$ of the portfolio are therefore given by,

$$d\Pi_t = \Delta_t dS_t + r(\Pi_t - \Delta_t S_t) dt.$$
Inserting the definition of the dynamics \( dS_t \) gives,
\[
\begin{align*}
\frac{d\Pi_t}{\partial t} &= \Delta_t (\alpha_t S_t dt + \sigma_t S_t dW_t) + r(\Pi_t - \Delta_t S_t) dt \\
&= r\Pi_t dt + \Delta_t S_t \sigma_t \left( \frac{\alpha_t - r}{\sigma_t} dt + dW_t \right) \\
&= r\Pi_t dt + \Delta_t S_t \sigma_t (\Theta_t dt + dW_t) \\
&= r\Pi_t dt + \Delta_t S_t \sigma_t dW_t
\end{align*}
\]
(3.6)

Using Ito’s Lemma once more, we find that the dynamics of the discounted portfolio are given by,
\[
d(\Pi_t/\alpha) = \sigma_t \Delta_t (S_t/\alpha) d\tilde{W}_t, \tag{3.7}
\]
i.e., the discounted portfolio is a martingale with respect to the risk-neutral measure.

Then, under the risk-neutral measure, the discounted asset price is a martingale, which implies,
\[
\Pi_t = \tilde{E}[e^{-r(T-t)}\Pi_T|\mathcal{F}_t] = \tilde{E}[e^{-r(T-t)}V_T|\mathcal{F}_t],
\]
where \( \tilde{E} \) is the expectation taken under the risk neutral measure \( \tilde{\mathbb{P}} \).

Thus, \( \Pi_t \) is the capital required at time \( t \) to create a hedging portfolio with payoff \( V_T \). Therefore, we call this the price \( V_t \) of the option at time \( t \). Thus, in the Black-Scholes model, we obtain the risk-neutral option pricing formula,
\[
V_t = \tilde{E}[e^{-r(T-t)}V_T|\mathcal{F}_t].
\]

We skipped over the existence of the trading strategy, which is key to the risk-neutral option valuation formula. In the one-dimensional case, i.e., where only one asset on one source of randomness exists, we can find an explicit formula for the hedging strategy \( \Delta_t \).

When we define \( V_t \) by the risk-neutral pricing formula, it is a \( \tilde{\mathbb{P}} \)-martingale, and therefore its dynamics are of the form,
\[
d(V_t/\alpha) = \tilde{\Gamma}_t d\tilde{W}_t,
\]
for some adapted process \( \tilde{\Gamma}_t \). On the other hand, our discounted replicating portfolio \( \Pi_t \), satisfies,
\[
d(\Pi_t/\alpha) = \Delta_t \sigma_t (S_t/\alpha) d\tilde{W}_t,
\]
and in order to have \( X_t = V_t \) for all \( t \), we should set \( X_0 = V_0 \) and,
\[
\Delta_t = \frac{\tilde{\Pi}_t}{\sigma_t (S_t/\alpha)}.
\]

Now, we have a hedge for the derivative security with payoff \( V_T \) at time \( T \).\(^3\) Note that this approach, known to as the martingale representation theorem, does not give a practical method of finding the hedging portfolio \( \Delta_t \).

### 3.2 Connection with partial differential equations

Another approach in option pricing is to see that the option value is a function of time \( t \) and the asset \( S_t \). Therefore, the dynamics of the option value \( V(S_t, t) \) at time \( t \) and asset value \( S_t \) can be found using Ito’s formula, assuming that the partial derivatives \( \frac{\partial V}{\partial S} (S_t, t) \), \( \frac{\partial V}{\partial t} (S_t, t) \) and \( \frac{\partial^2 V}{\partial S^2} (S_t, t) \) exists and are continuous. Thus, the dynamics of \( V(S_t, t) \) are given by,
\[
\begin{align*}
\frac{dV(S_t, t)}{\partial t} &= \frac{\partial V}{\partial S} (S_t, t) dS_t + \frac{\partial V}{\partial S} (S_t, t) dS_t + \frac{\partial^2 V}{\partial S^2} (S_t, t) dS_t^2.
\end{align*}
\]

Using that \( S_t \) is driven by a geometric Brownian motion,
\[
dS_t = \alpha_t S_t dt + \sigma_t S_t dW_t,
\]
\(^3\)Assuming that \( \sigma_t > 0 \) and that there is only one source of randomness.
we find that the dynamics of the option value are equivalent to,
\[ d\Pi_t = \left( \alpha_t S_t \frac{\partial V}{\partial S_t} + \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_t^2 S_t^2 \frac{\partial^2 V}{\partial S_t^2} \right) dt + \sigma_t S_t \frac{\partial V}{\partial S_t} dW_t, \]
where we omitted the dependence on \((S_t, t)\) for notation reasons on several places.

The approach here is again to create a replicating portfolio \(\Pi_t\), consisting of one \(V(S_t, t)\) and \(\Delta_t\) shares of asset. I.e., \(\Pi_t = V(S_t, t) - \Delta_t S_t\). By Ito’s formula,
\[
d\Pi_t = dV - \Delta_t dS_t
= \left( \alpha_t S_t \frac{\partial V}{\partial S_t} + \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_t^2 S_t^2 \frac{\partial^2 V}{\partial S_t^2} \right) dt + \sigma_t S_t \frac{\partial V}{\partial S_t} dW_t - \Delta_t (\alpha_t S_t dt + \sigma_t S_t dW_t). \tag{3.8} \]

Now, we can eliminate randomness from the portfolio by setting \(\Delta_t = \frac{\partial V}{\partial S_t}\), such that the \(dW_t\) terms cancel out. Then, the portfolio is a risk-neutral portfolio, and grows with the risk-neutral rate \(r\), i.e.,
\[ d\Pi_t = r\Pi dt = r(V - \frac{\partial V}{\partial S_t}) \text{ as in Section 3.1.2.} \]
With these two changes, (3.8) implies,
\[ \frac{\partial V}{\partial t} + \frac{1}{2} \sigma_t^2 S_t^2 \frac{\partial^2 V}{\partial S_t^2} + r S_t \frac{\partial V}{\partial S_t} - rV = 0, \tag{3.9} \]
which is the Black-Scholes pricing PDE for European options.

We thus presented two approaches to price options, one following the martingale representation theorem, resulting in the risk-neutral option valuation formula. The other approach presented here in a PDE setting. These two approaches are connected to each other through the Feynman-Kac theorem, stated below [Shr04].

**Theorem 3.3 (Discounted Feynman-Kac).** Consider the stochastic differential equation,
\[ dX_u = \beta(u, X_u) du + \gamma(u, X_u) dW_u. \]
Let \(h(y)\) be a Borel-measurable function and let \(r\) be constant. Fix \(T > 0\) and let \(t \in [0, T]\) be given. Define the function,
\[ f(t, x) = \mathbb{E}[e^{-rT} h(X_T) \mid \mathcal{F}_t]. \]
(We assume that \(E|h(X_t)| < \infty\) for all \(t\) and \(x\).) Then \(f(t, x)\) satisfies the partial differential equation,
\[ f_t(t, x) + \beta(t, x) f_x(t, x) + \frac{1}{2} \gamma^2(t, x) f_{xx}(t, x) = r f(t, x), \]
and the terminal condition,
\[ f(T, x) = h(x) \text{ for all } x. \]

### 3.3 Exponential Lévy Asset Dynamics

We focus on a specific subset of asset models called exponential Lévy processes [App04, Sch03], due to their known characteristic function and stationary increments. Exponential Lévy models generalize the classical Black-Scholes framework [BS73] by allowing the stock prices to jump, while preserving the independence and stationarity of returns. Jumps in price processes are introduced for a number of reasons, but mainly because asset prices do jump. Besides, some risks cannot be handled with continuous-path models [Tan09].

Lévy processes are named after the French mathematician Paul Lévy (1886–1971) [OR15], and can be seen as a continuous analog of a random walk, having independent and stationary increments. The best known Lévy processes are Brownian motion and the Poisson process. In fact, the only Lévy process with continuous (non-deterministic) paths is Brownian motion with drift.

**Definition 3.8.** A stochastic process \(\{L_t\}_{t \geq 0}\) is a Lévy process if it satisfies the following properties:

(i) \(L_0 = 0\), (with probability 1),

(ii) **Stationary increments:** For any \(s < t\), \(L_t - L_s\) is equal in distribution to \(L_{t-s}\),
3. EUROPEAN OPTION PRICING METHODS

(iii) **Independence of increments:** For any $0 \leq t_1 < t_2 < \cdots < t_n < \infty$, the random variables $L_{t_2} - L_{t_1}, L_{t_3} - L_{t_2}, \ldots, L_{t_n} - L_{t_{n-1}}$ are independent.

(iv) **Continuity in probability:** For any $\varepsilon > 0$ and $t \geq 0$ it holds that,

$$
\lim_{h \to 0} P (|L_{t+h} - L_t| > \varepsilon) = 0.
$$

Of the properties of the increments in Definition 3.8, (ii) and (iii), are the most important ones. Property (i) is just a normalization and property (iv) is a technicality allowing us to do serious analysis.

**Theorem 3.4.** A Lévy process $\{L_t\}_{t \geq 0}$ can be entirely specified in terms of its characteristic exponent, defined as,

$$
\psi_L(\omega) := \ln[\exp(i\omega L(1))].
$$

**Proof.** We refer the reader to [CT04] for a proof. □

The characteristic function of a Lévy process is known due to the celebrated Lévy-Khintchine formula.

**Theorem 3.5 (Lévy-Khintchine formula).** By the Lévy-Khintchine formula, the characteristic function of the Lévy process $\{L_t\}_{t \geq 0}$ defined by the characteristic exponent $\psi_L(\omega)$ equals,

$$
\hat{f}_L(\omega; t) = \mathbb{E}[e^{i\omega L_t}] = e^{i\psi_L(\omega)}. \quad (3.10)
$$

**Proof.** We refer the reader to [CT04] for a proof. □

To model the underlying randomness for the risky asset $\{S_t\}_{t \geq 0}$ with dividend yield $q$, we assume an exponential Lévy process of the form,

$$
S_t = S_0 e^{\mu t + L_t}, \quad (3.11)
$$

where $\{L_t\}_{t \geq 0}$ is a Lévy process defined by its characteristic exponent $\psi_L$, and $\mu$ is a drift parameter, defined such that the reinvested relative price $e^{rt} S_t / Z_t$ is a martingale under the risk-neutral measure. That is, $\mathbb{E}[e^{rt} S_t / Z_t] = S_0$. We use the deterministic bank account $Z_t = e^{rt}$, and by application of the Lévy-Khintchine formula, we find,

$$
\mathbb{E}[e^{i\omega S_t} / Z_t] = S_0 e^{-(r-q-\mu)t} \mathbb{E}[e^{\omega L_t}] = S_0 e^{-(r-q-\mu)t} e^{i\psi_L(-i)} = S_0 e^{\mu t - (r-q-\psi_L(-i))t},
$$

thus by defining the drift as,

$$
\mu := r - q - \psi_L(-i), \quad (3.12)
$$

the desired martingale property is obtained.

The characteristic function of $S_t$ itself is not known in general, but when we log-transform the asset domain, the resulting log-asset process is a Lévy process with the characteristic function known by Theorem 3.5. Let the log-asset process be given by $Y_t := \log(S_t/S_0)$. Then, by Equation (3.11), $Y_t$ is equal to the Lévy process with drift, $Y_t = \mu t + L_t$. The characteristic function4 $\hat{f}_Y(\omega; t)$ of $Y_t$ is given by,

$$
\hat{f}_Y(\omega; t) = \mathbb{E}[e^{i\omega Y_t}] = e^{i\omega \mu t} \mathbb{E}[e^{i\omega L_t}] = e^{t(i\omega \mu + \psi_L(\omega))} = e^{t\psi_Y(\omega)},
$$

thus $Y_t$ is a Lévy-process itself with characteristic exponent $\psi_Y(\omega)$.

A second process we use later on is the log-asset process $X_t := \log(S_t/K)$, where $K$ is a positive constant, later to be chosen as the strike price of the option. By defining $x := \log(S_0/K)$, it follows that $X_t = Y_t + x = x + \mu t + L_t$. Note that $X_t$ itself is not a Lévy process, as it fails requirement (i) when $x \neq 0$. However, due to the independent increments property of Lévy processes, we find that the probability density function of $X_t$ given $x$ is,

$$
\hat{f}_X(y; t|x) = \hat{f}_X(y-x; t|0) = \hat{f}_Y(y-x; t).
$$

Since, in general, the probability density function $\hat{f}_L(x; t)$ corresponding to the Lévy process $L_t$ is unavailable, it is hard to find the moments of the distribution. However, we will make use of the cumulants to capture the salient properties of the distribution.

4Note again that we distinguish between the characteristic function, denoted by $f(\omega)$ and the Fourier transform $\hat{f}(\omega)$. They are related by $f(\omega) = \hat{f}(-\omega)$.  

24
For the sake of clarity, we include the following definitions and theorems:

**Definition 3.9** (Cumulant generating function). Let \( \hat{f}_X \) be the characteristic function corresponding to a random variable \( X \). We define the cumulant-generating function of this random variable by,

\[
g(\omega) := \log E[e^{\omega X}] = \log \hat{f}_X(-i\omega) = \log \hat{f}_X(i\omega),
\]

and the \( i \)-th cumulant of \( X \), denoted by \( c_i \), is given by the \( i \)-th derivative at zero of \( g(\omega) \) i.e.,

\[
c_i := g^{(i)}(0).
\]

When numerically approximating integrals, truncation of integrals with infinite integration domain is required. However, it is important that ‘enough’ mass is captured, i.e.,

\[
\int_a^b f(x) \, dx \approx \int_a^b f(x) \, dx,
\]

for some interval \([a, b] \subset \mathbb{R}\). A proposed heuristic by Fang and Oosterlee [FO09] is to make use of the cumulants of the distribution. Let \( c_n \) denote the \( n \)-th cumulant of \( y = \ln(S_T/K) \) and let \( L > 0 \) be a scaling parameter, then

\[
[a, b] := \left[ c_1 - L\sqrt{c_2 + c_4}, c_1 + L\sqrt{c_2 + c_4} \right],
\]

where \( L \) is suggested to be chosen in the range 7.5 – 10. A truncation rule which also includes \( c_6 \) is more accurate for extreme short maturities, but the sixth cumulant is relatively difficult to derive for many models.

### 3.3.1 Risk-Neutral European option valuation formula

We transform the payoff function of the option to the log-asset domain, as only the characteristic function in log-space is known.

Denote with \( v(x,t) \) the option value at time \( t = 0 \) and log-asset value \( x = \log(S_0/K) \). The option payoff in (3.1) in terms of the log-asset at time \( T \) can be written as

\[
v(y,T) = \{\alpha \cdot K(e^y - 1)\}^+, \quad \text{with} \quad \alpha = \begin{cases} 1 & \text{for a call,} \\ -1 & \text{for a put,} \end{cases}
\]

The risk-neutral option valuation formula provides us, for \( t = 0 \), with the European option pricing formula in the log-asset domain,

\[
v(x,t) = e^{-r(T-t)}E[v(y,T) \, | \, x] = e^{-r(T-t)}\int_{\mathbb{R}} v(y,T) f(y|x) \, dy,
\]

where \( f(y|x) \) denotes the probability density function of \( X_T \) given \( X_0 \equiv x \).

We now discuss a few methods to approximate this integral. As no analytic solution exists for almost all underlying density functions, one of the major drawbacks is that this density function is often not available in explicit form. The methods we discuss below are based on recovery of the density function from the characteristic function in terms of a series expansion.

First, we discuss the COS method, which uses a Fourier Cosine expansion to recover the density. Then, the WA\([a,b]\) method uses Haar wavelets or linear B-Splines to recover the density. Finally, we focus on the method of our interest, the SWIFT method, based on the Shannon wavelet expansion.

### 3.4 COS method

The COS method for European options, introduced in [FO08], is based on the insight that the Fourier-Cosine series coefficients of the underlying density function are closely related to its characteristic function. Starting with the risk-neutral pricing formula,

\[
v(x,t) = e^{-r(T-t)}\int_{\mathbb{R}} v(y,T) f(y|x) \, dy,
\]
we approximate the density function \( f(y|x) \). Since \( f(y|x) \) decays rapidly as \( y \to \pm \infty \), we can truncate the infinite integration range in the risk-neutral valuation formula without losing significant accuracy. Suppose that we have, with \([a, b] \subset \mathbb{R}\),

\[
\int_{\mathbb{R}\setminus[a,b]} f(y|x) \, dy < \text{TOL},
\]

for some given tolerance \text{TOL}. The interval \([a, b]\) is determined by the cumulants method of (3.13). Then we can approximate \( v(x,t) \) in (3.15) by,

\[
v(x,t) \approx v_1(x,t) = e^{-r(T-t)} \int_a^b v(y,T) f(y|x) \, dy.
\]  

(3.16)

(The intermediate terms \( v_i \) are used to distinguish approximation errors.) As a second step, we replace the (unknown) density function \( f(y|x) \) by its Fourier-cosine expansion over \([a, b]\),

\[
f(y|x) = \sum_{k=0}^{\infty} D_k(x) \cos \left( k\pi \frac{y-a}{b-a} \right),
\]

where, \( D_k(x) = \frac{2}{b-a} \int_a^b f(y|x) \cos \left( k\pi \frac{y-a}{b-a} \right) \, dy, \)  

(3.17)

where the apostrophe (') after the summation sign denotes that the first term of the summation is divided by two. We will refer to \( D_k(x) \) as the (Fourier cosine) density coefficients. Inserting the Fourier cosine expansion of \( f(y|x) \) into (3.16), using Fubini’s Theorem, gives,

\[
v_1(x,t) = e^{-r(T-t)} \sum_{k=0}^{\infty} D_k(x) \left[ \int_a^b v(y,T) \cos \left( k\pi \frac{y-a}{b-a} \right) \, dy \right],
\]

(3.18)

where we note that the integral at the right-hand side is equal to the Fourier coefficients of \( v(y,T) \) in \( y \) (except for a constant). We therefore define the payoff coefficients \( V_k \) as the Fourier cosine-series coefficients of \( v(y,T) \) as,

\[
V_k := \frac{2}{b-a} \int_a^b v(y,T) \cos \left( k\pi \frac{y-a}{b-a} \right) \, dy,
\]

(3.19)

and obtain,

\[
v_1(x,t) = \frac{b-a}{2} e^{-r(T-t)} \sum_{k=0}^{\infty} D_k(x) V_k.
\]

Due to the rapid decay of the payoff and density coefficients, we can further truncate the series summation to obtain,

\[
v(x,t) \approx v_2(x,t) = \frac{b-a}{2} e^{-r(T-t)} \sum_{k=0}^{N-1} D_k(x) V_k.
\]

### 3.4.1 Density Coefficients

The strength of the COS method is the insight that the Fourier-cosine coefficients \( D_k(x) \) are closely related to the conditional characteristic function.

When we assume that the density function \( f(y|x) \) is an \( L^2(\mathbb{R}) \)-function, the characteristic function \( \hat{f}(\omega; x) := \hat{f}(-\omega; x) \) is also in \( L^2(\mathbb{R}) \), justifying the following approximation,

\[
\int_a^b f(y|x) e^{i\omega y} \, dy \approx \int_{\mathbb{R}} f(y|x) e^{i\omega y} \, dy = \hat{f}(\omega; x)
\]
We can now derive an approximation for the density coefficients,

\[ D_k(x) = \frac{2}{b-a} \int_a^b f(y|x) \cos \left( k\pi \frac{y-a}{b-a} \right) dy \]

\[ = \frac{2}{b-a} \Re \left\{ e^{-ik\pi \frac{a}{b-a}} \int_a^b f(y|x) \exp \left( ik\pi \frac{y}{b-a} \right) dy \right\} \quad (3.20) \]

In a final step, we replace \( D_k(x) \) by its approximation \( D_k^*(x) \) in \( v_2(x,t) \) to obtain the general COS pricing formula,

\[ v(x,t) \approx v_3(x,t) = e^{-r(T-t)} \sum_{k=0}^{N-1} \Re \left\{ \tilde{f} \left( k\pi \frac{y-a}{b-a} ; x \right) e^{-ik\pi \frac{a}{b-a}} \right\} V_k, \quad (3.21) \]

where the density coefficients \( V_k \) are determined by the payoff of the option.

### 3.4.2 Plain Vanilla Payoff Coefficients

The payoff coefficients \( V_k \), as defined in (3.19), for a European call (or put) with pay-off function (3.14) are given by,

\[ V_k = \frac{2}{b-a} \int_a^b [\alpha \cdot K(e^y - 1)]^+ \cos \left( k\pi \frac{y-a}{b-a} \right) dy. \]

Let us consider a European call option, \( \alpha = 1 \). For a put, the steps are similar. We distinguish two different cases. If \( a < b < 0 \), the integral equals zero, and \( V_k = 0 \) for all \( k \). In the other case, set \( \bar{a} = \max(0,a) \). We can then rewrite \( V_k \) as,

\[ V_k = K \left[ \frac{2}{b-a} \int_a^b e^y \cos \left( k\pi \frac{y-a}{b-a} \right) dy - \frac{2}{b-a} \int_a^b \cos \left( k\pi \frac{y-a}{b-a} \right) dy \right], \]

where the first term within the brackets represents the Fourier cosine coefficient of the function \( e^y \), and the second term the Fourier cosine coefficient of the constant function 1. Both of them can be solved analytically with their expressions in the next lemma.

**Lemma 3.1.** Let \( \chi_k(c,d) \) and \( \psi_k(c,d) \) be the cosine series coefficients on \( [c,d] \subset [a,b] \subset \mathbb{R} \) of respectively \( g(y) = e^y \) and \( g(y) = 1 \). Then, given by

\[ \chi_k(c,d) = \frac{2}{b-a} \left[ \frac{1}{k\pi} \begin{cases} \cos \left( k\pi \frac{d-a}{b-a} \right) e^d & k \neq 0 \\ \sin \left( k\pi \frac{d-a}{b-a} \right) & k = 0 \end{cases} \right] \]

\[ + \frac{k\pi}{b-a} \sin \left( k\pi \frac{d-a}{b-a} \right) e^d \left( k\pi \frac{c-a}{b-a} \right) e^c, \]

\[ \psi_k(c,d) = \begin{cases} \frac{2}{b-a} \left[ \sin \left( k\pi \frac{d-a}{b-a} \right) - \sin \left( k\pi \frac{c-a}{b-a} \right) \right] \frac{b-a}{k\pi} & k \neq 0 \\ \frac{2}{b-a} (d-c) & k = 0. \end{cases} \]

**Proof.** This can be solved using basic calculus, and for a proof the reader is referred to [FO08]. \( \square \)

With Lemma 3.1, the payoff coefficients can now be written as

\[ V_k = \begin{cases} K(\chi_k(\bar{a},b) - \psi_k(\bar{a},b)), & \text{for a call,} \\ K(-\chi_k(a,b) + \psi_k(a,b)), & \text{for a put,} \end{cases} \quad (3.22) \]

where \( \bar{b} := \min(0,b) \).
3. EUROPEAN OPTION PRICING METHODS

3.4.3 Pricing multiple strikes

It is worth mentioning that the COS pricing formula (3.21) is greatly simplified for the Lévy and Heston models, so that options for many strike prices can be computed simultaneously. We denote vectors with bold-faced characters. For a vector of strikes \( K \), the \( V \)-formulas for European options can be factored as

\[
V_k = K U_k,
\]

where \( U_k \) is, independent of the strike, given by,

\[
U_k = \left[ \frac{2}{b-a} \int_a^b e^{by} \cos \left( k\pi \frac{y-a}{b-a} \right) dy - \frac{2}{b-a} \int_a^b e^{by} \cos \left( k\pi \frac{y-a}{b-a} \right) dy \right].
\]

For Lévy processes, whose characteristic functions can be represented by,

\[
\hat{f}(\omega; x) = \hat{f}^{\text{levy}}(\omega) e^{i\omega x}, \quad \hat{f}^{\text{levy}}(\omega) := \hat{f}(\omega; 0),
\]

the pricing formula is simplified to,

\[
v(x, t) \approx e^{-r(T-t)} K \sum_{k=0}^{N-1} \text{Re} \left\{ \hat{f}^{\text{levy}} \left( \frac{k\pi}{b-a} \right) e^{ik\pi \frac{x-a}{b-a}} U_k \right\},
\]

(3.23)

where the summation can be written as a matrix-vector product if \( K \) (and therefore \( x \)) is a vector. We see that the evaluation of the characteristic function is independent of the strike. In general, the evaluation of the characteristic function is more expensive than the other computations.

In the section with numerical results, we show that with very small \( N \) we can achieve highly accurate results.

3.5 WA\([a,b]\) method

The WA\([a,b]\) method is introduced in [OGO13] and follows a similar approach as the COS method, but is based on a wavelet expansion using the \( j \)th order cardinal B-splines as the scaling function for \( j = 0 \) and \( j = 1 \). We already saw B-splines in Example 3, and the scaling functions are given by,

\[
\phi^0(x) := \begin{cases} 1, & \text{if } x \in [0,1), \\ 0, & \text{otherwise}, \end{cases} \quad \text{and} \quad \phi^1(x) := \begin{cases} x, & \text{if } x \in [0,1), \\ 2-x, & \text{if } x \in [1,2), \\ 0, & \text{otherwise}, \end{cases}
\]

and are referred to as Haar wavelets and linear B-splines, respectively.

Higher order B-splines are defined recursively by a convolution,

\[
\phi^j(x) = \int_0^1 \phi^{j-1}(x-t) dt, \quad j \geq 1,
\]

and are used for option pricing in [Kir15].

Following the MRA framework, we define the wavelet family \( \{\phi^j_{m,k}\}_{m,k \in \mathbb{Z}} \) with father wavelets,

\[
\phi^j_{m,k}(x) := 2^{m/2} \phi^j(2^mx - k),
\]

for a fixed wavelet scale \( m \). We discuss the choice of \( m \in \mathbb{N} \) in the numerical section at the end of this chapter.

Since splines are only piecewise polynomial functions, see Figure 2.2, they are very easy to implement. In [OGO13], two methods for approximating the density function are described. We focus on the method that applies a \textit{Wavelet Approximation} on a bounded interval \([a,b]\), the WA\([a,b]\) method.

We assume that the density function \( f(y|x) \) is an \( L^2(\mathbb{R}) \)-function, and thus the mass in the tails tends to zero when \( y \to \pm \infty \), so that it can be well approximated in a finite interval \([a,b]\) by,

\[
f^c(y|x) = \begin{cases} f(y|x), & \text{if } x \in [a,b], \\ 0, & \text{otherwise}. \end{cases}
\]
Following the theory of MRA on a bounded interval\(^5\), we can approximate \( f^c \approx f_{m,j}^c \) for all \( y \in [a, b] \), where,

\[
f_{m,j}^c(y|x) = \sum_{k=0}^{(j+1)\cdot(2^m-1)} D_{m,k,j}^i(x) \phi_{m,k}^j \left( \frac{j+1 \cdot y-a}{b-a} \right), \quad j \geq 0,
\]

with convergence in the \( L^2(\mathbb{R}) \)-norm and \( D_{m,k,j}^i(x) \) are the wavelet density coefficients.

**Remark 3.1 (Notation).** Generally, we use \( D \) to denote density coefficients, and \( V \) to denote payoff coefficients in the pricing of European options. From context, it should be clear which method is discussed at that point. Furthermore, they can be distinguished by their sub- and superscripts. The WA\(^{[A,B]} \) method has two subscripts and a superscript \( j \) indicating the order or the \( B \)-spline, while the COS method has only has a single subscript.

We use the cumulants approach to determine the interval \([a, b]\), and obtain the truncated risk-neutral option valuation formula for \( v_1(x, t) \) as in (3.16). Then, by substituting \( \hat{f} \) by \( f_{m,j}^c \), we obtain after interchange of integration and summation,

\[
v_2(x, t) = e^{-r(T-t)} \int_a^b v(y, T) f_{m,j}^c(y|x) \, dy
\]

\[
= e^{-r(T-t)} \sum_{k=0}^{(j+1)\cdot(2^m-1)} D_{m,k,j}^i(x) V_{m,k}^j,
\]

where we define the payoff coefficients as,

\[
V_{m,k}^j = \int_a^b v(y, T) \phi_{m,k}^j \left( \frac{j+1 \cdot y-a}{b-a} \right) \, dy,
\]

and the density coefficients \( D_{m,k}^j \) to be discussed in the subsequent section.

### 3.5.1 Density Coefficients

As in [OGO13], we use Cauchy’s integral formula to find an expression for the density coefficients \( D_{m,k}^j(x) \). An alternative approach, based on Parseval’s identity, is described in [Kir15].

The main idea behind the Wavelet Approximation method is to approximate \( \hat{f} \) by \( f_{m,j}^c \) and then to compute the coefficients \( D_{m,k}^j \) by inverting the Fourier Transform. Proceeding this way, we have,

\[
\hat{f}(\omega; x) = \int_\mathbb{R} f(y|x) e^{-i\omega y} \, dy \approx \int_\mathbb{R} f_{m,j}^c(y|x) e^{-i\omega y} \, dy
\]

\[
= \sum_{k=0}^{(j+1)\cdot(2^m-1)} D_{m,k}^j(x) \left[ \int_\mathbb{R} \phi_{m,k}^j \left( \frac{j+1 \cdot y-a}{b-a} \right) e^{-i\omega y} \, dy \right].
\]

Introducing a change of variables, \( u = (j+1) \cdot \frac{y-a}{b-a} \), gives us,

\[
\hat{f}(\omega; x) \approx \frac{b-a}{j+1} \cdot e^{-i\omega} \sum_{k=0}^{(j+1)\cdot(2^m-1)} D_{m,k}^j(x) \left[ \int_\mathbb{R} \phi_{m,k}^j(u) e^{-i\omega \frac{b-a}{j+1} u} \, du \right]
\]

\[
= \frac{b-a}{j+1} \cdot e^{-i\omega} \sum_{k=0}^{(j+1)\cdot(2^m-1)} D_{m,k}^j(x) \hat{\phi}_{m,k}^j \left( \frac{b-a}{j+1} \cdot \omega \right).
\]

Taking into account that \( \hat{\phi}_{m,k}^j(\omega) = 2^{-m} \hat{\phi}^j \left( \frac{\omega}{2^m} \right) e^{-i \frac{k}{2^m} \omega} \) and performing a change of variables, \( z = e^{-i \frac{k}{2^m} \omega} \), we find by rearranging terms,

\[
P_{m,k}^j(z; x) \approx Q_{m}^j(z; x),
\]

\(^5\)Scaling function in a bounded interval are discussed in detail in [Chu92].
where,

\[ P_{n}^{j}(z; x) := \sum_{k=0}^{(j+1)\cdot(2^n-1)} D_{m,k}^{j}(x)z^k, \]

\[ Q_{n}^{j}(z; x) := \frac{2\pi i (j+1)z^{-2^{m}(j+1)n}}{(b-a)\hat{\gamma}(i\cdot\log(z)).} \]

Since \( P_{n}^{j}(z; x) \) is a polynomial (in \( z \)), it is (in particular) analytic inside a disc of the complex plane \( \{z \in \mathbb{C} : |z| < \rho \} \) for \( \rho > 0 \). We can obtain expressions for the coefficients \( D_{m,k}^{j}(x) \) by means of Cauchy’s integral formula. This is,

\[ D_{m,k}^{j}(x) = \frac{1}{2\pi i} \int_{\gamma} \frac{P_{n}^{j}(z; x)}{z^{k+1}} dz, \quad k = 0, \ldots, (j+1)\cdot(2^n-1), \]

where \( \gamma \) denotes a circle of radius \( \rho \), \( \rho > 0 \), about the origin. We set \( \rho = 0.9995 \) [OGM14]. Considering now the change of variables \( z = \rho e^{iu} \), and the approximation \( P_{n}^{j}(z; x) \approx Q_{n}^{j}(z; x) \) gives us,

\[ D_{m,k}^{j}(x) \approx D_{m,k}^{j,\ast}(x) := \frac{1}{2\pi \rho^k} \int_{0}^{2\pi} Q_{n}(\rho e^{iu}; x)e^{-iku} du, \quad (3.28) \]

We approximate the above integral with the Trapezoidal Rule over the grid points \( u_n = n \frac{2\pi}{N} \) for \( N = 2^m(j+1) \) and \( n = 0, 1, 2, \ldots, N-1 \). Thus, the final approximation for the density coefficients is,

\[ D_{m,k}^{j,\ast}(x) \approx \frac{1}{N\rho^k} \text{Re} \left\{ \sum_{n=0}^{N-1} \tau_n Q_{n}(\rho e^{iu}; x)e^{-iku} \right\}, \quad (3.29) \]

where \( \tau_n = 1 - \frac{1}{2}(\delta_{1,n} + \delta_{x,n}) \) are the Trapezoidal Rule weights.

Note that we can directly apply the FFT-algorithm to compute the whole vector of coefficients \( \{D_{m,k}^{j}\}_{k=0}^{N} \) with a computational complexity of just \( \mathcal{O}(N \cdot \log_2(N)) \).

The resulting B-splines wavelet pricing formula for general European options is,

\[ v(x, t) \approx e^{-r(T-t)} \sum_{k=0}^{N} D_{m,k}^{j,\ast}(x)V_{m,k}^{j}. \quad (3.30) \]

### 3.5.2 Plain Vanilla Payoff Coefficients

We derive the payoff coefficients for a European call (or put) with payoff as in (3.14). The payoff coefficients, as defined in (3.26), are given by,

\[ V_{m,k}^{j} = \int_{a}^{b} [\alpha \cdot K(e^{y} - 1)]^{+} \phi_{m,k}^{j} \left( (j+1) \frac{y-a}{b-a} \right) dy. \]

Let us consider a European call option, \( \alpha = 1 \). For a put, the steps are similar.

We distinguish two different cases. If \( a < b < 0 \), the integral equals zero, and \( V_{m,k}^{j} = 0 \) for all \( k \). In the other case, set \( \tilde{a} = \max(0, a) \). We can then rewrite \( V_{m,k}^{j} \) as

\[ V_{m,k}^{j} = K \left[ \int_{a}^{b} e^{y}\phi_{m,k}^{j} \left( (j+1) \frac{y-a}{b-a} \right) dy - \int_{a}^{b} \phi_{m,k}^{j} \left( (j+1) \frac{y-a}{b-a} \right) dy \right]. \]

Both of the integrals can be solved analytically using basic calculus, and for a proof and the expression, the reader is referred to [OGO13].
3.6 SWIFT-method

The SWIFT-method, introduced by Ortiz-Gracia and Oosterlee in [OGO15], is similar to the WA[^a,b] method in methodology; the main novelty is the type of wavelet; the so-called Shannon wavelet is used, resulting in its name, the ‘Shannon Wavelets Inverse Fourier Technique’ (SWIFT) method.

Following the MRA framework, the truncated Shannon wavelet expansion of the density function \( f(y|x) \) is given by,

\[
f(y|x) \approx \mathcal{P}_m f(y|x) = \sum_{k \in \mathbb{Z}} D_{m,k}(x) \phi_{m,k}(y),
\]

(3.31)

with \( D_{m,k}(x) := \int_{\mathbb{R}} f(y|x) \phi_{m,k}(y) \, dy \),

where the scaling functions are defined from \( \phi(x) = \text{sinc}(x) \) for a fixed wavelet scale \( m \in \mathbb{N} \).

Since Shannon wavelets have infinite support, we take a different approach in truncating the wavelet series. We note that for \( h \in \mathbb{Z} \),

\[
f\left( \frac{h}{2^m}, x \right) \approx \mathcal{P}_m f\left( \frac{h}{2^m}, x \right) = 2^{\frac{m}{2}} \sum_{k \in \mathbb{Z}} D_{m,k}(x) \delta_{k,h} = 2^{\frac{m}{2}} D_{m,h}(x).
\]

Now, since \( f \in L^2(\mathbb{R}) \) and it is non-negative, and if we assume that \( \lim_{x \to \pm \infty} f(x) = 0 \) then we conclude that \( D_{m,k} \) vanishes as well as \( k \to \pm \infty \).

We therefore approximate the infinite series in Equation (3.31) by a finite summation without loss of considerable density mass,

\[
f(y|x) \approx f_m(y|x) := \sum_{k=k_1}^{k_2} D_{m,k}(x) \phi_{m,k}(y),
\]

(3.32)

for conveniently chosen integers \( k_1 < k_2 \). When setting \( \mathcal{I}_m = \left[ \frac{k_1}{2^m}, \frac{k_2}{2^m} \right] \), the option pricing formula becomes,

\[
v(x,t) = e^{-r(T-t)} \int_{\mathbb{R}} v(y,T) f(y|x) \, dy
\]

\[
\approx e^{-r(T-t)} \sum_{k=k_1}^{k_2} D_{m,k}(x) V_{m,k},
\]

(3.33)

where the payoff coefficients are defined as,

\[
V_{m,k} := \int_{\mathcal{I}_m} v(y,T) \phi_{m,k}(y) \, dy.
\]

(3.34)

We define the truncation parameters \( k_1 \) and \( k_2 \), as the smallest integers (in absolute sense), such that when \( a \) and \( b \) are the truncation parameters of Section 3.3, determined by the cumulants of the density function, we have,

\[
k_1 \leq a < b \leq k_2,
\]

which implies that \([a, b] \subset \mathcal{I}_m\).

3.6.1 Density Coefficients

In [OGO15], two methods are presented to compute the density coefficients \( D_{m,k} \) in Equation (3.31). We discuss the approach based on Vieta’s formula, as this approach allows us to control the numerical error.

**Theorem 3.6.** For \( J \in \mathbb{N} \), we can approximate the sinc function by,

\[
sinc(t) \approx \text{sinc}^*(t) := \frac{1}{2^{J-1}} \sum_{j=1}^{2^{J-1}} \cos \left( \frac{2j-1}{2^J} \pi t \right),
\]

(3.35)
where the absolute error is bounded by,

\[ |\text{sinc}(t) - \text{sinc}^\ast(t)| \leq \frac{(\pi e)^2}{2^{2j+1} - (\pi e)^2}, \tag{3.36} \]

for \( t \in [-c, c] \), where \( c \in \mathbb{R}, c > 0 \) and \( J \geq \log_2(\pi e) \).

**Proof.** We show how to find the expression for \( \text{sinc}^\ast(t) \). The proof of the error bound is Lemma 2 in [OGO15]. As shown by Vieta, and described in [GS90], the sinc function can be written as an infinite product,

\[ \text{sinc}(t) = \prod_{j=1}^{\infty} \cos \left( \frac{\pi t}{2^j} \right), \tag{3.37} \]

and by truncating the infinite product to a finite product with \( J \) factors, we can apply the cosine product-to-sum identity described in [QA13]. This gives the desired result,

\[ \text{sinc}(t) \approx \prod_{j=1}^{J} \cos \left( \frac{\pi t}{2^j} \right) = \frac{1}{2^{J-1}} \sum_{j=1}^{2^J-1} \cos \left( \frac{2j-1}{2^j} \pi t \right) =: \text{sinc}^\ast(t). \]

\[ \square \]

If we write out the definition of the coefficients \( D_{m,k} \) in (3.31), we get,

\[ D_{m,k}(x) = 2^m \int_\mathbb{R} f(y|x) \phi(2^m y - k) \, dy. \]

Using Vieta’s approximation \( \text{sinc}(x) \) by \( \text{sinc}^\ast(x) \) from Theorem 3.6 gives us,

\[ D_{m,k}(x) \approx D^\ast_{m,k}(x) := \frac{2^{m/2}}{2^{J-1}} \sum_{j=1}^{2^J-1} \int_\mathbb{R} f(y|x) \cos \left( \frac{2j-1}{2^j} \pi(2^m y - k) \right) \, dy. \]

We now note the resemblance between the integral at the right-hand side of the equation above and the integral in the COS method (3.20). In a similar way, we replace the integral over the unknown density function by its Fourier transform,

\[
\begin{align*}
\int_\mathbb{R} f(x) \cos \left( \frac{2j-1}{2^j} \pi(2^m x - k) \right) \, dx &= \text{Re} \left\{ \int_\mathbb{R} f(x) \exp \left( -i \frac{2j-1}{2^j} \pi(2^m x - k) \right) \, dx \right\} \\
&= \text{Re} \left\{ e^{i \frac{2j-1}{2^j} \pi k} \int_\mathbb{R} f(x) \exp \left( -i \frac{(2j-1)\pi 2^m x}{2^j} \right) \, dx \right\} \\
&= \text{Re} \left\{ e^{i \frac{2j-1}{2^j} \pi k} \hat{f} \left( \frac{(2j-1)\pi 2^m}{2^j} \right) \right\},
\end{align*}
\tag{3.38}
\]

Inserting this into the density coefficients gives us an expression for the density coefficients,

\[ D_{m,k}(x) \approx D^\ast_{m,k}(x) := \frac{2^{m/2}}{2^{J-1}} \sum_{j=1}^{2^J-1} \text{Re} \left\{ \hat{f} \left( \frac{(2j-1)\pi 2^m}{2^j} \right) e^{i \frac{k(2j-1)}{2^j}} \right\}. \tag{3.39} \]

A strategy for choosing \( J \) follows from Theorem 3.6, which implies that when we set \( M_{m,k} = \max(\|2^m a - k\|, \|2^m b + k\|) \) and \( M_m := \max_{k_1<k<k_2} M_{m,k} \). We set \( J := \lfloor \log_2(M_m) \rfloor \), where \( \lfloor x \rfloor \) denotes the smallest integer greater than or equal to \( x \). For a proof, the reader is referred to [OGO15].

Although for each \( k \) another \( J \) could be chosen, we decide to fix one \( J \) for all \( k \), so that we can benefit from the efficiency of the FFT algorithm to compute the vector of density coefficients \( \{D_{m,k}(x)\}_{k=1}^{k_2} \) at once, as described in [OGO15].

32
Density Coefficients by Parseval’s Identity

As mentioned before, thanks to the compact support of $\hat{\phi}_{m,k}$, we can also accurately compute the density coefficients by Parseval’s Identity.

**Theorem 3.7.** By Parseval’s Identity and the fact that the density coefficients $c_{m,k}$ in Equation (3.31) are real-valued coefficients, we find,

$$D_{m,k} = 2\pi \int_{-\frac{1}{2}}^{\frac{1}{2}} \Re \left\{ \hat{f}(2^{m+1}\pi t)e^{i2\pi kt} \right\} dt.$$

**Proof.** The proof of this theorem is Remark 2 in [OGO15].

By applying the Trapezoidal Rule to the representation of the density coefficients in Theorem 3.7, we can accurately compute these coefficients. When one chooses to discretize the domain $t = [-\frac{1}{2}, \frac{1}{2}]$ in $2^j$ steps of length $h = 1/2^j$, where $j$ is the same as the parameter in Vieta’s method, the computational complexity of Parseval’s method appears to be similar to Vieta’s method, with the equivalent rate of convergence.

### 3.6.2 Payoff coefficients

We show how to compute the payoff coefficients for a European call (or put), based on [OGO15]. In contrast to the COS and WA $[a,b]$ methods, we do not have an analytic expression for the payoff coefficients, but we can benefit from the FFT algorithm for an efficient approximation.

We look for an expression for the payoff of a European call. The steps for deriving a formula for the European put are similar. Recall that the payoff coefficients for a European call are defined by,

$$V_{m,k} = K \left[ \int_{\mathcal{I}_m} e^y \phi_{m,k}(y) \, dy - \int_{\mathcal{I}_m} \phi_{m,k}(y) \, dy \right],$$

as in Equation (3.34). Let us define,

$$I_{j,k}^1(a,b) := \int_a^b e^y \cos(\omega_j(2^m y - k)) \, dy,$$

and,

$$I_{j,k}^2(a,b) := \int_a^b \cos(\omega_j(2^m y - k)) \, dy,$$

where $\omega_j := \frac{2j-1}{2^m} \pi$. Note that these integrals represent just a change of variables of integrals we had to solve for the COS payoff coefficients. For a proof the reader is referred to [OGO15].

When $k_2 \leq 0$, the payoff coefficients vanish, i.e., $V_{m,k} = 0$ for every $k$. In case $0 < k_2$, we can write (3.40) as,

$$V_{m,k} \approx V_{m,k}^* := K \frac{2^m}{2^{j-1}} \sum_{j=1}^{2^{j-1}} \left[ I_{j,k}^1 \left( \frac{k_1}{2^m}, \frac{k_2}{2^m} \right) - I_{j,k}^2 \left( \frac{k_1}{2^m}, \frac{k_2}{2^m} \right) \right].$$

As in the case of the density coefficients, we consider a constant $J$, which we call $\bar{j}$, defined by $\bar{j} := \lceil \log_2(\pi N) \rceil$, where $N := \max_{k_1,k_2} N_k$ and $N_k := \max(|k_1-k|,|k_2-k|)$. This allows us to compute the whole vector of payoff coefficients $\{V_{m,k}^*\}_{k_1}^{k_2}$ with the help of the FFT algorithm [OGO15].

The resulting SWIFT pricing formula for European call options reads,

$$v(x,t) \approx e^{-r(T-t)} \sum_{k=k_1}^{k_2} D_{m,k}(x)V_{m,k}^*.$$

The payoff coefficients can be computed by means of the FFT, which is a slight reformulation of the approach described in [OGO15].

**Lemma 3.2.** The payoff coefficients $V_{m,k}^*$ as in (3.41) can be computed efficiently with a computational complexity of $O(N \log_2 N)$ using the FFT.
Proof. Let \( \omega_j := \frac{2j-1}{2^j} \pi \), and define \( B_j := \frac{1}{\omega_j^{2m}} \) and \( A_j := \frac{\omega_j^{2m}}{1+\omega_j^{2m}} \), then let,

\[
I_j^1(a, b) := -ie^{b+\omega_j^{2m}b} + ie^{a+\omega_j^{2m}a} + B_je^{b+\omega_j^{2m}b} - B_je^{a+\omega_j^{2m}a},
\]

\[
I_j^2(a, b) := -ie^{\omega_j^{2m}b} + e^{\omega_j^{2m}a},
\]

and define the vector \( \mathbf{X}(a, b) \) with entries given by \( X_j(a, b) := A_jI_j^1(a, b) - B_jI_j^2(a, b) \), where the overline denotes the complex conjugate. Then,

\[
\sum_{j=1}^{2^j-1} [I_{j,k}^1(a, b) - I_{j,k}^2(a, b)] = \text{Re} \left\{ \sum_{j=1}^{2^j-1} X_j(a, b)e^{\omega_j^{2m}k} \right\},
\]

which we can write in the required FFT-form by some basic manipulations, giving,

\[
\text{Re} \left\{ e^{i\pi \frac{k}{2^j}} \sum_{j=1}^{2^j} X_j(a, b)e^{-i\pi \frac{k}{2^j}(j-1)(k-1)} \right\} = \text{Re} \left\{ e^{i\pi \frac{k}{2^j}}D_k \{ \mathbf{X}(a, b) \} \right\},
\]

and the payoff coefficients (3.41) are given by,

\[
V^*_m = K \frac{2^m/2}{2^{j-1}} \text{Re} \left\{ e^{i\pi \frac{k}{2^j}}D_{1-k} \left\{ \mathbf{X} \left( \frac{k_1}{2^m}, \frac{k_2}{2^m} \right) \right\} \right\}.
\]

3.6.3 Pricing multiple strikes

As for the COS method, both the WA\(^{[a,b]}\) and SWIFT methods can be applied to efficiently price options at multiple strikes simultaneously. We describe the approach for the SWIFT method when the underlying process is a Lévy process or the Heston model. For these processes, we can apply the Shannon wavelet expansion to the density function \( f(y) \), instead of \( f(y|x) \),

\[
f(y|x) = f(y|x) = \sum_{k=k_1}^{k_2} D^*_m(k)x \phi_m(x - y),
\]

where \( D^*_m(x) \) are the density coefficients as in (3.39), evaluated at \( x = 0 \). The SWIFT option pricing formula then becomes,

\[
v(x, t) = e^{-r(T-t)} \sum_{k=k_1}^{k_2} D^*_m(k) V^*_m(x),
\]

where \( V^*_m(x) := \int_{I_m} v(y, T) \phi_m(y - x) dy \).

Compared to the original SWIFT pricing formula (3.42), the dependence on \( x \) has been moved from the density coefficients to the payoff coefficients. The density coefficients have to be computed only once. The payoff coefficients now depend on \( x \), and they are generally cheaper to compute, especially for the WA\(^{[a,b]}\) method.

3.7 Numerical Results

In this section, we compare the different methods. The COS method is a very competitive method, both in terms of speed and accuracy. Especially for the Black-Scholes-Merton (BSM) and other light-tailed distributions, convergence of the COS method is extremely fast, independent of essentially all problem parameters like the strike, volatility, maturity, etc. This is shown in Example 5. The WA\(^{[a,b]}\) method offers more flexibility and robustness, but due to its linear convergence it cannot in general compete in terms of speed with the exponential convergence of the COS method. The SWIFT method shows a similar fast convergence, although the method is more expensive in terms of CPU time.
Example 5 (Rate of convergence for GBM European call). We show that the methods based on B-splines cannot compete against the rate of convergence of the SWIFT and COS method for the GBM model. We price a European call on GBM with parameters,

\[ S_0 = 100, \sigma = 0.15, r = 0.1, T = 1, \text{ and } K = 110. \]

In Figure 3.2, the convergence results in terms of the number of coefficients is given. For the WA\([a,b]\) method, the number of coefficients is \(N_{WA}^j = (j + 1)(2^m - 1)\). For the SWIFT method, this is \(N_{Sw}^m := k_2 - k_1 + 1 \approx 2^m(b - a)\).

Rapid convergence is observable for the COS and SWIFT methods, where the WA\([a,b]\) method converges slower.

![Figure 3.2: The error convergence in terms of the number of coefficients for different methods in the pricing of a European call on GBM.](image)

The coefficients of the COS method are cheapest to compute, as both the payoff and density coefficients can be computed analytically. For the coefficients of the WA\([a,b]\) method, only the payoff coefficients can be computed analytically, but the density coefficients can be efficiently computed using FFT. The coefficients of the SWIFT method are the most expensive to compute, as the FFT is required for both the payoff and density coefficients.

Thus, the WA\([a,b]\) method can only compete against SWIFT and COS when it would need fewer coefficients. This is the case for example when pricing call options with long maturities, which correspond to pension or mortgage contracts.

Example 6 (Long Maturity Options under Heston). In this section, we repeat the example from [OGO13], where we focus on the Heston model [Hes93], see Appendix A.4.

In Figure 3.3, a long-maturity call is priced on the Heston model with parameters \(T = 45; S = 100; K = 100; r = 0; \lambda = 1.5768; \eta = 0.5751; \bar{u} = 0.0398; u = 0.0175; \rho = -0.5711; \). We observe linear convergence for all of the four methods.

Remark 3.2. In the case of the pricing of \(K\) simultaneous strikes, the WA method has an advantage over SWIFT, as the payoff coefficients can be computed analytically, thus a computational complexity of \(O(KN + N \log N)\) can be achieved, while the SWIFT method achieves \(O((K + 1)N \log N)\), and the COS method has \(O(KN)\). Thus, when \(K\) is big and the COS method needs many coefficients, the WA\([a,b]\) method can outperform COS, as shown in [Kir15].

The WA\([a,b]\) method is also attractive due to its robustness. Since wavelets series represent functions locally, we can easily adjust coefficients to match local difficulties. We demonstrate this robustness in Example 7 by pricing a very long maturity call option, which arises in economy and real options. Due to the unbounded payoff of a call option, extreme payoffs occur on the right hand side of the computation domain.

Example 7 (Robustness of WA\([a,b]\)). We price a European call with a very long \(T = 100\) maturity. Since the COS method uses a global basis, all coefficients are affected by the extreme payoff on the right hand side of this domain, and as is shown in Figure 3.4(left), payoff coefficients of magnitude \(10^{12}\) are multiplied by density coefficients of magnitude \(10^{-15}\), causing such round-off errors that the final option value has an absolute error of \(10^{-4}\).
3. EUROPEAN OPTION PRICING METHODS

Figure 3.3: Long-maturity European call option on Heston with parameters $T = 45; S = 100; K = 100; r = 0; \lambda = 1.5768; \eta = 0.5751; \bar{u} = 0.0398; \bar{u} = 0.0175; \rho = -0.5711$. Linear convergence is observed for all of the methods. Reference price by the COS method with $N = 50000$, Reference val. = 46.91153664...

We price the same long-maturity call using $WA^{[a,b]}$ with the Haar basis and scale $m = 5$ (32 coefficients), and the coefficients are shown in Figure 3.4(right). Wavelets form a local basis and, as can be seen from the figure, each coefficient $D^+_{m,k}$ only affects the points of the density locally, in the interval $\left[\frac{k}{2^m}, \frac{k+1}{2^m}\right]$. We can avoid big round-off errors by removing the payoff coefficients that cause very big round-off errors at the right hand side of the domain. We therefore consider the truncated series,

$$v_{\kappa_m}(x,t) := e^{-r(T-t)} \sum_{k=0}^{K_m} D^+_{m,k}(x)V_{m,k},$$

and by choosing $\kappa_m$ such that $v_{\kappa_m}(x,t) < S_0$, using that $S_0$ is an upper bound for the value of a call, we find an error of about $10^{-1}$.

Figure 3.4: Coefficients for from COS ($N = 32$) and $WA^{[a,b]}$ (Haar, $m = 5$) methods arising from the pricing of a long-maturity call on GBM with $S_0 = 100, K = 100, r = 0.1, q = 0, \sigma = 0.25, T = 100$ and $L = 10$. Reference price by the Black-Scholes formula.

One final difference we highlight before looking in more detail at the SWIFT method is how the density function is approximated. Density functions are generally very regular functions, so that approximating them with the non-smooth $B$-splines is less effective than approximating them with very smooth cosines or the sinc function, as we see in Figure 3.5.
Figure 3.5: Approximation error in the approximation of the probability density function of a CMGY driven random variable with parameters $C = 1, G = 5, M = 5, Y = 0.7, r = 0.05$ and $T = 0.01$. We truncate the integration domain to $[a, b]$ by the cumulants method with $L = 25$. The reference pdf is found by the COS method with $2^{18}$ coefficients.
Chapter 4

Analysis of the SWIFT Method

In this chapter, we further analyze the SWIFT method. In [OGO15], the SWIFT method was introduced for the first time, and when trying to extend it to the pricing of Bermudan options, we found a number of interesting properties that improve robustness, accuracy and speed.

The main contributions of this chapter are a new approach to compute payoff coefficients without truncating the integration range, heuristics to estimate all of the SWIFT parameters a priori and a new approach in multiple-strike pricing.

Furthermore, we perform an extensive analysis of the boundary behavior of the SWIFT method, and introduce the SWIFT-Whittaker method.

For the SWIFT method, there are two basic considerations to be made that determine a density approximation. The first choice is the resolution or scale $m$ of the wavelet approximation. Higher resolutions imply the ability to capture finer features of a density function, such as extreme peaks.

In theory, the wavelet approximation approximates the density function on the entire real line in terms of an infinite dimensional basis. In practice, finitely many basis functions are used, so the second consideration is to truncate the support. We note that in contrast to the COS method, the underlying density function itself is not truncated, we merely project it onto a space with a finite dimensional basis.

The simplicity of the Shannon wavelet in the Fourier domain gives us a hint on how to select a proper resolution $m$, which we demonstrate in Section 4.2. We also discuss a test to determine the suitability of the truncated domain by means of tail estimation.

We start with a reformulation of the SWIFT method such that we can individually choose $m$ and the corresponding domain truncation. The starting point is again the pricing of a European option contract, but we limit ourselves to the case where the option payoff is bounded, i.e., we assume that the payoff of the option satisfies $\|v(\cdot, T)\|_{\infty} = B$ for some real positive constant $B$.

The Multi Resolution (MRA) framework allows us to approximate a function on the whole real line, contrasting Fourier expansions, which are restricted to a bounded interval on the real line. Let the underlying conditional density function $f(y|x)$ be a function in $L^2(\mathbb{R}) \cap L^1(\mathbb{R})$. Then, its wavelet projection of resolution $m$ is given by,

$$f(y|x) \approx f_1(y|x) := P_m f(y|x) = \sum_{k \in \mathbb{Z}} D_{m,k}(x) \phi_{m,k}(y),$$

where the density coefficients are defined as $D_{m,k}(x) = \langle f(\cdot|x), \phi_{m,k} \rangle$. Subsequently, we truncate the infinite summation to find,

$$f_1(y|x) \approx f_2(y|x) := \sum_{k=1}^{k_2} D_{m,k}(x) \phi_{m,k}(y) dy.$$

The density coefficients are unchanged from the original SWIFT formulation, and are approximated using Vieta’s formula as in Section 3.6.1, i.e.,

$$D_{m,k}(x) \approx D^*_m(x) := \frac{2^{m/2}}{2^{j-1}} \sum_{j=1}^{2^{j-1}-1} \text{Re} \left\{ \hat{f} \left( \frac{(2j-1)\pi 2^m}{2^j}; x \right) e^{i k \pi (2j-1)} \right\}. \quad (4.1)$$
4.1 Payoff Coefficients

In this section, we discuss how to compute the payoff coefficients without truncating the infinite integration range, which we illustrate for a Cash-or-Nothing option and a vanilla put.

4.1.1 Cash-or-Nothing option

Cash-or-nothing options are priced using the SWIFT method in [OGO15], with a payoff function given by

\[ v(y, T) = \mathbb{1}_{(0,\infty)}(y). \]

Then, by the pricing strategy proposed, we apply a change of variables \( t := 2^m y - k \) to the payoff coefficients, resulting in,

\[ V_{m,k} = \int_{\mathbb{R}} \mathbb{1}_{(0,\infty)}(y) \phi_{m,k}(y) \, dy \]
\[ = \int_{0}^{\infty} \phi_{m,k}(y) \, dy \]
\[ = 2^m \int_{0}^{\infty} \text{sinc}(2^m y - k) \, dy \]
\[ = 2^{-m} \int_{-k}^{\infty} \text{sinc}(t) \, dt. \]

Now, we split this last integral, finding,

\[ V_{m,k} = 2^{-\frac{m}{2}} \left( 1 + \int_{0}^{k} \text{sinc}(t) \, dt \right). \]
where the remaining integral, known as the Sine integral $\text{Si}(k)$, is approximated using Vieta’s formula,
\[
\int_0^k \text{sinc}(t) \, dt \approx \frac{2}{\pi} \sum_{j=1}^{2^{J-1}} \frac{1}{2j-1} \sin \left( \frac{2j-1}{2^J} \pi k \right),
\]
and the whole vector of payoff coefficients can be computed using the FFT.

**Remark 4.1.** Using the error bound in Vieta’s formula we can derive that a suitable truncation $J$ is given by $J = \lfloor \log_2(\pi M) \rfloor$ where $M = \max(|k_1|, |k_2|)$. [OGO15].

### 4.1.2 European put

The same splitting argument as for the Cash-or-Nothing option can be used to compute the payoff coefficients of a European put option with payoff $v(y, T) = K(1 - e^y)^+$, with resulting coefficients,
\[
V_{m,k} = K \int_{-\infty}^{0} (1 - e^y) \phi_{m,k}(y) \, dy
\]
\[
= 2^{-\frac{N}{2}} K \int_{-\infty}^{0} (1 - e^y) \text{sinc}(2^m y - k) \, dy
\]
\[
= 2^{-\frac{N}{2}} K \int_{-\infty}^{-k} \left(1 - e^{\frac{y+k}{2^m}}\right) \text{sinc}(t) \, dt
\]
\[
= 2^{-\frac{N}{2}} K \int_{-\infty}^{0} \left(1 - e^{\frac{y+k}{2^m}}\right) \text{sinc}(t) \, dt - 2^{-\frac{N}{2}} K \int_{-k}^{0} \left(1 - e^{\frac{y+k}{2^m}}\right) \text{sinc}(t) \, dt.
\]

and by using the analytic solution of the first integral, we find,
\[
V_{m,k} = 2^{-\frac{N}{2}} K \left( \frac{1}{2} + e^{k/2^m} \arctan(2^m y)/\pi - 2^{-\frac{N}{2}} \int_0^{k/2^m} (1 - e^y) \phi_{m,k}(y) \, dy \right),
\]
and the remaining integral is similar to the original formulation of the option pricing problem as we saw in Section 3.6.2. However, we should take care, since the integration range now depends on $k$. Applying Vieta’s again results in,
\[
\int_a^{k/2^m} (1 - e^y) \phi_{m,k}(y) \, dy \approx \frac{2^m/2}{2^{J-1}} \sum_{j=1}^{2^{J-1}} I_{j,k}(0),
\]
where we solve the integral $I_{j,k}(a)$ analytically. Let $c_j := \frac{2j-1}{2^m} \pi$, $A_j = \frac{1}{1+1/(c_j 2^m)}$ and $B_j = \frac{1}{\pi c_j 2^m}$ then,
\[
I_{j,k}(a) := \int_a^{k/2^m} (1 - e^y) \cos(c_j(2^m y - k)) \, dy
\]
\[
= \text{Re} \left\{ B_j e^{a} - A_j e^{(2^m a-k)} - A_j e^{k/2^m} \right\}.
\]

We can once more apply the FFT to compute the vector of payoff coefficients at once.

**Remark 4.2.** This approach is robust, i.e., no domain truncation is required. However, due to the exponentially growing term $e^{k/2^m}$ that appears in this approach for positive $k$, we need $J \geq \log_2(\pi N) e^{M/2^m}$, where $N := k_2 - k_1 + 1$ and $M := \max(|k_1|, |k_2|)$. Due to the large value for $J$, this approach will be unattractive for fat-tailed distributions.

As a computationally faster alternative, we propose to truncate the integral in (4.5) to a large domain, i.e., use $V_{m,k} \approx \int_a^b g(y) \phi_{m,k}(y) \, dy$ with $[a, b] := [2k_1/2^m, 0]$.

### 4.1.3 European call

The payoff function of a European call option is unbounded, as it is given by $v_{\text{Call}}(y, T) = (e^y - K)^+$, thus applying the strategy from the previous section will result in undefined payoff coefficients, as,
\[
\int_{\mathbb{R}} v_{\text{Call}}(y, T) \phi_{m,k}(y) \, dy = \infty.
\]
To price European call options with higher accuracy, one should therefore price it through a put and then use the put-call parity.

### 4.1.4 Unbounded Payoff

In case of an option with unbounded payoff, assuming that the option value is finite, we one should truncate the payoff as follows. We assume that \( f(y|x) \) vanishes for \( y \) outside a domain \([a + x, b + x]\). Then, the option value can be written as,

\[
v(x, t) = e^{-r(T-t)} \int_{\mathbb{R}} f(y|x)v(y, T)\mathbbm{1}_{[a,b]}(y-x) \, dy
\]

\[
\approx \sum_{k=k_1}^{k_2} D_{m,k}^*(0) \int_{\mathbb{R}} v(y, T)\phi_{m,k}(y-x)\mathbbm{1}_{[a,b]}(y-x) \, dy.
\]

This approach is equivalent to the original SWIFT formulation [OGO15], where it is implicitly used that the integration range truncation should depend on \( x \). In our approach we highlight that the integration range in the payoff coefficients,

\[
V_{m,k}(x) := \int_{a+x}^{b+x} v(y, T)\phi_{m,k}(y-x) \, dy = \int_a^b v(y-x, T)\phi_{m,k}(y) \, dy,
\]

should be adjusted for \( x \).

In the old SWIFT formulation, the first approximation made was to truncate the integration range of the risk neutral option pricing formula. Therefore, both the payoff and density coefficients depend on this truncation. In the new SWIFT formulation, we do not have such a truncation of the integration range. In contrast, we truncate the summation range \( k \), but by increasing the range of \( k \), the coefficients do not change, we only have more of them.

### 4.2 Local error analysis

In this section, we discuss the errors introduced by the new formulation of the SWIFT method. We start this section with a local error analysis, i.e., we write the final pricing error of SWIFT for a fixed \( x \) (and \( t \)) as,

\[
|v(x, t) - v_4(x, t)| \leq |v(x, t) - v_3(x, t)| + |v_3(x, t) - v_4(x, t)|,
\]

where \( v_4(x, t) \) is given in (4.4) and \( v_3(x, t) \) in (4.3). The second term, \(|v_3(x, t) - v_4(x, t)|\), is related to the payoff function and depends on the approximation of the sinc function by Vieta’s formula and can be made arbitrarily small as a suitable truncation \( J_p \) is given by \( J_p = \lfloor \log_2(\pi M) \rfloor \), where \( M = \max(|k_1|, |k_2|) \), as explained in Section 4.1.

The first term, \(|v(x, t) - v_3(x, t)|\), is related to the approximation of the density coefficients, and can be written as,

\[
|v(x, t) - v_3(x, t)| \leq e^{-r(T-t)} \|v\|_{\infty} \|f - f_3\|_1 = e^{-r(T-t)} B \|f - f_3\|_1.
\]

Thus, we ought to find convergence of \( f_3 \) to \( f \) in the \( L^1(\mathbb{R}) \)-norm. For this purpose we split the error into three parts,

\[
\|f - f_3\|_1 \leq \|f - f_1\|_1 + \|f_1 - f_2\|_1 + \|f_2 - f_3\|_1 =: \varepsilon_1(m) + \varepsilon_2(k_1, k_2) + \varepsilon_3(J_d).
\]

In this way, we split up the error into three distinct parts, of which each can be controlled by a single parameter, allowing us to fix the suitable parameters one-by-one.

We have proven \( L^1(\mathbb{R}) \)-convergence within the MRA framework in Section 2.3. From Lemma 2.1, it follows that \( \varepsilon_1(m) = \|f - f_1\|_1 \to 0 \) when \( m \to \infty \). We now derive a method to choose \( m \).
4.2.1 Wavelet scale \( m \)

From Lemma 2.2, we recall that if \( f \) is a bandlimited function with bandlimit \( B < 2^m \), the function can be exactly replicated with a Shannon wavelet series at resolution \( m \). The probability density functions that often occur in finance are not bandlimited, but the characteristic function \( \hat{f}(\omega) \) does vanish for \( |\omega| \to \infty \). This suggests that there is an \( m \) value such that \( |\hat{f}(\omega)| < TOL \) for all \( |\omega| > 2^m \pi \), where \( TOL \) is a user defined tolerance. When selecting an \( m \) such that \( |\hat{f}(\omega)| < TOL \) holds for all \( |\omega| > 2^m \pi \), we can heuristically say that \( f \) is ‘close’ to a bandlimited function with bandlimit \( 2^m \), and thus approximated well at scale \( m \).

Let us illustrate this with an example.

Example 8 (Wavelet resolution for GBM). Consider the pricing of a European put with strike \( K \) on an asset driven by GBM.

The GBM characteristic function is given by \( \hat{f}(\omega;\sigma,T,r,q) = e^{T(i\omega\mu+\psi_L(\omega))} \), where \( \psi_L(\omega) = -\frac{\sigma^2}{4} \omega^2 \) and \( \mu = r - q - \psi_L(-i) \). In this case, the \( m \) value that solves \( |\hat{f}(\omega)| < TOL \) for all \( |\omega| > 2^m \pi \) can be found analytically and is given by \( m(TOL) = \log_2 \left( \frac{1}{2} \sqrt{-2 \log(TOL)/(\sigma^2)} \right) \).

Now, when large values are chosen for \( k_1, k_2 \) and \( J_d \), the error in the option price by the SWIFT method is dominated by \( \varepsilon_3(m(TOL)) \), and this can be controlled purely by the value for \( TOL \).

In Figure 4.1, the option price error is plotted as a function of \( TOL \), where \( TOL = 10^{-d} \), with \( d = 1 : 15 \). We plotted both the option pricing error as a function of \( m(TOL) \) and of \( \lceil m(TOL) \rceil \). We observe perfect predictability of the resulting option error, with the exception of some round-off errors close to machine accuracy.

![SWIFT GBM European Put Price error as function of Tolerance Level](image)

Figure 4.1: Pricing error with the SWIFT method for a GBM European put as a function of \( TOL \). Parameters: \( S = 100, K = 100, \sigma = 0.25, r = 0.1, T = 1, L = 10 \). The values for \( \lceil m \rceil \) are \( m = \{2, 3, 4\} \).

In the next example, we repeat this procedure of selecting an appropriate value for \( m \), but with CGMY as underlying model.

Example 9 (Wavelet resolution for CGMY). For CGMY, the characteristic exponent is given by,

\[ \psi_L(\omega) = C T(-Y) \left( (M - i\omega)^Y - M^Y + (G + i\omega)^Y - G^Y \right) . \]

Thus, to find the \( \omega \) where \( |\hat{f}(\omega)| < TOL \), we have to solve,

\[ \text{Re} \{ \psi_L(\omega) \} = \frac{1}{T} \log(TOL). \]

This is not as easy to solve analytically as the GBM case in the previous example. We could apply Newton’s algorithm to approximate the solution, but since \( m \) is limited to the natural numbers, we select \( m \) by trial and error, i.e., we plug in \( m = 0, 1, 2, \ldots \), until \( |\hat{f}(2^m \pi)| < TOL \) is satisfied. The results are shown in Figure 4.2 for both \( Y = 0.5 \) (left) and \( Y = 1.5 \) (right). We plotted the resulting option price error as a function of \( TOL \). The corresponding scale \( m \) is selected by iterating over \( m = 1, 2, \ldots \) (yellow line) and \( m = 0.01, 0.02, \ldots \) (red line).
4.2. LOCAL ERROR ANALYSIS

Figure 4.2: Pricing error for a CGMY European Put with respect to TOL with the borderless SWIFT. Parameters: \( S = 100, K = 100, \sigma = 0, r = 0.1, T = 1, C = 1, G = 5, M = 5, Y = 0.5, 1.5, L = 20. \) "m Natural" corresponds to \( m = \{0, \ldots, 5\} \) (left) and \( m = \{0, 1\} \) (right).

4.2.2 Truncation of \( k \)

The next error we look into is \( \varepsilon_2(k_1, k_2) \), i.e., the error introduced by the truncation of the summation range \( k \). This truncation to \((k_1, k_2)\) is the only parameter of which we cannot control the error in advance. However, we will provide a test to check a priori whether the truncation is appropriate.

We use the Shannon-Whittaker interpolation polynomial to get insight in how to choose \( k_1 \) and \( k_2 \). To justify the truncation we used that \( D_{m,k} \approx 2^k f(k/2^m) \). Since \( f(y|x) \) vanishes for \( y \to \pm \infty \), the coefficients \( D_{m,k} \) vanish accordingly.

Thus, truncation should be performed on the region \( y \) where \( |f(y|x)| < TOL \). However, we do not have a tool to find the domain where this inequality holds when only the characteristic function is known. We therefore use the cumulants method proposed by the COS method, as in Section 3.3, i.e., we determine an interval,

\[
[a, b] = [c_1 - L \sqrt{c_2 + c_4}, c_1 + L \sqrt{c_2 + c_4}],
\]

where \( c_i \) are the cumulants of the underlying density. Then, we set \( k_1 := \lfloor a/2^m \rfloor \) and \( k_2 := \lfloor b/2^m \rfloor \).

It is important that \([a, b]\) captures enough mass of the underlying density function, and we can perform two simple tests to check whether this is true. Our first option is to directly integrate \( f_2 \), which gives,

\[
H_1(L) := \left| 1 - \int_{\mathbb{R}} f_2(y|x) \, dy \right| = \left| 1 - \sum_{k = k_1}^{k_2} D_{m,k} \right| \leq 1 - \sum_{k = k_1}^{k_2} |D_{m,k}| < TOL \|v\|^{-1}. \tag{4.8}
\]

A second test is based on \( D_{m,k} \approx 2^\frac{k}{2^m} f(k/2^m) \), thus we could check whether,

\[
H_2(L) := |D_{m,k_1}| + |D_{m,k_2}| < TOL \|v\|^{-1}.
\]

We test how well both indicators, \( H_1(L) \) and \( H_2(L) \), perform in predicting the error for a European put under GBM and CGMY. Parameters are the same as used for Figures 4.1 and 4.2. The price convergence with respect to \( L \) is shown in Figure 4.3, as well as the indicators. We see that \( H_1 \) is a better predictor, but \( H_2 \) gives a useful upper bound to the error and is cheaper to evaluate.

Remark 4.3 (Iteration over \( L \)). To compute the correct value for \( L \) we make use of the fact that the option price error decays very predictable, as can be seen by the very straight lines in Figure 4.3. First, we select \([a, b]\) by the cumulants for \( L = 8 \). This is the recommended value by [FO08]. Then we set \((k_1, k_2) := ([a/2^m], [b/2^m])\) and compute \( H_1(L) \). If it does not satisfy (4.8), we compute the pair \((k_1, k_2)\) corresponding to a smaller value of \( L \), say \( L = 6 \) and compute \( H_1(6) \), which can be done using the already computed density coefficients. Then we extrapolate to find the \( L \) corresponding to the desired tolerance \( TOL \) and we compute the corresponding pair \((k_1, k_2)\) and \( D_{m,k} \) once more. In this way, the computational complexity is just two times the FFT complexity.
Figure 4.3: Test $H_1$ and $H_2$ to predict pricing error for a European put with parameters as in Figures 4.1 and 4.2 with desired $TOL = 10^{-14}$.

The third and only remaining error, $\varepsilon_3(J_d)$, depends on the difference between $D_{m,k}$ and $D_{m,k}^*$. It is shown in [OGO15] that this error vanishes when $J_d$ goes to infinity, and a proper $J_d$ is given by

$$J_d := \lceil \log_2(\pi M) \rceil,$$

where $M = \max(|k_1|, |k_2|)$.

Using the proposed rule of thumb for the wavelet scale $m$ and the cumulants method in combination with check for the domain truncation $(k_1, k_2)$ makes that we can determine appropriate values for all the required parameters of the SWIFT method a priori.

One remaining and interesting error is the spatial boundary error, which is prominent for the COS method, and plays an important role in the pricing of Bermudan options.

### 4.3 Spatial Boundary Error

In the error analysis above, we considered the pointwise convergence of the final option value for a fixed $x$ value. In this section, we consider the pricing for a range of $x$ values. For the COS method, some errors occur along the boundary of the integration interval. We first describe the problem with the COS method, and then show that we do not suffer from this problem.

These boundary issues for the COS method are described by M. Ruijter in [ROA13], and are a consequence of the periodicity of the cosine basis. The COS method depends heavily on a predetermined domain $[a, b]$. This domain, typically determined by the cumulants of the characteristic function, determines which basis is used in the approximation, i.e.,

$$\left\{ \cos \left( k \pi \frac{x - a}{b - a} \right) \right\}_{k \in \mathbb{N}}.$$

These basis functions are used to compute the density coefficients as well as the payoff coefficients. When, along the process, one determines that the domain $[a, b]$ is insufficient, and should thus be increased, the basis changes and therefore all coefficients have to be recomputed.

Even for a properly chosen interval $[a, b]$, errors occur along the boundary of this domain. This is not a problem when pricing European options since one is interested only in the option value at $x_0$, which is
often ‘safely’ chosen to be in the middle of \([a, b]\). However, in stochastic control problems, the pricing of Bermudan options and multiple strike pricing, these boundary errors might become significant.

### 4.3.1 Boundary errors for call options

Let us replicate the example given by M. Ruijter [ROA13], for both the COS and SWIFT methods. We price a European call option driven by GBM with a payoff function,

\[ g(y) = (e^y - K)^+ \]

where the state variable \( y \) is defined as \( y = \log(S_T) \). (Note that \( y \) is not scaled by the strike). For the call option under geometric Brownian motion, the analytic solution is available, i.e., the Black-Scholes price, against which we compare our numerical option value. The parameters we use are

\[ K = 100, S_0 = 100, (x_0 \approx 4.6), r = 0.1, T = 0.1, \sigma = 0.25, L = 10. \]

The option pricing formula is given by

\[
v(x_0, t) = e^{-r(T-t)} \int_{\mathbb{R}} g(y) f(y|x_0) \, dy
\]

and thus, the option value, with respect to \( x \), represents a similarity with convolutions. In fact, it is simply the convolution between \((-f)\) and \(g\). Thus, to price an option for a different \( x_0 \), the density is shifted along the spatial axis. This is represented in Figure 4.4.

![Figure 4.4: The European call payoff function (scaled to fit the image) and density function on the computation domain \([a, b]\) = \([4, 5]\), and its behavior outside this domain. Due to the even periodic extension of the payoff approximated by the COS method, option prices are low biased.](image)

The replication of the payoff function by the COS method is significantly lower than the true payoff. Thus we can expect undervaluation of the option value for a large initial value \( x \), i.e., near the the right side boundary of the spatial domain. This is shown in Figure 4.5.
4. ANALYSIS OF THE SWIFT METHOD

In multiple strike pricing, the boundary problem is not disastrous, as we can simply increase the domain side \([a, b]\). However, when pricing Bermudan options, error propagation (and CPU time) remains a problem.

A remedy for the COS method is proposed by M.Ruiter [ROA13], who explored different types of extrapolations to the Fourier cosine coefficients.

Figure 4.5: GBM European call priced on the domain \([a, b] = [4, 5]\) with the COS method with \(N = 2^{10}\). The error at \(x_0 = \log(S_0)\) is \(10^{-12}\), but is significant at the boundaries, even within the computation domain.

We replicate the call option pricing example with the SWIFT method. To get the most accurate results, one should price this call option through a put option and employ the put-call parity. However, we illustrate here that it is straightforward to solve boundary issues with the SWIFT method, even for option contracts with unbounded payoffs, as described in Section 4.1. Point of departure is the formulation of the SWIFT pricing formula in (4.6) and payoff coefficients \(V_{m,k}(x)\) as defined in (4.7). For each \(x \in [a, b]\) we evaluate, we recompute the whole set of payoff coefficients. Therefore, we adjust the integration range of the payoff coefficients for each \(x\). This is something that cannot be done with the COS method, as this changes the basis functions as well.

This approach is shown in Figure 4.6, where \([a, b] = [4, 5]\), and where we see that COS exhibits boundary issues on the right hand side of the domain, the accuracy of the SWIFT method is not bounded by the domain.

Figure 4.6: GBM European call priced on the domain \([a, b] = [4, 5.55]\) with the SWIFT (\(m = 5\)) and COS method (\(N = 2^{10}\)).

An advantage of the SWIFT method over the COS method is that the domain truncation for the payoff coefficients is independent of the number of coefficients. The only additional costs lie in the fact that when the integral in \(V_k(T)\) is approximated by Vieta’s formula, and that when the integration range
is bigger, a larger value for $J_d$ should be chosen to accurately compute all the payoff coefficients at once using the FFT.

### 4.3.2 Boundary Errors for put options

In this section, we check the boundary behavior for a European put option. As mentioned in the previous section, the exponentially growing payoff function of a call option causes significant round off errors when multiplying density coefficients that are almost zero with huge payoff coefficients. This is an issue where the bounded payoff of the put option does not suffer from.

The COS method has also an advantage when pricing put options, since the payoff function is nearly constant along both the boundaries of the domain, the periodic extension of the payoff is *not that unsatisfactory* for put options as it is for call options.

For the SWIFT method, we use the derivation of the put payoff coefficients from the previous section. The underlying asset model is CGMY, with parameters as in Table 4.1. For the SWIFT method, we observe a constant accuracy along the complete computational domain in Figure 4.7.

<table>
<thead>
<tr>
<th>Model</th>
<th>$S_0$</th>
<th>$K$</th>
<th>$T$</th>
<th>$r$</th>
<th>$\sigma$</th>
<th>Other Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGMY</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>$C = 1, G = 5, \alpha = 5, Y = 0.5$</td>
</tr>
</tbody>
</table>

Table 4.1: Test parameters for pricing a European Put

![Graph](image)

**Figure 4.7:** CGMY European put priced on the domain $[a, b] \approx [-6.5, 6.5]$ with the SWIFT ($m = 5$) and COS method ($N = 2^{10}$).

### 4.3.3 Domain Comparison with COS

Both the COS and SWIFT methods require a truncation of the log-asset domain. However, the impact of this truncation is different on both methods. In this section, we compare the required domain for both methods.

We select the domain $[a, b]$ with the cumulants method, such that,

$$ [a, b] := [c_1 - L \sqrt{c_2 + \sqrt{c_4}}, c_1 + L \sqrt{c_2 + \sqrt{c_4}}] $$

where $c_i$ are the cumulants of the underlying density. And we set $k_1 := \lfloor a/2^m \rfloor$ and $k_2 := \lceil b/2^m \rceil$. We can now write both option pricing formulas as a function of the parameter $L$, and price convergence is shown in Figure 4.8. We have chosen large values for $m$ and $N$, such that the error is dominated by the log-asset domain truncation.
At first sight, it seems remarkable that the SWIFT method requires a bigger parameter $L$ to obtain the same accuracy as the COS method. However, this behavior can be explained by noting that the Shannon wavelet has non-compact support. Thus each of the coefficients $\phi_{m,k}$ influences the function-approximation on $[a,b]$, even if $2^{-m}k \notin [a,b]$.

Until now, we only focused on the impact of the truncation $(k_1, k_2)$ to the density approximation, but also the payoff approximation is influenced. We slightly reorder the approximations we make to obtain the SWIFT pricing formula, to gain more insight.

The density function $f$ is approximated by $f_3$ as in (4.2), and we define the function $g_1$ as,

$$g_1(y) := \sum_{k=k_1}^{k_2} V_{m,k}^* \phi_{m,k}(y),$$

where $V_{m,k}^*$ are the payoff coefficients as in (4.4). Then, by the SWIFT option pricing formula, the option value can be written as,

$$v_4(x,t) = \sum_{k=k_1}^{k_2} D_{m,k}^*(x)V_{m,k}^* = \int_{\mathbb{R}} f_3(y|x)g_1(y) \, dy,$$

and thus we should have that $g_1(y) \approx g(y) = v(y,T)$. However, we see that truncation of the summation range $k$ causes so-called sinc wiggles at the non-zero boundary of the payoff function, which is shown in Figure 4.9. We recovered the payoff function of a put option by $g_1(y)$. The wiggles are caused by the truncated range $k = k_1, \ldots, k_2$. Due to the non-compact support of the Shannon wavelet, even the dismissed coefficients for $k \notin (k_1, \ldots, k_2)$ influence the function approximation on the interval of interest, i.e., $[a,b] \subset [2^{-m}k_1, 2^{-m}k_2]$.

Figure 4.9: The put payoff function in log-asset space. Recovered by a truncated SWIFT series expansion at scale $m = 5$. Wiggles are caused by the truncation of $k = k_1, \ldots, k_2$. 

48
4.3. SPATIAL BOUNDARY ERROR

The wiggles occur between any two points \(2^{-m}j\) and \(2^{-m}(j + 1)\), although they are more significant when the payoff is large. Thus, when \(m\) increases, the wiggles get compressed, but they will never vanish by increasing \(m\), as the payoff function does not vanish when \(x \to -\infty\).

To get a little more into detail, we recall from Remark 4.2 that for computational reasons, the integration range of the payoff coefficients is truncated. In case of a put option with payoff function \(g(y) := K(1 - e^y)^+\), the payoff coefficients are approximated by,

\[
V_{m,k} := K \int_{-\infty}^{0} (1 - e^y) \phi_{m,k}(y) \, dy \approx K \int_{a}^{0} (1 - e^y) \phi_{m,k}(y) \, dy,
\]

where \(a \ll 0\). This generates a discontinuity at \(a\) of a size about equal to \(K\), and the resulting wiggles are due to the Gibbs phenomenon in Shannon sampling expansions [JJ07].

Multiple solutions have been proposed in sampling theory and wavelet literature [Dau92, Mal09], by considering wavelets on a compact domain or by artificially damp the payoff function, such that it vanishes when \(x \to -\infty\). In this application, we carelessly truncated the integral, which appears to be fine for European options. However, we will see later on that in the case of Bermudan options, wiggles cause significant errors.

A natural solution in terms of damping follows from the use of the FFT to compute the payoff coefficients, which we present in the following section.

**Remark 4.4.** One might remark that the SWIFT method has issues along the boundary of the computational domain, in contrast to what is explained above in the example of multiple strike pricing. The reason multiple strike pricing is insensitive to these wiggles is that the payoff coefficients are recomputed for each of the requested strikes. Thus, the region where the wiggles are most prominent is where the density function is basically zero.

### 4.3.4 Diminishing Wiggles by Efficient FFT Implementation

We discuss here an efficient implementation of the SWIFT method for European options, exploiting that we have an excess of coefficients due to the Fast Fourier transform, which we use to damp the payoff function and reduce wiggles.

Point of departure is again the SWIFT pricing formula,

\[
v(x, t) \approx e^{-r(T-t)} \sum_{k=k_1}^{k_2} D^*_{m,k}(x)V^*_{m,k},
\]

where \(D^*_{m,k}(x)\) are the density coefficients, and \(V^*_{m,k}\) the payoff coefficients, both approximated with Vieta’s formula, and both computed using the FFT. We use the Matlab FFT implementation, see Definition 2.6, which is defined for \(k = 1 - N/2, \ldots, N/2\) as,

\[
D_k(\{z_j\}_{j=1}^{N}) := \sum_{j=1}^{N} z_j e^{-i \frac{2\pi}{N} (j-1)(k-1)},
\]

which is equivalent to \(Z = \text{fftshift(fft(z))}\) in Matlab-code. In [OGO15], formulas for the FFT implementation of both the payoff and density coefficients are given, but for sake of illustration, we give the formula for the density coefficients,

\[
D^*_{m,k}(x) := \frac{2^{m/2}}{N/2} \sum_{j=1}^{N/2} \text{Re} \left\{ f \left( \frac{2j - 1}{N} \pi 2^m; x \right) e^{ik \frac{2j-1}{N}} \right\},
\]

where we defined \(N := 2^J\), and where,

\[
z_j := \begin{cases} f \left( \frac{2j-1}{N} \pi 2^m; x \right), & \text{for } j = 1, \ldots, N/2, \\ 0 & \text{for } j = N/2 + 1, \ldots, N. \end{cases}
\]
Thus, by applying the FFT, we obtain the set of density coefficients

\[ \{ D_{m,k}^* \text{ for } k = 1 - N/2, \ldots, N/2 \} . \]

However, the SWIFT method only uses the subset \( \{ D_{m,k}^* \text{ for } k = k_1, \ldots, k_2 \} \). It is inefficient to not use all these additional computed coefficients. When using the full set of coefficients \( k = 1 - N/2, \ldots, N/2 \), we accurately approximate the function on the domain \([a, b]\), as shown in Figure 4.10, and since both the payoff and density function vanish at \( k \to \pm \infty \), no wiggles occur.

![Figure 4.10: SWIFT recovery of the payoff function. \((k_1, k_2) = (-13, 19), N = 128\). The payoff function is naturally damped using the coefficients \( k < k_1\).](image)

We demonstrate the the use of all of the coefficients by pricing a European put on CGMY dynamics from Table 4.1, and compare convergence to the COS method and the previous SWIFT implementation, as shown in Figure 4.11. We see that the COS and SWIFT method cannot reach machine accuracy with \( L = 8 \), while the SWIFT method with extended use of coefficients reaches machine accuracy with the same value for \( L \).

To make optimal use of all the coefficients, we set the same values \( J = J_d = J_p \) for both the payoff and density coefficients, and we integrate the payoff coefficients over \( [x_1, x_2] := [c_1 - L\sqrt{c_2}, c_1 + L\sqrt{c_2}] \). In that way, none of the coefficients has to be discarded. When the two truncation coefficients \( J_d \) and \( J_p \) are the same, we can save CPU time as well, since some of the computations for the payoff and density coefficients can be reused, which results in the fact that we have an equivalently fast implementation of the SWIFT method with higher accuracy.

![Figure 4.11: European put on CGMY from Table 4.1, without throwing away any coefficient, we reach a higher accuracy with the same domain truncation parameter \( L = 8 \). For both the SWIFT methods, \( m = 0 : 6 \).](image)

To show the benefits of this approach, we have plotted the dependence of the efficient implementation on the parameter \( L \). This is shown in Figure 4.12 for the CGMY European put with parameters from Table 4.1. Thus, we see that half the domain size in this example is enough for a sufficient approximation.
4.4 Multiple strike pricing on a grid

We propose a new approach to multiple strike pricing in this section. This method can be applied to any wavelet basis as long as the payoff coefficients can be factored as $V_{m,k} = KU_{m,k}$, which is the case for European put and call options.

This approach, which we call multiple strike pricing on a grid, depends on the strikes lying on a grid in the asset domain. Later, interpolation is used to approximate values for arbitrary strikes.

This approach is based on the barrier option pricing approach in [Kir14] and the efficient Toeplitz-matrix algorithm in [FO09]. Although this method is suitable for any scaling function, we demonstrate it with the SWIFT method.

Let us consider the pricing of a European call with the SWIFT method. We consider a vector $K = \{K_j\}_j$ of strikes of the form $K_j := S_0 e^{-j/2^m}$, where $m$ is the wavelet resolution, so that in log-asset space, we have $x_j := \log(S_0/K_j) = 2^{-m}j$. Then, the payoff function can be written as,

$$g(y, T, K_j) = K(e^y - 1)^+ = e^{-2^{-m}j}g(y, T, S_0).$$

By the translation property of wavelets, we find the useful property,

$$\phi_{m,k}(y - x_j) = 2^{m/2} \phi \left( 2^m(y - \frac{j}{2^m}) - k \right) = 2^{m/2} \phi(2^m(y - (k + j))) = \phi_{m,k+j}(y).$$

Combining these two facts, we can reformulate the SWIFT pricing formula for a grid point $x_j$ as,

$$v(x_j, t) \approx v_4(x_j, t) = e^{-r\Delta t} \sum_{k=K_1}^{K_2} D_{m,k}^{*}(0)V_{m,k}^{*}(x_j),$$

where the payoff coefficients are given by,

$$V_{m,k}^{*}(x_j) := \int_{\mathbb{R}} g(y, T, K_j)\phi_{m,k}(y - x_j) \, dy = e^{-2^{-m}j}V_{m,k+j}^{*}(0).$$

Note that the vector of density coefficients is unaffected by the number of strikes to be priced, and only the length of the vector of payoff coefficients is affected by the number of strikes. We thus have to apply the FFT only two times to compute the coefficients.

Additionally, the vector of option prices $v(x, t)$ can be written as a matrix-vector multiplication with a Toeplitz matrix, which can be computed by means of two FFT computations, a Hadamard product and an inverse Fourier transform, similar to the approach in [FO09].

Interpolation is required when not the option price surface, but the option value for specific strikes is required. We found that spline interpolation gives desirable results.

In total, to value $K$ strikes when we use $N$ coefficients, the computational complexity of the grid approach is just $O((K + N) \log_2(K + N))$, compared to a computational complexity of $O(KN')$ for the...
COS method with $N'$ coefficients. In Figure 4.13, the CPU times for COS and SWIFT and SWIFT on a GRID (plus interpolation) are compared, for multiple strikes. Due to the interpolation, the SWIFT grid method loses some accuracy, but we see that its computational time is basically constant when the number of strikes increases. For the COS method, we used a large computational domain $[a, b]$ to overcome the boundary issues as described in the previous section.

![Graph](image1.png)

Figure 4.13: Pricing of a range of strikes $K = [50 : 3 : 150]$, with $m = 5$ for a European call on GBM with $S_0 = 100, r = 0.1, T = 1$ and $\sigma = 0.25$. For the COS method, $N = 50$ and $L = 10$. No interpolation is required here for the SWIFT GRID approach, thus we observe smaller errors and less CPU time compared to Figure 4.13.

![Graph](image2.png)

Figure 4.14: Pricing of a range of strikes $K = j/2^m$ with $m = 5$ for a European call on GBM with $S_0 = 100, r = 0.1, T = 1$ and $\sigma = 0.25$. For the COS-method, $N = 50$ and $L = 10$. No interpolation is required here for the SWIFT GRID approach, thus we observe smaller errors and less CPU time compared to Figure 4.13.
4.5 SWIFT-Whittaker method

In this section, we introduce a simplified version of the SWIFT method, which we call the SWIFT-Whittaker method. The main difference is that we use the Whittaker-Shannon interpolation polynomial to approximate the payoff coefficients in a very cheap way. This is useful in cases where the payoff coefficients are unavailable analytically or when they are expensive to compute. The SWIFT-Whittaker method is very easy to apply to European options, but its main advantage is in more complex options like Bermudans (see Section 5.5) and two-dimensional options (see Appendix B).

We numerically show that the method has algebraic convergence with order two. This is a slower convergence than the exponential convergence of the COS and SWIFT methods, and when a high accuracy is required, any of the other methods is more suitable. The SWIFT-Whittaker method is useful however, when only a low accuracy is required. In all the examples we discuss in this chapter, the SWIFT-Whittaker method is faster than either the COS or SWIFT methods for low accuracy.

Also, the SWIFT-Whittaker method can be a solution when no expression for the option value coefficients can be found using the original SWIFT approach.

Remark 4.5. The SWIFT-Whittaker method is also easily extended to the pricing of options on two underlying assets. The derivation of SWIFT-Whittaker 2D is a first step into the application of SWIFT to 2D options. Since the results of the SWIFT-Whittaker 2D method itself are not competitive, we only mention them in Appendix B.

The idea behind SWIFT-Whittaker is the following. From the Whittaker-Shannon interpolation polynomial of Theorem 2.4, we know that a function $g$ can be approximated by

$$g(x) \approx \sum_{k \in \mathbb{Z}} 2^{-\frac{m}{2}} g \left( \frac{k}{2^m} \right) \phi_{m,k}(x).$$

On the other hand, following Multi Resolution Analysis, the projection of $g$ onto the space $V_m$ results in

$$g(x) \approx \mathcal{P}_m g(x) = \sum_{k \in \mathbb{Z}} G_{m,k} \phi_{m,k}(x),$$

where $G_{m,k} := \langle g, \phi_{m,k} \rangle$. Thus, comparing both series approximations, we find that we can approximate the wavelet coefficients with

$$G_{m,k} \approx 2^{-\frac{m}{2}} g \left( \frac{k}{2^m} \right).$$

Remark 4.6. This approximation is only useful when the function $g$ is known. Thus, we cannot apply it to the computation of the density coefficients. An equality $G_{m,k} = 2^{-\frac{m}{2}} g \left( \frac{k}{2^m} \right)$ occurs when the function $g$ is bandlimited with bandlimit $B < 2^m$, as shown in Lemma 2.2. However, payoff functions in finance are generally not bandlimited, which is the reason for the low-accuracy mentioned.

We start by applying the SWIFT-Whittaker method by pricing European options and price a put option using the SWIFT pricing formula in (4.3), i.e.,

$$v(x,t) \approx v_3(x,t) = e^{-r(T-t)} \sum_{k=k_1}^{k_2} D^*_{m,k}(x)V_{m,k},$$

(4.13)

where the density coefficients are unchanged from the original SWIFT formulation and they are approximated using Vieta’s formula as in Section 3.6.1. The payoff coefficients are defined as $V_{m,k} := \int_{\mathbb{R}} v(y,T)\phi_{m,k}(y) \, dy$, and are approximated by

$$V_{m,k} \approx V^W_{m,k} := 2^{-\frac{m}{2}} v \left( \frac{k}{2^m}, T \right).$$

The SWIFT-Whittaker pricing formula for any European option with payoff function $g(x)$ is given by,

$$v(x,t) \approx e^{-r(T-t)} 2^{-\frac{m}{2}} \sum_{k=k_1}^{k_2} g \left( \frac{k}{2^m} \right) D^*_{m,k}(x),$$

(4.14)

In terms of CPU time, for the same approximation scale $m$, the SWIFT-Whittaker method is twice as fast as SWIFT, since the FFT has to be applied only once, in the computation of the density coefficients, and not for the payoff coefficients.
Example 10 (GBM European call). To demonstrate the convergence of the SWIFT-Whittaker method, we price a European put on GBM, as shown in Figure 4.15. We observe a remarkable precise linear convergence in the log-domain, where the price error is of order $O(N^{-2})$, when $N$ denotes the number of coefficients.

We also see from the right subfigure that the SWIFT-Whittaker method is faster than both SWIFT and COS up to an accuracy of $10^{-3}$.

In the SWIFT-Whittaker method, the density coefficients are accurately approximated using Vieta’s formula, while the payoff coefficients are badly approximated using the Whittaker-Shannon interpolation formula. The SWIFT-Whittaker method is more efficient when both approximation errors are of the same order. This is the case when a high wavelet scale is required to approximate the density function, for example when the density function is highly peaked, as in the following example

Example 11 (CGMY European put). We price a European put option on CGMY dynamics with parameters as in Table 4.1. We used the efficient implementation as described in Section 4.3.4 using $L = 8$.

Figure 4.16 illustrates that the SWIFT-Whittaker method converges linearly in $m$, whereas the SWIFT method converges exponential, but it also shows an advantage of the SWIFT-Whittaker method, that is, it is less dependent on the domain boundary. For SWIFT, it is expensive to compute payoff coefficients on a large domain, and therefore, we damp the payoff function, see Remark 4.2 and combined with the resulting damping as in Section 4.3.4, low-scale errors occur of order $10^{-10}$. The SWIFT-Whittaker method does not suffer from this, and can achieve higher accuracy, although at a very high wavelet scale.

The advantage of the SWIFT-Whittaker method is mainly in the case where the payoff coefficients of the SWIFT method are unavailable, or expensive to compute. This is what we will see in the next chapter for Bermudan options.
Chapter 5

Bermudan option valuation

In this section, we generalize the SWIFT method to the pricing of Bermudan options for exponential Lévy processes. The SWIFT method is a Fourier method, and the fast Fourier transform can be applied for an efficient implementation.

European options, as explained in Chapter 3, allow the holder of the option contract to exercise the option at a fixed time $T$. Not before, and certainly not after that moment. Bermudan options, on the contrary, specify a certain number of exercise dates on which the holder can choose whether to exercise or not. Thus, the holder needs an exercise strategy to decide which is the optimal moment to exercise. This style of option is called an early-exercise option.

An option contract which allows the holder to exercise the contract at any time before maturity $T$ is called an American option.

The name ‘Bermudan option’ is a pun. Bermuda is a British overseas territory, being somewhat American, somewhat European. Both the option style and physical location are in the between American and European.

It is easy to see that the value of Bermudan option is bounded below by the value of its equivalent European option. When the holder chooses the exercise policy: “Exercise at time $T_n$, when the option is in-the-money”, it replicates its European counterpart. Any other strategy we look for should be an improvement over this one.

5.1 Bermudan Option Pricing formula

We describe the Bermudan option pricing problem for a vanilla Bermudan put. Consider a vanilla Bermudan put option with strike $K$. The option has $M$ exercise opportunities, denoted by $0 = t_0 < t_1 < \cdots < t_M = T$, where $T$ is the options maturity. For ease of notation only, we use a constant time between two exercise dates, i.e., $\Delta t_n := t_{n+1} - t_n = \Delta t$. As in the European case, we consider again the log-asset domain, and define,

$$x := \log(S_{t_{n-1}}/K), \quad \text{and} \quad y := \log(S_{t_n}/K),$$

where $S_t$ is the underlying asset, which follows an exponential Lévy process. The payoff function for a put option is, as before, given by,

$$g(x) := g(x,t,K) = K(1-e^x)^+. $$

Since a Bermudan option can be exercised at any exercise opportunity $t_n$, it can never be worth less than the payoff at that point. Using this insight, we value the option by a dynamic programming approach, processing backwards in time, starting at maturity $t_n = T$. At maturity, the option value is given by the payoff,

$$v(y,t_n) = g(y). \quad (5.1)$$

Then we step backwards in time over the discrete set of exercise dates $n = M, M - 1, \ldots, 2$. At each exercise date, the option value is the maximum between immediate exercise and a continuation value,
We show that the vector of value coefficients \( V \) and the continuation coefficients \( G \) are now time dependent and defined by,

\[
\begin{align*}
V(x, t_n) &= \max(g(x), c(x, t_n)), \\
C(x, t_n) &= e^{-r\Delta t} \int_R v(y, t_n) f_{\Delta t}(y|x) \, dy,
\end{align*}
\]  

where \( c(x, t_n) \) is referred to as the continuation value. The option value at initial time \( t_0 := 0 \) can be found by solving,

\[
v(x, t_0) = e^{-r\Delta t} \int_R v(y, t_1) f_{\Delta t}(y|x) \, dy.
\]  

In the methods we describe in the following sections, we assume that the underlying prices are driven by an exponential Lévy process. This implies that in log-asset space, the price increments are independent and stationary.

### 5.2 The COS method

Bermudan option pricing with the COS method is described for Lévy processes by F. Fang and Oosterlee [FO09].

We follow the dynamic programming approach as described above. The goal is to compute the option value at initial time \( t_0 = 0 \), as described in (5.3). We recognize in (5.3) a European option pricing problem, where the (unknown) value function \( v(y, t_1) \) plays the role of the payoff function. We can thus approximate this integral on a domain \([a, b]\) using the COS method. Following the COS approach, described in Section 3.4, we approximate \( v(x, t_0) \) by,

\[
v(x, t_0) = e^{-r\Delta t} \int_R v(y, t_1) f_{\Delta t}(y|x) \, dy \\
\approx \frac{b-a}{2} e^{-r\Delta t} \sum_{k=0}^{N-1} D_k(x) V_k(t_1),
\]  

where the density coefficients \( D_k(x) \) can be computed from the characteristic function of the conditional transitional density, \( f_{\Delta t}(y|x) \), i.e.,

\[
D_k(x) := \frac{2}{b-a} \Re \left\{ \hat{f}_{\Delta t} \left( \frac{k\pi}{b-a} ; x \right) e^{-ik\pi \frac{x}{a}} \right\}.
\]

The value coefficients \( V_k(t_1) \) are now time dependent and defined by,

\[
V_k(t_n) := \frac{2}{b-a} \int_a^b v(y, t_n) \cos \left( k\pi \frac{y-a}{b-a} \right) \, dy.
\]

We show that the vector of value coefficients \( V(t_n) := \{V_k(t_n)\}_{k=0,1,\ldots,N-1} \) can be recovered from \( V(t_{n+1}) \). First, we note that there exists an early exercise point \( x^*_n \) at time \( t_n \) such that \( g(x^*_n) = c(x^*_n, t_n) \). Assume that \( x^*_n \) is known. We can then split up the integration that defines \( V_k(t_n) \) into two parts: one on the interval \([a, x^*_n]\) and the other on \((x^*_n, b]\), i.e.,

\[
V_k(t_n) = G_k(a, x^*_n) + C_k(x^*_n, b, t_n),
\]

where the payoff coefficients \( G_k(x_1, x_2) \) are defined by,

\[
G_k(x_1, x_2) := \frac{2}{b-a} \int_{x_1}^{x_2} g(x) \cos \left( k\pi \frac{x-a}{b-a} \right) \, dx,
\]

and the continuation coefficients \( C_k(x_1, x_2, t_n) \) are defined by,

\[
C_k(x_1, x_2, t_n) := \frac{2}{b-a} \int_{x_1}^{x_2} c(x, t_n) \cos \left( k\pi \frac{x-a}{b-a} \right) \, dx.
\]
When \( x_n^* \) is known, the integral in the payoff coefficients can be computed analytically, similar to the European case as described in Section 3.4. The recursion between time steps will follow from the continuation coefficients.

Using the insight that the continuation value \( c(x, t_n) \) itself is a European option pricing problem, we apply the COS method to \( c(x, t_n) \) and approximate it by,

\[
c(x, t_{n-1}) = e^{-r\Delta t} \int_R v(y, t_n) f_{\Delta t}(y|x) \, dy
\]

\[
\approx \frac{b-a}{2} e^{-r\Delta t} \sum_{k=0}^{N-1} D_k(x) V_k(t_n)
\]

\[
e^{-r\Delta t} \sum_{k=0}^{N-1} \text{Re} \left\{ \hat{f}_{\Delta t} \left( \frac{j\pi}{b-a} ; x \right) e^{-ij\pi \frac{x}{b-a}} \right\} V_j(t_n)
\]

\[
e^{-r\Delta t} \sum_{j=0}^{N-1} \text{Re} \left\{ \hat{f}_{\Delta t} \left( \frac{j\pi}{b-a} ; 0 \right) e^{ij\pi \frac{x}{b-a}} \right\} V_j(t_n),
\]

where the final equality arises from the Lévy property that \( f(y|x) = f(y(0))e^{iyx} \). We insert this approximation of the continuation value into the continuation coefficients and by exchanging summation and integration we obtain,

\[
C_k(x_1, x_2, t_{n-1}) = \frac{2}{b-a} \int_{x_1}^{x_2} c(x, t_{n-1}) \cos \left( k\pi \frac{x-a}{b-a} \right) \, dx
\]

\[
\approx e^{-r\Delta t} \text{Re} \left\{ \sum_{j=0}^{N-1} V_j(t_n) \hat{f}_{\Delta t} \left( \frac{k\pi}{b-a} ; 0 \right) \frac{2}{b-a} \int_{x_1}^{x_2} e^{ij\pi \frac{x}{b-a}} \cos \left( k\pi \frac{x-a}{b-a} \right) \, dx \right\}
\]

\[
e^{-r\Delta t} \text{Re} \left\{ \sum_{j=0}^{N-1} F_j(t_n) \cdot \mathcal{M}_{k,j}(x_1, x_2) \right\},
\]

where we defined \( F_j(t_n) := V_k(t_n) \hat{f}_{\Delta t} \left( \frac{j\pi}{b-a} ; 0 \right) \) and

\[
\mathcal{M}_{k,j}(x_1, x_2) := \frac{2}{b-a} \int_{x_1}^{x_2} e^{ij\pi \frac{x}{b-a}} \cos \left( k\pi \frac{x-a}{b-a} \right) \, dx.
\]

These integrals can be solved analytically, and we can thus compute the continuation coefficients. However, in [FO09], an efficient method for the computation of the whole vector of continuation coefficients using the FFT is derived, which we discuss below.

### 5.2.1 Continuation coefficients

The continuation coefficients \( C_k(x_1, x_2, t_{n-1}) \) can be computed efficiently by the application of the FFT algorithm by the insight that the matrix \( \mathcal{M} := \{ \mathcal{M}_{k,j}(x_1, x_2) \} \) is the sum of a Toeplitz and a Hankel matrix, and an efficient algorithm is derived in [FO09], which we present here.

Replacing \( e^{i\alpha} = \cos(\alpha) + i\sin(\alpha) \) in (5.7) gives the representation,

\[
\mathcal{M}_{k,j}(x_1, x_2) = -\frac{i}{\pi} \left( \mathcal{M}_{k,j}^c(x_1, x_2) + \mathcal{M}_{k,j}^s(x_1, x_2) \right),
\]

where,

\[
\mathcal{M}_{k,j}^c := \begin{cases} \frac{(x_2-x_1)\pi i}{b-a} & \text{if } k = j = 0, \\ \frac{\exp(i(j+k)(x_2-x_1)\pi/j+k)}{j+k} - \exp(i(j+k)(x_1-a)\pi/j+k) & \text{otherwise}, \end{cases}
\]

and

\[
\mathcal{M}_{k,j}^s := \begin{cases} \frac{(x_2-x_1)\pi i}{b-a} & \text{if } k = j, \\ \frac{\exp(i(j-k)(x_2-a)\pi/j-k)}{j-k} - \exp(i(j-k)(x_1-a)\pi/j-k) & \text{otherwise}. \end{cases}
\]
Thus, we obtain a matrix-vector product representation for $C(x_1, x_2, t_m)$, i.e.,

$$
C(x_1, x_2, t_m) = e^{-r(T-t)} \frac{\pi}{\pi} \text{Im}\left\{ (\mathcal{M}_c + \mathcal{M}_s)u \right\}.
$$

In particular, $\mathcal{M}_c$ is a Hankel matrix, and $\mathcal{M}_s$ is a Toeplitz matrix, therefore, we can compute these matrix-vector products efficiently in $O(N \log_2 N)$ operations using the Fast Fourier Transform.

### 5.2.2 Computation of the early-exercise point

Each timestep $n$, we have to determine the early exercise point $x^*_n$, which is the $x$ value that solves $g(x) = c(x, t_n)$. The payoff function $g(x)$ is known, and the continuation value $c(x, t_n)$ can be recovered by the COS method in terms of,

$$
c(x, t_n) \approx b - \frac{a}{2} e^{-r \Delta t} \sum_{k=0}^{N-1} D_k(x) V_k(t_n),
$$

and for Lévy processes the derivative of $D_k(x)$ with respect to $x$ can be computed analytically and is given by,

$$
\frac{\partial}{\partial x} D_k(x) = \frac{ik\pi}{b-a} D_k(x).
$$

Therefore, to find $x^*_n$, we can apply Newton’s method, i.e., we iterate over $x_j$ for $j = 1, 2, \ldots$,

$$
x_{j+1} = x_j - \frac{g(x_j) - c(x_j, t_n)}{\frac{\partial}{\partial x} g(x_j) - \frac{\partial}{\partial x} c(x_j, t_n)},
$$

where we start with $x_0 := x^*_{n+1}$, and we know that at maturity, $x^*_M = 0$.

### 5.2.3 COS Bermudan Put pricing algorithm

We are now ready to formulate the COS pricing algorithm, which is shown in Algorithm 1.

```
Algorithm 1: COS method for pricing a Bermudan Put
```

| At maturity $T$, $V_k(t_M) = G_k(a, 0)$ for $k = 0, 1, \ldots, N - 1$; |
| **for** $n = M-1, \ldots, 1$ **do** |
| Determine early-exercise point $x^*_n$ by Newton’s method; |
| Compute $C_k(x^*_n, b, t_n)$ from $V_k(t_{n+1})$ using the FFT as in (5.6) and (5.7); |
| Set $V_k(t_n) = G_k(a, x^*_n) + C_k(x^*_n, b, t_n)$; |
| **end** |
| Finally, reconstruct $v(x, t_0)$ by inserting $V_k(t_1)$ in Equation (5.4) |

### 5.2.4 Boundary Behavior

As we saw in Section 4.3.2, the COS method has difficulties in approximating the true option value around the boundary of the asset domain $[a, b]$. Option values close to the boundary are of increased interest in the Bermudan pricing problem, where the continuation value is of the form,

$$
c(x, t_{n-1}) = e^{-r \Delta t} \int_R v(y, t_n) f_{\Delta t}(y|x) \, dy,
$$

and thus depends on all $v(y, t_n)$ for $y \in [a, b]$. Since the pricing of Bermudans is done backwards in time, using the previous continuation value each time step to determine the early exercise point, an inaccurate exercise policy may occur when errors propagate. This is shown in Figure 5.1. We price a Bermudan put on GBM with $L = 10$. Focusing on the left sub-figure, we see on the left hand side of the computational domain that the option value is underestimated, as became clear in the European case.
5.3. The SWIFT Bermudan Method

We can apply the strategy of the COS method to the SWIFT method as well to price Bermudan put options. In a similar fashion as the COS method in the previous section, we can approximate \( v(x,t_0) \) with the SWIFT method, such that,

\[
v(x,t_0) = e^{-r\Delta t} \int_{\mathbb{R}} v(y,t_1) f_{\Delta t}(y|x) \, dy \\
\approx e^{-r\Delta t} \sum_{k=k_1}^{k_2} D_{m,k}^*(x)V_{m,k}(t_1),
\]

(5.10)

where the density coefficients \( D_{m,k}^*(x) \) now arise from the wavelet expansion at scale \( m \) of \( f_{\Delta t}(y|x) \), given by,

\[
f_{\Delta t}(y|x) \approx \sum_{k=k_1}^{k_2} D_{m,k}^*(x)\phi_{m,k}(y),
\]

as has been described in the SWIFT method for Europeans in Section 3.6.1. We will argue later how to choose \( m, J \) and \( (k_1, k_2) \). The density coefficients are, as in the European case, given by,

\[
D_{m,k}^*(x) := \frac{2^m}{2^{j-1}} \sum_{j=1}^{2^j-1} \text{Re} \left\{ \hat{f}(c_j 2^m; x) e^{ikc_j} \right\},
\]

and \( c_j := \frac{2j-1}{2^j}\pi \). We propose a similar scheme as in the COS method to recursively determine \( V_k(t_1) \). Starting at maturity, the option value equals the payoff, and thus,

\[
V_{m,k}(T) := \int_{\mathbb{R}} v(y,T)\phi_{m,k}(y) \, dy \\
= \int_{-\infty}^{0} g(y)\phi_{m,k}(y) \, dy \\
=: G_{m,k}(-\infty, 0, T),
\]

(5.11)

where the payoff coefficients can be approximated by application of Vieta’s formula, similar to the European case, and can be computed using the FFT algorithm (See Section 4.1). At any time \( t_n \) for
5. BERMDAN OPTION VALUATION

$n = M - 1, \ldots, 1$, we have that,

$$V_{m,k}(t_n) := \int_\mathbb{R} v(y, t_n) \phi_{m,k}(y) \, dy$$

$$= \int_\mathbb{R} \max \{ g(y), c(y, t_n) \} \phi_{m,k}(y) \, dy$$

$$= \int_{x_n^*}^{+\infty} g(y) \phi_{m,k}(y) \, dy + \int_{x_n^*}^{\infty} c(y, t_n) \phi_{m,k}(y) \, dy$$

$$=: G_{m,k}(-\infty, x_n^*, t_n) + C_{m,k}(x_n^*, \infty, t_n),$$

where \( x_n^* \) is again the early-exercise point at time \( t_n \), and the continuation coefficients \( C_{m,k}(a, x_n^*, t_n) \) can be approximated efficiently with the use of the FFT algorithm, as we see in the next section.

### 5.3.1 Continuation coefficients

The continuation coefficients can be approximated when one notes that the continuation value is a European option pricing problem, which, when inserted into the expression for the continuation coefficients, results in,

$$C_{m,k}(x_1, x_2, t_{n-1}) := \int_{x_1}^{x_2} c(x, t_{n-1}) \phi_{m,k}(x) \, dx$$

$$\approx e^{-r\Delta t} \int_{x_1}^{x_2} \left( \sum_{p=k_1}^{k_2} D_{m,p}^* \cdot V_{m,p}(t_n) \right) \phi_{m,k}(x) \, dx$$

$$e^{-r\Delta t} \sum_{p=k_1}^{k_2} V_{m,p}(t_n) \int_{x_1}^{x_2} D_{m,p}^* \phi_{m,k}(x) \, dx. \quad (5.12)$$

Next, we insert the expression for the density coefficients and make use of the translation property of Lévy processes, i.e., \( \hat{f}(y; x) = \hat{f}(y; 0) e^{-iyx} \). Then, by interchanging summations and integration we find,

$$C_{m,k}(x_1, x_2, t_{n-1}) \approx e^{-r\Delta t} \sum_{p=k_1}^{k_2} V_{m,p}(t_n) \int_{x_1}^{x_2} D_{m,p}^* \phi_{m,k}(x) \, dx$$

$$= \frac{2^{m/2}}{2^{j-1}} e^{-r\Delta t} \sum_{p=k_1}^{k_2} V_{m,p}(t_n) \int_{x_1}^{x_2} \sum_{j=1}^{2^{j-1}} \text{Re} \left\{ \hat{f}(c_j 2^m; 0) e^{-ic_j 2^m x} e^{ip_c j} \right\} \phi_{m,k}(x) \, dx$$

$$= \frac{2^{m/2}}{2^{j-1}} e^{-r\Delta t} \sum_{j=1}^{2^{j-1}} \text{Re} \left\{ \hat{f}(c_j 2^m; 0) \cdot \left( \sum_{p=k_1}^{k_2} V_{m,p}(t_n) e^{ip_c j} \right) \cdot \int_{x_1}^{x_2} e^{ic_j 2^m x} \phi_{m,k}(x) \, dx \right\}$$

$$= \frac{2^{m/2}}{2^{j-1}} e^{-r\Delta t} \sum_{j=1}^{2^{j-1}} \text{Re} \left\{ \mathcal{V}_j(t_n) \cdot \int_{x_1}^{x_2} e^{-ic_j 2^m x} \phi_{m,k}(x) \, dx \right\}$$

$$= \frac{2^{m/2}}{2^{j-1}} e^{-r\Delta t} \sum_{j=1}^{2^{j-1}} \int_{x_1}^{x_2} e^{-ic_j 2^m x} \phi_{m,k}(x) \, dx \quad (5.13)$$

where we defined \( \mathcal{V}_j(t_n) := \sum_{p=k_1}^{k_2} V_{m,p}(t_n) e^{ip_c j} \), and note that the whole vector of payoff coefficients \( \{ \mathcal{V}_j(t_n) \}_{j=1, \ldots, 2^{j-1}} \) can be computed using the FFT algorithm.

Then, \( \mathcal{F}_j(t_n) := \hat{f}(c_j 2^m; 0) \mathcal{V}_j(t_n) \), which is a simple product, thus can be formed cheaply. Now, we focus on the remaining integral,

$$I_{j,k}(x_1, x_2) := \int_{x_1}^{x_2} e^{-ic_j 2^m x} \phi_{m,k}(x) \, dx$$

$$\approx \frac{2^{m/2} 2^{j-1}}{2^{j-1}} \sum_{p=1}^{2^{j-1}} \int_{x_1}^{x_2} e^{-ic_j 2^m x} \cos(c_q(2^m(x - k)) \, dx \quad (5.14)$$
The integral here can be solved analytically, but we make use of the equality \( \cos(t) = \frac{1}{2} \left( e^{it} + e^{-it} \right) \), to make it independent of \( k \). By noting that \(-c_q = c_{1-q}\) we find,

\[
I_{j,k}(x_1, x_2) \approx \frac{2^{m/2}}{2^J} \sum_{q=1-2^J} 2^{J-1} e^{ic_qk} M_{j,q}(x_1, x_2), \tag{5.15}
\]

where,

\[
M_{j,q} := \int_{x_1}^{x_2} e^{-i(c_j+c_q)2^m x} dx.
\]

The integrals \( M_{j,q} \) can be solved analytically, and the solutions are given by,

\[
M_{j,q}(x_1, x_2) = \begin{cases} \frac{x_2 - x_1}{2^{m-1} e^{-i(c_j+c_q)2^m(x_2-x_1)/2}}, & \text{for } q = 1-j, \\ \text{else}. \end{cases}
\]

Reordering the summations and integrals once more using this representation of \( I_{j,k}(x_1, x_2) \) gives us the final expression for the continuation coefficients,

\[
C_{m,k}(x_1, x_2, t_{n-1}) \approx \frac{2^{m/2}}{2^J} e^{-r\Delta t} \sum_{j=1}^{2^{J-1}} \text{Re} \left\{ F_j(t_n) \cdot I_{j,k}(x_1, x_2) \right\}
\]

\[
= \frac{2^m}{2^{2J-1}} e^{-r\Delta t} \text{Re} \left\{ \sum_{q=1-2^J} 2^{J-1} e^{ic_qk} \cdot \left( \sum_{j=1}^{2^{J-1}} F_j(t_n) M_{j,q}(x_1, x_2) \right) \right\} \tag{5.16}
\]

where this summation over \( q \) can be computed for the whole vector \( \{C_{m,k}\}_k \) using the FFT, with coefficients,

\[
J_q(x_1, x_2, t_n) := \left( \sum_{j=1}^{2^{J-1}} F_j(t_n) M_{j,q}(x_1, x_2) \right).
\]

Finally, the coefficients \( J_q \) can be computed efficiently by noting that \( M_{j,q} \) only depends on its indices through \( j+q \), and thus this summation represents a matrix-vector product where the matrix is a Hankel matrix. As described in Section 2.1, this matrix-vector product can be computed with the application of three times the FFT.

### 5.3.2 Domain truncation

As in the European case, see Section 3.3, we use the cumulants method to determine a suitable domain truncation. Let \( L \) be a positive constant and define \([a, b] := [c_1 - L\sqrt{c_2}, c_1 + L\sqrt{c_2}]\), where \( c_i \) is the \( i \)th cumulant of the underlying characteristic function. This heuristically chosen domain is generally sufficient to capture sufficient probability mass.

Vieta’s truncation parameter \( J \) is chosen the same for the payoff, density and continuation coefficients. Following [OGO15], we set,

\[
J := \left\lfloor \log_2(2^m \pi (\bar{b} - \bar{a})) \right\rfloor = \left| 1 + m + \log_2(\pi L \sqrt{c_2}) \right|.
\]

Then, the application of the FFT results in the computation of a vector of \( N := 2^J \) coefficients, corresponding to \( k = 1 - \frac{1}{2} N, \ldots, \frac{1}{2} N \). However, we are interested in only \( k := k_1, \ldots, k_2 \), where \( k_1 := \lceil 2^m a \rceil \) and \( k_2 := \text{ceil}2^m a \).

**Remark 5.1.** To optimally benefit from the FFT, it is advised to use all of the coefficients \( k = 1 - \frac{1}{2} N, \ldots, \frac{1}{2} N \), as they are readily computed; in stead of only using \( k = k_1, \ldots, k_2 \). We do not do use it, as it reduces the flexibility of the method, which complicates the analysis in the next section.
5.3.3 Early exercise point

Each timestep $n$, we have to determine the early exercise point $x^*_n$, which is the $x$ value that solves $g(x) = c(x, t_n)$. The payoff function $g(x)$ is known, and the continuation value $c(x, t_n)$ can be recovered by the SWIFT method (4.3), in terms of,

$$c(x, t_n) \approx \sum_{p=k_1}^{k_2} D_{m,p}^*(x)V_{m,p},$$

and the derivative of $D_{m,p}^*(x)$ with respect to $x$ can be computed analytically for Lévy processes and is given by,

$$\frac{\partial}{\partial x}D_{m,p}^*(x) = \frac{2^{m/2}}{2^{j-1}} \sum_{j=1}^{2^{j-1}} \text{Re}\left\{-ic_j 2^m \hat{f}(c_j 2^m; x) e^{ikc_j}\right\}.$$ 

Therefore, to find $x^*_n$, we can apply Newton’s method, i.e., we iterate over $x_j$ for $j = 1, 2, \ldots$,

$$x_{j+1} = x_j - \frac{g(x_j) - c(x_j, t_n)}{\frac{\partial}{\partial x}g(x_j) - \frac{\partial}{\partial x}c(x_j, t_n)},$$

where we start with $x_0 := x^*_{n+1}$, and we know that at maturity, $x^*_M = 0$.

5.3.4 The SWIFT Bermudan Algorithm

The SWIFT Bermudan algorithm is summarized in Algorithm 2.

Choosing a wavelet scale $m$ and domain truncation parameter $L$;
Set $J, N, [a, b]$ and $k = k_1, \ldots, k_2$ as in Section 5.3.2;
At maturity $T$, $V_{m,k}(t_n) = G_{m,k}(a,0)$, where $G_{m,k}(a,0)$ are computed by (5.11) with the help of the FFT;
for $n = M - 1, \ldots, 1$ do
\hspace{1cm} Determine early-exercise point $x^*_n$ by Newton’s method as described in Section 5.3.3;
\hspace{1cm} Compute $C_{m,k}(x^*_n, b, t_n)$ using the FFT approach (5.16) in Section 5.3.1;
\hspace{1cm} Compute $G_{m,k}(a, x^*_n)$ by (5.11) using the FFT;
Set $V_{m,k}(t_n) = G_{m,k}(a, x^*_n) + C_{m,k}(x^*_n, b, t_n)$;
end
Finally, reconstruct $v(x, t_0)$ by inserting $V_{m,k}(t_1)$ in Equation (5.10)

Algorithm 2: SWIFT method for pricing a Bermudan Put

The COS method requires the application of the FFT five times for each time step. The SWIFT method is more expensive. At each time step, to determine the early-exercise point, we employ 5 iterations of Newton’s algorithm, and for each iteration we recompute the density coefficients with the use of the FFT. To compute the continuation coefficients with SWIFT we require the FFT 5 times as well. Thus, when the number of coefficients of the SWIFT and COS methods are about the same, both methods have the same order of computational time, but the SWIFT method is about 2 times slower.

As we will see in the next section, there are many cases in which the SWIFT method requires fewer coefficients than the COS method, and thus can compete in terms of speed.

Also, we show that the SWIFT method is more robust in terms of the computational domain $[a, b]$, which we demonstrate with numerical examples.

Remark 5.2. In the following section, we numerically show exponential convergence of the SWIFT method in the pricing of Bermudan options. A formal proof should be given, and we are sure this can be done. This however is future work.
5.4 Numerical Results for Bermudan Options

We test the SWIFT method that we derived in this chapter against the COS method in the pricing of Bermudan options. We start with showing two pricing examples where we compare computational time with accuracy.

<table>
<thead>
<tr>
<th>Test No.</th>
<th>Model</th>
<th>( S_0 )</th>
<th>( K )</th>
<th>( T )</th>
<th>( r )</th>
<th>Other Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BS</td>
<td>100</td>
<td>110</td>
<td>1</td>
<td>0.1</td>
<td>( \sigma = 0.2 )</td>
</tr>
<tr>
<td>2</td>
<td>CGMY</td>
<td>100</td>
<td>80</td>
<td>1</td>
<td>0.1</td>
<td>( C = 1, G = 5, M = 5, Y = 1.5 )</td>
</tr>
<tr>
<td>3</td>
<td>CGMY</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
<td>( C = 1, G = 5, M = 5, Y = 0.5 )</td>
</tr>
</tbody>
</table>

Table 5.1: Test parameters for pricing Bermudan Options

The first test is the pricing of a Bermudan put with 10 exercise moments and parameters are given in Table 5.1. The reference price is \( 10^{479520123} \), obtained by the COS method with \( L = 50 \) and \( N = 10^5 \) coefficients. Convergence results are shown in Figure 5.2. Rapid convergence is observed for both the methods, and the computational complexity is similar.

![GBM Bermudan Put - 10 exercise moments - Accuracy](image1)

![CPU Time](image2)

Figure 5.2: Pricing a GBM Bermudan put with 10 exercise moments and parameters as in Table 5.1. Other parameters: \( L = 8 \), For COS, \( N = 32d \) with \( d = 1 : 5 \) coefficients are used. For SWIFT, \( m = 3 : 6 \).

In the next test, Test 2, we price a Bermudan put with 10 exercise moments, but the underlying is driven by CGMY with parameters in Table 5.1. The reference value is found to be \( 28.829781986 \). We observe a similar pattern as for the Bermudan put on GBM.

![CGMY Bermudan Put - Error](image3)

![CPU Time](image4)

Figure 5.3: Pricing a CGMY Bermudan Put with 10 exercise moments and parameters as in Table 5.1, Test No.2. Other parameters: \( L = 8 \), for COS, \( N = 32d \) with \( d = 1 : 5 \) coefficients are used. For SWIFT, \( m = 0 : 3 \).

Although the COS method is faster, the SWIFT method exhibits the same order of computational costs, thus the SWIFT method is only a few times slower. We benefit from the fact that we can test the suitability of the parameters of the SWIFT method \textit{a priori}. We start of by selecting a proper wavelet scale.
5.4.1 Wavelet Scale

For European option pricing, we gave a rule of thumb to select an appropriate value for the wavelet scale \( m \), i.e., we select a value for \( m \) such that \( |\hat{f}(\omega)| < \text{TOL} \) holds for all \( |\omega| > 2^m \pi \). We can then heuristically say that \( f \) is 'close' to a bandlimited function with bandlimit \( 2^m \), and is thus approximated well at scale \( m \).

From the uncertainty principle of the Fourier transform we know that when a density function is highly peaked, its characteristic function has heavy tails, and thus requires a large value for the wavelet scale \( m \) to approximate it accurately.

Highly peaked density function occur in Bermudan option pricing when many exercise moments are considered. In Figure 5.4, we show, for the test parameters from Table 5.1, the required scale \( m \) to price an option with maturity \( T = 1 \) and \( M \) exercise moments, thus \( \Delta t = 1/T \). We observe that for CGMY with \( Y = 1.5 \), a fat-tailed density function, a low scale \( m \) is sufficient, but for CGMY with \( Y = 0.5 \), a very high scale is required, up to \( m = 16 \).

![Figure 5.4: The required wavelet scale to obtain an accuracy of \( TOL = 1e-08 \) of the final option price approximation for each of the three test models from Table 5.1. Maturity of the option is fixed at \( T = 1 \), \( \Delta t = 1/M \), where \( M \) is the number of exercise moments.](image)

We conclude that when the time between exercise moments decreases, the approximation scale needs to increase. When the computation domain \([a, b]\) is fixed, the number of coefficients is approximately given by \( N_{SWIFT} \approx 2^m (b - a) \). To control the number of coefficients, it is therefore important to find an accurate bound on the computation domain.

Remark 5.3. The COS method shows similar behavior with respect to the number of exercise moments. When the underlying density is very peaked, a large value \( N \) is required for an accurate approximation.

5.4.2 Domain Truncation

We analyze the behavior of the SWIFT Bermudan method with respect to the domain boundary. In Chapter 4, we explained for European options that the SWIFT method is less vulnerable to the domain truncation as the COS method. Due to the locality of wavelets, it the domain is naturally extended by added more coefficients.

As in the European case, we consider the coefficients \( k = k_1, \ldots, k_2 \), representing the option value on a domain \([a, b] = 2^{-m}[k_1, k_2] \). When this domain is chosen too small, probability mass is lost, and due to the recursive pricing algorithm, this mass might be significant when the option has many exercise moments. To verify this statement, we plotted the test \( H_1(L) \) from (4.8), given by,

\[
H_1(L) = \left| 1 - \sum_{k=k_1}^{k_2} D_{m,k} \right| \leq 1 - \sum_{k=k_1}^{k_2} |D_{m,k}| < \frac{TOL}{K},
\]

and the results are shown in Figure 5.6. We see that for small values of \( L \) the option error is dominated by the missing probability mass, as for the COS method, and this is in line met the test \( H_1 \).
5.4. NUMERICAL RESULTS FOR BERMUDAN OPTIONS

However, we observe a second error source, which is dominating the overall error when $L$ is about 3 and larger. The cause of these errors are the wiggles as explained in Section 4.3.4. A remedy is found in extending the computational domain, making use of the additionally computed coefficients by the FFT.

We propose a characterization of the computational domain in the following example, which optimally uses the FFT and all of the resulting coefficients.

**Example 12** (SWIFT Bermudan Extended domain). As in the European case, see Section 3.3, we use the cumulants method to determine a suitable domain truncation. Let $L$ be a positive constant and define $[\tilde{a}, \tilde{b}] := [c_1 - L\sqrt{c_2}, c_1 + L\sqrt{c_2}]$, where $c_i$ is the $i$th cumulant of the underlying characteristic function. This heuristically chosen domain is generally sufficient to capture sufficient probability mass. In our method however, we compute simultaneously $2^J$ coefficients using the FFT, where the parameter $J$ results from the application of Vieta’s formula. Therefore, we extend the domain truncation to $[a, b]$, such that it is covered by exactly $2^J$ coefficients.

Vieta’s truncation parameter $J$ is chosen the same for the payoff, density and continuation coefficients. Following [OGO15], we set,

$$ J := \left\lceil \log_2(2^m \pi |\tilde{b} - \tilde{a}|) \right\rceil = \left\lceil 1 + m + \log_2(\pi L\sqrt{c_2}) \right\rceil. $$

Then, the application of the FFT results in the computation of a vector of $N := 2^J$ coefficients, corresponding to $k = k_1, \ldots, k_2$, where $k_1 := 1 - \frac{1}{2}N$ and $k_2 := \frac{1}{2}N$. Now, the domain we are interested in is given by $[a, b] := 2^{-m}[k_1, k_2]$.

It is shown that using these additional coefficients results in a method that converges twice as fast with respect to $L$; see the pink line in Figure 5.6.

In Figure 5.5, we priced a Bermudan put option on GBM dynamics with parameters from Table 5.1 and $M = 10$ exercise moments. The COS method suffers more heavily from the boundary issues than the SWIFT method, although we do see that error propagation occurs as well with SWIFT on the right-hand side of the domain.

![Continuation Surface](image)

Figure 5.5: The reconstructed option value of a Bermudan Put at grid point on GBM with 10 exercise moments and parameters from Test No.1, Computed with both SWIFT ($m = 7$) and COS ($N = 200$).
5.5 The SWIFT-Whittaker method for Bermudan options

The SWIFT-Whittaker method, as described in Chapter 4, heavily simplifies the application of the SWIFT method to Bermudan options. In Section 4.4, we developed a grid approach to compute a whole range of initial option values \( x = \{ x_p \}_{p \in \mathbb{Z}} \) efficiently when \( x_p := p/2^m \). Then, by the translation property of wavelets, \( \phi_{m,k}(x-x_p) = \phi_{m,k+p}(x) \), we can write the SWIFT pricing formula as,

\[
v(x_p, t_0) \approx e^{-r \Delta t} \sum_{k=k_1}^{k_2} D^*_{m,k} \int_{\mathbb{R}} v(y, t_1) \phi_{m,k+p}(y) \, dy \]

\[
=: e^{-r \Delta t} \sum_{k=k_1}^{k_2} D^*_{m,k} V_{m,k+p}(t_1),
\]

where the value coefficients \( V_{m,k}(t_1) \) are defined as,

\[
V_k(t_n) := \int_{\mathbb{R}} v(y, t_n) \phi_{m,k}(y) \, dy,
\]

and as a result of the translation property, these are independent of \( x \). To compute these value coefficients, we make use of the Whittaker-Shannon interpolation polynomial, stating that \( V_{m,k}(t_n) \approx 2^{-\frac{m}{2}} v(x_k, t_n) \).

At maturity, the option value equals the payoff value, and thus we find \( V_{m,k}(T) \approx 2^{-\frac{m}{2}} g(x_k) \). Next, stepping backwards in time, we find the following recursion for \( V_{m,k}(t_{n-1}) \),

\[
V_{m,k}(t_{n-1}) = \int_{\mathbb{R}} v(x, t_{n-1}) \phi_{m,k}(x) \, dx,
\]

\[
\approx 2^{-\frac{m}{2}} v(x_k, t_{n-1})
\]

\[
= 2^{-\frac{m}{2}} \max \left( g(x_k), c(x_k, t_{n-1}) \right).
\]

Here, the only unknown quantity is the continuation value \( c(x_k, t_{n-1}) \), which is a European option pricing problem, and by the application of the SWIFT method for European options we find,

\[
c(x_k, t_{n-1}) \approx e^{-r \Delta t} \sum_{p=k_1}^{k_2} D_{m,p} \int_{\mathbb{R}} v(y, t_n) \phi_{m,p+k}(y) \, dy
\]

\[
= e^{-r \Delta t} \sum_{p=k_1}^{k_2} D_{m,p} V_{m,p+k}(t_n).
\]
5.5. THE SWIFT-WhITTAKER METHOD FOR BERMUDAN OPTIONS

As we recall from the pricing of a grid of initial asset values \( x \), we can compute the whole vector of continuation values efficiently using the FFT when we realize that we can write (5.19) as a matrix-vector product with a Hankel matrix, as in Section 2.1.1, and we set,

\[
m_h := [F_0, F_{-1}, \ldots, F_k, 0, F_{k-1}, F_{k-2}, \ldots, F_1]^T,
\]
\[
x_h(t_n) := [V_1(t_n), V_{1+1}(t_n), \ldots, V_{k+1}(t_n), 0, \ldots, 0]^T,
\]

such that we find the continuation values \( c(x_k, t_{n-1}) \) by means of the FFT algorithm as the first \( N \) elements, in reversed order, of \( M = \text{ifft} \left( \text{fft}(x_h(t_n)) \odot \text{fft}(m_h) \right) \).

We summarize the SWIFT-Whittaker algorithm with application of the FFT in detail in the following algorithm.

Fix a wavelet scale \( m \) and determine the integral truncation parameters \( k_1 \) and \( k_2 \) by the cumulants method [FO08], and let \( N = k_2 - k_1 + 1 \);

Compute the density coefficients \( D_{m,k} \) for \( k = k_1, \ldots, k_2 \), by application of the fft algorithm;

Construct the vector \( m_h \) as in (5.21), and compute \( \mu_c := \text{fft}(m_h) \);

For maturity \( T \), set the value coefficients equal to the payoff value, i.e., \( V_{m,k}(T) = g(x_k) \);

for Now, iterate back in time, \( n = M - 1, \ldots, 1 \) do

Construct \( x_h \) as in (5.21) from the payoff coefficients in (5.19) ;

Compute the continuation values \( c(x_k, t_{n-1}) \) by means of the FFT algorithm as the first \( N \) elements, in reversed order, of \( M = \text{ifft} \left( \mu_c \odot \text{fft}(x_h(t_n)) \right) \); ;

Compute \( V_{m,k}(t_{n-1}) = 2^{-m/2} \max \{ g(x_k), c(x_k, t_{n-1}) \} \) for \( k = k_1, \ldots, k_2 \);

end

At \( t_0 = 0 \), we find the option value is given by \( v(x_p, t_0) = e^{-r \Delta t} \sum_{k=k_1}^{k_2} D_{m,k} V_{m,k+p}(t_n) \).

Algorithm 3: SWIFT-Whittaker method for pricing a Bermudan Put

Remark 5.4. Following the recursion we obtain the option value at initial time \( t_0 \) for a whole range of initial asset values \( x_p \). However, we are interested in \( x = \log(S/K) \). If there is no integer \( p \) such that \( x_p = x \), one could apply a spline interpolation on the vector \( v(x_0, t_0) \) to approximate the option value. This should be avoided when possible since this results in a loss of accuracy and increase of computation time. A better approach is to use an alternative log-asset transformation, i.e, use \( X_t = \log(S_t/S_0) \), such that \( x = X_0 = 0 \), which is a grid point.

The proposed algorithm is very efficient, as the FFT can be applied in every matrix-vector computation. We just require 2 FFT’s per time step, whereas COS requires 5, and determination of the early exercise point is free. The downside is accuracy of the option price approximation, since the approximation \( V_n(t_n) \approx \text{ifft} \left( \text{fft}(v(x, t_n)) \right) \) is not very accurate.

Also, due to the fact that we evaluate the payoff function only on grid points, the early exercise point is found for free, but it has to be on a grid point, which is not precise.

Example 13 (GBM Bermudan put). We price a Bermudan put option with 10 exercise moments on GBM. Convergence results with respect to the number of coefficients are shown in Figure 5.7. Again, we observe a convergence of about \( \mathcal{O}(N^{-2}) \) for the SWIFT-Whittaker method, although convergence is less regular due to inaccuracy in the determination of the early exercise point.

Again we see that up to an accuracy of \( 10^{-4} \), the SWIFT-Whittaker method is faster than both the COS and SWIFT method.

Example 14 (GBM Bermudan put with many exercise opportunities). An advantage of the SWIFT-Whittaker becomes clear when we price options with a lot of early-exercise opportunities. We price the same option as in Example 13, but with \( M = 100 \) exercise opportunities. Then, the boundary errors for the COS method cause slow convergence for the COS method, and the SWIFT method requires many coefficients to capture the peaked density function. Therefore, the SWIFT-Whittaker method is faster compared to the two other methods up to an accuracy of 5 decimals.

Example 15. CGMY Bermudan put We see a clear advantage for the SWIFT-Bermudan method when very irregular densities are approximated. In that case, a large wavelet scale \( m \) is required for the density.
approximation, which implies that this error dominates both the SWIFT and SWIFT-Whittaker method, resulting in a similar convergence for low approximation scales.

We price the Bermudan put option on CGMY dynamics from Test 3, and convergence is shown in Figure 5.9. The SWIFT-Whittaker method is significantly faster for an accuracy up to $10^{-7}$.

5.6 Bermudan and American Options

American style options are options which can be exercised at any moment prior to expiry, in contrast to Bermudan options, which can only be exercised at a few exercise moments prior to maturity.

American options also require an exercise strategy, and we price American options using the insight that the price of a Bermudan option converges to the price of an American option when the number of exercise opportunities $M$ goes to infinity.

The price of American options can be obtained by applying a Richardson extrapolation on the prices of a few Bermudan options with a small number of exercise moments. Let $v(M)$ denote the value of a Bermudan option with $M$ early-exercise moments. We use a 4-point Richardson extrapolation scheme [FO09],

$$v_{AM}(d) = \frac{1}{21} \left( 64v(2^d+3) - 56v(2^d+2) + 14v(2^d+1) - v(2^d) \right),$$

(5.22)
5.7 Barrier options

Another class of options closely related to Bermudan options is the class of discretely monitored barrier options. The payoff of a barrier option depends on whether or not the price of the underlying asset crosses a certain level (the barrier) during the options lifetime.

Barrier contracts are specified with an expiry date, a strike price and a barrier price. We discuss the four most common types of barriers, which can be written as either puts or calls.

(i) **Down-and-Out**: A down-and-out option gives the holder the right but not the obligation to buy or sell shares of an underlying asset at a pre-determined strike price so long as the price of that asset did not go below a pre-determined barrier during the option lifetime.

That is, once the price of the underlying asset falls below the barrier, the option is “knocked-out” and no longer carries any value. Hence the name down-and-out.

(ii) **Down-and-In**: Down-and-in options only carry value if the price of the underlying asset falls below the barrier during the options lifetime.

(iii) **Up-and-Out**: An up-and-out barrier option is similar to a down-and-out barrier option, the only difference being the placement of the barrier. Rather than being knocked out by falling below the
5. BERMDAN OPTION VALUATION

barrier price, up-and-out options are knocked out if the price of the underlying asset rises above
the predetermined barrier.

(iv) **Up-and-In:** An up-and-in barrier option is similar to a down-and-in option, however the barrier is
placed above the current price of the underlying asset and the option will only be valid if the price
of the underlying asset reaches the barrier before expiration.

In this section, we focus on *discretely monitored* barrier options, i.e., only on a discrete set of monitor
dates it is checked if the underlying asset has crossed the barrier.

We focus in this section on the pricing of discretely monitored down-and-out put (DOP) and call
(DESC) options and up-and-out put (UOP) and call (UOC) options.

Discretely monitored barrier options can be seen as Bermudan options with a pre-determined
exercise policy, i.e., the barrier. In the Bermudan case, when the asset price crosses the early exercise point, a
payoff occurs and the option expires immediately. Similar to barrier options, where the option expires
immediately when the barrier is crossed. In the case of barrier options however, no payoff occurs when
the barrier is breached.

We use the SWIFT method to price discretely monitored barrier options. We explain the method by
pricing a Down-and-Out put option (DOP) with barrier \(H\), strike \(K\) and \(M\) monitoring dates \(t_0 < t_1 < \ldots < t_M = T\) with \(\Delta t := t_n - t_{n-1}\). When, at any of the monitoring dates \(t_n\), the option value has
crossed the barrier, i.e. \(S_{t_n} \leq H\), the option expires worthless.

Taking the usual steps (and notation) as in the previous chapters, we apply a transform to the log-
asset domain, where \(X_t := \log(S_t/K)\), and thus the log-asset barrier becomes \(H := \log(H/K)\) and the
payoff is denoted \(g(x)\). Let \(W\) be the continuation region, in log-asset space, implied by any knock-out
barriers, i.e., if, for any \(t_n\), \(X_{t_n} \notin W\), the option expires worthless.

We price this option backwards in time via a recursion, and starting at maturity where the option
value is given by the payoff,

\[
v(y,T) := g(y) = K(e^y - 1)1_{y \in W}.
\]

Assuming that no rebates are awarded upon barrier breach, the option value at time \(t_{n-1}\) for \(n = 1, \ldots, M\)
is now given by,

\[
v(x,t_{n-1}) = e^{-r \Delta t} \int_{\mathbb{R}} v(y,t_n) 1_{y \in W} f_{\Delta t}(y|x) \, dy \\
= e^{-r \Delta t} \int_{W} v(y,t_n) f_{\Delta t}(y|x) \, dy.
\] (5.23)

This recursion is a simplification of the recursion for Bermudan options, where we have a pre-determined
exercise points, i.e., \(x^*_n \equiv H\), and upon ‘exercising’, the payoff of the option is zero.

The option value at time \(t_0\), assuming that the underlying asset is in the continuation region, is
approximated by the SWIFT method at scale \(m\),

\[
v(x,t_0) = e^{-r \Delta t} \int_{\mathbb{R}} v(y,t_1) 1_{y \in C} f_{\Delta t}(y|x) \, dy \\
\approx e^{-r \Delta t} \sum_{k=k_1}^{k_2} D^*_{m,k}(x) \int_{\mathbb{R}} v(y,t_1) 1_{y \in W} \phi_{m,k}(y) \, dy \\
:= e^{-r \Delta t} \sum_{k=k_1}^{k_2} D^*_{m,k}(x)V_{m,k}(W,t_n),
\] (5.24)

where the density coefficients \(D^*_{m,k}(x)\) are the SWIFT coefficients as usual, see Section 3.6.1. The value
coefficients \(V_{m,k}(t_1,W)\) are now time-dependent, and we added the continuation region as one of the
parameters. We propose a recursive scheme to compute \(V_{m,k}(t_1,W)\).
At maturity, $t_M = T$, the option value equals the payoff and we find,

$$V_{m,k}(W, T) := \int_{\mathbb{R}} v(y, t_M) 1_{\{y \in W\}} \phi_{m,k}(y) \, dy$$

$$= \int_{W} g(y) \phi_{m,k}(y) \, dy,$$

$$= G_{m,k}(W, T)$$

and these are just the payoff coefficient as we have seen them in the European case, Section 4.1, and can be approximated up to any desired accuracy with Vieta’s formula.

Then, to compute any $V_{m,k}(W, t_n)$ for $n = M - 1, \ldots, 1$, we use the insight that a barrier option between to exercise dates behaves like a European option, and we use the SWIFT method to approximate these integrals,

$$V_{m,k}(W, t_n) := \int_{\mathbb{R}} v(x, t_{n-1}) 1_{\{x \in W\}} \phi_{m,k}(x) \, dx$$

$$= \int_{W} \left( e^{-r \Delta t} \int_{\mathbb{R}} v(y, t_n) f_{\Delta t}(y|x) \, dy \right) \phi_{m,k}(x) \, dx,$$

$$\approx e^{-r \Delta t} \sum_{p=k_1}^{k_2} V_{m,p}(W, t_n) \int_{W} D_{m,p}(x) \phi_{m,k}(x) \, dx.$$

$$= C_{m,k}(W, t_{n-1})$$

We note that the value coefficients $V_{m,k}(t_{n-1}, W)$ are equivalent to the continuation coefficients in the Bermudan option pricing approach, see (5.12), where $[x_1, x_2] \equiv W$.

### 5.7.1 SWIFT Barrier pricing Algorithm

The SWIFT Barrier algorithm is summarized in Algorithm 4.

<table>
<thead>
<tr>
<th>Test No.</th>
<th>Model</th>
<th>$S_0$</th>
<th>$K$</th>
<th>$T$</th>
<th>$r$</th>
<th>$q$</th>
<th>Other Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>CGMY</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>0.05</td>
<td>0.02</td>
<td>$C = 4, G = 50, M = 60, Y = 0.7$</td>
</tr>
<tr>
<td>5</td>
<td>NIG</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>0.05</td>
<td>0.02</td>
<td>$\alpha = 15, \beta = -5, \delta = 0.5$</td>
</tr>
</tbody>
</table>

Table 5.2: Test parameters for pricing Barrier Options

We start by pricing a DOP under CGMY (Test 4), and the numerical results are shown in Table 5.3. As expected, the SWIFT method is more efficient for discretely monitored barriers than for Bermudan options because the barrier is known in advance. Exponential convergence is observed, an accuracy up to 10 decimal places is obtained in less than 10 milliseconds.
Next, we focus on the NIG model (see Appendix A.3), with parameters from Test 5 and repeat the barrier option tests in in Table 5.4. To reach the same level of accuracy as for the CGMY model, we observe that a slightly larger scale of approximation is required under NIG. This is because the NIG density function is more peaked with the parameters from Test 4.
Chapter 6

Conclusion

In this thesis, we have generalized the SWIFT option pricing method, based on Shannon wavelet expansions, to Bermudan and discretely-monitored barrier options.

We started with comparing three different methods for the pricing of European options. All of these methods are Fourier methods, based on the recovery of the underlying density function from the characteristic function via a series expansion. The difference of these methods lies in the basis used for this series expansion. We considered the state-of-the-art COS method [FO08], based on Fourier cosine expansions, the WA^[a,b] method [OGO13] based on B-spline expansions and the SWIFT method, based on Shannon wavelet expansions.

The global Fourier cosine basis is capable of very fast approximations, but it depends heavily on the a priori determination of a computation domain. Wavelet bases are used for local approximations, and depend less significantly on the a priori domain. The WA^[a,b] (j = 0) method uses the Haar wavelet basis, which is a non-smooth basis. Density functions in finance are however often very regular. The SWIFT method uses the smooth Shannon wavelet basis to approximate density functions with exponential converge, as the COS method, but still possesses the local properties of wavelets. The cost of this is that the Shannon wavelets are non-compactly supported, which causes truncation errors that appear when recursive pricing is applied, like for Bermudan options.

We reformulate the SWIFT method for options with bounded payoffs, like put options or binary options, such that no truncation is required, and thus further reducing the approximation error. To price call options, the put-call parity can be applied.

We derived a rule-of-thumb to determine a priori the required approximation scale m. Also, we present a test to determine a proper truncation of the asset-domain with a computational cost of O(N log_2 N), which is slightly expensive in the pricing of European options, but relatively cheap in the pricing of Bermudan options with many exercise moments. Therefore, we are able to determine all of the parameters of the SWIFT method in advance in a cheap manner.

We derived a novel approach with the help of the FFT to price a vector of options with different strikes, when the strikes prices are on an equidistant grid in the log-asset domain. The resulting order of complexity is O((N + K) log_2(N + K)), where N are the used coefficients and K is the number of strikes. This approach is faster than any of the other methods we considered when K > log_2 N.

We also present the SWIFT-Whittaker method, which is a simplified version of the SWIFT method using insights from sampling theory. The SWIFT-Whittaker method can be easily extended to the pricing of many option types, and we present the pricing of European and Bermudan options. The rate of convergence of the SWIFT-Whittaker method is generally low, but due to its extreme efficiency it outperforms the COS and SWIFT methods in many cases when only a few decimals of accuracy are required. Another advantage of this method is that it is easy to implementation and straightforward to adapt to new payoff styles.

The main result of this thesis is that the SWIFT pricing formula for European options from [OGO15] can be used for pricing Bermudan and barrier options, if the value coefficients at the first early-exercise (or monitoring) date are known. We presented an algorithm to recursively recover these coefficients from the payoff function.

The resulting computational complexity is O(MN log_2 N) for Bermudan and barrier options with M exercise, or monitoring, dates. The SWIFT method therefore has the same order of computational
complexity as the COS method, although about two times slower. The advantage of the SWIFT method is that all of the parameters can be determined a priori. We show that, although the SWIFT method also suffers from error propagation at the boundaries of the domain, this is less than the COS method does, and it is possible to ad hoc extend the computational domain.

We also propose an alternative method called the SWIFT-Whittaker method for Bermudan options, which is a computationally faster algorithm, but at a slower convergence rate. This method is competitive when payoff coefficients are relatively expensive, as in the case of Bermudan options with many exercise dates.

Future Research

- In this thesis we numerically showed exponential convergence for the SWIFT method in terms of the wavelet scale $m$. We have also shown how to heuristically select an appropriate $m$, however, further research is required to prove exponential convergence convergence in $m$, both for the European and Bermudan SWIFT method.

- There are more possibilities to exploit the local behavior of the wavelets. We compute all the payoff and continuation coefficients, even though many of them are close to zero. By using a compactly supported wavelet or a windowed Shannon wavelet, higher efficiency might be obtained.

- We did not do research into the computation of any of the Greeks with the SWIFT method, nor for Europeans, nor for Bermudans. This may be a point of interest.

- In the pricing of Bermudan options, the determination of the early-exercise point is done by means of Newton’s algorithm. This is relatively expensive since the density coefficients have to be recomputed each iteration with help of the FFT. We did not exploit the local properties of the wavelet approximation here. One could for example apply the multiple strikes pricing on a grid approach to approximate the early-exercise point in a cheaper manner.
Bibliography


Appendix A

Exponential Lévy models

As described in Section 3.3, we model the price process of the ‘risky’ asset \( S_t \) by,
\[
S_t = S_0 \exp (\mu t + L_t),
\]
where the Lévy process \( L_t \) is uniquely determined by its characteristic exponent \( \psi_L \) and \( \mu \) is a constant drift parameter. Each of the processes is uniquely defined by their characteristic exponent, \( \psi_L \), as in Theorem 3.4, given by,
\[
\psi_L(\omega) := \ln \mathbb{E}[\exp(i\omega L(1))].
\]

In Table A.1 a number of common Lévy process in finance and their parameters are listed. Below, we discuss their properties separately.

A.1 Geometric Brownian Motion (GBM)

The best known asset model is Geometric Brownian Motion (GBM). For GBM, the characteristic exponent is given by,
\[
\psi_{L[GBM]}(\omega; \sigma) := -\frac{1}{2} \sigma^2 \omega^2,
\]
where the volatility \( \sigma \) is a positive constant. GBM is basis for the Black-Scholes framework [BS73], and is the solution of the stochastic differential equation,
\[
dS_t = \mu S_t \, dt + \sigma S_t \, dW_t,
\]
with initial position \( S_0 \). When the underlying asset is modeled by GBM, the prices of a European put and call options can be computed analytically, and their expression is known as the Black-Scholes formula.

Brownian motion with drift is the only Lévy process with continuous paths. The processes discussed below belong to the class of jump models.

A.2 CGMY model

The CGMY model is introduced by Carr, Geman, Madan and Yor in [CGMY02], and named after the authors. This model for asset returns is a continuous time model that allows for both diffusions and for jumps of both finite and infinite activity.

The CGMY model is characterized by four parameters: \( C \geq 0 \) accounts for the overall activity level, \( G \geq 0 \) and \( M \geq 0 \) control the skewness, \( Y < 2 \) dictates the fine structure of the process. Specifically, \( Y < 0 \) determines a finite activity process, \( 0 \leq Y \leq 1 \) a process with finite variation but infinite activity, and \( 1 \leq Y < 2 \) a process of infinite activity and variation.

The CGMY risk-neutral characteristic exponent \( \psi_L \) is given by,
\[
\psi_{L[CGMY]}(\omega; C, G, M, Y) := C \Gamma(-Y) \left((M - i\omega)^Y - M^Y + (G + i\omega)^Y - G^Y\right).
\]
A. EXPONENTIAL LÉVY MODELS

<table>
<thead>
<tr>
<th>Model</th>
<th>( \psi_L(\omega) )</th>
<th>Param. Restric.</th>
<th>Cumulants</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSM</td>
<td>(-\frac{\omega^2}{2})</td>
<td>( \sigma &gt; 0 )</td>
<td>( c_1 = \mu t, c_2 = \sigma^2 t, c_4 = 0 )</td>
</tr>
<tr>
<td>NIG</td>
<td>( \delta \left( \sqrt{\alpha^2 - (\beta + i\omega)^2} - \sqrt{\alpha^2 - \beta^2} \right) )</td>
<td>( \alpha, \delta &gt; 0 )</td>
<td>( c_1 = \mu t + \delta \beta / \sqrt{\alpha^2 - \beta^2} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \beta \in (-\alpha, \alpha - 1) )</td>
<td>( c_2 = \delta \alpha \beta (\alpha^2 - \beta^2)^{-3/2} )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( c_4 = 3 \delta \alpha^2 (\alpha^2 + 4 \beta^2)(\alpha^2 - \beta^2)^{-7/2} )</td>
</tr>
<tr>
<td>Kou</td>
<td>(-\frac{\omega^2}{2} + \lambda \left( \frac{(1-p)\eta_1 + \eta_1}{\eta_1 + \omega} + \frac{\eta_1}{\eta_1 + \omega} - 1 \right) )</td>
<td>( \lambda &gt; 0 ), ( \eta_1, \eta_2 &gt; 0 )</td>
<td>( c_1 = t \left( \mu \frac{\eta_2}{\eta_1} + \frac{\lambda(1-p)}{\eta_2} \right) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \eta_1 &gt; 1, \eta_2 &gt; 0 )</td>
<td>( c_2 = t \left( \sigma^2 + 2 \lambda \frac{\eta_2}{\eta_1} + 2 \lambda(1-p) \right) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( p \in [0,1] )</td>
<td>( c_4 = 24 t \lambda \left( \frac{\eta_2}{\eta_1} + \frac{1-p}{\eta_2} \right) )</td>
</tr>
<tr>
<td>VG</td>
<td>( \frac{a^2}{2} \omega^2 - \frac{1}{\sigma^2} \log \left( 1 - i \nu \theta \omega + \nu^2 \omega^2 \right) )</td>
<td>( \nu &gt; 0 ), ( \sigma &gt; 0 )</td>
<td>( c_1 = t(\mu + \lambda \bar{\theta}) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( c_2 = t(\sigma^2 + \lambda \nu^2 + \bar{\theta}^2 \lambda) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( c_4 = 3 t^3(\sigma^2 \nu + 2 \nu^2 \bar{\theta} + 4 \nu^2 \lambda) )</td>
</tr>
<tr>
<td>CGMY</td>
<td>( CT(-\nu) \left( (M - i\omega)Y - M^Y \right) )</td>
<td>( C, G &gt; 0 )</td>
<td>( c_1 = \mu^T )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( M &gt; 1 )</td>
<td>( + C \Gamma(1-Y)(M^{Y-1} - G^{Y-1}) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( c_2 = \sigma^2 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( + C \Gamma(2-Y)(M^{Y-2} - G^{Y-2}) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( c_4 = C \Gamma(4-Y)(M^{Y-4} - G^{Y-4}) )</td>
</tr>
</tbody>
</table>

Table A.1: Symbols, parameters restrictions and cumulants for tractable Lévy processes. The drift parameter \( \mu \) is defined in (3.12).

A.3 Normal Inverse Gaussian (NIG)

The NIG(\( \alpha, \beta, \delta \)) return process \( Y_t \) [BN07] is constructed by time changing a standard Brownian motion via the inverse Gaussian subordinator \( I_t \), which has parameter \( a = 1 \) and \( b = \delta \sqrt{\alpha^2 - \beta^2} \):

\[
Y_t = \beta \delta^2 I_t + \delta W_t,
\]

where \( \alpha > 0 \), \( \beta \in (-\alpha, \alpha), \delta > 0 \). The continuous density function, while known in closed form, is given in terms of the modified Bessel function of the third kind. The risk-neutral Lévy symbol is given by

\[
\psi_{L[NIG]}(\omega; \alpha, \beta, \delta) := \delta \left( \sqrt{\alpha^2 - (\beta + i\omega)^2} - \sqrt{\alpha^2 - \beta^2} \right).
\]  

(A.1)

The class of NIG distributions contains fat-tailed and skewed distributions. The normal distribution belongs to this class as a special case by setting \( \beta = 0, \delta = \sigma^2 \alpha \), and letting \( \alpha \to \infty \).

A.4 Heston Model

The Heston model [Hes93] is proposed as an extension of the Black-Scholes model, where the asset is modeled by a geometric Brownian motion, but with stochastic variance. The variance process is a CIR process. Under the Heston model, the volatility, denoted by \( \sqrt{\theta_t} \), and is given by the system of equations,

\[
\begin{align*}
\frac{dx_t}{x_t} &= (\mu - \frac{1}{2} \bar{u}_t)dt + \sqrt{\theta_t}dW_t^1, \\
\frac{d\theta_t}{\theta_t} &= \lambda(\bar{u}_t - \theta_t)dt + \eta \sqrt{\theta_t}dW_t^2, \\
\frac{dW_t^1 dW_t^2}{W_t^2} &= \rho dt,
\end{align*}
\]

where \( x_t \) denotes the log-asset price variable and \( u_t \) the variance of the asset price process. Parameters \( \lambda \geq 0, \bar{u} \geq 0, \eta \geq 0 \) and \( \rho \in [-1,1] \) are called the speed of mean reversion, the mean level of variance, the volatility of variance and correlation, respectively.
The characteristic function of the log-asset price reads [FO08],
\[
\hat{f}_{\text{Heston}}(\omega; x) = e^{-i\omega x} \cdot \exp \left( -i\omega \mu dt + \frac{u_0}{\eta^2} \left( 1 - e^{-D\Delta t} \right) \left( \lambda + i\rho \eta \omega - D \right) \right) \cdot \exp \left( \frac{\lambda \bar{u}}{\eta^2} \left( \lambda + i\rho \eta \omega - D \right) \Delta t - 2 \log \left( \frac{1 - Ge^{-D \Delta t}}{1 - G} \right) \right),
\]
where the time to maturity \( \Delta t = T - t \),
\[
D = \sqrt{(\lambda - i\rho \eta \omega)^2 + (\omega^2 + i\omega)\eta^2)} \quad \text{and} \quad G = \frac{\lambda - i\rho \eta \omega - D}{\lambda - i\rho \eta \omega + D}.
\]
We note that we can write the characteristic function as \( \hat{f}_{\text{Heston}}(\omega; x) = e^{-i\omega x} \hat{f}_{\text{Heston}}(\omega; 0) \), just as for a Lévy process. Thus, the pricing formulas can be simplified equally. For Heston, the cumulants are given by,
\[
c_1 = \mu \Delta t + (1 - e^{-\lambda \Delta t}) \bar{u} - \frac{u_0}{2\lambda} - \frac{1}{2} \bar{u} \Delta t
\]
\[
c_2 \approx \bar{u}(1 + \eta) \Delta t,
\]
where \( c_2 \) is simplified for ease of implementation.
Appendix B

SWIFT-Whittaker 2D method

This work is based on early work by L. Ortiz-Gracia and G. Coldeolm-Papiol [OGCP15], where a bivariate density function is recovered using SWIFT. First, we introduce the required definitions for option pricing on two underlying assets. Then, we present the derivation of 2D SWIFT density coefficients from [OGCP15]. We use this derivation to show how the SWIFT-Whittaker method can be applied to the pricing of European and Bermudan style options on 2 underlying assets.

Remark B.1. The SWIFT-Whittaker 2D method is a first step in the pricing of 2D rainbow options with the SWIFT method. The results in Figure B.1 show that the method already requires many grid points to only obtain a low accuracy. However, the method presented here can be used as a reference method or guidance to the derivation of the SWIFT 2D method.

The bivariate Fourier transform is defined as,
\[ \hat{f}(u|x) = \int_{\mathbb{R}^2} f(y|x) e^{-iu \cdot y} dy, \]
where \( u = (u_1, u_2) \), \( x = (x_1, x_2) \) and \( y = (y_1, y_2) \).

MRA in higher dimensions is discussed in [Dau92]. The Multi Resolution Analysis can be extended to two dimensions with the 2D-Shannon scaling function,
\[ \phi_{m,k_1,k_2}(x_1, x_2) := \text{sinc}(2^m x_1 - k_1) \text{sinc}(2^m x_2 - k_2), \]
where we have chosen to use the same resolution \( m \) in both dimensions for simplicity. Then, a function \( f(x) \in L^2(\mathbb{R}^2) \) can be written as,
\[ f(x) = \lim_{m \to \infty} \sum_{k_1 \in \mathbb{Z}} \sum_{k_2 \in \mathbb{Z}} c_{m,k_1,k_2} \phi_{m,k_1,k_2}(x), \]
with convergence in the \( L^2(\mathbb{R}^2) \)-norm, and the 2D wavelet coefficients \( c_{m,k_1,k_2} \) are defined as,
\[ c_{m,k_1,k_2} := \int_{\mathbb{R}^2} f(x) \phi_{m,k_1,k_2}(x) dx. \]

B.1 European option pricing

Let \( X_t = (X^1_t, X^2_t) \) denote a 2D stochastic process representing the log-asset prices. We assume that a bivariate characteristic function of the stochastic process is known, which is the case for affine jump-diffusions [RO12]. The value of a European rainbow option, with payoff function \( g(\cdot) \), is given by the risk-neutral option valuation formula ([Shr04]),
\[ v(t_0, x) = e^{-r \Delta t} \int_{\mathbb{R}^2} g(y) f(y|x) dy. \]
Here, \( x = (x_1, x_2) \) is the current state, \( f(y_1, y_2|x_1, x_2) \) is the conditional density function, \( r \) is the risk-free rate and the time to expiry \( \Delta t := T - t_0 \).
Similar to the one dimensional case, we replace the conditional density function by its series expansion at resolution \( m \), and by the translation property of Lévy functions, we find,

\[
f(y|x) = f(y - x|0) \approx f_1(y - x|0) := \sum_{k_1 \in \mathbb{Z}} \sum_{k_2 \in \mathbb{Z}} D_{m,k_1,k_2} \phi_{m,k_1,k_2}(y - x),
\]

where the 2D density coefficients are defined as

\[
D_{m,k_1,k_2} := \int_{\mathbb{R}^2} f(y|0) \phi_{m,k_1,k_2}(y) \, dy.
\]

**Remark B.2.** We can use the SWIFT-Whittaker method for options on any underlying process with known characteristic function, and are not limited to Lévy processes in the European case. Only in the Bermudan case or when pricing multiple strikes, the independent and stationary increments of Lévy processes are of importance.

Then, the first approximation of the option value becomes,

\[
v(t_0, x) \approx v_1(t_0, x) := e^{-r \Delta t} \sum_{k_1 \in \mathbb{Z}} \sum_{k_2 \in \mathbb{Z}} D_{m,k_1,k_2} \int_{\mathbb{R}^2} g(y) \phi_{m,k_1,k_2}(y - x) \, dy.
\]

We define the payoff coefficients as,

\[
G_{m,k_1,k_2}(x) := \int_{\mathbb{R}^2} g(y) \phi_{m,k_1,k_2}(y - x) \, dy.
\]

We truncate the summation range such that \( k_1 = l_1, \ldots, u_1 \) and \( k_2 = l_2, \ldots, u_2 \), and thus the 2D SWIFT pricing formula becomes,

\[
v(t_0, x) \approx v_2(t_0, x) := e^{-r \Delta t} \sum_{k_1=l_1}^{u_1} \sum_{k_2=l_2}^{u_2} D_{m,k_1,k_2} G_{m,k_1,k_2}(x).
\]

Next, we use the Whittaker-Shannon interpolation polynomial to approximate the payoff coefficients,

\[
G_{m,k_1,k_2}(x) \approx G^W_{m,k_1,k_2}(x) := 2^{-m} g(2^{-m} k - x).
\]

Thus, the 2D SWIFT-Whittaker pricing formula is given by

\[
v(t_0, x) \approx v_3(t_0, x) := e^{-r \Delta t} 2^{-m} \sum_{k_1=l_1}^{u_1} \sum_{k_2=l_2}^{u_2} D_{m,k_1,k_2} g(2^{-m} k - x).
\]

In the next section, we discuss how to efficiently compute the density coefficients using the FFT.

### B.2 2D SWIFT Density coefficients

As in [OGCP15], an efficient algorithm to approximate the density coefficients,

\[
D_{m,k_1,k_2} := \int_{\mathbb{R}^2} f(y|0) \phi_{m,k_1,k_2}(y) \, dy,
\]

is presented when the characteristic function is known. The approach is very similar to the density coefficients for the 1D case. We approximate the sinc functions with Vieta’s formula into a sum of cosine function. Then, we recognize the real part of the characteristic function. The difficult part here is to write everything in a form such that the 2D FFT can be applied.

By application of Vieta’s formula (3.35), we approximate the 2D Shannon scaling function as,

\[
\phi_{m,k_1,k_2}(y) \approx \frac{2^m + 2}{N^2} \sum_{j_1=1}^{N/2} \sum_{j_2=1}^{N/2} \cos(\omega_{j_1} (2^m y_1 - k_1)) \cos(\omega_{j_2} (2^m y_2 - k_2)),
\]
where \( \omega_j := \frac{2j-1}{N} \pi \). We use in both dimensions the same truncation parameter \( N \) for simplicity reasons only.

Then, the density coefficients can be approximated by,

\[
D_{m,k_1,k_2} := \int f(y)\phi_{m,k_1,k_2}(y)
\]

\[
\approx \frac{2^{m+2}N/2}{N^2} \sum_{j_1,j_2=1}^{N/2} \int f(y)\cos(\omega_{j_1}(2^m y_1 - k_1))\cos(\omega_{j_2}(2^m y_2 - k_2))
\]

\[
=: D_{m,k_1,k_2}^*.
\]

Now, we use the goniometric relation ([RO12]),

\[
2\cos(\alpha)\cos(\beta) = \cos(\alpha + \beta) + \cos(\alpha - \beta)
\]

d and we obtain,

\[
D_{m,k_1,k_2}^* = \frac{2^{m+1}N/2}{N^2} \sum_{j_1,j_2=1}^{N/2} \left[ \int f(y)\cos(\omega_{j_1}(2^m y_1 - k_1)) + \omega_{j_2}(2^m y_2 - k_2))
\]

\[
+ \int f(y)\cos(\omega_{j_1}(2^m y_1 - k_1)) - \omega_{j_2}(2^m y_2 - k_2))
\]

\[
= \frac{2^{m+1}N/2}{N^2} \sum_{j_1,j_2=1}^{N/2} \sum_{j_2=1-N/2}^{N/2} \int f(y)\cos(\omega_{j_1}(2^m y_1 - k_1)) + \omega_{j_2}(2^m y_2 - k_2))
\]

where the last equality follows from \(-\omega_{j_2} = \omega_{1-j_2}\), which allows us to write both integrals in a single summation over negative \( j_2 \). In a final step, we use that \( \cos(\alpha) = \Re \{e^{-i\alpha}\} \) to find,

\[
D_{m,k_1,k_2}^* = \frac{2^{m+1}N/2}{N^2} \Re \left\{ \sum_{j_1=1}^{N/2} \sum_{j_2=1-j_2}^{N/2} \int f(y) e^{-i\omega_{j_1}(2^m y_1 - k_1)) - i\omega_{j_2}(2^m y_2 - k_2))}
\]

\[
= \frac{2^{m+1}N/2}{N^2} \Re \left\{ \sum_{j_1,j_2=1}^{N/2} \sum_{j_2=1-N/2}^{N/2} e^{i\omega_{j_1}k_1+i\omega_{j_2}k_2} \int f(y) e^{-i\omega_{j_1}2^m y_1 - \omega_{j_2}2^m y_2}
\]

\[
= \frac{2^{m+1}N/2}{N^2} \Re \left\{ \sum_{j_1,j_2=1}^{N/2} \sum_{j_2=1-j_2}^{N/2} \hat{f}(\omega_{j_1}2^m,\omega_{j_2}2^m|0) e^{i\omega_{j_1}k_1+i\omega_{j_2}k_2}
\]

where \( \hat{f}(u|0) \) is the Fourier transform of the density function \( f(y|0) \). The characteristic function is given by \( \hat{f}(u|0) = \hat{f}(-u|0) \). This representation of the density coefficients can be directly implemented as the characteristic function is known. We now show how to efficiently compute the coefficients with help of the FFT. We use the Matlab FFT implementation, which is defined for \( k = 1,\ldots,N \) as,

\[
D_k(\{x_j\}_{j=1}^N) := \sum_{j=1}^N x_j e^{-i\frac{2\pi}{N}(j-1)(k-1)}.
\]

First, we define an intermediate value, which we can compute with the help of the FFT,

\[
H_{j_1,k_2} := \sum_{j_2=1-N/2}^{N/2} \hat{f}(\omega_{j_1}2^m,\omega_{j_2}2^m) e^{i\omega_{j_1}k_2}
\]

\[
= \sum_{j_2=1}^N \hat{f}(\omega_{j_1}2^m,\omega_{j_2-N/2}2^m) e^{i\frac{2\pi}{N}(j_2-1)k_2 - i\frac{N-1}{2}\pi k_2}
\]

\[
= e^{-i\frac{N-1}{2}\pi k_2} D_{1-k_2} \left( \left\{ \hat{f}(\omega_{j_1}2^m,\omega_{j_2-N/2}2^m) \right\}_{j_2=1}^N \right).
\]
Now, the density coefficients $D^*_{m,k_1,k_2}$ are given by,

$$D^*_{m,k_1,k_2} = \frac{2^{m+1}}{N^2} \text{Re} \left\{ \sum_{j_1=1}^{N/2} \mathcal{H}_{j_1,k_2} e^{i\omega_j k_1} \right\},$$

which we can rewrite in a form suitable for Matlab’s FFT as well by basic algebraic manipulations, resulting in,

$$D^*_{m,k_1,k_2} = \frac{2^{m+1}}{N^2} \text{Re} \left\{ e^{i\frac{\pi}{N}k_1 \mathcal{D}_{1-k_1}} \left( \{\mathcal{H}_{j_1,k_2}\}_{j_1=1}^N \right) \right\},$$

where we appended zeros to $\mathcal{H}_{j_1,k_2} = 0$ for $j_1 = N/2 + 1, \ldots, N$.

Define the number of coefficients in each dimension as $K_1 := u_1 - l_1 + 1$ and $K_2 := u_2 - l_2 + 1$. Then, the straightforward approach to compute the 2D density coefficients would result in a computational complexity of $\mathcal{O}(K_1 K_2 N^2) \sim \mathcal{O}(N^4)$. Using this FFT approach, the computational complexity is only $\mathcal{O}(N^2 \log_2 N)$.

We shortly mention the well known Geometric Brownian motion in two dimensions in the next example.

**Example 16 (2D Geometric Brownian Motion).** Under GBM, the risk-neutral asset prices $S = (S_1, S_2)$ evolve according to the following dynamics,

$$dS_i^t = \mu_i S_i^t dt + \sigma_i S_i^t dW_i^t, \quad i = 1, 2,$$

where $W_t = (W_1^t, W_2^t)$ is a 2D correlated Wiener process with correlation $dW_i^t dW_j^t = \rho_{ij} dt$. We switch to the log-process $X_i^t := \log S_i^t$, driven by,

$$dX_i^t = (r - \frac{1}{2} \sigma_i^2) dt + \sigma_i dW_i^t.$$

The log-asset prices at time $t$ given the current state at $t_0 = 0$ are bivariate normally distributed,

$$X_t \sim \mathcal{N}(X_0 + \mu \Delta t, \Sigma),$$

where $\mu_i = r - \frac{1}{2} \sigma_i^2$ and covariance matrix $\Sigma_{ij} = \sigma_i \sigma_j \rho_{ij} \Delta t$. The characteristic function reads as $\hat{f}(\mathbf{u}|\mathbf{x}) = \hat{f}(-\mathbf{u}|\mathbf{x}) = e^{-i \mathbf{u}^T \mathbf{x}} f(-\mathbf{u}|0) = e^{-i \mathbf{u}^T \mathbf{u}} f_{\text{levy}}(-\mathbf{u})$, and for GBM,

$$\hat{f}_{\text{levy}}(\mathbf{u}) = \exp(i \mathbf{u}^T \Sigma \mathbf{u} - \frac{1}{2} \mathbf{u}^T \Sigma \mathbf{u}).$$

We are now ready to run a numerical example.

**Example 17 (2D European call spread on GBM).** We price a 2D European call spread option with strike $K$ on two assets driven by 2D correlated GBM. We use the parameters from problem 6 of the BenchOP project [vSea15], i.e., $r = 0.03$, $\sigma_1 = \sigma_2 = 0.15$, $\rho = 0.5$, $K = 0$, and $T = 1$. The payoff function for the European call spread is given by,

$$g(y_1, y_2) = \max(e^{y_1} - e^{y_2}, 0).$$

We use the COS method with $L = 10$ and $N = 200$ to find a reference price, and the results are shown in Figure B.1. We observe the same convergence rate as with all other examples, a price error convergence of $\mathcal{O}(N^{-2})$, although the SWIFT-Whittaker is only faster up to an accuracy of $10^9$.

The obvious next step to increase accuracy is to efficiently compute the payoff coefficients,

$$G_{m,k_1,k_2}(x) := \int_{\mathbb{R}^2} g(y) \phi_{m,k_1,k_2}(y-x) \, dy$$

without the use of the Shannon-Whittaker approximation. This however is future work.
Figure B.1: Pricing a 2D GBM European Call Spread with parameters \( r = 0.03, \sigma_1 = \sigma_2 = 0.15, \rho = 0.5, K = 0, \) and \( T = 1. \) Reference price by the COS method.

### B.3 SWIFT-Whittaker 2D Bermudan

We generalize the SWIFT-Whittaker method to Bermudan rainbow options with a 2D underlying log-asset price process \( \mathbf{X}_t = (X_1^t, X_2^t) \), that is in the class of Lévy processes. A Bermudan option can be exercised at a fixed set of \( M \) early-exercise times \( t_0 < t_1 < \cdots < t_n < \cdots < t_M = T \), with \( \Delta t := t_{n+1} - t_n \).

The payoff function is denoted by \( g(\cdot) \). The problem is solved backwards in time, with
\[
\begin{align*}
\nu(t_M, \mathbf{x}) &= g(\mathbf{x}), \\
c(t_{n-1}, \mathbf{x}) &= e^{-r\Delta t}[E[g(t_M, \mathbf{X}_{t_{n-1}})|\mathbf{X}_{t_{n-1}} = \mathbf{x}]], \\
v(t_{n-1}, \mathbf{x}) &= \max(g(\mathbf{x}), c(t_{n-1}, \mathbf{x})), \quad 2 \leq n \leq M \\
v(0, \mathbf{x}_0) &= c(0, \mathbf{x}_0).
\end{align*}
\]

Function \( c(t_{n-1}, \mathbf{x}) \) is called the continuation value and is approximated by the 2D SWIFT-Whittaker formula on the grid points \( \mathbf{x} = 2^{-m}\mathbf{k} \), with \( \mathbf{k} = (k_1, k_2) \) and \( k_1 = l_1, \ldots, u_1 \), and \( k_2 = l_2, \ldots, u_2 \), giving
\[
c(t_{n-1}, 2^{-m}\mathbf{p}) \approx c^*(t_{n-1}, 2^{-m}\mathbf{p}) := e^{-r\Delta t} \sum_{k_1=l_1}^{u_1} \sum_{k_2=l_2}^{u_2} D_{m,k} V_{m,k}(t_n, 2^{-m}\mathbf{p}),
\]
where the value coefficients are time dependent and given by
\[
V_{m,k}(t_n, \mathbf{x}) := \int_{\mathbb{R}^2} \nu(t_n, \mathbf{y}) \phi_{m,k}(\mathbf{y} - \mathbf{x}) \, d\mathbf{y}.
\]

Now, at a grid point \( 2^{-m}\mathbf{p} \), we find \( V_{m,k}(t_n, 2^{-m}\mathbf{p}) = V_{m,k+p}(t_n, 0) =: V_{m,k+p}(t_n) \).

We approximate the value function by \( \nu(t_{n-1}, \mathbf{x}) \approx \tilde{\nu}^*(t_{n-1}, \mathbf{x}) := \max(g(\mathbf{x}), c^*(t_{n-1}, \mathbf{x})) \) and we propose a backward recursion to recover the value coefficients. Using Shannon-Whittaker, we find for \( 2 \leq n \leq M \) the relation
\[
V_{m,k}(t_n) \approx 2^{-\frac{m}{2}} \nu(t_n, 2^{-m}\mathbf{k}) = 2^{-\frac{m}{2}} \max(g(2^{-m}\mathbf{k}), c(t_{n-1}, 2^{-m}\mathbf{k})).
\]

Thus, the resulting SWIFT-Whittaker recursion is given by,
\[
\begin{align*}
V_{m,p}(t_M) &= 2^{-\frac{m}{2}} g(2^{-m}\mathbf{p}), \\
c^*(t_{n-1}, 2^{-m}\mathbf{p}) &= e^{-r\Delta t} \sum_{k} D_{m,k} V_{m,k+p}(t_n), \\
V_{m,p}(t_{n-1}) &= \max(g(2^{-m}\mathbf{p}), c^*(t_{n-1}, 2^{-m}\mathbf{p})), \quad 2 \leq n \leq M \\
v(0, \mathbf{x}_0) &= c^*(0, \mathbf{x}_0).
\end{align*}
\]

In the previous section, we discussed how to compute the 2D SWIFT Density coefficients using the FFT. Now, the remaining step is to compute the vector of continuation values \( c^*(t_{n-1}, 2^{-m}\mathbf{p}) \) efficiently. For each of the coefficients, a double summation is required. This would result in a computational complexity of \( \mathcal{O}(N^3) \), which will be most expensive part of the computation, thus should be avoided.
To speed up the computation, we rewrite the computation of the continuation values as a matrix-vector product with a Hankel matrix. Let $K_1 := u_1 - l_1 + 1$, i.e., the number of coefficients in the first asset dimension, and similarly, let $K_2 := u_2 - l_2 + 1$. Then we define a change of variables,

$$j := (1 - l_1 + k_1) + (K_1 - 1)(1 - l_2 + k_2),$$

such that $j = 1, 2, \ldots, K_1 K_2$. We can now write $c^*(t_{n-1}, 2^{-m}p)$ in terms of this new notation, which uniquely maps the matrix of continuation values into a vector. We do the same for the double summation, where we uniquely map $k$ into $q = 1, \ldots, K_1 K_2$ such that it becomes a single summation, i.e.,

$$c_j(t_{n-1}) = e^{-r \Delta t} \sum_{q=1}^{K_1 K_2} D_{m,q} V_{m,q+j}(t_n).$$

Now, we obtained the same matrix-vector notation as in the one-dimensional case (5.20), thus we can compute the whole vector of continuation coefficients using the FFT as it resembles a matrix-vector multiplication with a Hankel matrix.