On the Application of Shannon Wavelet Inverse Fourier Techniques

An Extension to Asian Option Valuation and European Option Pricing under the SABR Model

Emma Ingeleiv Wagner

TU Delft

On the Application of Shannon Wavelet Inverse Fourier Techniques

An Extension to Asian Option Valuation and European Option Pricing under the SABR Model

by

Emma Ingeleiv Wagner

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Friday October 27, 2017 at 16:00 PM.

Student number:	4004949	
Project duration:	February, 2017 – October, 2017	
Thesis committee:	Prof. dr. ir. C. W. Oosterlee,	TU Delft, (supervisor)
	Dr. L. Ortiz-Gracia,	Universitat de Barcelona (supervisor)
	Dr. P. Cirillo,	TU Delft

This thesis is confidential and cannot be made public until October 27, 2017.

TU Delft

Acknowledgments

On Monday morning September 6th, 2010 the adventure Applied Mathematics began. Now, 7 years later, I am writing this from the library of the Delft University of Technology two weeks before I present my master thesis project and will receive my master's degree. The last 7 years as a student Applied Mathematics have been a wonderful experience and I would like to take some time to thank all the people that have contributed to the amazing journey I have embarked on.

First, I would like to thank the Board of Examiners of Applied Mathematics, that have made it possible for me to continue with the second year of the bachelor Applied Mathematics. Without the exception they made for me, I would not have been so close to receiving my masters degree in Applied Mathematics. After receiving the bachelor's degree, I continued with the master's program and because of the minor Finance, I knew which specialization to choose: Financial Engineering. Within the minor Finance, the interesting lectures on applied mathematics and derivative pricing, taught by professor Kees Oosterlee and Pasquale Cirillo, sparked my interest in Computational Finance.

Before starting this project, I had not seized the opportunity to study abroad. That is why I went to prof. Kees Oosterlee to discuss if there was a possibility to do research in the field of computational finance abroad. Prof. Kees Oosterlee introduced me to Luis Ortiz-Gracia and I was more than welcome to do research at the University of Barcelona under his supervision. Last February I hopped on a plane to Barcelona to work on the SWIFT method. I felt right at home at the University of Barcelona, thanks to the great guidance of Luis Ortiz-Gracia. When I returned to Delft to finish the thesis, prof. Kees Oosterlee has been a great supervisor. Without his advise and guidance, this thesis would not have been as it is. Thus, my most sincere gratitude goes out to both my supervisors Luis Ortiz-Gracia and prof. Kees Oosterlee for their support and advice throughout this thesis. Also regarding this thesis, I would like to thank my Uncle Kjell (with a capital U) for his corrections on my written English. As well am I indebted to Alvaro Leitao, who has helped with the Matlab implementation of the valuation of arithmetic Asian options, without his help the program would not have ran so fast.

The 6 months I lived in Barcelona have been a great experience, thanks to my friends that made Barcelona feel like home. Guillem, Toni, Jaime and Estrella for being the best flatmates ever. Gemma, thank you for introducing me to Catalan hot chocolate, for the good conversations (in both Dutch and Spanish) and for all the fun times we had in Catalunya! I am beyond grateful to have spent 6 months in the same office with you.

Furthermore, I would like to thank my friends in Delft. The adventures we experienced together the past 8 years in Delft, were awesome because I got to experience them with you. Especially the girls from Vesper: Marloes, Simone, Anne Louise, Anne, Liona, Marijn en Karien. Together we embarked on a marvelous journey, containing of weekend getaways, D.S.V. Sint Jansbrug parties and weekly dinners on Tuesdays. Then, I would like to thank "De wiskundemeisjes", as well as Vera and Jasper, for the great projects we did together during our studies and the coffee breaks on the faculty.

Last, but most certainly not least, I would like to thank Odd and Tineke, my parents, for providing a safe haven in Heemskerk, where I was always welcome for a good meal, a good night sleep and fun family time.Finally, I would like to thank Anne and Florine, my sister and best friend. You have been

the best support crew, whenever I felt insecure and/or was stressing about an upcoming exam, you were just one call away.

Within two weeks, my Applied Mathematics adventure will come to an end, but new adventures are just around the corner!

Emma Ingeleiv WagnerDelft, 15 October 2017

Abstract

This work is on the extension of the SWIFT method to option pricing problems where the sum of lognormals occurs. The SWIFT method (Shannon Wavelet Inverse Fourier Technique) is extended to the valuation of geometric Asian options and arithmetic Asian options with a Lévy process as underlying price process and the valuation of European options under SABR dynamics. In both applications a sum of lognormals (or sum of increments) occurs. The main result in this thesis is the SWIFT-SIA method (*SWIFT sinc integral approximation*), which is applied to the valuation of arithmetic Asian options as well as to the valuation of European options under the SABR model. Within the SWIFT-SIA method the recovery of the probability density function is obtained by an approximation, instead of a numerical integration method, which results in a very fast method compared to an alternative method based on cosine expansions as well as high accuracy in the option values.

Keywords: Option pricing, Asian options, Lévy processes, SABR model, Shannon wavelets, Fourier transform.

Contents

List of Figures

List of Tables

1

Introduction

In the field of Computational Finance, options often need to be priced at high speed with high accuracy since the resulting prices are used to calibrate financial models and determine a price to a complex financial product. Option pricing has been introduced by Merton, Black and Scholes [4] and their Black-Scholes formula has been popular ever since. The Black-Scholes formula is an analytical solution of the price of an European option under the Geometric Brownian Motion (GBM) asset price process. The simplest example of an option is the European option, which under the correct assumptions results in an analytical solution, for example under GBM dynamics as computed by the Black-Scholes formula. Nowadays, there are a lot of different types of derivative contracts on the market, each type of contract requires its own method in order to compute the fair price of this contract.

The start of each option pricing problem is the risk neutral pricing formula, which states that the fair price of an option is determined by the discounted expectation of the payoff function under the risk neutral measure. In order to apply this formula, one needs the probability density function corresponding to the underlying stochastic process. For some underlying stochastic processes the probability density function is not known, which makes option pricing involved. Therefore, option pricing techniques have been developed that approximate the probability density function. Option pricing methods can be divided into three categories: *partial differential equation pricing methods* (PDE methods), *Monte Carlo methods* and *Fourier and numerical integration methods*.

In this thesis we focus on the class of Fourier and numerical integration methods. This class of methods is known for its computational efficiency, since these methods rely on recovering the characteristic function instead of the probability density function. For several log-asset price processes the characteristic function does exist, whereas the probability density function is not known in closed-form. The characteristic function of a stochastic process coincides with the Fourier transform of the probability density function, thus the computation takes place in Fourier space.

The Fourier pricing methods have become more popular after the publication of the Carr-Madan method, [5]. Nowadays, the COS method is a state-of-the-art pricing method, which is based on a cosine expansion of the probability density function. The COS method has been applied to several types of options and is a highly efficient method to compute European option prices and the method has been extended to the valuation of path dependent options, such as Asian options [25].

Another way to approximate the probability density function is by applying wavelet functions. The first method to introduce a wavelet expansion to price options was the $\text{WA}^{[a,b]}$ method [19]. In [19] the recovery of the probability density function is done by Haar wavelets and B-spline wavelet approximations. Shannon wavelets, which are based on the cardinal sine (*sinc*) function can also be used to approximate the probability density function of the underlying, which results in the method known as the SWIFT method [20]. The SWIFT method has shown great results in terms of accuracy and speed

compared to the COS method in the valuation of European options [20] and has been extended to the valuation of Bermudan options [17].

This thesis extends the applications of the SWIFT method to option pricing problems where the sum of lognormals is present. The sum of lognormals (in the case of a GBM underlying process) or the sum of independent increments (in the case of general Lévy processes) occurs in the valuation of Asian options. Therefore the SWIFT method will be applied to the valuation of Asian options and in particular we will focus on the arithmetic Asian options. The same techniques developed for Asian options can be applied to the Stochastic Alpha Beta Rho (SABR) model.

The SABR model is a stochastic volatility model which is often used to price interest rate derivatives, since the model tries to mimic the volatility smile in the market. The SABR model consists of one forward S_t with volatility σ_t, both S_t and σ_t are random variables. Due to the stochastic volatility of the SABR model, option pricing under SABR dynamics is quite involved. Hagan et al. [14] have developed the model, and provided an approximation for the implied volatility, which is often used by practitioners to calibrate their market parameters. Unfortunately, the approximation is not always as accurate as desired, for example in pricing long maturity options. Leitao et al. [15] have developed a Monte Carlo method which shows high accuracy for contracts up to two years. In their work a sum of lognormals occurs, which shows a good opportunity to test the SWIFT method under SABR dynamics. The method shows great similarity with the recovery of the price of an arithmetic Asian option, thus the application of the SWIFT density approximation could result in a more robust and efficient method to compute European options under SABR dynamics.

1.1. Outline

The thesis is organized as follows. Chapter 2 provides the mathematical and financial background that will be used in this work, such as the definition of the characteristic function and the class of Lévy processes. With all the mathematical and financial background discussed, Chapter 3 will introduce the SWIFT method. First an introduction to wavelet theory will be given, then the SWIFT method to recover the probability density function of the underlying will be introduced. Chapter 3 discusses one application of an option pricing problem to illustrate the method in practice.

Chapters 4 and 5 form the heart of the thesis. In these two chapters the SWIFT method will be applied to the two aforementioned option pricing problems. Chapter 4 focuses on the pricing of arithmetic Asian options with different underlying Lévy processes. The chapter will conclude with numerical results. To check the performance of the SWIFT method in terms of CPU time and accuracy in arithmetic Asian option pricing, experiments will be conducted for three different underlying processes and compared to the COS method.

The paper by Leitao et al.[15] offers another way to analyze the performance of the SWIFT method. Therefore Chapter 5 illustrates in what way the one time-step Monte Carlo method can benefit from the SWIFT density approximation. At first, the method will be explained, where the focus is on Section 5.2.1. The application of SWIFT will be dealt with and the chapter concludes with numerical results.

In Chapter 6 we will draw conclusions and the results on the difference between the COS method and the SWIFT method will be discussed. Finally we will give recommendations for future research on this subject.

2

The Sum of Lognormals Problem

This thesis focuses on derivative pricing problems in which a *sum of lognormals* occurs. A *derivative* is a security which price depends on, or derives from, one or more underlying assets. The most common underlying assets of derivative contracts are stocks, interest rates, commodities and foreign exchange rates. Most known derivative contracts are *bonds*, *futures*, *swaps*, *credit derivatives* and *options*. If the price of a future or an option depends on a stock at the stock exchange, that stock will be denoted as the underlying asset of the derivative here.

The motivation in this thesis are option pricing problems, therefore let us further develop our knowledge of options. Options can basically be divided in two types: call options and put options. A call option gives the holder (buyer) of the option the right to sell the underlying asset for a fixed price to the writer of the option (seller). A put option gives the holder the right to buy the underlying asset for a fixed price from the writer. The fixed price is known as the strike price, K. Note that in both cases the holder has the *right* to exercise the option, not the *obligation*, which results in a non-negative payoff. The payoff functions of European options (vanilla options) are given by

$$V_{Call}(S(T),T) = \max\{S(T)-K,0\} \;\&\; V_{Put}(S(T),T) = \max\{K-S(T),0\},$$

with $S(\cdot)$ the price of the underlying and T the maturity date of the contract. We have discussed the difference between call and put options, but options come in different shapes and sizes. Another way to categorize options is by their allowed exercise date. The holder of an European option is only allowed to exercise the option at maturity T, whereas the holder of an American option may exercise the option from the day of the purchase up to and including the maturity day T. Any option that differs from the European or American option is denoted as an *exotic option*. These options are more involved, which is implied by the name 'exotic'. The Asian option is an example of an exotic option, where the payoff is dependent on the average of the underlying stock process. The path-dependency of the option results in a more complicated pricing procedure compared to European options.

For each derivative security that is being traded, the right price has to be determined. The starting point for an option pricing problem is the risk neutral pricing formula, which states that the price of the option is the discounted expectation of the payoff-function, i.e.

$$V(S_0,t_0) = e^{-r(T-t_0)}\mathbb{E}^{\mathbb{Q}}[V(S(T),T)], \tag{2.1}$$

where r the risk neutral interest rate and $\mathbb{E}$ is the expectation with respect to the risk neutral measure $\mathbb{Q}$.

As mentioned, this thesis focuses on option pricing problems in the presence of a sum of lognormals. The sum of lognormally distributed random variables (*lognormals*) occurs in several option pricing problems. Unfortunately, the distribution of the sum of lognormals is not known in closed-form, therefore numerical approximation techniques are necessary to compute the distribution. This work consists of an approximation technique for the probability density function by means of its characteristic function.

This chapter provides the necessary background in order to apply the SWIFT method. The sum of lognormals will be defined, as well as the class of Lévy processes, which will be used throughout this work as the class of stochastic processes for the underlying. The characteristic function plays an important role in the approximation of the distribution, therefore we will define the characteristic function, the Fourier transform and state some useful properties of the two.

2.1. Sum of Lognormals

Intuitively one might expect that the sum of lognormally distributed random variables has a closed form distribution as in the case of the sum of normal random variables and squared normal random variables. Let us define $N \in \mathbb{N}$ independent normal random variables $X_i, i = 1,\dots,N$, with mean μ_i and variance σ_i^2, i.e. $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

Lemma 2.1. *The sum* $X = \sum_{i=1}^{N} X_i \sim \mathcal{N}\left(\sum_{i=1}^{N} \mu_i, \sum_{i=1}^{N} \sigma_i^2\right)$.

Lemma 2.1 will be proven for $N = 2$, but the extension to $N \in \mathbb{N}$ follows by repeating the same procedure.

Proof. Let us define two normal random variables X_1, X_2 with mean and variance μ_1, σ_1^2 and μ_2, σ_2^2 respectively. Then let us define the sum of the two by Z, $Z = X_1 + X_2$. The probability density function of the random variable Z can be computed by the convolution

$$f_Z(z) = \int_{\mathbb{R}} f_{X_2}(z - x_1) f_{X_1}(x_1) dx_1.$$

Since X_1 and X_2 are normally distributed, it follows that

$$\begin{aligned}
f_Z(z) &= \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(z - x_1 - \mu_2)^2}{2\sigma_2^2}\right) \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(x_1 - \mu_1)^2}{2\sigma_1^2}\right) dx_1 \\
&= \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\sqrt{2\pi}\sigma_1\sigma_2} \exp\left(-\frac{\sigma_1^2(z - x_1 - \mu_2)^2 + \sigma_2^2(x_1 - \mu_1)^2}{2\sigma_1^2\sigma_2^2}\right) dx_1 \\
&= \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\sqrt{2\pi}\sigma_1\sigma_2} \exp\left(-\frac{\sigma_1^2(z^2 + x_1^2 + \mu_2^2 - 2x_1 z - 2z\mu_2 + 2x_1\mu_2) + \sigma_2^2(x_1^2 + \mu_1^2 - 2x_1\mu_1)}{2\sigma_2^2\sigma_1^2}\right) dx_1 \\
&= \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\sqrt{2\pi}\sigma_1\sigma_2} \exp\left(-\frac{x_1^2(\sigma_1^2 + \sigma_2^2) - 2x_1(\sigma_1^2(z - \mu_2) + \sigma_2^2\mu_1) + \sigma_1^2(z^2 + \mu_2^2 - 2z\mu_2) + \sigma_2^2\mu_1^2}{2\sigma_1^2\sigma_1^2}\right) dx_1.
\end{aligned} \tag{2.2}$$

Let us define $\sigma_Z = \sqrt{\sigma_1^2 + \sigma_2^2}$, such that

$$
\begin{aligned}
f_Z(z) &= \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\sigma_Z} \frac{1}{\sqrt{2\pi}\frac{\sigma_1\sigma_2}{\sigma_Z}} \exp\left(- \frac{x_1^2 - 2x_1 \frac{\sigma_1^2(z-\mu_2)+\sigma_2^2\mu_1}{\sigma_Z^2} + \frac{\sigma_1^2(z^2+\mu_2^2-2z\mu_2)+\sigma_2^2\mu_1^2}{\sigma_Z^2}}{2\left(\frac{\sigma_1\sigma_2}{\sigma_Z}\right)^2} \right) dx_1 \\
&= \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\sigma_Z} \frac{1}{\sqrt{2\pi}\frac{\sigma_1\sigma_2}{\sigma_Z}} \exp\left(- \frac{\left(x_1 - \frac{\sigma_1^2(z-\mu_2)+\sigma_2^2\mu_1}{\sigma_Z^2}\right)^2 - \left(\frac{\sigma_1^2(z-\mu_2)+\sigma_2^2\mu_1}{\sigma_Z^2}\right)^2 + \frac{\sigma_1^2(z-\mu_2)^2+\sigma_2^2\mu_1^2}{\sigma_Z^2}}{2\left(\frac{\sigma_1\sigma_2}{\sigma_Z}\right)^2} \right) dx_1 \\
&= \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\sigma_Z} \exp\left(- \frac{\sigma_Z^2\left(\sigma_1^2(z-\mu_2)^2+\sigma_2^2\mu_1^2\right) - \left(\sigma_1^2(z-\mu_2)+\sigma_2^2\mu_1\right)^2}{2\sigma_Z^2(\sigma_1\sigma_2)^2} \right) \frac{1}{\sqrt{2\pi}\frac{\sigma_1\sigma_2}{\sigma_Z}} \exp\left(- \frac{\left(x_1 - \frac{\sigma_1^2(z-\mu_2)+\sigma_2^2\mu_1}{\sigma_Z^2}\right)^2}{2\left(\frac{\sigma_1\sigma_2}{\sigma_Z}\right)^2} \right) dx_1 \\
&= \frac{1}{\sqrt{2\pi}\sigma_Z} \exp\left(- \frac{(z-(\mu_1+\mu_2))^2}{2\sigma_Z^2} \right) \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\frac{\sigma_1\sigma_2}{\sigma_Z}} \exp\left[- \frac{\left(x_1 - \frac{\sigma_1^2(z-\mu_2)+\sigma_2^2\mu_1}{\sigma_Z^2}\right)^2}{2\left(\frac{\sigma_1\sigma_2}{\sigma_Z}\right)^2} \right] dx_1 .
\end{aligned}
\tag{2.3}
$$

Note that the integral in (2.3) is the integral over the real line for a normal random variable x_1, which by definition is equal to 1, such that

$$
\begin{aligned}
f_Z(z) &= \frac{1}{\sqrt{2\pi}\sigma_Z} \exp\left(- \frac{(z-(\mu_1+\mu_2))^2}{2\sigma_Z^2} \right) \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}\frac{\sigma_1\sigma_2}{\sigma_Z}} \exp\left(- \frac{\left(x_1 - \frac{\sigma_1^2(z-\mu_2)+\sigma_2^2\mu_1}{\sigma_Z^2}\right)^2}{2\left(\frac{\sigma_1\sigma_2}{\sigma_Z}\right)^2} \right) dx_1 \\
&= \frac{1}{\sqrt{2\pi}\sigma_Z} \exp\left(- \frac{(z-(\mu_1+\mu_2))^2}{2\sigma_Z^2} \right).
\end{aligned}
\tag{2.4}
$$

As a result Z is a normal random variable with mean $\mu_Z = \mu_1 + \mu_2$ and variance $\sigma_Z^2 = \sigma_1^2 + \sigma_2^2$. □

Lemma 2.1 states that the sum of independent normal random variables follows again a normal distribution and is easily obtained. Another example is the distribution of the sum of squared standard normal random variables. Let us define $Y_i = \frac{X_i - \mu_i}{\sigma_i}, i = 1, \ldots, N$, such that Y_i is a standard normal random variable.

Lemma 2.2. *The sum* $Y = \sum_{i=1}^{N} Y_i^2 \sim \chi_N^2$.

Instead of giving a full proof, an illustrative proof of Lemma 2.2 will be provided, which can easily be extended to a formal proof.

Proof. Let $X = Y^2$, where $Y \sim \mathcal{N}(0,1)$, then the cumulative distribution function of X is defined as

$$
\begin{aligned}
F_X(x) &= \mathscr{P}(X \leq x) \\
&= \mathscr{P}\left(Y^2 \leq x\right) = \mathscr{P}\left(-\sqrt{x} \leq Z \leq \sqrt{x}\right).
\end{aligned}
\tag{2.5}
$$

Since X is the square of a random variable it only admits non-negative values, thus $F_X(x) = 0$ for $x < 0$ and by the Leibniz rule for integration it follows that

$$
\begin{aligned}
f_X(x) &= \frac{dF_X(x)}{dx} = \frac{d\left(\int_{-\sqrt{x}}^{\sqrt{x}} f_Z(z)dz\right)}{dx} \\
&= f_Z(\sqrt{x})\frac{d\left(\sqrt{x}\right)}{dx} - f_Z(\sqrt{x})\frac{d\left(\sqrt{x}\right)}{dx} \\
&= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}x^2\right)\frac{1}{2}x^{-\frac{1}{2}} - \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}x^2\right)\cdot -\frac{1}{2}x^{-\frac{1}{2}} \\
&= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}x\right)x^{-\frac{1}{2}} \\
&= \frac{1}{2^{\frac{1}{2}}\Gamma\left(\frac{1}{2}\right)}\exp\left(-\frac{1}{2}x\right)\frac{1}{2}x^{\frac{1}{2}-1},
\end{aligned}
\tag{2.6}
$$

for $x < 0$ it must hold that $f_X(x) = 0$, such that the density of X equals the density of a $\chi^2(1)$ density with 1 degree of freedom,

$$
f_X(x) = \begin{cases} \frac{1}{2^{\frac{1}{2}}\Gamma\left(\frac{1}{2}\right)}\exp\left(-\frac{1}{2}x\right)\frac{1}{2}x^{\frac{1}{2}-1}, & x \geq 0 \\ 0, & x < 0. \end{cases}
\tag{2.7}
$$

The $\chi^2(1)$ distribution has the extra property that the sum of $N \in \mathbb{N}$ independent $\chi^2(1)$ distributed random variables is a $\chi^2(N)$ random variable. Thus by combining the previous results, the lemma holds. □

Thus, a sum of squared normal variables follows a Chi-Squared distribution with N degrees of freedom. Lemmas 2.1 and 2.2 illustrate two examples where the sum of Gaussian random variables is available in closed form. This is not the case for the sum of lognormally distributed random variables.

Definition 2.1. A random variable Y is **lognormally distributed** if the random variable $X = \log Y$ has a normal distribution, thus the pdf of X is defined as

$$
f_X(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}}\exp\left(-\frac{(x-\mu)^2}{2\sigma_X^2}\right),
\tag{2.8}
$$

such that the probability density function of Y is given by

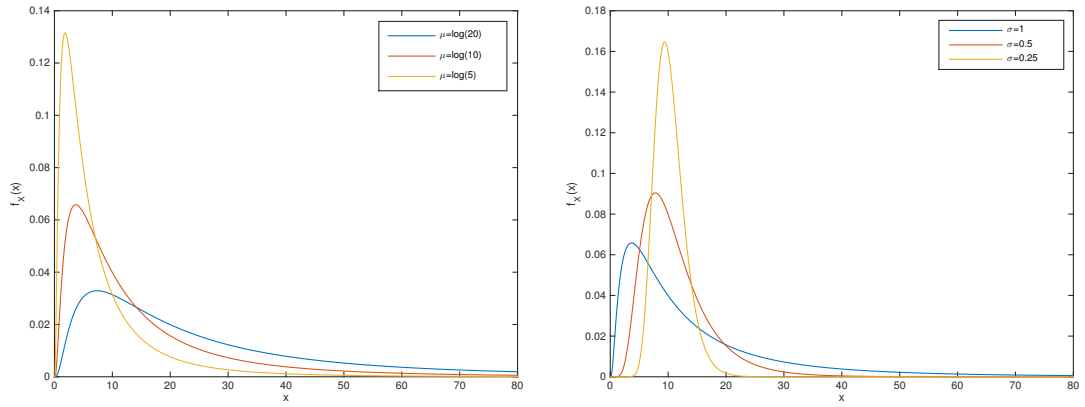
$$
f_Y(y) = \frac{1}{y\sigma_X\sqrt{2\pi}}\exp\left(-\frac{(y-\mu)^2}{2\sigma_X^2}\right).
\tag{2.9}
$$

Figure 2.1 illustrates the probability density function of lognormally distributed random variables. As the figure shows, depending on the parameters μ and σ the distribution is a heavy-tailed distribution with positive skewness.

Let us define Y as the sum of $N \in \mathbb{N}$ lognormal random variables

$$
Y = \sum_{i=1}^{N} Y_i = Y_1 + \ldots + Y_N,
$$

with $\log Y_i$ normally distributed with mean μ_i and variance σ_i^2, $\log Y_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, in that case the distribution of Y is difficult to determine. Since the probability density function of the sum of lognormals



(a) μ (b) σ

Figure 2.1: Probability density function $f_X(x)$ of lognormal random variable X, for several μ and σ.

is not known in closed form, numerous numerical techniques have been proposed to approximate the distribution function, for example in [3] and [23]. One way to approximate the distribution of the sum of lognormals Y is by determining the characteristic function of Y.

2.2. Characteristic function

If a random variable admits a probability density function, the probability density function can be defined by its characteristic function. In that way recovering the probability density function is equivalent to the recovery of the characteristic function. The inverse Fourier transform of the characteristic function results in the probability density function. The characteristic function approach allows us to approximate the probability density function of an underlying stochastic model without a closed-form expression for the probability density function.

Having defined the distribution of a single lognormally distributed random variable we can now turn to the determination of the characteristic function of a lognormally distributed random variable. The following definitions will be used throughout this work.

Definition 2.2. A function $f : \mathbb{R} \to \mathbb{C}$ is said to be **Lebesgue-integrable** if

$$\int_{\mathbb{R}} |f(x)| dx < \infty. \tag{2.10}$$

The space of all integrable functions is defined as $L^1(\mathbb{R})$.

Furthermore, $L^2(\mathbb{R})$ is the space of square integrable functions defined by all functions that satisfy

$$\int_{\mathbb{R}} |f(x)|^2 dx < \infty. \tag{2.11}$$

The $L^2(\mathbb{R})$ space is equipped with the inner product

$$\langle f, g \rangle = \int_{\mathbb{R}} f \bar{g} dx, \tag{2.12}$$

where $\bar{g}$ denotes the complex conjugate of g. Next, let us define the Fourier transform of an integrable function f.

Definition 2.3. The *Fourier transform* of an integrable function $f : \mathbb{R} \to \mathbb{C}$ denoted by $\hat{f}$ is defined as

$$\hat{f}(\omega) := \int_{\mathbb{R}} f(x) e^{-\imath\omega x} dx, \tag{2.13}$$

for $\omega \in \mathbb{R}$ and $\imath$ the complex unit.

In practice, different definitions of the characteristic function are being used, we have chosen to use the one in definition 2.4.

Definition 2.4. Let X be a random variable with probability density function f_X, its characteristic function is defined as

$$\begin{aligned} \hat{f}_X(\omega) :=& \mathbb{E}\left[e^{-\imath\omega X}\right] \\ =& \int_{\mathbb{R}} f_X(x) e^{-\imath\omega x} d.x \end{aligned} \tag{2.14}$$

We can conclude that the Fourier transform of a probability density function coincides with the characteristic function of the random variable corresponding to that probability density function. Since we are interested in the characteristic function of the sum of lognormals, we propose the use of Lemma 2.3.

Lemma 2.3. *For independent random variables X and Y with characteristic functions $\hat{f}_X(\omega)$ and $\hat{f}_Y(\omega)$ and $a, b \in \mathbb{R}$, it follows that*

$$\hat{f}_{aX+bY}(\omega) = \hat{f}_X(a\omega) \cdot \hat{f}_Y(b\omega). \tag{2.15}$$

Proof. Since X and Y are independent random variables, by the definition of the characteristic function it follows that

$$\begin{aligned} \hat{f}_{aX+bY}(\omega) =& \mathbb{E}\left[e^{-\imath\omega(aX+bY)}\right] \\ =& \mathbb{E}\left[e^{-\imath\omega aX} e^{-\imath\omega bY}\right] \\ =& \mathbb{E}\left[e^{-\imath\omega aX}\right] \mathbb{E}\left[e^{-\imath\omega bY}\right] \\ =& \hat{f}_X(a\omega) \cdot \hat{f}_Y(b\omega) \end{aligned} \tag{2.16}$$

and the lemma holds. □

Let us assume that Y is the sum of N independent lognormally distributed random variables $Y_1, \ldots, Y_N$, then by Lemma 2.3 its characteristic function can be written as

$$\begin{aligned} \hat{f}_Y(\omega) =& \hat{f}_{\sum_{i=1}^N Y_i}(\omega), \\ =& \prod_{i=1}^N \hat{f}_{Y_i}(\omega). \end{aligned} \tag{2.17}$$

Furthermore if the $Y_1, \ldots, Y_N$ are identically distributed, we can write

$$\hat{f}_Y(\omega) = \left(\hat{f}_{Y_i}(\omega)\right)^N. \tag{2.18}$$

These results show that the recovery of the probability density function $f_Y(y)$ can rely on the recovery of the characteristic function of the individual random variables Y_i. The stochastic processes dealt with in this work are in the class of *Lévy processes*. The next section will define this class.

2.3. Lévy processes

The class of Lévy processes is a particular class of stochastic processes. Before defining the characteristics of Lévy processes, let us start with the definition of a stochastic process.

Definition 2.5. Let $(\Omega, \mathscr{F}, \mathscr{P})$ be a probability space and let $X : \Omega \to \mathbb{R}$ be a random variable. A **stochastic process** is the set of real random variables on the sample space Ω indexed by the ordered time set T,

$$X = \{X_t : t \in T\}.$$

The time set is then defined by $\{t_0, t_1, \ldots, t_{M_d}\}$, with $0 = t_0$ and $t_{M_d} = T$.

Definition 2.6. The stochastic process X is **stable** if for each $N \in \mathbb{N}$, $X_1, \ldots, X_N$ independently and identically distributed copies of X, it holds that

$$bX + c \overset{d}{=} X_1 + \ldots + X_N,$$

with $b \in \mathbb{R}^N_{>0}$, $c \in \mathbb{R}^N_{\geq 0}$ constants.

The normal distribution, the Lévy distribution and the Cauchy distribution satisfy definition 2.6 and are special cases of Lévy stable distributions. Lévy processes have shown to be a better fit for the financial market compared to Gaussian processes, since Lévy processes can model jumps, skewness and excess kurtosis.

Definition 2.7. A stochastic process $X = \{X_t : t \geq 0\}$ with $X_0 = 0$ a.s. is said to be a **Lévy process** if it satisfies the following properties:

- X has independent increments, i.e. for $0 \leq t_1 < \ldots < t_N < \infty$, $X_{t_2} - X_{t_1}, X_{t_3} - X_{t_2}, \ldots, X_{t_N} - X_{t_{N-1}}$ are independent.
- X has stationary increments, i.e. for any $s < t$, $X_t - X_s$ has the same distribution as X_{t-s}.
- X is continuous in probability, i.e. for any $\epsilon > 0$ and $t \geq 0$, $\lim_{h \to 0} P(|X_{t+h} - X_t| > \epsilon) = 0$.

Having stated the definition of a Lévy process, we can introduce infinitely divisible distributions. It turns out that Lévy processes are generated by infinitely divisible distributions.

Definition 2.8. A real valued random variable X, with probability density function $f_X(x)$ is said to be **infinitely divisible** if there exist independently and identically distributed random variables $X_1, \ldots, X_N$ for all $N \in \mathbb{N}$, such that X has the same distribution as the sum of the X_i, $i = 1, \ldots, N$, i.e.

$$X \overset{d}{=} X_1 + \ldots + X_N.$$

Lévy processes are related to the class of infinitely divisible distributions, the relationship between the two has been defined by Sato in [22] and is stated here as Theorem 2.4

Theorem 2.4. *If $\{X_t\}_{t\in[0,T]}$ is a Lévy process on probability space $(\Omega,\mathscr{F},\mathscr{P})$, then X_t has an infinitely divisible distribution $\forall\, t\in[0,T]$.*

Proof. Let $t_0 = 0 < t_1 < \ldots < t_N = t$ be a realization of the stochastic process $\{X_t\}_{t\in[0,T]}$, then we can define X_t as the sum of its increments,

$$X_t = (X_{t_1} - X_{t_0}) + (X_{t_2} - X_{t_1}) + \ldots + (X_t - X_{t_{N-1}}), \tag{2.19}$$

Since the increments are independent and statistically identical for increments with the same length $t_i - t_{i-1}$, $\left(\{X_t\}_{t\in[0,T]}\text{ is a Lévy process}\right)$, we can write for the characteristic function of X_t,

$$\begin{aligned}\hat{f}_{X_t}(\omega) &= \hat{f}_{\sum_{i=1}^{N}\left(X_{t_i}-X_{t_{i-1}}\right)}(\omega)\\ &= \prod_{i=1}^{N}\hat{f}_{\left(X_{t_i}-X_{t_{i-1}}\right)}(\omega)\\ &= \left(\hat{f}_{\left(X_{t_i}-X_{t_{i-1}}\right)}(\omega)\right)^N.\end{aligned} \tag{2.20}$$

If we define new random variables $Y_i = X_{t_i} - X_{t_{i-1}}$, it follows from Lemma 2.3 that

$$X_t \overset{d}{=} Y_1 + \ldots + Y_n. \tag{2.21}$$

As a result the Lévy process is infinitely divisible. □

Another way to determine whether a distribution belongs to the Lévy family is by its characteristic function, which is known from the *Lévy-Khinchine formula*. The Lévy-Khinchine formula gives an analytical expression for the characteristic function of an infinitely divisible distribution. Since each Lévy process admits an infinitely divisible distribution, the characteristic function is known for each Lévy process.

Theorem 2.5 (Lévy-Khinchine formula)**.** *Let $\{X_t\}_{t\in[0,T]}$ be a Lévy process defined on probability space $(\Omega,\mathscr{F},\mathscr{P})$, then there exists a triplet (b,c,ν) uniquely determined by X_t, such that for all $\omega\in\mathbb{R}$ and $x\geq 0$ the characteristic function of X can be written as,*

$$\begin{aligned}\hat{f}_{X_t}(\omega) &= \mathbb{E}\left[e^{-\imath\omega x}\right]\\ &= \mathbb{E}\left[e^{x\Psi_\omega}\right],\end{aligned} \tag{2.22}$$

with Ψ_ω defined as

$$\Psi_\omega = -\frac{1}{2}\omega^2 c - \imath\omega b + \int_{\mathbb{R}}\left(e^{-\imath\omega y} - 1 + \imath\omega y\right)\nu(dy), \tag{2.23}$$

where $b\in\mathbb{R}, c\in\mathbb{R}_{\geq 0}$ and ν the Lévy measure satisfying

$$\nu(\{0\}) = 0 \text{ and } \int_{\mathbb{R}}\min\left\{|x|^2, 1\right\}\nu(dx) < \infty.$$

The triplet (b, c, ν) uniquely defines the Lévy process and is known as the Lévy triplet and the exponent Ψ_ω is known as the Lévy (or characteristic) exponent. Lévy processes can be categorized in several ways. One way is to consider (in)finite variation, another way is to check whether a Lévy process admits (in)finite activity. To be able to distinguish between infinite and finite activity, we need to provide for more explanation on the Lévy measure ν.

Definition 2.9. Let $\{X_t\}_{t\in[0,T]}$ be a Lévy process defined on probability space $(\Omega, \mathscr{F}, \mathscr{P})$ with Lévy triplet (b, c, ν). The **Lévy measure** is a unique positive measure ν on $\mathbb{R}$ which counts the expected number of jumps per unit time interval, $[0, 1]$, i.e.,

$$\nu(c) = \mathbb{E}\left[\#\{t \in [0,1] : \Delta X_t = X_t - X_{t^-} \neq 0\}\right]. \tag{2.24}$$

Although the number of jumps is important, the size of the jump matters too. A jump is considered large if the jump size is greater than 1, thus if $|\Delta X_t| > 1$, which results in the following important property of a Lévy measure,

$$\int_{\mathbb{R}} \min\left\{|x|^2, 1\right\} \nu(dx) < \infty. \tag{2.25}$$

For a large jump equation (2.25) reduces to

$$\int_{|x|>1} \nu(dx) < \infty$$

and for a small jump the condition can be written as

$$\int_{|x|<1} |x|^2 \nu(dx) < \infty.$$

Equation (2.25) tells us that for a Lévy process the expected number of large jumps must be finite in $[0, 1]$. But the expected number of small jumps may be infinite, which leads to the distinction between Lévy processes of finite- or infinite activity.

Definition 2.10. A Lévy process $\{X_t\}_{t\in[0,T]}$ is a **finite activity** Lévy process with triplet (b, c, ν) if for the Lévy measure ν it holds that

$$\int_{\mathbb{R}} \nu(dx) < \infty,$$

thus if the measure has a finite integral.

In other words, a finite activity Lévy process has a finite amount of expected small jumps in the interval $[0, 1]$. An infinite activity Lévy process has an infinite number of expected jumps in the unit interval and its formal definition is stated as follows.

Definition 2.11. A Lévy process $\{X_t\}_{t\in[0,T]}$ is an **infinite activity** Lévy process with triplet (b, c, ν) if for the Lévy measure ν it holds that

$$\int_{\mathbb{R}} \nu(dx) = \infty,$$

thus if the measure has infinite integral.

We have covered the definitions of Lévy processes with finite and infinte activity, but we did not discuss the zero activity Lévy process. If a Lévy process has zero activity, its measure ν equals 0. An example of a zero activity Lévy process is the Brownian Motion (with drift). The Brownian Motion is a continuous process, there are no jumps present in the process. The characteristic function of a Brownian motion is given by

$$\hat{f}(\omega,t)_{\text{GBM}} = \exp\left(-\imath\omega\mu t - \frac{1}{2}\sigma^2\omega^2 t\right).$$

By the Lévy-Khinchine formula, Theorem 2.5, indeed implies that $\nu = 0$. As mentioned another way to categorize Lévy processes is to check their level of variation. A Lévy process can have finite or infinite variation, but in order to distinguish between these two, the definition of total variation of a stochastic process is given.

Definition 2.12. The **total variation** of a stochastic process $\{X_t\}_{t\in[0,T]}$, with the partitioning of the interval $[0,T]$ given by

$$0 = t_0 < t_1 < \ldots < t_N = T,$$

is defined by

$$T(X) = \sup \sum_{i=1}^{N} |X_{t_i} - X_{t_{i-1}}|.$$

By this definition we can define (in)finite variation for Lévy processes.

Definition 2.13. Let $\{X_t\}_{t\in[0,T]}$ be a Lévy process with triplet (b,c,ν) on the interval $[0,T]$. The process is said to be a Lévy process of **finite variation** if

$$\mathscr{P}\left(T(X) = \sup \sum_{i=1}^{N} |X_{t_i} - X_{t_{i-1}}| < \infty\right) = 1.$$

Thus a Lévy process is of finite variation if the total variation on the interval $[0,T]$ is finite with probability one. The Lévy process is of infinite variation if the total variation on a sample path in $[0,T]$ is infinite with probability one.

Definition 2.14. Let $\{X_t\}_{t\in[0,T]}$ be a Lévy process with triplet (b,c,ν) on the interval $[0,T]$. The process is said to be Lévy process of **infinite variation** if

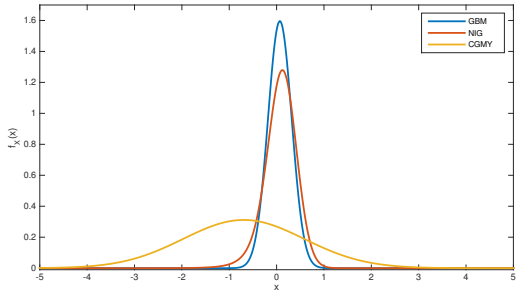
$$\mathscr{P}\left(T(X) = \sup \sum_{i=1}^{N} |X_{t_i} - X_{t_{i-1}}| = \infty\right) = 1.$$

Another way to distinguish between finite and infinite variation Lévy processes is with help of Theorem 2.5 and the corresponding triplet (b,c,ν). If a Lévy process is of finite variation, it holds that

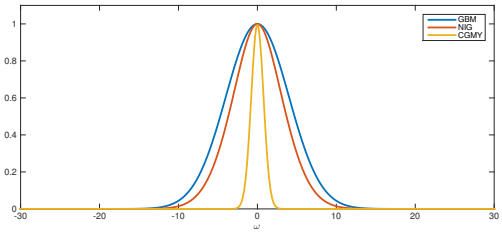
$$c = 0 \text{ and } \int_{|x|<1} |x|\nu(dx) < \infty.$$

If the process admits infinite variation, the process satisfies

$$c \neq 0 \text{ or } \int_{|x|<1} |x|\nu(dx) = \infty.$$



(a) Probability density function.



(b) Modulus of the characteristic function.

Figure 2.2: Approximated probability density functions of the GBM, NIG and CGMY processes and the modulus of the corresponding characteristic functions.

	GBM	NIG	CGMY
Variation	∞	∞	∞ if $Y \in [1,2)$
Activity	0	∞	∞ if $Y \in [0,2)$

Table 2.1: Three Lévy processes categorized by their activity and variation.

Concluding the section on the Lévy process, three different Lévy processes will be stated which will act as underlying stochastic processes for the option pricing problems. The first is the well-known Geometric Brownian Motion (GBM), a continuous Lévy process. The second process is the Normal Inverse Gaussian process (NIG) and finally the CGMY process will be introduced. Table 2.1 categorizes the three Lévy processes based on the variation and activity.

2.3.1. GBM process

The *Geometric Brownian Motion* is the asset price model which is used in the Black-Scholes setting. Due to assumptions made in the Black-Scholes model, calculating standard European option prices under GBM dynamics are straightforward and admit an analytical solution. A stochastic process $S = \{S_t : t \geq 0\}$ follows a Geometric Brownian Motion if it satisfies the following stochastic differential equation

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

with μ the drift, σ the volatility and W_t a Wiener process. The characteristic function of the GBM process is defined as

$$\hat{f}(\omega, t)_{\text{GBM}} = \exp\left(-\imath\omega\mu t - \frac{1}{2}\sigma^2\omega^2 t\right). \tag{2.26}$$

2.3.2. NIG process

One of the more modern processes of the Lévy processes is the NIG process, which is short for *the Normal Inverse Gaussian* process. The process was introduced in 1977, but found its application in financial mathematics in 1997 [2]. The Normal Inverse Gaussian process is mixture of a Gaussian distribution and an Inverse Gaussian process. The process is controlled by four parameters α, β, ∂ and μ. α controls the steepness of the density, β is the skewness parameter, ∂ is the scale parameter and μ represents the percentage drift of the process.

$$dS_t = \mu dt + dW_{h(t)}, \tag{2.27}$$

where W_t is a Wiener process, $h(t)$ an *Inverse Gaussian* process with parameters $IG\left(\frac{\nu}{\gamma}, \nu^2\right)$, with

$$\nu = \partial dt \text{ and } \gamma = \sqrt{\alpha^2 - \beta^2}.$$

The characteristic function of the NIG process is defined by

$$\hat{f}(\omega, t, \alpha, \beta, \partial)_{\text{NIG}} = \hat{f}(\omega, t)_{\text{GBM}} \cdot \exp\left[\partial t\left(\sqrt{\alpha^2 - \beta^2} - \sqrt{\alpha^2 - (\beta + \imath\omega)^2}\right)\right]. \tag{2.28}$$

2.3.3. CGMY process

The *Carr-Geman-Madan-Yor* (CGMY) process is inspired by jump diffusion models, which describe assets as jump diffusions. Diffusion models capture small moves, while in financial markets large jumps could also occur. The CGMY process can capture both small jumps as well as large moves in the underlying process. The model is named after the four writers of [6], which have developed the model. The CGMY process is a Lévy process defined by four parameters, C, G, M and Y, where $C, G, M > 0$ and $Y \in (-\infty, 2)$. The CGMY model allows both infinite and finite activity as well as infinite and finite variation, depending on the parameter Y. The CGMY process has infinite activity if and only if $Y \in [0, 2)$ and paths of the CGMY process have infinite variation if and only if $Y \in [1, 2)$. Parameter C represents the kurtosis of the distribution, where G controls the exponential decay on the right tail of the density function and M the exponential decay on the left tail of the density function.

The process admits a closed-form expression for the characteristic function, but unfortunately not for the probability density function. Nevertheless, by means of the SWIFT method and the COS method an approximation to the density function will be obtained. The characteristic function of the CGMY process is defined by

$$\hat{f}(\omega, t, C, G, M, Y)_{\text{CGMY}} = \hat{f}(\omega, t)_{\text{GBM}} \cdot \exp\left[C t \Gamma(-Y)\left((M - \imath\omega)^Y - M^Y + (G - \imath\omega)^Y - G^Y\right)\right] \tag{2.29}$$

2.4. Sum of increments of a Lévy process

This chapter has defined the sum of lognormals, of which we will approximate the distribution by means of the SWIFT method. Previously we have stated three Lévy processes, the GBM process, the NIG process and the CGMY process. In the application of SWIFT for the arithmetic Asian option valuation and European option valuation under SABR dynamics, we will define a new state variable which is related to the increment process of the underlying stochastic process $S(t_i)$ at $[0, T]$,

$$R(t_i) = \log\left(\frac{S(t_i)}{S(t_{i-1})}\right).$$

If the underlying stochastic process has GBM dynamics, the increments are lognormally distributed and the sum of increments is equivalent to the sum of lognormals. However, if the underlying stochastic process has NIG or CGMY dynamics, the increments are not lognormally distributed. Luckily the SWIFT method is applicable to both the sum of lognormals as the sum of independent increments. Because the NIG and the CGMY process are Lévy processes, they have independent and identically distributed increments.

3

The SWIFT Method

SWIFT is short for *Shannon Wavelet Inverse Fourier Technique*, which implies that the SWIFT method is a Fourier method combined with a Shannon wavelet approximation. Wavelet bases to approximate density functions have found their way to the field of computational finance recently. Different types of wavelet bases have been proposed for different financial problems. Haar wavelets have been used to quantify credit portfolio losses [18] and have been applied to the computation of market risk measures, such as the *VaR* and the *Expected Shortfall* [9]. The use of a Shannon wavelet base, which relies on the cardinal sine function, has shown to result in a robust numerical method to compute European option prices [20]. The SWIFT method has shown exponential convergence for European option valuation, for heavy-tailed distributions and for long maturity options. One benefit of the SWIFT method is that the method is robust, compared to other Fourier methods, such as the COS method. A priori the integral does not need to be truncated, since the scale of approximation, m, results automatically in the number of terms in the wavelet expansion needed to give an accurate approximation of the density function.

This chapter will introduce the SWIFT method, first of all we will explain the basics of wavelet theory and define multiresolution analysis with the Shannon wavelet basis. For this explanation we follow the theory presented by Ingrid Daubechies in [10]. Having looked at the theory, the SWIFT method to approximate the probability density function will be addressed. We are interested in the numerical benefits of the SWIFT method, therefore we will compare the SWIFT method to a state of the art method for option pricing: the COS method. Both the SWIFT method abd the COS method relies on a Fourier expansion of the probability density function based on the characteristic function. Section 3.5 will give a short introduction to the COS method. To gain full insight in the SWIFT method in practice, we will apply the SWIFT method as well as the COS method to European option pricing.

3.1. Introduction to wavelet theory

The aim is to approximate the probability density function by a finite number of Shannon scaling functions. This is all based on multiresolution analysis.

3.1.1. Multiresolution analysis

A *multiresolution analysis* includes a family of closed nested subspaces in $L^2(\mathbb{R})$,

$$\ldots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots$$

with the following properties

$$\bigcap_{j\in\mathbb{Z}} V_j = \{0\} \text{ and } \overline{\bigcup_{j\in\mathbb{Z}} V_j} = L(\mathbb{R})^2. \tag{3.1}$$

Multiresolution results from the two extra properties:

$$\begin{aligned} f(x) \in V_j &\Leftrightarrow f(2x) \in V_{j+1}, \\ f(x) \in V_j &\Leftrightarrow f(x - 2^{-j}k) \in V_j, \forall k \in \mathbb{Z}. \end{aligned} \tag{3.2}$$

As a result we can write the following

$$f(x) \in V_0 \Leftrightarrow f(2^j x) \in V_j,$$

thus all spaces V_j are scaled versions of the space V_0. At this point a multiresolution analysis can be defined. The core of a multiresolution analysis is that if a collection of subspaces V_j satisfy the aforementioned properties, then there exists an orthonormal wavelet basis $\{\psi_{m,k}\}_{m,k\in\mathbb{Z}}$.

Definition 3.1. Let us define $\phi \in L^2(\mathbb{R})$ the generator of the family $\{\phi_{m,k}\}_{m,k\in\mathbb{Z}}$ with $\phi_{m,k}(x) = 2^{\frac{m}{2}}\phi(2^m x - k)$, furthermore let us define the space V_m as

$$V_m := \text{closure}_{L^2(\mathbb{R})}\left(\{\phi_{m,k}\}_{k\in\mathbb{Z}}\right).$$

If V_m satisfies the aforementioned properties (3.1) and (3.2) and $\phi_{0,k}$ forms an orthonormal basis of V_0, then we call ϕ the **scaling function** or **father wavelet**.

For any $f \in L^2(\mathbb{R})$ a projection of $L^2(\mathbb{R})$ onto V_m, $P_m : L^2(\mathbb{R}) \to V_m$ is defined as

$$P_m f(x) = \sum_{k\in\mathbb{Z}} \langle f, \phi_{m,k}\rangle \phi_{m,k}(x). \tag{3.3}$$

If ϕ is a scaling function (father wavelet) then there exists an orthonormal wavelet basis $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ of $L^2(\mathbb{R})$, $\psi_{j,k}(x) = 2^{\frac{j}{2}}\psi\left(2^j x - k\right)$, such that, for all $f \in L^2(\mathbb{R})$,

$$P_j f(x) = P_{j+1} f(x) + \sum_{k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k}(x). \tag{3.4}$$

The function ψ is called the *mother wavelet* and can be determined explicitly. In order to derive the mother wavelet, let us define subspaces W_j, with the following properties

$$V_{j+1} = V_j \bigoplus W_j \text{ and } L^2(\mathbb{R}) \sum_{j\in\mathbb{Z}} \bigoplus W_j, \tag{3.5}$$

such that $\{\psi_{j,k}\}_{m,k\in\mathbb{Z}}$ forms an orthonormal basis for W_j and it follows that the projection map can be written as

$$P_m f(x) = \sum_{j=-\infty}^{m-1} \sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}(x) = \sum_{k\in\mathbb{Z}} c_{m,k}\phi_{m,k}(x), \tag{3.6}$$

where $d_{j,k} = \int_{\mathbb{R}} f(x)\psi_{j,k}(x)dx$ are the wavelet coefficients and $c_{m,k} = \int_{\mathbb{R}} f(x)\phi_{m,k}(x)dx$ the scaling coefficients. The projection map can be expressed in both the scaling functions as well as the wavelet functions. The higher the scale of approximation m, the more accurate the truncated series representation becomes.

3.1.2. Shannon Wavelets

In this thesis we will consider the *Shannon* scaling function or the *Sinc* function to be the scaling function. The Shannon wavelet father function is defined as

$$\phi(x) = \text{sinc}(x) = \begin{cases} \dfrac{\sin(\pi x)}{\pi x}, & x \neq 0, \\ 1, & x = 0. \end{cases} \tag{3.7}$$

The Shannon mother wavelet is defined as

$$\psi(x) = \text{sinc}\left(x - \frac{1}{2}\right) - 2\text{sinc}\left(2x - 1\right). \tag{3.8}$$

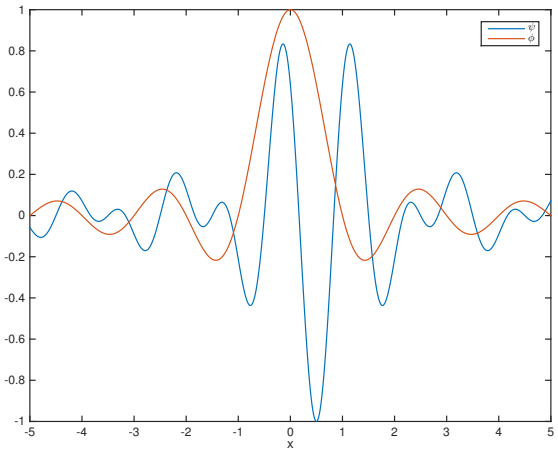


Figure 3.1: Shannon mother (ψ) and father wavelet (ϕ) functions.

Figure 3.1 illustrates the Shannon mother and father wavelet functions. The corresponding basis $\forall k \in \mathbb{Z}$ for subspaces for V_m and W_m are the following,

$$\begin{aligned} \phi_{m,k}(x) =& 2^{\frac{m}{2}}\phi\left(2^m x - k\right) \\ =& 2^{\frac{m}{2}} \frac{\sin\left(\pi\left(2^m x - k\right)\right)}{\pi\left(2^m x - k\right)}, \end{aligned} \tag{3.9}$$

$$\psi_{m,k}(x) = 2^{\frac{m}{2}} \frac{\sin\left(\pi\left(2^m x - k - \frac{1}{2}\right)\right) - \sin\left(2\pi\left(2^m x - k - \frac{1}{2}\right)\right)}{\pi\left(2^m x - k - \frac{1}{2}\right)}. \tag{3.10}$$

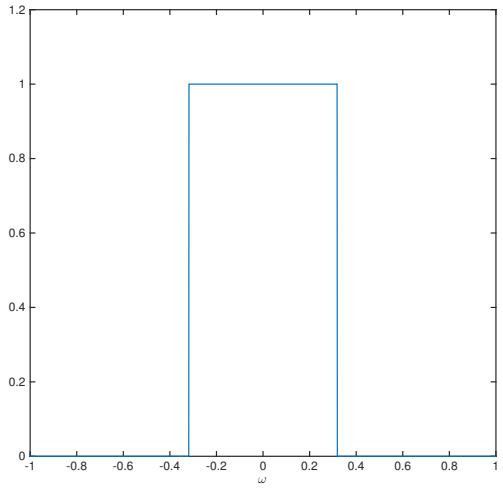


Figure 3.2: The Fourier transform of the Shannon scaling function, $\hat{\phi}(\omega)$.

The Fourier transforms are given by

$$\begin{aligned}\hat{\phi}_{m,k}(\omega) &= \int_{\mathbb{R}} \phi_{m,k} e^{-1\omega x} dx \\ &= \frac{e^{-1\frac{k}{2^m}\omega}}{2^{m/2}} \cdot \text{rect}\left(\frac{\omega}{2^{m+1}\pi}\right),\end{aligned} \tag{3.11}$$

and

$$\begin{aligned}\hat{\psi}_{m,k}(\omega) &= \int_{\mathbb{R}} \psi_{m,k} e^{-i\omega x} dx \\ &= \frac{e^{-1\frac{k+\frac{1}{2}}{2^m}\omega}}{2^{\frac{m}{2}}} \left(\text{rect}\left(\frac{\omega}{2^m\pi} - \frac{3}{2}\right) + \text{rect}\left(-\frac{\omega}{2^m\pi} - \frac{3}{2}\right)\right),\end{aligned} \tag{3.12}$$

with rect$(\cdot)$ the rectangular function

$$\text{rect}(x) = \begin{cases} 1 & \text{if } |x| < \frac{1}{2}, \\ \frac{1}{2} & \text{if } |x| = \frac{1}{2}, \\ 0 & \text{if } |x| > \frac{1}{2}. \end{cases} \tag{3.13}$$

Since the father wavelets ϕ are rather-straightforward, the focus is on the father wavelet instead of the mother wavelet. Furthermore, the Shannon wavelet function has a slow decay in the time domain, but a sharp decay in the Fourier (frequency) domain. Figure 3.2 illustrates the compact support of the Fourier transform of the wavelet scaling function.

At this stage, we have all the tools to use Shannon wavelets to approximate the probability density function $f(\cdot)$ by means of the SWIFT method.

3.2. The approximation of the probability density function

The density function f_X will be expanded by Shannon scaling functions with approximation level m and the coefficients $c_{m,k}$ in (3.6) will be determined by the Fourier transform. The computation of the coefficients $c_{m,k}$ is the core of the SWIFT method and we will determine the coefficients with the help of Vieta's formula.

The probability density function of a continuous random variable X, $f_X(\cdot)$, is defined on $L^2(\mathbb{R})$ and its projection $P_m f(x)$ (3.3) is defined as

$$f_X(x) \approx P_m f_X(x) = \sum_{k\in\mathbb{Z}} c_{m,k}\phi_{m,k}(x), \tag{3.14}$$

with m the scale of approximation. By Lemma 1 in [20], i.e. the assumption that the density function f_X tends to zero at plus and minus infinity, the infinite summation can be truncated to a finite summation without loss of significant density mass. We will truncate the infinite sum (3.14) by $k = k_1, \ldots, k_2$, such that

$$P_m f_X(x) \approx f_X^*(x) = \sum_{k=k_1}^{k_2} c_{m,k}\phi_{m,k}(x), \tag{3.15}$$

for well-chosen k_1 and k_2. The scaling (or density) coefficients are defined by

$$\begin{aligned} c_{m,k} =& \langle f_X, \phi_{m,k}\rangle = \int_{\mathbb{R}} f_X(x)\phi_{m,k}(x)dx \\ =& \int_{\mathbb{R}} f_X(x) 2^{\frac{m}{2}} \phi(2^m x - k)dx \\ =& 2^{\frac{m}{2}} \int_{\mathbb{R}} f_X(x)\operatorname{sinc}(2^m x - k)dx. \end{aligned} \tag{3.16}$$

3.2.1. Vieta's formula

In order to obtain the density coefficients in (3.16), we use Vieta's formula to write the cardinal sine function as a product of cosine functions. By [13] it follows that

$$\begin{aligned} \operatorname{sinc}(x) =& \frac{\sin(\pi x)}{\pi x} \\ =& \cos\left(\pi\frac{x}{2}\right)\frac{\sin\left(\pi\frac{x}{2}\right)}{\pi\frac{x}{2}} \\ =& \cos\left(\pi\frac{x}{2}\right)\operatorname{sinc}\left(\pi\frac{x}{2}\right) \\ =& \ldots \\ =& \prod_{n=1}^{\infty}\cos\left(\pi\frac{x}{2^n}\right). \end{aligned} \tag{3.17}$$

We truncate this infinite product by a finite product, with $N \in \mathbb{N}$ and apply the known trigonometric relation

$$\cos(x)\cos(y) = \frac{1}{2}\left(\cos(x+y) + \cos(x-y)\right), \tag{3.18}$$

and obtain

$$\begin{aligned}\operatorname{sinc}(x) &= \prod_{n=1}^{\infty} \cos\left(\pi \frac{x}{2^n}\right) \\ &\approx \prod_{n=1}^{N} \cos\left(\pi \frac{x}{2^n}\right) \\ &= \frac{1}{2^{N-1}} \sum_{n=1}^{2^{N-1}} \cos\left(\frac{2n-1}{2^N}\pi x\right) \\ &=: \operatorname{sinc}^*(x).\end{aligned} \tag{3.19}$$

Substituting (3.19) to (3.16), we obtain the following expression for the density coefficients

$$\begin{aligned}c_{m,k} &= 2^{\frac{m}{2}} \int_{\mathbb{R}} f_X(x) \operatorname{sinc}(2^m x - k)\, dx \\ &\approx \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=1}^{2^{N-1}} \int_{\mathbb{R}} f_X(x) \cos\left(\frac{2n-1}{2^N}\pi\left(2^m x - k\right)\right) dx \\ &=: c^*_{m,k}.\end{aligned} \tag{3.20}$$

Let us define the error resulting from Vieta's formula approximation as

$$\epsilon_{Vieta} = \operatorname{sinc}(x) - \operatorname{sinc}^*(x),$$

then Lemma 1 in [20] states that for $c \in \mathbb{R}_{>0}$ and $x \in [-c, c]$ the error ϵ_{Vieta} is bounded by

$$|\epsilon_{Vieta}| \leq \frac{(\pi c)^2}{2^{2(N+1)} - (\pi)^2}, \tag{3.21}$$

if $N \geq \log_2(\pi c)$. This lemma allows us to choose a sufficiently large N. Let us consider $N \geq \log\left(\pi M_{m,k}\right)$, where $M_{m,k} = \max\left(|2^m a - k|, |2^m a + k|\right)$ for a constant $a > 0$ such that

$$F_X(-a) + 1 - F_X(-a) < \epsilon,$$

for an $\epsilon > 0$ and F_X the distribution of the random variable X, then by Theorem 1 in [20] it follows that

$$\lim_{N \to +\infty} c^*_{m,k} \to c_{m,k}.$$

In order to speed up the computation of the density coefficients and to allow the use of the Fast Fourier Transform (FFT), we choose (following [20]) a constant N for each k. Recall that for the Fourier transform of $f_X(\cdot)$, $\hat{f}_X(\cdot)$, it follows that

$$\begin{aligned}
&\hat{f}_X(\omega) = \int_{\mathbb{R}} e^{-\imath\omega x} f_X(x)\,dx \\
&\Leftrightarrow \\
&\hat{f}_X(\omega)\cdot e^{-\imath k\pi\frac{2n-1}{2^N}} = e^{-\imath k\pi\frac{2n-1}{2^N}}\cdot\int_{\mathbb{R}} e^{-\imath\omega x} f_X(x)\,dx \\
&\Leftrightarrow \\
&\hat{f}_X(\omega)\cdot e^{\imath k\pi\frac{2n-1}{2^N}} = \int_{\mathbb{R}} e^{\imath\omega x+\imath k\pi\frac{2n-1}{2^N}} f_X(x)\,dx,
\end{aligned} \tag{3.22}$$

and using Euler's identity, we obtain

$$\text{Real}\left\{\hat{f}_X(\omega)\right\} = \int_{\mathbb{R}} \cos(\omega x) f_X(x)\,dx. \tag{3.23}$$

Applying the results of (3.23), (3.22) and the substitution $\omega = \frac{2n-1}{2^N}\pi 2^m$ at (3.20), we obtain the following approximation of the probability density coefficients $c_{m,k}$

$$c_{m,k} \approx \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=1}^{2^{N-1}} \text{Real}\left\{\hat{f}_X\left(\frac{2n-1}{2^N}\pi 2^m\right) e^{\frac{\imath k\pi(2n-1)}{2^N}}\right\}. \tag{3.24}$$

To be able to apply the Fast Fourier Transform algorithm, we need to make an extra adjustment to (3.24),

$$\begin{aligned}
c_{m,k} &\approx \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=1}^{2^{N-1}} \text{Real}\left\{\hat{f}_X\left(\frac{2n-1}{2^N}\pi 2^m\right) e^{\frac{\imath k\pi(2n-1)}{2^N}}\right\} \\
&= \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=0}^{2^{N-1}-1} \text{Real}\left\{\hat{f}_X\left(\frac{2n+1}{2^N}\pi 2^m\right) e^{\frac{\imath k\pi(2n+1)}{2^N}}\right\},
\end{aligned} \tag{3.25}$$

with the extra assumption that $\hat{f}_X\left(\frac{2n+1}{2^N}\pi 2^m\right) = 0$ from 2^{N-1} to $2^N - 1$, we obtain

$$c_{m,k} \approx \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=0}^{2^N-1} \text{Real}\left\{\hat{f}_X\left(\frac{2n+1}{2^N}\pi 2^m\right) e^{\frac{\imath k\pi(2n+1)}{2^N}}\right\}. \tag{3.26}$$

3.2.2. Probability density function

Equation (3.15) results in a way to approximate the probability density function. If the density coefficients are computed, the probability density function can be recovered in one step. Let us recall the wavelet approximation to the density function (3.15),

$$\begin{aligned}
f_X(x) &\approx \sum_{k=k_1}^{k_2} c_{m,k}\phi_{m,k}(x) \\
&= \sum_{k=k_1}^{k_2} c_{m,k} 2^{\frac{m}{2}} \text{sinc}\left(\pi\left(2^m x - k\right)\right).
\end{aligned} \tag{3.27}$$

If we set $x = \frac{k}{2^m}$, for $k = k_1, \ldots, k_2$, it follows that

$$\begin{aligned} f_X\left(\frac{k}{2^m}\right) &= \sum_{k=k_1}^{k_2} c_{m,k} 2^{\frac{m}{2}} \operatorname{sinc}\left(\pi\left(2^m\left(\frac{k}{2^m}\right) - k\right)\right) \\ &= 2^{\frac{m}{2}} \sum_{k=k_1}^{k_2} c_{m,k} \cdot 1, \end{aligned} \tag{3.28}$$

such that the approximation to the probability density function is the cumulative sum of the density coefficients $c_{m,k}$ scaled by $2^{\frac{m}{2}}$.

3.3. Scale of approximation m

In this section we will derive a method to choose the scale parameter m a priori. From Lemma 2.2 in [17] we can state that if a function f is bandlimited with $B < 2^m$, then there exists an ω value such that $|\hat{f}_X(\omega)| < TOL$ for all $|\omega| > 2^m\pi$.

Definition 3.2. A function f is **bandlimited** if there exists a constant $B < \infty$, such that the Fourier transform is zero for $|\omega| > B\pi$, i.e.

$$\hat{f}(\omega) = \int_{-B\pi}^{B\pi} f(x) e^{-\imath\omega x} dx. \tag{3.29}$$

The variable B is called the bandwidth of the function. In our application in financial mathematics the most common density functions are not bandlimited, however their characteristic functions do tend to zero for $|\omega| \to \infty$. Thus we can assume that the density function is 'close' to a bandlimited function and we can choose the scale of approximation m such that the density coefficients are negligibly small for $|\omega| > B\pi$. If we define a tolerance level TOL a priori, we can search for the value $m \in \mathbb{Z}$, such that $|\hat{f}_X(\omega)| < TOL$. Let us define the projection error ϵ_p as

$$\begin{aligned} \epsilon_p &:= \left| f(x) - P_m\left(f(x)\right)\right| \\ &= \left| f(x) - \sum_{k\in\mathbb{Z}} c_{m,k}\phi_{m,k}(x)\right|. \end{aligned} \tag{3.30}$$

In the case of the GBM process as underlying an analytical expression to compute m is available, however in the case of the CGMY process there is no analytical inversion available and the proper parameter m is obtained by plugging in $m = 1, 2, \ldots$ until we find $|\hat{f}(2^m\pi)| < TOL$.

3.3.1. Computation of m under the GBM model

The characteristic function of an asset, X_t, driven by the GBM model is given by (2.26), i.e.,

$$\hat{f}_{X_t}(\omega) = \exp\left(-\imath\omega\mu - \frac{1}{2}\sigma^2\omega^2 t\right). \tag{3.31}$$

Where we look for the value m such that $|\hat{f}_{X_t}(\omega)| < TOL$ for all $|\omega| > 2^m\pi$. By [17] the following analytical solution is obtained

$$m(TOL) = \log\left(\frac{1}{\pi}\sqrt{\frac{-2\log(TOL)}{(t\sigma^2)}}\right). \tag{3.32}$$

The appropriate scale of approximation is then given by $m = \lceil m(TOL) \rceil$.

3.3.2. Computation of m under the NIG and CGMY model

Since there is no analytical solution available in the case of the NIG process or the CGMY process, the scale of approximation at tolerance level TOL, $m(TOL)$, is obtained with Algorithm 3.1 in which ω_m is defined as $\omega_m = 2^m \pi$, with 2^m the 'bandwidth' of the density function f_X.

Initialization:

- Define $\omega_m = 2^m \pi$.

Main loop: for $m = 1$ to 10

- Compute $\hat{f}_X(-\omega_m)$, $\hat{f}_X(\omega_m)$.
- Compute $\epsilon_m = \frac{1}{2\pi} \left| \hat{f}_X(-\omega_m) + \hat{f}_X(\omega_m) \right|$.
- **If** $\epsilon_m < TOL$ **Break**

Final Step:

- Scale of approximation m.

Algorithm 3.1: A priori computation of scale of approximation m.

3.4. The SWIFT method applied to European option pricing

The starting point in option pricing methods is the risk neutral option valuation formula,

$$V(S_0, t_0) = e^{-r(T-t_0)} \mathbb{E}^{\mathbb{Q}} \left[V(y, T) | S_0 \right] \tag{3.33}$$

with $V(S_0, t)$ the option value at time t with initial asset price S_0, $V(\cdot, \cdot)$ the payoff function, maturity of the option T, risk neutral interest rate r and $\mathbb{E}^{\mathbb{Q}}$ the expectation under the risk neutral measure $\mathbb{Q}$. We can express(3.33) in terms of the transitional density function $f_{y|S_0}$, we obtain

$$\begin{aligned} V(S_0, t_0) &= e^{-r(T-t_0)} \mathbb{E}^{\mathbb{Q}} \left[V(y, T) | S_0 \right] \\ &= e^{-r(T-t_0)} \int_{\mathbb{R}} V(y, T) f_{y|S_0} \, dy. \end{aligned} \tag{3.34}$$

The integral on $\mathbb{R}$ will be truncated by $[a, b]$ and the density function can be approximated by the SWIFT wavelet approximation, it follows that

$$\begin{aligned} V^*(S_0, t_0) &= e^{-rT} \sum_{k=k_1}^{k_2} \left(c_{m,k} \int_a^b V(y, T) \phi_{m,k}(y|S_0) \, dy \right) \\ &= e^{-rT} \sum_{k=k_1}^{k_2} c_{m,k} \mathcal{V}_{m,k}(S_0) \end{aligned} \tag{3.35}$$

with the payoff coefficients $\mathcal{V}_{m,k}(S_0)$ defined as:

$$\mathcal{V}_{m,k}(S_0) = \int_a^b V(y,T)\phi_{m,k}(y|S_0)\,dy. \tag{3.36}$$

The coefficients $k = \{k_1, \ldots, k_2\}$ can be directly computed from the truncated interval $[a,b]$, with

$$k_1 := \lfloor 2^m a \rfloor \text{ and } k_2 := \lceil 2^m b \rceil.$$

3.4.1. Integration range $[a,b]$

The real line on which the probability density function is approximated is truncated by an arbitrary interval or by use of the cumulant generating function, following [12]. For the underlying GBM, NIG and CGMY processes the cumulants are available in closed-form, such that truncation will be based on the cumulant generating function to obtain the finite interval $[a,b]$. Let us define

$$[a,b] := \left[x_0 + \eta_1 - L\sqrt{\eta_2 + \sqrt{\eta_4}}, x_0 + \eta_1 + L\sqrt{\eta_2 + \sqrt{\eta_4}} \right], \tag{3.37}$$

where $x_0 = \log\left(\frac{S_0}{K}\right)$, default value $L = 10$ and η_t represents the $t-$th cumulant of $\log\left(\frac{S_T}{K}\right)$. Table 3.1 shows the accuracy of the cumulant approach to truncate the integration range of a Normal random variable X, $X \sim \mathcal{N}(\mu, \sigma^2)$. The cumulant approach results in the smallest error, which shows that the area under the density function is very close to the value 1.

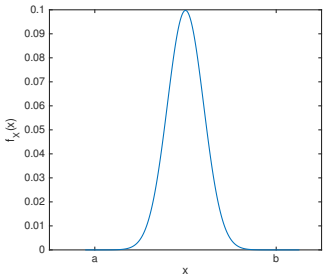


Figure 3.3: Probability density of a Normal random variable with the cumulant interval $[a,b]$ shown.

a	b	Error (area)
-4	15	0.105
-6	20	$6.241e-03$
-8	25	$9.181-05$
-10	30	$5.733e-07$

Table 3.1: Absolute error of the area under the probability density function with on the truncated interval $[a,b]$, cumulant approach gives $a = -10$ and $b = 30$. Parameters: $\mu = 10$, $\sigma = 4$.

Let us define the cumulant generating function

$$\eta_t^X := \log \mathbb{E}\left[e^{tX}\right], \tag{3.38}$$

where η_t^X denotes the t-th cumulant of the random variable X. Table 3.2 summarizes the formulas for the first, second and fourth cumulants of the GBM, the NIG and the CGMY process.

	η_1	η_2	η_4
GBM	$\left(\mu - \frac{1}{2}\sigma^2\right)T$	$\sigma^2 T$	0
NIG	$\left(\mu - \frac{1}{2}\sigma^2 + \upsilon_{NIG}\right)T + \frac{\partial T\beta}{\sqrt{\alpha^2-\beta^2}}$	$\partial T\alpha^2\left(\alpha^2-\beta^2\right)^{-\frac{3}{2}}$	$3\partial T\alpha^2\left(\alpha^2+4\beta^2\right)\left(\alpha^2-\beta^2\right)^{-\frac{7}{2}}$
CGMY	$\mu T + CT\Gamma(1-Y)\left(M^{Y-1} - G^{Y-1}\right)$	$\sigma^2 T + CT\Gamma(2-Y)\left(M^{Y-2} + G^{Y-2}\right)$	$CT\Gamma(4-Y)\left(M^{Y-4} + G^{Y-4}\right)$

Table 3.2: First, second and fourth cumulants of three Lévy processes.

The drift correction terms for the NIG and the CGMY processes, such that $e^{-\upsilon T} = \hat{f}(-1)$, respectively υ_{NIG} and υ_{CGMY}, are defined by

$$\upsilon_{NIG} = -\partial\left(\sqrt{\alpha^2-\beta^2}-\sqrt{\alpha^2-(\beta+1)^2}\right),$$
$$\upsilon_{CGMY} = -C\Gamma(-Y)\left[(M-1)^Y - M^Y + (G+1)^Y - G^Y\right].$$

Throughout this thesis the value of L in (3.37) will be fixed to $L = 10$. To illustrate the dependence of the interval on the parameter L, Table 3.3 shows the absolute error of the area under the density function of a normal random variable. $L = 8$ already gives a very accurate approximation of the density function, such that $L = 10$ is sufficient. Furthermore the increase of L, does not affect the absolute error, it will only increase the CPU time.

	8	10	12	14	16	18
Abs. error	$3.2196e-15$	$8.8818e-16$	$1.1102e-15$	$1.2212e-15$	$1.5543e-15$	$1.2212e-15$

Table 3.3: The absolute error of the area under the normal density dependent on the value of L. Parameters: $\mu = 3$, $\sigma = 0.5$.

3.4.2. Payoff coefficients

The payoff coefficients are defined by (3.36),

$$\begin{aligned}\mathcal{V}_{m,k} &:= \int_a^b V(y,T)\phi_{m,k}(y|x_0)\,dy \\ &= K2^{\frac{m}{2}}\int_a^b \left(e^y-1\right)\operatorname{sinc}\left(2^m y-k\right)dy \\ &\approx K\frac{2^{\frac{m}{2}}}{2^{N-1}}\sum_{n=1}^{2^{N-1}}\left[I^1_{n,k}\left(\frac{\bar{k}_1}{2^m},\frac{k_2}{2^m}\right)-I^2_{n,k}\left(\frac{\bar{k}_1}{2^m},\frac{k_2}{2^m}\right)\right],\end{aligned} \tag{3.39}$$

with $\omega_n = \frac{2n-1}{2^N}\pi$ and

$$\begin{aligned}I^1_{n,k}(a,b) &= \int_a^b e^y\cos\left(\omega_n\left(2^m y-k\right)\right)dy \\ &= \frac{\omega_n 2^m}{1+(\omega_n 2^m)^2}\left[e^b\sin\left(\omega_n\left(2^m b-k\right)\right)-e^a\sin\left(\omega_n\left(2^m a-k\right)\right)\right. \\ &\quad \left.+\frac{1}{\omega_n 2^m}\left(e^b\cos\left(\omega_n\left(2^m b-k\right)\right)-e^a\cos\left(\omega_n\left(2^m a-k\right)\right)\right)\right], \\ I^2_{n,k}(a,b) &= \int_a^b \cos\left(\omega_n\left(2^m y-k\right)\right)dy \\ &= \frac{1}{\omega_n 2^m}\left(\sin\left(\omega_n\left(2^m b-k\right)\right)-\sin\left(\omega_n\left(2^m a-k\right)\right)\right).\end{aligned} \tag{3.40}$$

We can compute the payoff coefficients by means of the FFT algorithm available in Matlab, which will result in an efficient and fast implementation. For the details we will refer the reader to [20].

3.4.3. Density coefficients

In order to retrieve the probability density function, we will start with (3.26)

$$c_{m,k} \approx \frac{2^{\frac{m}{2}}}{2^{N-1}}\sum_{n=0}^{2^N-1}\text{Real}\left\{\hat{f}_{y|S_0}\left(\frac{2n+1}{2^N}\pi 2^m\right)e^{\frac{\mathrm{i}k\pi(2n+1)}{2^N}}\right\}. \tag{3.41}$$

Thus in order to determine the price of the option, we need the characteristic function of the underlying process. Section 2.3 contains the details of the characteristic function for each of the three processes considered.

3.5. Alternative: the COS Method

An alternative Fourier method to price European options is the COS method, [25]. The COS method relies on the Fourier cosine expansion of the transitional density function $f_{y|S_0}$, which is given by

$$f_{y|S_0}(y) \approx \frac{2}{b-a} \sum_{k=0}^{N-1} \text{Real}\left\{\hat{f}_{y|S_0}\left(\frac{k\pi}{b-a}\right) \exp\left(-1\frac{ak\pi}{b-a}\right)\right\} \cos\left(k\pi\frac{y-a}{b-a}\right), \tag{3.42}$$

with N the number of expansion terms, $\hat{f}_{y|S_0}(\omega)$ the characteristic function of $f_{y|S_0}$ and the truncated domain $[a,b]$ is based on the cumulants of the underlying distribution. If we apply this substitution in the risk neutral pricing formula (3.34) and after interchanging the summation and integration, we obtain

$$\begin{aligned} V_{\cos}(S_0,t_0) =& e^{-r(T-t_0)} \int_{\mathbb{R}} V(y,T) f_{y|S_0}(y) dy \\ \approx& e^{-r(T-t_0)} \int_{\mathbb{R}} V(y,T) \frac{2}{b-a} \sum_{k=0}^{N-1} \text{Real}\left\{\hat{f}_{y|S_0}\left(\frac{k\pi}{b-a}\right) \exp\left(-1\frac{ak\pi}{b-a}\right)\right\} \cos\left(k\pi\frac{y-a}{b-a}\right) dy \\ =& \sum_{k=0}^{N-1} \text{Real}\left\{\hat{f}_{y|S_0}\left(\frac{k\pi}{b-a}\right) \exp\left(-1\frac{ak\pi}{b-a}\right)\right\} \mathcal{V}_k^{\cos}, \end{aligned} \tag{3.43}$$

where $\mathcal{V}_k^{\cos}$ are defined as the *payoff coefficients* for the COS method,

$$\mathcal{V}_k^{\cos} := \frac{2}{b-a} \int_a^b V(y,T) \cos\left(k\pi\frac{y-a}{b-a}\right) dy, \tag{3.44}$$

which are the Fourier cosine coefficients of the payoff function $V(y,T)$, such that we can simplify (3.43) with the definition of the *density coefficients* D_k,

$$D_k(S_0) := \frac{2}{b-a} \text{Real}\left\{\hat{f}_{y|S_0}\left(\frac{k\pi}{b-a}\right) \exp\left(-1\frac{ak\pi}{b-a}\right)\right\}, \tag{3.45}$$

and the approximation to the option price becomes

$$V_{\cos}(S_0,t_0) = \frac{b-a}{2} e^{-r(T-t_0)} \sum_{k=0}^{N-1} D_k(S_0) \mathcal{V}_k^{\cos}. \tag{3.46}$$

3.5.1. COS Payoff coefficients of the European call option

If we consider the *log-asset* prices, the payoff at maturity T of a plain vanilla call option (with strike price K) is defined by

$$V(y,T) = K \cdot \max\left\{e^y - 1, 0\right\}. \tag{3.47}$$

This result applied to the definition of the payoff coefficients (3.44) yields

$$\begin{aligned} \mathcal{V}_k^{\cos} &:= \frac{2}{b-a}\int_0^b K(e^y-1)\cos\left(k\pi\frac{y-a}{b-a}\right)dy \\ &= \frac{2}{b-a}K\left(\chi_k(0,b)-\psi_k(0,b)\right), \end{aligned} \tag{3.48}$$

with

$$\begin{aligned} \chi_k(x_1,x_2) := \frac{1}{1+\left(\frac{k\pi}{b-a}\right)^2}\Bigg[&\cos\left(k\pi\frac{x_2-a}{b-a}\right)e^{x_2}-\cos\left(k\pi\frac{x_1-a}{b-a}\right)e^{x_1} \\ &+\frac{k\pi}{b-a}\sin\left(k\pi\frac{x_2-a}{b-a}\right)e^{d}+\frac{k\pi}{b-a}\sin\left(k\pi\frac{x_1-a}{b-a}\right)e^{x_1}\Bigg], \end{aligned} \tag{3.49}$$

$$\psi_k(x_1,x_2) := \begin{cases} \frac{b-a}{k\pi}\left[\sin\left(k\pi\frac{x_2-a}{b-a}\right)-\sin\left(k\pi\frac{x_1-a}{b-a}\right)\right], & k\neq 0 \\ (x_2-x_1), & k=0. \end{cases} \tag{3.50}$$

3.6. Numerical results

In order to test the SWIFT method for European option pricing, we will compute the absolute error of the option value and the CPU time [1] for short and long-maturity contracts. The results are presented in Table 3.4 and Table 3.5. Table 3.4 shows the performance of the SWIFT method for several scales of approximation m. We can conclude exponential convergence (in accordance with [20]) of the SWIFT method and a competitive method in terms of CPU time required. In less than a millisecond, the European option ($T=1$) can be computed with an absolute error of 10^{-14}.

		$T=1$	$T=10$	$T=100$
$m=0$	Abs. error	7.4465	137.1936	$5.3350e-05$
	CPU time	1.38	1.06	3.67
$m=2$	Abs. error	0.0887	$5.4499e-12$	$4.7785e-07$
	CPU time	0.89	0.66	0.72
$m=4$	Abs. error	$5.6843e-14$	$2.8990e-12$	$9.6864e-07$
	CPU time	0.79	0.64	1.74

Table 3.4: Absolute error and CPU time (milliseconds) of the price of an European call option with several scales of approximation m. Reference values are 10.160052368788676 ($T=1$), 62.533054649055678 ($T=10$), and 99.995013366508417 ($T=100$), which are computed by the well known Black-Scholes pricing formula. Parameters: $S_0=100$, $K=110$, $\sigma=0.25$, $r=0.1$.

The next example illustrates the performance of the SWIFT method for long maturity options. Table 3.5 lists the absolute error and the required CPU time of the valuation of a European call option under GBM dynamics for both the SWIFT and the COS method. We can conclude that the SWIFT method is a robust method, since it can price the long-maturity option accurately at low scales of approximation m. The payoff function of an European option is not bounded, which could result to round off errors that decrease the level of accuracy. If we use a Shannon wavelet approximation, the large payoff coefficients are compensated by really small density coefficients. This results in accurate option prices for contracts with long maturity.

[1] All programs in this thesis were coded in Matlab R2017b and run on a MacBook Pro with 2,4 GHz Intel Core i5 processor and 8GB RAM.

			$T = 50$	$T = 100$
SWIFT	$m = 0$	Abs. error	$6.6451e-01$	$5.7711e-05$
		CPU time	0.67	0.65
COS	$N = 35$	Abs. error	1.3007	$4.6391e+02$
		CPU time	0.39	0.38
SWIFT	$m = 1$	Abs. error	$6.7814e-09$	$4.1550e-06$
		CPU time	0.68	0.70
COS	$N = 70$	Abs. error	$1.5282e-08$	$2.7067e-05$
		CPU time	0.59	0.61

Table 3.5: Absolute error and CPU time (milliseconds) corresponding to the valuation of a European call option with the GBM price process for the underlying, for both the COS and the SWIFT method. Reference values are 99.202592852553181 ($T = 50$) and 99.995013366508417 ($T = 100$), which are computed by the well known Black-Scholes pricing formula. Parameters: $S_0 = 100$, $K = 120$, $\sigma = 0.25$, $r = 0.1$.

4

Asian Option Valuation

In this chapter the SWIFT method will be applied to the valuation of Asian options under Lévy processes. This chapter contains both the pricing of the geometric Asian option and the computation of the arithmetic Asian option price. First of all the definition of an Asian option will be given in the next section, after which the risk neutral pricing formula will be the starting point of the SWIFT method to price Asian options. Zhang et al. [25] have extended the general COS framework to the valuation of Asian options. The COS method for Asian option, known as the ASCOS method, will provide reference values for the SWIFT method. The chapter will conclude with a numerical comparison between the ASCOS and the SWIFT implementation for several underlying Lévy processes.

4.1. Asian Options

Asian options can basically be divided into two types: *fixed strike* and *floating strike* Asian options. The difference is best illustrated by their payoff functions.

- Payoff function of a fixed strike Asian call option:

$$V(S(T),T) = \max\{A(0,T) - K, 0\}, \tag{4.1}$$

- Payoff function of a floating strike Asian call option:

$$V(S(T),T) = \max\{S(T) - A(0,T), 0\}, \tag{4.2}$$

where $V(S(T),T)$ represents the option value at maturity T, $S(T)$ the price of the underlying at maturity T, K the predetermined fixed strike price and $A(0,T)$ the average price of the underlying in $[0,T]$. For the fixed strike options, the strike is set in advance, whereas for the floating strike options the strike is path dependent by means of the average price of the underlying $A(0,T)$. The average price of the underlying can be determined in several ways. Let us define the *geometric average* and the *arithmetic average*:

- Geometric average:

$$\begin{aligned} A_{geo}(0,T) &= \left(\prod_{i=0}^{M_d} S(t_i)\right)^{\frac{1}{M_d+1}} \\ &= \exp\left[\frac{1}{M_d+1}\sum_{i=0}^{M_d} \log S(t_i)\right], \end{aligned} \tag{4.3}$$

- Arithmetic average:

$$A_{ari}(0,T) = \frac{1}{M_d+1}\sum_{i=0}^{M_d} S(t_i). \tag{4.4}$$

The price process of the underlying is discretely monitored at M_d monitoring dates, such that $0 = t_0, T = t_{M_d}$ and $t_i = i \cdot \frac{T}{M_d}$. In this thesis we will focus on the fixed strike Asian option, for both the geometric average and the arithmetic average. The pricing of an floating strike Asian options follows immediately from a symmetry relation between the floating strike and fixed strike Asian option. Papantoleon and Eberlein [11] have proposed a symmetry relationship between the fixed strike and floating strike Asian option under Lévy asset price processes, based on a change of numeraire and the Lévy triplet.

Figure 4.1 illustrates the difference between the geometric running mean and the arithmetic running mean. The figure shows that the arithmetic mean is here higher than the geometric mean.

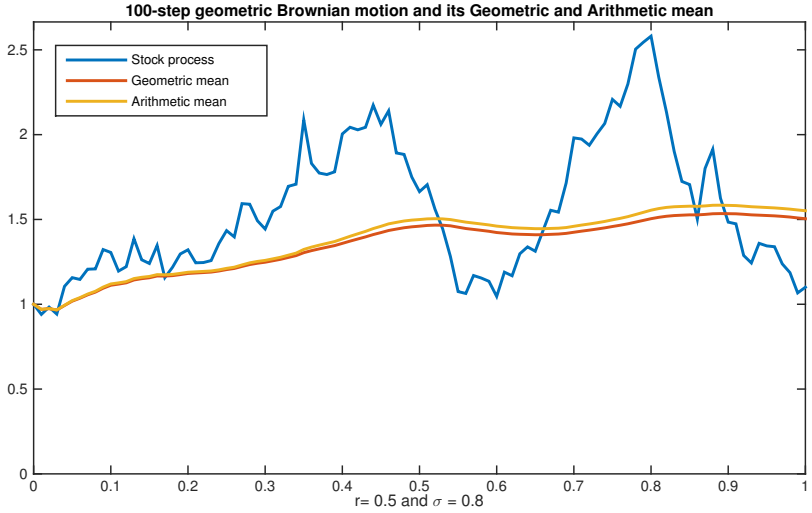


Figure 4.1: 100-step Geometric Brownian Motion with its running geometric and arithmetic mean. Parameters: $T = 1$, $r = 0.5$ and $\sigma = 0.8$.

One of the benefits of trading in Asian options is that the payoff function is less volatile compared to the payoff function of a plain vanilla option, since averaging decreases the impact of volatility. However, the computation of Asian option values can be cumbersome and expensive. Common methods to price Asian options are based on Monte Carlo simulations, which may be expensive ways to obtain accurate option prices. For continuously monitored geometric Asian options under GBM dynamics an analytical solution is available. For this analytical solution the reader is referred to Appendix A. Since the Asian options in this thesis are discretely monitored, with M_d monitoring dates, we propose the SWIFT method to compute the value of an discretely monitored geometric Asian option and we show fast convergence of the SWIFT method, with the GBM as underlying asset price process. The SWIFT method will be applied to geometric Asian options under NIG and CGMY dynamics as well.

In the case of an arithmetic Asian option, there is no closed-form solution available and we will rely on the characteristic function of the density to obtain the option value with the SWIFT method. To approximate the probability density function of the option price, we have to deal with a sum of asset prices, that for Lévy asset price processes results in a sum of lognormals or sum of independent increments. The results obtained by the SWIFT method will be compared to the results of the ASCOS method, which admits as well satisfactory results in terms of speed and accuracy.

4.2. Pricing formula for Asian options

The risk neutral pricing formula forms again the starting point for the valuation of Asian options. Let us consider an arithmetic Asian option with M_d monitoring dates, $t_0 < t_1 < \ldots < t_{M_d}$, with $t_0 = 0$, maturity $t_{M_d} = T$ and fixed strike K. The time between monitoring dates is assumed to be constant, thus $\Delta t = \frac{T}{M_d}$. The payoff function of a geometric Asian call option is given by

$$V(S(T),T) = \max\left\{\exp\left[\frac{1}{M_d+1}\sum_{i=0}^{M_d}\log S(t_i)\right] - K, 0\right\}. \tag{4.5}$$

The payoff function of an arithmetic Asian call option is given by

$$V(S(T),T) = \max\left\{\frac{1}{M_d+1}\sum_{i=0}^{M_d} S(t_i) - K, 0\right\}. \tag{4.6}$$

By the risk neutral pricing formula we can evaluate the value of the option with underlying stochastic process $\{S(t)\}_{t\geq 0}$ as the discounted expected value of the payoff function at time T under the risk neutral measure $\mathbb{Q}$, i.e.,

$$\begin{aligned} V(S_0,t_0) &= e^{-r(T-t_0)}\mathbb{E}^{\mathbb{Q}}\left[V(S(T),T)|S(0),t_0\right] \\ &= \begin{cases} e^{-r(T-t_0)}\int_{\mathbb{R}} V(y,T) f_{A_{geo}(0,T)}(y)dy, & \text{geometric average} \\ e^{-r(T-t_0)}\int_{\mathbb{R}} V(y,T) f_{A_{ari}(0,T)}(y)dy, & \text{arithmetic average} \end{cases} \end{aligned} \tag{4.7}$$

where $V(y,T)$ represents the payoff function, $f_{A_{geo}(0,T)}$ denotes the probability density function of the geometric average underlying and $f_{A_{ari}(0,T)}$ is the probability density function corresponding to the arithmetic average of the underlying price process.

The characteristic functions of $f_{A_{geo}(0,T)}$ and $f_{A_{ari}(0,T)}$ will be computed recursively, where we can benefit from the properties of Lévy processes. The computation of these characteristic functions differs for the two cases, thus the two types of Asian options are dealt with separately. Section 4.3 computes the value of a geometric Asian option and Section 4.4 contains the SWIFT method applied to arithmetic Asian options. This chapter concludes with numerical results for both the geometric and the arithmetic Asian option valuation.

4.3. Geometric Asian option valuation

Geometric Asian options under Lévy processes benefit from the fact that their characteristic function is analytically available, thus the SWIFT method can be applied easily. In order to compute the characteristic function, let us define the new state variable x

$$\begin{aligned} x &:= \log\left(\left(\prod_{i=0}^{M_d} S(t_i)\right)^{\frac{1}{M_d+1}}\right) - \log(K) \\ &= \left(\frac{1}{M_d+1}\sum_{i=0}^{M_d}\log S(t_i)\right) - \log(K). \end{aligned} \tag{4.8}$$

Let us define an increment process

$$R_{t_i} := \log\left(\frac{S(t_i)}{S(t_{i-1})}\right) = \log(S(t_i)) - \log(S(t_{i-1})),$$

for $i = 1,2,\ldots,M_d$. Since the increments of a Lévy process are independent and with constant time interval Δt the increments are statistically identical, it follows that for each $i, j = 1,\ldots,M_d$ with $i \neq j$

$$R_{t_i} \overset{d}{=} R_{t_j} :\overset{d}{=} R.$$

Substituting the increment process for the underlying price process, it follows that

$$\begin{aligned} x &= \left(\frac{1}{M_d+1}\sum_{i=0}^{M_d}\log S(t_i)\right) - \log(K) \\ &= \left(\sum_{i=0}^{M_d}\frac{M_d-i+1}{M_d+1}R_{t_i}\right) + \log\frac{S_0}{K} \\ &= \left(\sum_{i=0}^{M_d}\frac{M_d-i+1}{M_d+1}R_{t_i}\right) + x_0, \end{aligned} \tag{4.9}$$

where $x_0 = \log\frac{S_0}{K}$. With help of Lemma 2.3 we obtain the characteristic function for the state variable x,

$$\begin{aligned} \hat{f}_x(\omega; x_0) =& \mathbb{E}\left[e^{-\imath\omega\left(\left(\sum_{i=0}^{M_d}\frac{M_d-i+1}{M_d+1}R_{t_i}\right)+x_0\right)}\right] \\ =& e^{-\imath\omega x_0}\prod_{i=1}^{M_d}\hat{f}_R\left(\frac{M_d-i+1}{M_d+1}\omega\right). \end{aligned} \tag{4.10}$$

Substituting characteristic function expression (4.10) to the definition of the density coefficients $c_{m,k}$, it follows that

$$\begin{aligned} c_{m,k} \approx& \frac{2^{\frac{m}{2}}}{2^{N-1}}\sum_{n=1}^{2^{N-1}}\text{Real}\left\{\hat{f}_y\left(\frac{2n-1}{2^N}\pi 2^m\right)e^{\frac{\imath k\pi(2n-1)}{2^N}}\right\} \\ =& \frac{2^{\frac{m}{2}}}{2^{N-1}}\sum_{n=1}^{2^{N-1}}\text{Real}\left\{e^{-\imath\omega_n 2^m x_0}\prod_{i=1}^{M_d}\hat{f}_R\left(\frac{M_d-i+1}{M_d+1}\omega_n 2^m\right)e^{\frac{\imath k\pi(2n-1)}{2^N}}\right\}, \end{aligned} \tag{4.11}$$

with $\omega_n = \frac{2n-1}{2^N}\pi$. Equation (4.11) will be modified in the same way as (3.26) such that we can apply the Fast Fourier Transform to compute the density coefficients in an efficient way.

4.3.1. Integration range $[a, b]$

The integration range is based on the cumulants of the underlying random variable. Since the underlying in this case is the state-variable y, we have to determine the cumulants of the geometric average in (4.9). The cumulant generating function is defined by (3.38) and applied to the geometric average results in,

$$
\begin{aligned}
\eta_t^y &:= \log\mathbb{E}\left[e^{ty}\right] \\
&= \log\mathbb{E}\left[e^{t\left(\sum_{i=1}^{M_d}\frac{M_d-i+1}{M_d+1}R_{t_i}\right)}\right] \\
&= \log\mathbb{E}\left[\exp\left(\frac{tM_d}{M_d+1}R_{t_1}\right)\right] + \log\mathbb{E}\left[\exp\left(\frac{t(M_d-1)}{M_d+1}R_{t_2}\right)\right] + \ldots + \log\mathbb{E}\left[\exp\left(\frac{2t}{M_d+1}R_{t_{M_d-1}}\right)\right] + \log\mathbb{E}\left[\exp\left(\frac{t}{M_d+1}R_{t_{M_d}}\right)\right] \\
&= \left(\frac{M_d}{M_d+1}\right)^t \eta_t^{R_{t_1}} + \left(\frac{M_d-1}{M_d+1}\right)^t \eta_t^{R_{t_2}} + \ldots + \left(\frac{2}{M_d+1}\right)^t \eta_t^{R_{t_{M_d-1}}} + \left(\frac{1}{M_d+1}\right)^t \eta_t^{R_{t_{M_d}}} \\
&= \frac{\sum_{i=1}^{M_d}(M_d-i+1)^t}{(M_d+1)^t}\eta_t^R,
\end{aligned}
\tag{4.12}
$$

where we have used the properties of the cumulant generating function and the fact that for Lévy processes the increments are independent identically distributed, with cumulants η_t^R. We apply the same integration range as in the European case (3.37), i.e.,

$$
[a,b] := \left[x_0 + \eta_1^y - L\sqrt{\eta_2^y + \sqrt{\eta_4^y}}, x_0 + \eta_1^y + L\sqrt{\eta_2^y + \sqrt{\eta_4^y}}\right]. \tag{4.13}
$$

As a result of (4.12), the following cumulants are necessary to compute the integration range $[a,b]$

- $t = 1$:
$$
\eta_1^y = \frac{\sum_{i=1}^{M_d} M_d - i + 1}{M_d+1}\eta_1^R = \frac{M_d}{2}\eta_1^R,
$$
- $t = 2$:
$$
\eta_2^y = \frac{\sum_{i=1}^{M_d}(M_d-i+1)^2}{(M_d+1)^2}\eta_2^R = \frac{2M_d^2+M_d}{6M_d+6}\eta_2^R,
$$
- $t = 4$:
$$
\eta_2^y = \frac{\sum_{i=1}^{M_d}(M_d-i+1)^4}{(M_d+1)^4}\eta_4^R = \frac{M_d\left(6M_d^3+9M_d^2+M_d-1\right)}{30\,(M_d+1)^3}\eta_4^R,
$$

The cumulants of the increment process $\{R_{t_i}\}_{i=1,\ldots,M_d}$ are known and stated in Section 3.4.1.

4.3.2. Payoff coefficients

The payoff coefficients are defined by (3.36) and for the European-style geometric Asian call option result in

$$
\begin{aligned}
\mathcal{V}_{m,k}(x_0) &= \int_a^b V(y,T)\phi_{m,k}(y|x_0)\,dy \\
&= K2^{\frac{m}{2}}\int_a^b \left(e^y - 1\right)\operatorname{sinc}\left(2^m y - k\right)dy \\
&\approx K\frac{2^{\frac{m}{2}}}{2^{N-1}}\sum_{n=1}^{2^{N-1}}\left[I_{n,k}^1\left(\frac{\bar{k}_1}{2^m},\frac{k_2}{2^m}\right) - I_{n,k}^2\left(\frac{\bar{k}_1}{2^m},\frac{k_2}{2^m}\right)\right],
\end{aligned}
\tag{4.14}
$$

with (3.40)

$$I^1_{n,k}(x_1,x_2) = \frac{\omega_n 2^m}{1+(\omega_n 2^m)^2}\left[e^{x_2}\sin\left(\omega_n\left(2^m x_2-k\right)\right)-e^{x_1}\sin\left(\omega_n\left(2^m x_1-k\right)\right)\right. \\ \left. + \frac{1}{\omega_n 2^m}\left(e^{x_2}\cos\left(\omega_n\left(2^m x_2-k\right)\right)-e^{x_1}\cos\left(\omega_n\left(2^m x_1-k\right)\right)\right)\right], \tag{4.15}$$

$$I^2_{n,k}(x_1,x_2) = \frac{1}{\omega_n 2^m}\left(\sin\left(\omega_n\left(2^m x_2-k\right)\right)-\sin\left(\omega_n\left(2^m x_1-k\right)\right)\right), \tag{4.16}$$

where $\omega_n = \frac{2n-1}{2^N}\pi$.

With the computation of the payoff coefficients we can conclude this section. The necessities for the SWIFT application to the valuation of geometric Asian options are the density coefficients and the payoff coefficients, stated in (4.11) and (4.14) respectively. We have as well determined the integration interval $[a,b]$ by the cumulant approach, such that we will capture the mass of the density function with the SWIFT method. The next section is devoted to the valuation of arithmetic Asian options by the SWIFT method.

4.4. Arithmetic Asian option valuation

For the arithmetic Asian option let us again denote the increment process $R(t_i)$ of the underlying process as

$$R_{t_i} = \log\left(\frac{S(t_i)}{S(t_{i-1})}\right) = \log(S(t_i)) - \log(S(t_{i-1})), \tag{4.17}$$

for $i = 1,\ldots,M_d$. As already stated in the valuation of geometric Asian options, the increments of a Lévy process are identically and independently distributed and the characteristic function of $R_{t_i} \overset{d}{=} R$ is the same for all $i = 1,\ldots,M_d$. We can apply the Carverhill-Clewlow-Hodges [7] factorization scheme to a newly defined stochastic process Y_i,

$$\begin{aligned} Y_i :=& R_{M_d+1-i} + \log\left(1+\exp(Y_{i-1})\right) \\ =& R_{M_d+1-i} + Z_{i-1}, \end{aligned} \tag{4.18}$$

for $i = 2,\ldots,M_d$, we define $Z_{i-1} = \log\left(1+\exp(Y_{i-1})\right)$ and $Y_1 = R_{t_{M_d}}$. In this case we obtain

$$Y_i = \log\left(\frac{S(t_{M_d+1-i})}{S(t_{M_d-i})} + \frac{S(t_{M_d+2-i})}{S(t_{M_d-i})} + \ldots + \frac{S(t_{M_d})}{S(t_{M_d-i})}\right), \tag{4.19}$$

and it follows that

$$\frac{1}{M_d+1}\sum_{i=0}^{M_d} S(t_i) = \frac{\left(1+\exp(Y_{M_d})\right)S_0}{M_d+1}. \tag{4.20}$$

We will recover the characteristic function of Y_{M_d} by a recursive procedure and will use $\hat{f}_{Y_{M_d}}$ to determine the approximation to the probability density function $f_{Y_{M_d}}$. Since a new variable Y_i ($i = 1,\ldots,M_d$) has been defined, the payoff function (4.6) has changed accordingly. The option value is defined as the expected discounted payoff of the price process Y_{M_d} with

$$V(Y_{M_d}, T) = \max\left\{\frac{S_0\left(1+\exp(Y_{M_d})\right)}{M_d+1} - K, 0\right\}, \tag{4.21}$$

and the price of the option is given by the risk neutral pricing formula

$$\begin{aligned} V(S_0, t_0) =& e^{-r(T-t_0)} \int_{\mathbb{R}} V(y, T) f_{Y_{M_d}}(y)\, dy \\ \approx& e^{-r(T-t_0)} \int_{\mathbb{R}} V(y, T) \sum_{k=k_1}^{k_2} c_{m,k}^{M_d} \phi_{m,k}(y)\, dy \end{aligned} \tag{4.22}$$

4.4.1. Characteristic function

In the previous section we defined the new variable Y_{M_d} by a recursive scheme. The aim is to recover the characteristic function of Y_{M_d} in a recursive way as well. Let us first start with Y_1,

$$Y_1 = R_{M_d} = R \Leftrightarrow \hat{f}_{Y_1}(\omega) = \hat{f}_R(\omega). \tag{4.23}$$

For Lévy processes the characteristic function of the increment process R is available in closed-form. The approximation of $\hat{f}_{Y_{M_d}}$ is more involved. By equation (4.18) it follows that the characteristic function of Y_i, for $i = 2, \ldots, M_d$, can be recovered by

$$\begin{aligned} \hat{f}_{Y_i}(\omega) =& \hat{f}_{R_{M_d+1-i}+Z_{i-1}}(\omega) \\ =& \hat{f}_{R_{M_d+1-i}}(\omega) \cdot \hat{f}_{Z_{i-1}}(\omega) \\ =& \hat{f}_R(\omega) \cdot \hat{f}_{Z_{i-1}}(\omega), \end{aligned} \tag{4.24}$$

where we have used that for Lévy processes R_{M_d+1-i} and Z_{i-1} are independent. The characteristic function of $Z_{i-1} = \log\left(1+\exp(Y_{i-1})\right)$ is given by

$$\hat{f}_{Z_{i-1}}(\omega) = \mathbb{E}\left[e^{-\mathrm{i}\omega\log(1+\exp(Y_{i-1}))}\right] = \int_{\mathbb{R}} \left(1+e^{y}\right)^{-\mathrm{i}\omega} f_{Y_{i-1}}(y)\, dy. \tag{4.25}$$

The probability density function $f_{Y_{i-1}}$ in (4.25) will be approximated by the ASCOS and the SWIFT method. Within the ASCOS method the integral will be computed by a numerical integration method. The computation of the integral within the SWIFT method can be replaced by an approximation.

4.5. SWIFT method: arithmetic Asian options

The SWIFT method provides a wavelet approximation function for a probability density function of a random variable. In order to arrive at the characteristic function of Y_{M_d}, the characteristic function of Y_i for each monitoring date, $i = 1, \ldots, M_d$, has to be computed. In this section we will apply the wavelet approximation. For $i = 2, \ldots, M_d$

$$f_{Y_i}(y) \approx \sum_{k=k_1}^{k_2} c_{m,k}^{i} \phi_{m,k}(y), \tag{4.26}$$

such that the characteristic function of $\hat{f}_{Z_{i-1}}$ (4.25) can be approximated as

$$\begin{aligned}\hat{f}_{Z_{i-1}}(\omega) &= \int_{\mathbb{R}} \left(1+e^{y}\right)^{-\imath\omega} f_{Y_{i-1}}(y)\,dy \\ &\approx \int_{\mathbb{R}} \left(1+e^{y}\right)^{-\imath\omega} \sum_{k=k_1}^{k_2} c_{m,k}^{i-1}\phi_{m,k}(y)\,dy.\end{aligned} \tag{4.27}$$

In this section we will define the pricing formula for arithmetic Asian options by SWIFT, with the computation of the payoff coefficients and the density coefficients.

4.5.1. Payoff coefficients

The payoff coefficients are defined by (3.36), such that we obtain for arithmetic Asian options

$$\begin{aligned}\mathcal{V}_{m,k} :=& \int_a^b V(y,T)\phi_{m,k}(y)\,dy \\ =& 2^{\frac{m}{2}} \int_a^b \left(\frac{S_0\left(e^{y}+1\right)}{M_d+1} - K\right) \operatorname{sinc}\left(2^{m}y-k\right) dy \\ =& 2^{\frac{m}{2}} \int_a^b \left(\frac{S_0}{M_d+1}e^{y} + \left(\frac{S_0}{M_d+1} - K\right)\right) \operatorname{sinc}\left(2^{m}y-k\right) dy.\end{aligned} \tag{4.28}$$

The application of Vieta's formula and the truncation of the summation according to (3.19) gives

$$\begin{aligned}\mathcal{V}_{m,k} =& 2^{\frac{m}{2}} \int_a^b \left(\frac{S_0}{M_d+1}e^{y} + \left(\frac{S_0}{M_d+1} - K\right)\right) \operatorname{sinc}\left(2^{m}y-k\right) dy \\ \approx& 2^{\frac{m}{2}} \int_a^b \frac{S_0}{M_d+1} e^{y} \frac{1}{2^{N-1}} \sum_{n=1}^{2^{N-1}} \cos\left(\frac{2n-1}{2^{N}}\pi x\right) dy + 2^{\frac{m}{2}} \int_a^b \left(\frac{S_0}{M_d+1} - K\right) \frac{1}{2^{N-1}} \sum_{n=1}^{2^{N-1}} \cos\left(\frac{2n-1}{2^{N}}\pi x\right) dy \\ =& \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=1}^{2^{N-1}} \left[\frac{S_0}{M_d+1}\int_a^b e^{y}\cos\left(\frac{2n-1}{2^{N}}\pi\left(2^{m}y-k\right)\right) dy + \left(\frac{S_0}{M_d+1} - K\right)\int_a^b \cos\left(\frac{2n-1}{2^{N}}\pi\left(2^{m}y-k\right)\right) dy\right].\end{aligned} \tag{4.29}$$

The definite integrals in (4.29) have analytical solutions, which results in the following payoff coefficients $\mathcal{V}_{m,k}$

$$\mathcal{V}_{m,k} \approx \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=1}^{2^{N-1}} \left[\frac{S_0}{M_d+1} I_{n,k}^{1}(a,b) + \left(\frac{S_0}{M_d+1} - K\right) I_{n,k}^{2}(a,b)\right], \tag{4.30}$$

with $\omega_n = \frac{2n-1}{2^N}\pi$ and

$$
\begin{aligned}
I^1_{n,k}(a,b) &= \int_a^b e^y \cos\left(\omega_n\left(2^m y - k\right)\right) dy \\
&= \frac{\omega_n 2^m}{1+(\omega_n 2^m)^2}\left[e^b \sin\left(\omega_n\left(2^m b - k\right)\right) - e^a \sin\left(\omega_n\left(2^m a - k\right)\right)\right. \\
&\quad \left. + \frac{1}{\omega_n 2^m}\left(e^b \cos\left(\omega_n\left(2^m b - k\right)\right) - e^a \cos\left(\omega_n\left(2^m a - k\right)\right)\right)\right], \qquad (4.31)\\
I^2_{n,k}(a,b) &= \int_a^b \cos\left(\omega_n\left(2^m y - k\right)\right) dy \\
&= \frac{1}{\omega_n 2^m}\left(\sin\left(\omega_n\left(2^m b - k\right)\right) - \sin\left(\omega_n\left(2^m a - k\right)\right)\right).
\end{aligned}
$$

Equations (4.30) and (4.31) allow us to implement the Fast Fourier Transform, which reduces the CPU time of the computation of the payoff coefficients. Let us define

$$
A_n = \frac{\omega_n 2^m}{1+(\omega_n 2^m)^2\,\omega} \text{ and } B_n = \frac{1}{\omega_n 2^m},
$$

if we expand (4.31) by the trigonometric relations

$$
\sin(x+y) = \sin(x)\cos(y) - \cos(x)\sin(y) \text{ and } \cos(x+y) = \sin(x)\sin(y) + \cos(x)\cos(y),
$$

we obtain for $I^1_{n,k}(a,b)$

$$
\begin{aligned}
I^1_{n,k}(a,b) = A_n\Big(&\cos(\omega_n k)\left[e^b \sin(\omega_n 2^m b) - e^a \sin(\omega_n 2^m a) + B_n e^b \cos(\omega_n 2^m b) - B_n e^a \cos(\omega_n 2^m a)\right] \\
&+ \sin(\omega_n k)\left[e^b \cos(\omega_n 2^m b) - e^a \cos(\omega_n 2^m a) - B_n e^b \sin(\omega_n 2^m b) + B_n e^a \sin(\omega_n 2^m a)\right]\Big). \qquad (4.32)
\end{aligned}
$$

For $I^2_{n,k}(a,b)$ it follows that

$$
I^2_{n,k}(a,b) = B_n\left(\cos(\omega_n k)\left[\sin(\omega_n 2^m b) - \sin(\omega_n 2^m a)\right] + \sin(\omega_n k)\left[\cos(\omega_n 2^m b) - \cos(\omega_n 2^m a)\right]\right). \qquad (4.33)
$$

Let us substitute (4.32) and (4.33) in (4.30),

$$
\begin{aligned}
\mathcal{V}_{m,k} \approx &\frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=1}^{2^{N-1}} \cos(\omega_n k)\left(\frac{S_0}{M_d+1}\left[A_n\left(e^b \sin(\omega_n 2^m b) - e^a \sin(\omega_n 2^m a) + B_n e^b \cos(\omega_n 2^m b) - B_n e^a \cos(\omega_n 2^m a)\right)\right]\right. \\
&\left. + \frac{S_0}{M_d+1}\left[B_n \sin(\omega_n 2^m b) - B_n \sin(\omega_n 2^m a)\right] - K\left[B_n \sin(\omega_n 2^m b) - B_n \sin(\omega_n 2^m a)\right]\right) \\
&+ \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=1}^{2^{N-1}} \sin(\omega_n k)\left(\frac{S_0}{M_d+1}\left[A_n\left(e^b \cos(\omega_n 2^m b) - e^a \cos(\omega_n 2^m a) - B_n e^b \sin(\omega_n 2^m b) + B_n e^a \sin(\omega_n 2^m a)\right)\right]\right. \\
&\left. + \frac{S_0}{M_d+1}\left[B_n \cos(\omega_n 2^m b) - B_n \cos(\omega_n 2^m a)\right] - K\left[B_n \cos(\omega_n 2^m b) - B_n \cos(\omega_n 2^m a)\right]\right). \qquad (4.34)
\end{aligned}
$$

Equation (4.34) allows us to compute the payoff coefficients with the Fast Fourier Transform. The FFT has to be applied twice, once for the cosine terms and once for the sine terms.

4.5.2. Integration range $[a,b]$

We will depart from equation (3.37) with the integration range $[a,b]$ which holds for the European option valuation and was modified for the geometric Asian option valuation. For Y_i, $i=1,\dots,M_d$, it follows that the integration range $[a_i,b_i]$ is given by

$$[a_i,b_i]=\left[\eta_1^{Y_i}-L\sqrt{\eta_2^{Y_i}+\sqrt{\eta_4^{Y_i}}},\eta_1^{Y_i}+L\sqrt{\eta_2^{Y_i}+\sqrt{\eta_4^{Y_i}}}\right]. \tag{4.35}$$

Unfortunately, this implies that for each monitoring date $i=1,\dots,M_d$ the corresponding cumulants have to be computed, which is rather expensive and could slow down the method. In [25] a different (cheap) method to compute one single interval $[a,b]$ for all monitoring dates was proposed. For $i=1,\dots,M_d$ it holds that

$$\begin{aligned} 0\le i\frac{S(t_{M_d-i+1})}{S(t_{M_d-i})}\le e^{Y_i}\le\frac{S(t_{M_d})}{S(t_{M_d-i})} \\ \Leftrightarrow \\ 0\le\log\left(i\frac{S(t_{M_d-i+1})}{S(t_{M_d-i})}\right)\le Y_i\le\log\left(\frac{S(t_{M_d})}{S(t_{M_d-i})}\right). \end{aligned} \tag{4.36}$$

By the properties of the cumulant generating function, it follows that

$$\eta_1^{\log\left(\frac{iS(t_{M_d-i+1})}{S(t_{M_d-i})}\right)}\le\eta_1^{Y_i}\le\eta_1^{\log\left(\frac{iS(t_{M_d})}{S(t_{M_d-i})}\right)}\Leftrightarrow\log(i)+\eta_1^R\le\eta_1^{Y_i}\le\log(i)+i\eta_1^R, \tag{4.37}$$

$$0\le\eta_2^{Y_i}\le\eta_2^{\log\left(\frac{iS(t_{M_d})}{S(t_{M_d-i})}\right)}\Leftrightarrow 0\le\eta_2^{Y_i}\le i\eta_2^R, \tag{4.38}$$

$$0\le\eta_4^{Y_i}\le\eta_4^{\log\left(\frac{iS(t_{M_d})}{S(t_{M_d-i})}\right)}\Leftrightarrow 0\le\eta_4^{Y_i}\le i\eta_4^R. \tag{4.39}$$

Thus, the interval $[a_i,b_i]$ is bounded by

$$\begin{aligned} a_i=&\eta_1^{Y_i}-L\sqrt{\eta_2^{Y_i}+\sqrt{\eta_4^{Y_i}}} \\ \ge&\log(i)+\eta_1^R-L\sqrt{i\eta_2^R+\sqrt{i\eta_4^R}}, \end{aligned} \tag{4.40}$$

and

$$\begin{aligned} b_i=&\eta_1^{Y_i}+L\sqrt{\eta_2^{Y_i}+\sqrt{\eta_4^{Y_i}}} \\ \le&\log(i)+i\eta_1^R+L\sqrt{i\eta_2^R+\sqrt{i\eta_4^R}}. \end{aligned} \tag{4.41}$$

The cumulants of the increment process are available, therefore the integration range $[a_i, b_i]$ for $i = 1, \ldots, M_d$ can be computed. In order to speed up the method, the same integration range $[a, b]$ is chosen for all $1 \le i \le M_d$ by a maximization of the interval, i.e.,

$$\begin{aligned}[a,b] &:= \left[\min_{1\le i\le M_d} a_i, \max_{1\le i\le M_d} b_i\right] \\ &\approx \left[\eta_1^R - L\sqrt{\eta_2^R + \sqrt{\eta_4^R}}, \log(M_d) + M_d\eta_1^R + L\sqrt{M_d\eta_2^R + \sqrt{M_d\eta_4^R}}\right].\end{aligned} \tag{4.42}$$

4.5.3. Density coefficients

The density coefficients are defined by equation (3.26). Instead of state variable X, we have defined by the increment process the new state variable Y_{M_d}, thus the adjustment to equation (3.26) is as follows

$$c_{m,k}^{M_d} \approx \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=0}^{2^N-1} \text{Real}\left\{\hat{f}_{Y_{M_d}}\left(\frac{2n+1}{2^N}\pi 2^m\right) e^{\frac{\imath k\pi(2n+1)}{2^N}}\right\}. \tag{4.43}$$

Following the Carverhill-Clewlow-Hodges factorization scheme, the characteristic function $\hat{f}_{M_d}$ has to be obtained recursively. For each monitoring date $i = 1, \ldots, M_d$ we have to compute the characteristic function of Y_i, such that the density coefficients at monitoring date i, $c_{m,k}^i$, can be computed. Following (4.24), $\hat{f}_{Y_i}(\omega) = \hat{f}_R(\omega) \cdot \hat{f}_{Z_{i-1}}(\omega)$, therefore we have to iteratively compute the characteristic function of Z_{i-1}, which is given by equation (4.25).

$$\begin{aligned}\hat{f}_{Z_{i-1}}(\omega) &= \int_{\mathbb{R}} \left(1+e^x\right)^{-\imath\omega} f_{Y_{i-1}}(x)\,dx \\ &\approx \int_{\mathbb{R}} \left(1+e^x\right)^{-\imath\omega} \sum_{k=k_1}^{k_2} c_{m,k}^{i-1}\phi_{m,k}(x)\,dx \\ &= \sum_{k=k_1}^{k_2} c_{m,k}^{i-1} \int_{\mathbb{R}} \left(e^x+1\right)^{-\imath\omega} \phi_{m,k}(x)\,dx \\ &= \sum_{k=k_1}^{k_2} c_{m,k}^{i-1} \int_{\mathbb{R}} \left(e^x+1\right)^{-\imath\omega} 2^{\frac{m}{2}} \operatorname{sinc}\left(2^m x - k\right) dx \\ &= \sum_{k=k_1}^{k_2} c_{m,k}^{i-1} 2^{\frac{m}{2}} \int_{\mathbb{R}} \left(e^x+1\right)^{-\imath\omega} \operatorname{sinc}\left(2^m x - k\right) dx.\end{aligned} \tag{4.44}$$

Equation (4.44) illustrates the application of the SWIFT wavelet approximation of the probability density function of Y_{i-1}, i.e. $f_{Y_{i-1}}$. To summarize the result, we have obtained characteristic function $\hat{f}_{Z_{i-1}}$, as

$$\hat{f}_{Z_{i-1}}(\omega) \approx \sum_{k=k_1}^{k_2} c_{m,k}^{i-1} 2^{\frac{m}{2}} \int_{\mathbb{R}} g(x) \operatorname{sinc}\left(2^m x - k\right) dx, \tag{4.45}$$

where $g(x) = (e^x + 1)^{-\imath\omega}$, and thus we find for the characteristic function $\hat{f}_{Y_i}$

$$\begin{aligned}\hat{f}_{Y_i}(\omega) &= \hat{f}_R(\omega)\hat{f}_{Z_{i-1}}(\omega) \\ &\approx \hat{f}_R(\omega)\sum_{k=k_1}^{k_2} c_{m,k}^{i-1} 2^{\frac{m}{2}} \int_{\mathbb{R}} g(x)\operatorname{sinc}\left(2^m x-k\right)dx,\end{aligned} \tag{4.46}$$

with the density coefficients $c_{m,k}^i$ from equation (3.26) given by

$$c_{m,k}^i \approx \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=0}^{2^N-1} \text{Real}\left\{\hat{f}_{Y_i}\left(\frac{2n+1}{2^N}\pi 2^m\right) e^{\frac{\mathbb{i}k\pi(2n+1)}{2^N}}\right\}, \tag{4.47}$$

with $\omega_n = \frac{2n+1}{2^N}\pi 2^m$ the vector of length 2^N, $n = 0,1,\ldots,2^N-1$. To decrease the CPU time of the computation of the density coefficients, we implement a matrix-vector multiplication. Let us define matrix $M_{(n,k)}$, as

$$M_{(n,k)} = 2^{\frac{m}{2}} \int_{\mathbb{R}} \left(e^x+1\right)^{-\mathbb{i}\omega_n} \operatorname{sinc}\left(2^m x-k\right)dx, \tag{4.48}$$

with $n = 0,\ldots 2^N-1$, $k = k_1,\ldots,k_2$ and substituting the matrix $M_{(n,k)}$ in (4.45) yields

$$\hat{f}_{Z_i}(\omega) \approx M_{(n,k)} c_{m,k}^i. \tag{4.49}$$

Algorithm 4.1 summarizes the recovery of the density coefficients $c_{m,k}^{M_d}$ by the SWIFT method. In the computation of the density coefficients at every monitoring date we can benefit from the Fast Fourier Transform (FFT) in the same way as in the European option valuation by SWIFT. Note that the matrix $M_{(n,k)}$ does not change over time, which allows us to compute the matrix only once.

Initialization:

- Compute $M_{(n,k)}$ as in (4.48).
- Compute $\hat{f}_R(\omega)$.
- Set $\hat{f}_{Y_1}(\omega) = \hat{f}_R(\omega)$.

Loop to recover $\hat{f}_{Y_{m_d}}$: for $i = 1,\ldots,M_d$

- Compute $c_{m,k}^i$ from (5.19).
- Compute $\hat{f}_{Z_i}$ with matrix-vector multiplication $\hat{f}_{Z_i}(\omega) = Mc_{m,k}^i$.
- Compute $\hat{f}_{Y_{i+1}}(\omega) = \hat{f}_R(\omega)\cdot\hat{f}_{Z_i}(\omega)$.

Final Step:

- Compute the option value $V(S(0),t_0)$ by inserting $c_{m,k}^{M_d}$ in the pricing formula (4.22).

Algorithm 4.1: The recursion procedure to determine the density coefficients.

4.6. Computation of the matrix $M_{(n,k)}$

The matrix $M_{(n,k)}$ defined in (4.48) is constant, therefore it can be computed in advance and stored for later use. Unfortunately, the computation of $M_{(n,k)}$ itself can be quite expensive, because a numerical integration method is used to obtain the values for the integral of each combination (n,k). The numerical integration method used is a Clenshaw-Curtis quadrature rule, which admits exponential convergence, like other numerical integration methods do, but is cheaper.

The integral (4.45) consists of the function $g(x) = (e^x+1)^{-\imath\omega}$ and the Shannon scaling function. If the function g is bandlimited, with exponential decay of the Fourier transform, $\hat{g}(v)$, [24] provides an accurate approximation to the integral.

Both methods to compute the matrix $M_{(n,k)}$ will be implemented and will be compared for the accuracy of the resulting option price and the CPU time required.

4.6.1. Clenshaw-Curtis quadrature rule

In order to use the Clenshaw-Curtis quadrature with N_q integration points, we will truncate the integral in (4.45) for $[a,b]$ and by a variable transformation, we will change the integration interval from $[a,b]$ to $[-1,1]$. The following integral results

$$\begin{aligned} M_{(n,k)} &= \int_a^b \left(e^x+1\right)^{-\imath\omega_n} 2^{\frac{m}{2}} \operatorname{sinc}\left(2^m x-k\right) dx \\ &= \int_{-1}^1 2^{\frac{m}{2}} \frac{b-a}{2} \left(\exp\left[\frac{b-a}{2}x+\frac{a+b}{2}\right]+1\right)^{-\imath\omega_n} \operatorname{sinc}\left(2^m\left[\frac{b-a}{2}x+\frac{a+b}{2}\right]-k\right) dx. \end{aligned} \tag{4.50}$$

This integral can be approximated as following by a Clenshaw-Curtis quadrature rule,

$$\begin{aligned} M_{(n,k)} &= \int_{-1}^1 2^{\frac{m}{2}} \frac{b-a}{2} \left(\exp\left[\frac{b-a}{2}x+\frac{a+b}{2}\right]+1\right)^{-\imath\omega} \operatorname{sinc}\left(2^m\left[\frac{b-a}{2}x+\frac{a+b}{2}\right]-k\right) dx \\ &\approx \left(D^T d\right)^T y \\ &=: w^T y, \end{aligned} \tag{4.51}$$

with D an $\left(\frac{N_q}{2}+1\right)\times\left(\frac{N_q}{2}+1\right)$ matrix with elements

$$D(i,j) = \frac{2}{N_q}\left(\frac{\pi j i}{\frac{N_q}{2}}\right)\cdot\begin{cases}\frac{1}{2}, & \text{if } j=\left\{0,\frac{N_q}{2}\right\}\\ 1, & \text{otherwise.}\end{cases} \tag{4.52}$$

The vectors d and the elements y_j, $j=0,1,\ldots,\frac{N_q}{2}$, are defined as

$$\begin{aligned} d &:= \left(1,\frac{2}{1-4},\frac{2}{1-16},\ldots,\frac{2}{1-(N_q-2)^2},\frac{1}{1-N_q^2}\right) \\ y_j &:= h\left(\cos\left(\frac{j\pi}{N_q}\right)\right)+h\left(-\cos\left(\frac{j\pi}{N_q}\right)\right), \end{aligned} \tag{4.53}$$

where

$$h(x) = 2^{\frac{m}{2}} \frac{b-a}{2} \left(\exp\left[\frac{b-a}{2} x + \frac{a+b}{2} \right] + 1 \right)^{-\mathrm{i}\omega_n} \operatorname{sinc}\left(2^m \left[\frac{b-a}{2} x + \frac{a+b}{2} \right] - k \right).$$

The computation of the matrix $D^T d$ is $\mathcal{O}\left(N_q \log(N_q)\right)$, since it is a Discrete Cosine Transform of type 1. Furthermore the coefficients y_n need to be calculated for each pair (n,k), which costs $\mathcal{O}\left(N_q\right)$ operations. Since $n = 0, 2^{N-1}$ and $k = k_1, \ldots, k_2$ it follows that the total complexity of the computation of y_n is given by $\mathcal{O}\left(N_q 2^N (k_2 - k_1)\right)$. For each monitoring date the matrix vector product in (4.49) requires $\mathcal{O}\left(2^N (k_2 - k_1)\right)$ operations and $\mathcal{O}(2^N)$ computations to find $\hat{f}_{Y_i}$, which gives a total of $\mathcal{O}\left(M_d 2^N (k_2 - k_1)\right)$. The recovery of the characteristic function by means of the Clenshaw-Curtis quadrature rule sums up to

$$\mathcal{O}\left(N_q 2^N (k_2 - k_1)\right) + \mathcal{O}\left(2^{2N} (k_2 - k_1)\right) + \mathcal{O}\left(M_d 2^N (k_2 - k_1)\right).$$

We will refer to the SWIFT-CC method if we apply the SWIFT method with the numerical integration of matrix $M_{(n,k)}$.

4.6.2. Sinc integral approximation

Next we will state a very interesting and helpful theorem to approximate the integral in (4.48). This theorem originates from [24].

Theorem 4.1. *If f defined on $\mathbb{R}$ and its Fourier transform $\hat{f}$ are such that for a constant $d > 0$, $\left|\hat{f}(\omega)\right| = \mathcal{O}\left(e^{-d|\omega|}\right)$ for $|\omega| \to \infty$. Then as $h \to 0$,*

$$\frac{1}{h} \int_{\mathbb{R}} f(x) S_{j,h}(x) dx - f(h \cdot j) = \mathcal{O}\left(e^{-\frac{\pi d}{h}}\right), \tag{4.54}$$

where $S_{j,h}(x) = sinc\left(\frac{x}{h} - j\right)$ for $j \in \mathbb{Z}$.

If the function g in (4.48) satisfies the assumptions of Theorem 4.1 it allows us to approximate the integral in (4.45). We have to determine whether the Fourier transform of $g = (e^x + 1)^{-\mathrm{i}\omega_n}$ has exponential decay. Thus we have to show that

$$\left|\hat{g}(v)\right| = \mathcal{O}\left(e^{-d|v|}\right). \tag{4.55}$$

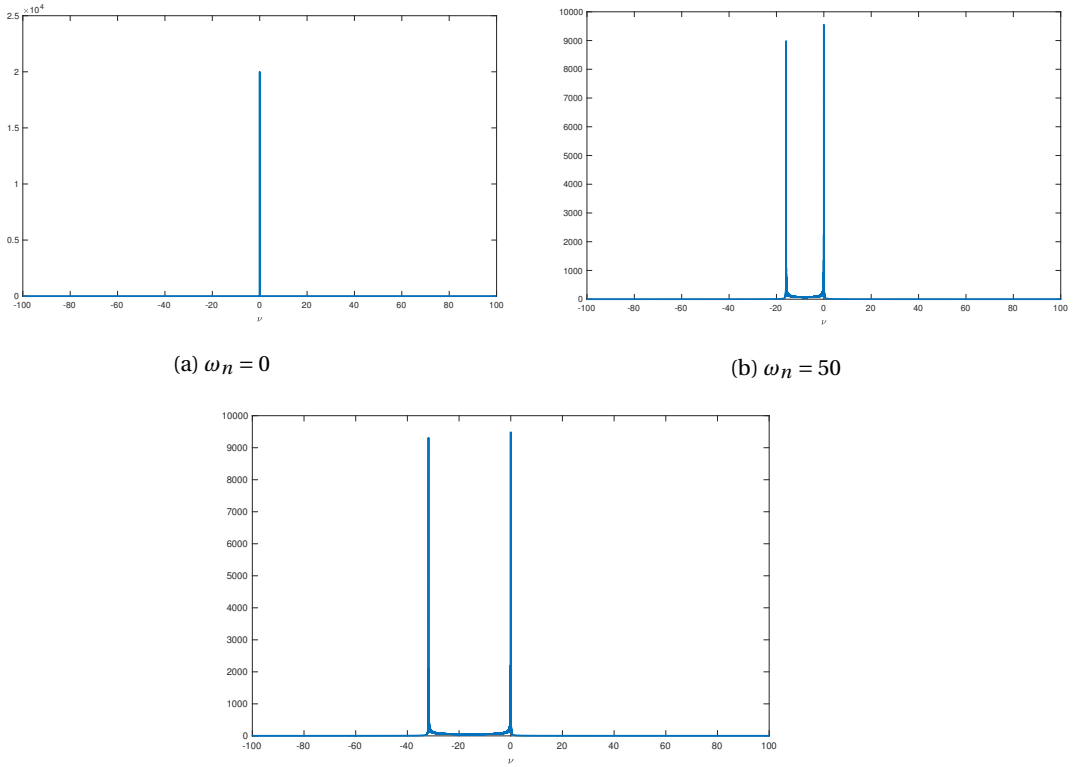
The Fourier transform of g for a fixed ω_n is given by

$$\hat{g}(v) := \int_{\mathbb{R}} \left(e^x + 1\right)^{-\mathrm{i}\omega_n} e^{-\mathrm{i}vx} dx. \tag{4.56}$$

Figure 4.2 shows the modulus of the Fourier transform of the function g for fixed $\omega_n = \{0, 50, 100\}$ [1]. The figure illustrates the decay of the characteristic function with two Dirac delta functions. Clearly this function is bandlimited, since the Fourier transform of the function will be zero except at the peaks in the Dirac delta functions, $\delta(\cdot)$. Therefore, we are allowed to approximate (4.45) by Theorem 4.1.

Let us define $h = \frac{1}{2^m}$, $j = k$ for $k = k_1, \ldots, k_2$. By Theorem 4.1 it follows that

[1] These particular values for ω_n are considered, since they represent the lowest, the middle and the highest value of ω_n in the pricing of an arithmetic Asian call option by the SWIFT-SIA method at scale $m = 4$ under GBM dynamics.



(a) $\omega_n = 0$

(b) $\omega_n = 50$

(c) $\omega_n = 100$

Figure 4.2: The modulus of the Fourier transform of g, for $\omega_n \in \{0, 50, 100\}$.

$$\begin{aligned} &\int_{\mathbb{R}} g(x)\operatorname{sinc}\left(2^m x-k\right)dx \approx hg\left(h\cdot k\right) \\ &\qquad\qquad \Leftrightarrow \\ &\int_{\mathbb{R}} \left(e^x+1\right)^{-\imath\omega_n}\operatorname{sinc}\left(2^m x-k\right)dx \approx \frac{1}{2^m}\left(e^{\frac{k}{2^m}}+1\right)^{-\imath\omega_n}, \end{aligned} \tag{4.57}$$

such that the matrix $M_{(n,k)}$ defined in (4.48) can be approximated by

$$M_{(n,k)} \approx 2^{-\frac{m}{2}}\left(e^{\frac{k}{2^m}}+1\right)^{-\imath\omega_n}. \tag{4.58}$$

Substituting (4.58) in (4.45) yields,

$$\begin{aligned} \hat{f}_{Z_{i-1}}(\omega_n) &\approx \sum_{k=k_1}^{k_2} c_{m,k}^{i-1} 2^{\frac{m}{2}} \int_{\mathbb{R}} \left(e^x+1\right)^{-\imath\omega_n}\operatorname{sinc}\left(2^m x-k\right)dx \\ &\approx \sum_{k=k_1}^{k_2} c_{m,k}^{i-1} 2^{-\frac{m}{2}}\left(e^{\frac{k}{2^m}}+1\right)^{-\imath\omega_n}. \end{aligned} \tag{4.59}$$

Recall (4.24) where the characteristic function $\hat{f}_{Y_i}$ was stated. When we apply the previous result, we find

$$\begin{aligned} \hat{f}_{Y_i}(\omega_n) &= \hat{f}_R(\omega_n)\hat{f}_{Z_{i-1}}(\omega_n) \\ &\approx \hat{f}_R(\omega_n)\sum_{k=k_1}^{k_2} c_{m,k}^{i-1} 2^{-\frac{m}{2}}\left(e^{\frac{k}{2^m}}+1\right)^{-\imath\omega_n}, \end{aligned} \tag{4.60}$$

where the density coefficients, defined in (3.26) are given by

$$c_{m,k}^{i} \approx \frac{2^{\frac{m}{2}}}{2^{N-1}}\sum_{n=0}^{2^N-1}\text{Real}\left\{\hat{f}_{Y_i}\left(\frac{2n+1}{2^N}\pi 2^m\right)e^{\frac{\imath k\pi(2n+1)}{2^N}}\right\}. \tag{4.61}$$

The computation of the matrix $M_{(n,k)}$ is no longer an expensive procedure. The numerical integration method required a total of $\mathcal{O}\left(N_q 2^N(k_2-k_1)\right)+\mathcal{O}\left(2^{2N}(k_2-k_1)\right)$ computations, whereas the entries of matrix $M_{(n,k)}$ are now found by one single function evaluation, which reduces the computational complexity tremendously. The SWIFT method with the sinc integral approximation implemented will be denoted as the SWIFT-SIA method.

4.7. Alternative: the ASCOS method

The ASCOS method (*ASian COSine*) is a Fourier based method proposed by B. Zhang and C.W. Oosterlee [25]. The ASCOS method was proposed for the valuation of geometric Asian options and arithmetic Asian options under Lévy asset price processes. This section will explain the alternative approach for both the geometric Asian option and the arithmetic Asian option. The COS method approximates the density function of a random variable by its Fourier cosine expansion, as explained in Section 3.5.

4.7.1. ASCOS method: valuation of geometric Asian options

For the valuation of a geometric Asian option by the SWIFT method, the state variable x was defined in (4.8), such that the characteristic function of this variable resulted in (4.10), i.e.

$$\hat{f}_x(\omega; x_0) = e^{-\mathrm{i}\omega x_0} \prod_{i=1}^{M_d} \hat{f}_R\left(\frac{M_d - i + 1}{M_d + 1}\omega\right). \tag{4.62}$$

Substituting (4.62) in the COS pricing formula, (3.43), results in the ASCOS pricing formula for European-style geometric Asian call options,[25],

$$\begin{aligned} V_{\text{ASCOS}}(S_0, t_0) &= \sum_{k=0}^{N-1} \text{Real}\left\{\hat{f}_x\left(\frac{k\pi}{b-a}\right)\exp\left(-\mathrm{i}\frac{ak\pi}{b-a}\right)\right\} \mathcal{V}_k^{\text{ASCOS}}, \\ &= \sum_{k=0}^{N-1} \text{Real}\left\{e^{-\mathrm{i}\frac{k\pi}{b-a}x_0} \prod_{i=1}^{M_d} \hat{f}_R\left(\frac{M_d - i + 1}{M_d + 1}\frac{k\pi}{b-a}\right)\exp\left(-\mathrm{i}\frac{ak\pi}{b-a}\right)\right\} \mathcal{V}_k^{\text{ASCOS}}. \end{aligned} \tag{4.63}$$

The ASCOS payoff coefficients, $\mathcal{V}_k^{\text{ASCOS}}$ are defined by

$$\mathcal{V}_k^{\text{ASCOS}} = \frac{2}{b-a}\left(\chi_k\left(\log(K), b\right) - K\psi_k\left(\log(K), b\right)\right), \tag{4.64}$$

with

$$\chi_k(x_1, x_2) = \int_{x_1}^{x_2} e^y \cos\left(k\pi\frac{y-a}{b-a}\right) dy$$

and

$$\psi_k(x_1, x_2) = \int_{x_1}^{x_2} \cos\left(k\pi\frac{y-a}{b-a}\right) dy,$$

which admit analytical solutions.

4.7.2. ASCOS method: valuation of arithmetic Asian options

In the ASCOS method, the characteristic function of state variable Y_{M_d} was obtained recursively by means of a Fourier cosine expansion, after which the Clenshaw-Curtis quadrature rule was applied to compute the integral (4.25).

For each monitoring date $i = 2, \ldots, M_d$ the probability density function f_{Y_i} will be approximated by its characteristic function. The probability density function of Y_i is approximated by its Fourier cosine expansion (3.42), with N expansion points, i.e.,

$$f_{Y_i}(y) \approx \frac{2}{b-a} \sum_{k=0}^{N-1} \text{Real}\left\{\hat{f}_{Y_i}\left(\frac{k\pi}{b-a}\right)\exp\left(-\mathrm{i}\frac{ak\pi}{b-a}\right)\right\} \cos\left(k\pi\frac{y-a}{b-a}\right). \tag{4.65}$$

This approximation applied to (4.25) results in

$$\hat{f}_{Z_{i-1}}(\omega) \approx \frac{2}{b-a}\sum_{k=0}^{N-1}{}'' \text{Real}\left\{\hat{f}_{Y_i}\left(\frac{k\pi}{b-a}\right)\exp\left(-\mathrm{i}\frac{ak\pi}{b-a}\right)\right\}\int_a^b \left(e^y+1\right)^{-\mathrm{i}\omega}\cos\left((y-a)\frac{k\pi}{b-a}\right)dy. \tag{4.66}$$

The integral in (4.66) reads,

$$\int_a^b \left(e^y+1\right)^{-\mathrm{i}\omega}\cos\left((y-a)\frac{k\pi}{b-a}\right)dy$$

and will be computed with the Clenshaw-Curtis quadrature rule with N_q integration points. The approach is similar to the computation of matrix $M_{(n,k)}$ in the SWIFT method, but it evaluates (4.66) instead of (4.48). So,

$$\begin{aligned}&\int_a^b \left(e^y+1\right)^{-\mathrm{i}\omega}\cos\left((y-a)\frac{k\pi}{b-a}\right)dy\\ &=\int_{-1}^1 \frac{b-a}{2}\left(\exp\left(\frac{b-a}{2}y+\frac{a+b}{2}\right)+1\right)^{-\mathrm{i}\omega}\cos\left(\left[\left(\frac{b-a}{2}y+\frac{a+b}{2}\right)-a\right]\frac{k\pi}{b-a}\right)dy.\end{aligned} \tag{4.67}$$

Following [25] the integral can be approximated by

$$\begin{aligned}&\int_{-1}^1 \frac{b-a}{2}\left(\exp\left(\frac{b-a}{2}y+\frac{a+b}{2}\right)+1\right)^{-\mathrm{i}\omega}\cos\left(\left[\left(\frac{b-a}{2}y+\frac{a+b}{2}\right)-a\right]\frac{k\pi}{b-a}\right)dy\\ &\approx \left(D^T d\right)^T y\\ &= w^T y,\end{aligned} \tag{4.68}$$

with D a $\left(\frac{N_q}{2}+1\right)\times\left(\frac{N_q}{2}+1\right)$ matrix with elements

$$D(l,p) = \frac{2}{N_q}\left(\frac{\pi l p}{\frac{N_q}{2}}\right)\cdot\begin{cases}\frac{1}{2}, & \text{if } p=\left\{0,\frac{N_q}{2}\right\}\\ 1, & \text{otherwise.}\end{cases} \tag{4.69}$$

The vectors d and the elements y_p, $\left(p=0,1,\ldots,\frac{N_q}{2}\right)$, are defined as

$$\begin{aligned}d &:= \left(1,\frac{2}{1-4},\frac{2}{1-16},\ldots,\frac{2}{1-(N_q-2)^2},\frac{1}{1-N_q^2}\right)\\ y_p &:= h\left(\cos\left(\frac{p\pi}{N_q}\right)\right)+h\left(-\cos\left(\frac{n\pi}{N_q}\right)\right),\end{aligned} \tag{4.70}$$

where

$$h(x) = \frac{b-a}{2}\left(\exp\left[\frac{b-a}{2}x+\frac{a+b}{2}\right]+1\right)^{-\mathrm{i}\omega}\cos\left(\left[\left(\frac{b-a}{2}x+\frac{a+b}{2}\right)-a\right]\frac{k\pi}{b-a}\right).$$

By the ASCOS method, the price of the arithmetic Asian call can be written as

$$V(S(0),t_0) = e^{-r(T-t_0)} \sum_{k=0}^{N-1} \text{Real}\left(\hat{f}_{Y_{M_d}}\left(\frac{k\pi}{b-a}\right)\exp\left(-\imath k\pi \frac{a}{b-a}\right)\right)\cdot \mathcal{V}_k^{\text{ASCOS}}, \tag{4.71}$$

with $\mathcal{V}_k^{\text{ASCOS}}$ the payoff coefficients of the ASCOS method, given by

$$\mathcal{V}_k^{\text{ASCOS}} := \frac{2}{b-a}\left[\frac{S(0)}{M_d+1}\chi(x^*,b) + \left(\frac{S(0)}{M_d+1} - K\right)\psi(x^*,b)\right], \tag{4.72}$$

in which χ,ψ satisfy equations (3.49), (3.50) and $x^* = \log\left(\frac{K(M_d+1)}{S(0)}\right)$.

Summarizing, the previous sections have extended the SWIFT method to the pricing of geometric Asian options and arithmetic Asian options. The valuation of geometric Asian options was straightforward, whereas the valuation of arithmetic Asian options have shown to be more involved. We have found two ways to determine the matrix $M_{(n,k)}$. A numerical integration technique (Clenshaw-Curtis quadrature) resulted in the SWIFT-CC method and an integral approximation (Theorem 4.1) resulted in the SWIFT-SIA method. In the next section numerical results will be stated for both determinations of $M_{(n,k)}$ and will be compared in terms of speed and accuracy.

4.8. Numerical results

This section contains several numerical examples of Asian option pricing with the SWIFT method. With these examples we can show robustness and high accuracy of the SWIFT method, while the necessary CPU time to compute the option values remains small. The performance of the SWIFT method will be compared to the ASCOS method, for European-style Asian options under GBM, NIG and CGMY dynamics. First of all the SWIFT method will be applied to price geometric Asian options, after which arithmetic Asian options prices will be computed.

4.8.1. Geometric Asian option pricing

The test cases for geometric Asian options with the GBM process, the NIG process and the CGMY process as underlying have the following data sets and reference values dependent on the number of monitoring dates M_d. The reference values of the valuation of geomtetric Asian call options under GBM dynamics are obtained by the ASCOS method with $N = 4096$ cosine expansion terms. In the case of an geometric Asian option with NIG or CGMY dynamics, we have used a reference value obtained by the SWIFT method with a high scale of approximation $m = 10$. Figure 4.3 illustrates the probability density functions of the three test cases. As the figure shows are the NIG and the CGMY processes more heavy tailed compared to the Geometric Brownian Motion.

- Data set GBM test case and reference values:

$$S_0 = 100, K = 110, r = 0.0367, \sigma = 0.17801, T = 1$$

  - $M_d = 12$: $V_{GBM}(S_0,0) = 1.251141891921760$
  - $M_d = 50$: $V_{GBM}(S_0,0) = 1.299030113811593$
  - $M_d = 100$: $V_{GBM}(S_0,0) = 1.307126980801588$

- Data set NIG test case and reference values:

$$S_0 = 100, K = 110, r = 0.1, \sigma = 0.25, T = 1, \alpha = 6.1882, \beta = -3.8941, \partial = 0.1622.$$

  - $M_d = 12$: $V_{NIG}(S_0,0) = 5.181988525438173$,

- $M_d = 50$: $V_{NIG}(S_0, 0) = 5.296683183954018$,
- $M_d = 100$: $V_{NIG}(S_0, 0) = 5.316056485053149$.

- Data set CGMY test case and reference values:

$$S_0 = 100, K = 110, r = 0.1, \sigma = 0, T = 1, C = 1, G = 5, M = 5, Y = 0.1,$$

- $M_d = 12$: $V_{CGMY}(S_0, 0) = 3.774380007547238$,
- $M_d = 50$: $V_{CGMY}(S_0, 0) = 3.859060576648810$,
- $M_d = 100$: $V_{CGMY}(S_0, 0) = 3.873409411006411$.

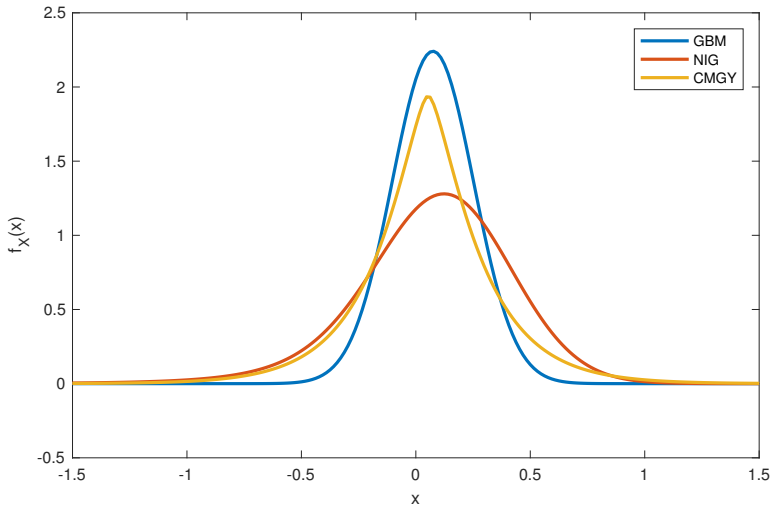


Figure 4.3: The approximated probability density functions of the three parameter sets for the GBM process, the NIG process and the CGMY process.

First we will show exponential convergence of the SWIFT method under GBM dynamics. Figure 4.4 confirms exponential convergence, since for increasing m-values the error decreases exponentially up to $m = 6$. Table 4.1 shows that the SWIFT method is an excellent method, since it computes the option value in 0.53 milliseconds up to basis points accuracy at scale of approximation $m = 4$. If the number of monitoring dates increases, the accuracy of the SWIFT method remains excellent, however the time required to compute the option value increases significantly.

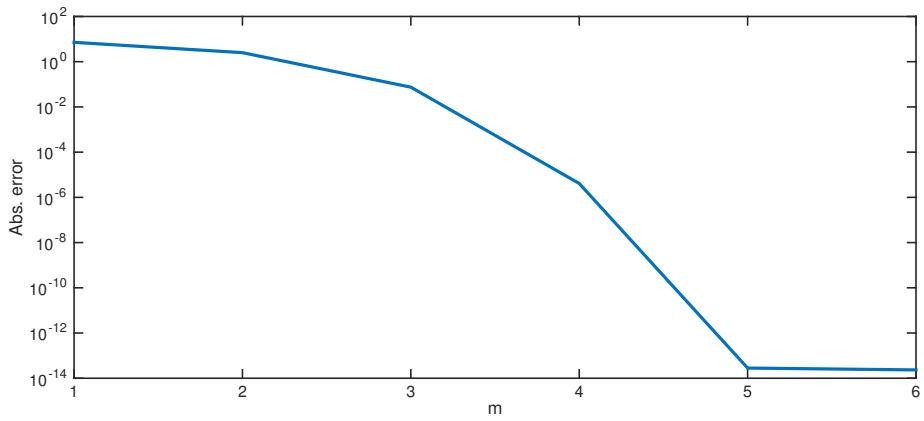


Figure 4.4: Convergence of the geometric Asian options for the GBM test case with $M_d = 12, S_0 = 100, K = 110, r = 0.0367, \sigma = 0.17801, T = 1$.

		SWIFT					
M_d	Time and error	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
12	Abs error	7.182	2.509	$7.464e-02$	$4.111e-06$	$2.819e-14$	$2.331e-14$
	CPU time	0.45	0.42	0.53	0.68	0.85	0.93
50	Abs error	7.195	2.571	$8.060e-02$	$2.483e-06$	$1.332e-14$	$1.576e-14$
	CPU time	0.39	0.41	0.56	0.73	0.89	1.75
100	Abs error	7.190	2.573	$8.061e-02$	$2.178e-06$	$3.996e-15$	$5.329e-15$
	CPU time	0.43	0.58	0.66	0.94	1.41	3.33

Table 4.1: The absolute error and the CPU time (in milliseconds) of the European style geometric Asian option under **GBM** dynamics by the SWIFT method. The reference values are obtained by the ASCOS method with $N = 4096$.

Table 4.2 shows that the SWIFT method is comparable to the ASCOS method. However, in the ASCOS method to obtain an option value with absolute error up to thirteen digits accurately we need only $N = 64$ Fourier expansion terms. To obtain such accurate option values by the SWIFT method, the scale of approximation needs to be $m = 6$, which corresponds to 134 approximation terms for the aforementioned GBM test case. This means that the SWIFT density approximation requires more than double the amount of terms in the expansion compared to the ASCOS method to reach basis point accuracy. In terms of CPU time the ASCOS method also beats the SWIFT method, especially if the number of monitoring dates increases. The CPU time required for the ASCOS method remains of the same order (< 1 millisecond), whereas the necessary CPU time of the SWIFT method increases drastically. If we consider $M_d = 100$, the SWIFT method is ten times slower than the ASCOS method. On the other hand, the scale of approximation m is the only parameter that needs to be chosen in advance, which underwrites the robustness of the SWIFT method.

		ASCOS	SWIFT		
M_d	Time and error	$N=64$	$m=4$	$m=5$	$m=6$
12	Abs error	$2.220e-16$	$4.111e-06$	$2.819e-14$	$2.331e-14$
	CPU time	0.26	0.68	0.85	0.93
50	Abs error	$1.443-14$	$2.483e-06$	$1.332e-14$	$1.576e-14$
	CPU time	0.23	0.73	0.89	1.75
100	Abs error	$5.107e-15$	$2.178e-06$	$3.996e-15$	$5.329e-15$
	CPU time	0.33	0.94	1.41	3.33

Table 4.2: The absolute error and the CPU time (in milliseconds) of the European style geometric Asian option under **GBM** dynamics by the ASCOS method and the SWIFT method. The reference values are obtained by the ASCOS method with $N = 4096$.

The performance of the SWIFT method for the NIG and CGMY examples is presented in Table 4.3 and Table 4.4. The geometric Asian call prices are shown for $M_d = 12, 50, 100$ monitoring dates and increasing scale of approximation m. Due to the confirmed exponential convergence, we will compare the resulting option values to the reference value obtained by the SWIFT method with $m = 10$. In less than a millisecond we can price the value of an European-style geometric Asian option by means of SWIFT for both $M_d = 12$ and $M_d = 50$. Table 4.3 reveals that increasing the number of monitoring dates, increases the CPU time significantly. If we compare our results for NIG and CGMY dynamics to the results in [25], we see that our methods gives the same option prices up to basis point accuracy, but our method is slower compared to the ASCOS method. To reach basis point accuracy by the SWIFT method, the valuation takes more than two milliseconds. Similar to the valuation of geometric Asian options under GBM dynamics, the SWIFT method is outperformed by the ASCOS method in the case of NIG and CGMY underlying price processes.

		SWIFT					
M_d	Time and error	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
12	Abs error	2.868	0.6866	$1.114e-03$	$2.970e-05$	$7.708e-07$	$2.945e-06$
	CPU time	0.44	0.85	0.52	0.59	1.26	1.29
50	Abs error	3.326	0.565	$6.224e-04$	$4.182e-06$	$5.681e-06$	$1.231e-06$
	CPU time	0.48	0.60	0.73	0.99	1.73	2.96
100	Abs error	3.433	0.547	$5.502e-04$	$7.942e-06$	$4.800e-06$	$3.738e-07$
	CPU time	0.90	1.03	0.94	2.51	6.50	7.05

Table 4.3: The absolute error and the CPU time (in milliseconds) of the European style geometric Asian option under **NIG** dynamics by the SWIFT method. The reference values are obtained by the SWIFT method with $m = 10$.

		SWIFT					
M_d	Time and error	$m=1$	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$
12	Abs error	2.598	2.737	0.1006	$1.267e-02$	$4.134e-03$	$2.769e-04$
	CPU time	0.50	0.91	1.19	1.10	1.32	2.21
50	Abs error	0.9877	3.195	0.3507	$5.309e-02$	$5.176e-03$	$3.1705e-04$
	CPU time	0.91	1.10	1.41	2.41	4.14	7.26
100	Abs error	0.7112	3.2250	0.3850	$5.337e-02$	$5.377e-03$	$4.1758e-04$
	CPU time	1.69	2.63	2.74	4.05	7.28	14.81

Table 4.4: The absolute error and the CPU time (in milliseconds) of the European style geometric Asian option under **CGMY** dynamics by the SWIFT method. The reference values are obtained by the SWIFT method with $m = 10$.

4.8.2. Arithmetic Asian option pricing

In this section we present several numerical examples of the valuation of arithmetic Asian options by the SWIFT method. We present three Lévy processes for the underlying: the GBM process, the NIG process and the CGMY process. All reference values are obtained by the ASCOS method with $N = 1000$ Fourier expansion terms and $N_q = 1500$ points in the numerical integration by the Clenshaw-Curtis quadrature rule.

- Data set GBM test case and reference values:

$$S_0 = 100, K = 90, r = 0.0367, \sigma = 0.17801, T = 1$$

  - $M_d = 12$: $V(S_0, 0) = 11.904915748797190$
  - $M_d = 50$: $V(S_0, 0) = 11.932938204587398$
  - $M_d = 100$: $V(S_0, 0) = 11.937676122149343$

- Data set NIG test case and reference values:

$$S_0 = 100, K = 110, r = 0.0367, \sigma = 0, T = 1, \alpha = 6.1882, \beta = -3.8941, \delta = 0.1622$$

  - $M_d = 12$: $V(S_0, 0) = 1.013550867167349$
  - $M_d = 50$: $V(S_0, 0) = 1.037700798283591$
  - $M_d = 100$: $V(S_0, 0) = 1.041904347350710$

- Data set CGMY test case and reference values:

$$S_0 = 100, K = 100, r = 0.1, \sigma = 0.25, T = 1, C = 1, G = 5, M = 5, Y = 0.1$$

  - $M_d = 12$: $V(S_0, 0) = 11.502766525425102$

- $M_d = 50$: $V(S_0,0) = 11.592233230967537$
- $M_d = 100$: $V(S_0,0) = 11.607481902427793$

Let us recall the matrix $M_{(n,k)}$ in equation (4.48):

$$M_{(n,k)} = M_{(n,k)} = 2^{\frac{m}{2}} \int_{\mathbb{R}} \left(e^x + 1\right)^{-1\omega_n} \operatorname{sinc}\left(2^m x - k\right) dx, \tag{4.73}$$

Matrix $M_{(n,k)}$ has to be computed once, however the computation of this matrix is the bottleneck in the speed of the method, since numerical integration is expensive to perform. Section 4.6 contains two different approaches to compute the matrix. The first computation is done by the numerical integration technique stated in Section 4.6.1 and denoted by the SWIFT-CC method, the second method to recover the characteristic function is by the SWIFT-SIA mehtod, which is based on the sinc integral approximation resulting from Theorem 4.1, Section 4.6.2. The performance of both SWIFT methods will be stated and the results will be compared to the ASCOS method.

SWIFT-CC method

The convergence of the SWIFT-CC method is shown in Figure 4.5. We can conclude that the SWIFT method shows rapid convergence in the scale of approximation m. Since the matrix $M_{(n,k)}$ is computed by the Clenshaw-Curtis quadrature rule, it is very important to select a sufficient amount of integration points N_q. Table 4.5 shows that the increase in the number of integration points N_q results in a smaller absolute error in the option price, as expected. However, if we select more than $N_q = 400$ integration points, the absolute error remains of the same order, whereas the CPU time required to compute the option value increases drastically. Therefore the choice has been made to set $N_q \leq 400$ integration points for each scale m and we can conclude from the results in Table 4.5 that we do not lose significant accuracy in the final option value.

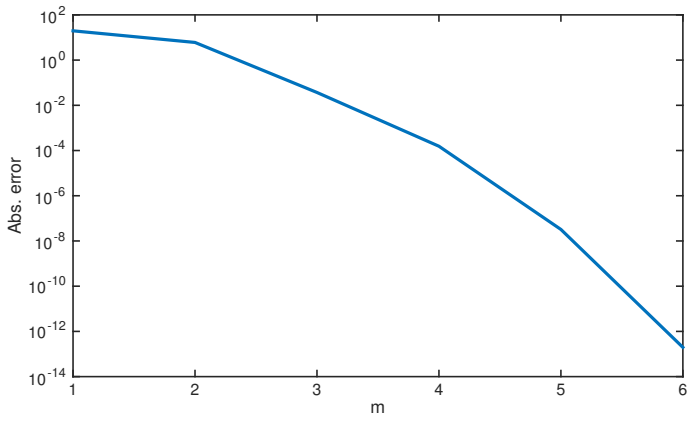


Figure 4.5: Convergence of SWIFT-CC method of the valuation of an European-style arithmetic Asian options under GBM dynamics with $M_d = 12, S_0 = 100, K = 90, r = 0.0367, \sigma = 0.17801, T = 1$.

Comparing the results of the valuation of arithmetic Asian options by the SWIFT-CC method to the ASCOS method, we can see that the SWIFT method is outperformed by the ASCOS method in terms of speed and accuracy. Table 4.6 states that where the ASCOS method computes a six decimals accurate option value in less than two seconds, the SWIFT-CC method needs approximately thirty seconds to reach the same level of accuracy for $M_d = 12$.

Another difference between the ASCOS method and the SWIFT-CC method we can conclude from Table 4.6 is that for the ASCOS method the CPU time does not depend much on the number of monitoring dates M_d. The speed of the method for $M_d = 12$ is of the same order as the speed of the method

		$N_q = 100$	$N_q = 200$	$N_q = 400$	$N_q = 600$
$m = 4$	Abs error	0.0022	$1.5477e-04$	$1.5477e-04$	$1.5477e-04$
	CPU time	4.33	5.27	7.18	8.95
$m = 5$	Abs error	0.0688	$2.8267e-08$	$3.2320e-08$	$3.2320e-08$
	CPU time	20.18	20.84	28.71	35.79
$m = 6$	Abs error	0.0688	$3.6680e-09$	$2.7711e-13$	$1.9007e-13$
	CPU time	77.98	95.89	118.89	147.28

Table 4.5: Absolute error and the required CPU time (in seconds) for the arithmetic Asian call option with several scales of approximation m of the SWIFT-CC with several number of integration points N_q. The underlying process is **GBM** with $M = 12$.

at $M_d = 100$. The speed of the SWIFT-CC method is affected by the number of monitoring dates considered, the computation of $M_d = 100$ is approximately twice as slow as the case is for $M_d = 12$. Since $M_{(n,k)}$ has to be computed only once outside the for-loop in Algorithm 4.1, we can conclude that the SWIFT method is (without the integration) slower compared to the COS method. This has also been shown in the performance of the SWIFT method applied to geometric Asian options.

		ASCOS			SWIFT-CC		
M_d	Time and error	$N = 128$	$N = 256$	$N = 384$	$m = 4$	$m = 5$	$m = 6$
		$N_q = 200$	$N_q = 400$	$N_q = 600$	$N_q = 200$	$N_q = 400$	$N_q = 400$
12	Abs error	$8.36e-07$	$4.68e-13$	$3.69e-13$	$1.5477e-04$	$3.23e-08$	$1.90e-13$
	CPU time	1.60	8.64	23.67	8.67	37.21	230.10
50	Abs error	$7.42e-06$	$5.32e-07$	$1.24e-09$	0.0015	0.0011	$1.6065e-08$
	CPU time	1.72	8.45	24.58	15.64	83.40	382.85
100	Abs error	$1.139e-05$	$3.307e-06$	$1.436e-07$	$2.029e-02$	$6.201e-03$	$2.232e-05$
	CPU time	1.68	8.94	24.47	17.61	83.33	436.40

Table 4.6: The underlying process is **GBM** and the reference values are computed by the ASCOS method with $N = 1000$ and $N_q = 1500$.

SWIFT-SIA method

Since the function $g(x) = (e^x + 1)^{-\imath\omega_n}$ is bandlimited and satisfies the property of Theorem 4.1, the matrix $M_{(n,k)}$ can be approximated, which can speed up the SWIFT method tremendously.

$$\begin{aligned} M_{(n,k)} =& M_{(n,k)} = 2^{\frac{m}{2}} \int_{\mathbb{R}} \left(e^x + 1\right)^{-\imath\omega_n} \operatorname{sinc}\left(2^m x - k\right) dx \\ \approx& \frac{1}{2^m} \left(e^{\frac{k}{2^m}} + 1\right)^{-\imath\omega_n} \\ =:& M^*_{(n,k)}. \end{aligned} \tag{4.74}$$

The computation of the matrix $M^*_{(n,k)}$ is no longer expensive and its computational time within the SWIFT-SIA method is negligible. The most interesting part of this extension of the SWIFT framework is whether we loose accuracy in the resulting option value due to this integral approximation or not. Figure 4.6 shows the convergence of the SWIFT-SIA method. As we can see, the convergence remains in m.

In order to check whether we lose accuracy, Table 4.7 shows the SWIFT-CC method next to the SWIFT-SIA method. As expected, the CPU time required to compute the value of an arithmetic Asian option has decreased drastically, for an option with $M_d = 12$ monitoring dates, it takes the SWIFT-SIA method 0.066 seconds to compute the option value at the same accuracy as the SWIFT-CC reaches at 230 seconds. Surprisingly, the SWIFT-SIA method performs in some cases even better than the SWIFT-CC

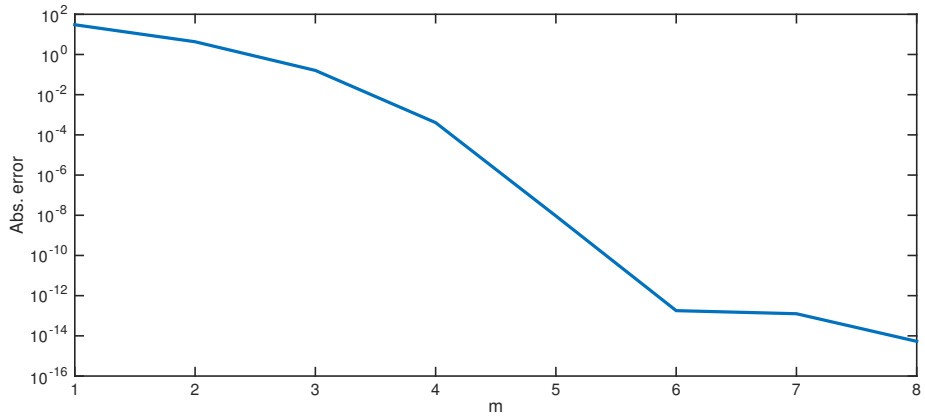


Figure 4.6: Convergence of the SWIFT-SIA method of the valuation of an European-style arithmetic Asian options under GBM dynamics with $M_d = 12, S_0 = 100, K = 90, r = 0.0367, \sigma = 0.17801, T = 1$.

method at the same scale of approximation m. This means that the approximation $M^*_{(n,k)}$ must be better than the numerical integration with the Clenshaw-Curtis quadrature rule.

		SWIFT-SIA			SWIFT-CC		
M_d	Time and error	$m = 4$	$m = 5$	$m = 6$	$m = 4$ $N_q = 200$	$m = 5$ $N_q = 400$	$m = 6$ $N_q = 400$
12	Abs error	$4.17e-04$	$9.24e-09$	$1.82e-13$	$1.5477e-04$	$3.23e-08$	$1.90e-13$
	CPU time	0.023	0.031	0.066	8.67	37.21	230.10
50	Abs error	$1.46e-02$	$9.35e-05$	$2.21e-10$	0.0015	0.0011	$1.6065e-08$
	CPU time	0.026	0.077	0.28	15.64	83.40	382.85
100	Abs error	$1.445e-02$	$9.335e-05$	$2.832e-06$	$2.029e-02$	$6.201e-03$	$2.232e-05$
	CPU time	0.042	0.13	0.45	17.61	83.33	436.40

Table 4.7: The absolute error and CPU time (in seconds) of the SWIFT-CC method and the SWIFT-SIA method. The underlying process is **GBM**.

Since the SWIFT-SIA method outperforms the SWIFT-CC method, we will compare the SWIFT-SIA method to the ASCOS method. Figure 4.7 illustrates the performance comparison. On the x-axis the CPU time is listed, on the y-axis the absolute error is stated. The SWIFT-SIA method reaches a higher level of accuracy in less time compared to the ASCOS method, as the red line is always on the left of the blue line. We can conclude that the SWIFT-SIA method performs better than the ASCOS method in terms of speed and accuracy. Note that Figure 4.7 represents the valuation of an arithmetic Asian option value monitored with $M_d = 12$.

So far, in the results stated the underlying process has been the GBM process. The numerical experiments were replicated for the NIG and the CGMY process. Table 4.8 lists the absolute error of the valuation of arithmetic Asian options under NIG dynamics and Table 4.9 lists the results on the valuation of arithmetic Asian options under CGMY dynamics. If we compare the results of the NIG and CGMY test cases to the GBM test case, we can conclude that the results of the GBM test case appear to be much better.

Figure 4.8 and Figure 4.9 illustrate the difference between the speed and the accuracy of the methods for the three underlying processes. The CPU time for the computation of arithmetic Asian options by means of the SWIFT-SIA method is comparable with the ASCOS method, however the results in terms of accuracy of the NIG and CGMY underlying test cases are not competitive to the ASCOS method.

We have shown exponential convergence of the SWIFT-SIA method for the valuation of arithmetic Asian options under GBM dynamics. From Tables 4.8 and 4.9 we see that if we consider $m > 4$, the

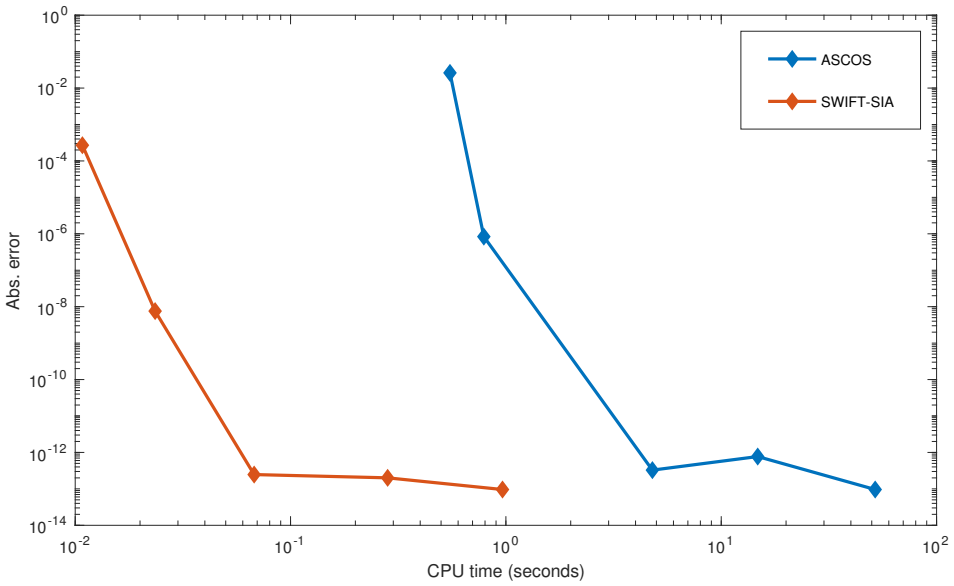


Figure 4.7: The absolute error and CPU time corresponding to the valuation of an European-style arithmetic Asian option with **GBM** dynamics by means of the SWIFT-SIA method (red) and the ASCOS method (blue).

		SWIFT-SIA			
M_d	CPU time and Abs.error	$m=2$	$m=4$	$m=6$	$m=8$
12	Abs error	4.487	$2.334e-02$	$2.904e-04$	$1.307e-06$
	CPU time	0.0027	0.014	0.077	1.06
50	Abs error	5.295	0.2307	$9.509e-04$	$8.091e-05$
	CPU time	0.0087	0.027	0.30	4.23
100	Abs error	10.21	0.1596	$1.089e-03$	$9.872e-05$
	CPU time	0.015	0.043	0.52	6.75

Table 4.8: Absolute error of the SWIFT-SIA method for an arithmetic Asian call option under the **NIG** dynamics.

absolute error remains of the same order. This might be due to the accuracy of the reference value we have taken into consideration. The reference value has been obtained with the ASCOS method with $N = 1500$ expansion terms and $N_q = 1000$ integration points. These reference values result in the stagnation of the accuracy of the SWIFT-SIA method. If we consider our reference value from the SWIFT-SIA method with a high scale of approximation ($m \in \{9,10\}$), it becomes clear that the SWIFT-SIA method is more accurate than the previously stated comparison might suggest. The reference value for the valuation of an European-style arithmetic call option under NIG dynamics is obtained at scale $m = 10$ and $M_d = 12$. Due to the heavier tails of the CGMY process considered in these numerical experiments, the program ran out-of-memory if we considered $m = 10$. The number of approximation terms became too big considering the cumulant based interval $[a,b]$. Therefore the reference value of the CGMY test case is obtained with $m = 9$.

Table 4.10 and Table 4.11 present the convergence and the CPU time of an arithmetic Asian option for the NIG and CGMY test case with $M_d = 12$. Increasing the scale of approximation m, results in option values converging to the reference value obtained with the SWIFT method with $m \in \{9,10\}$. The hypothesis that the reference values obtained by the ASCOS method were not representative to the SWIFT-SIA method appears to be correct for the underlying NIG and CGMY dynamics. The difference between between the absolute error listed in Table 4.8 and the error presented in Table 4.10 is three digits at scale $m = 8$.

		SWIFT-SIA			
M_d	CPU time and Abs.error	$m=2$	$m=4$	$m=6$	$m=8$
12	Abs error	3.617	$9.0869e-05$	$4.4084e-07$	$4.679e-07$
	CPU time	0.0059	0.016	0.18	3.17
50	Abs error	5.689	$2.371e-02$	$3.795e-07$	$4.219e-07$
	CPU time	0.012	0.038	0.40	5.71
100	Abs error	11.64	$7.573e-02$	$4.159e-07$	$4.274e-07$
	CPU time	0.016	0.057	0.61	8.80

Table 4.9: Absolute error of the SWIFT-SIA method for an arithmetic Asian call option under the **CGMY** dynamics.

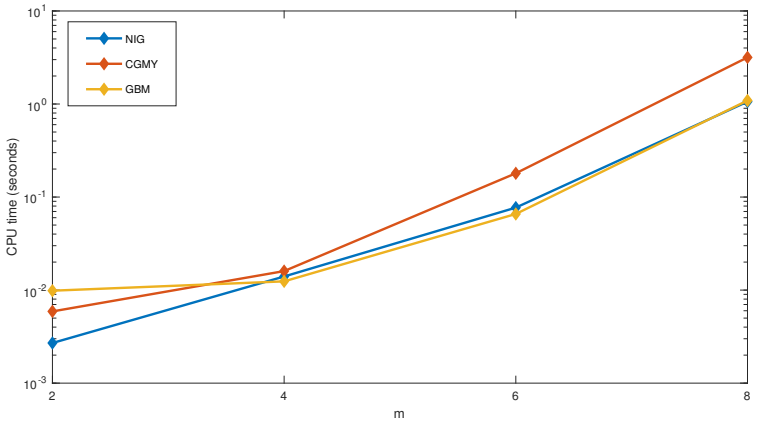


Figure 4.8: CPU time (in seconds) of the valuation of an arithmetic Asian option under GBM, NIG and CMGY dynamics for various m.

4.8.3. The put-call parity of arithmetic Asian options

The put-call parity is the relationship between a put and a call option that rely on the same stock price $S(t)$, the same strike K and have the same time to maturity T. The relationship originates from the no-arbitrage principle, which states that two portfolios with the same payoff at $t=T$, must admit the same value at $t<T$. Let us define the difference between the payoff of a fixed strike arithmetic Asian call option $V_C(S(T),T)$ and a fixed strike arithmetic Asian put option $V_P(S(T),T)$. Both options have the same strike K and maturity T, then it follows,

$$\begin{aligned} V_C(S(T),T)-V_P(S(T),T) &= \max\{A(t_0,T)-K,0\}-\max\{K-A(t_0,T),0\} \\ &= A(t_0,T)-K. \end{aligned} \tag{4.75}$$

Applying the risk neutral pricing formula, we obtain

$$\begin{aligned} V_C(S_0,t_0)-V_P(S_0,t_0) &= e^{-r(T-t_0)}\mathbb{E}^{\mathbb{Q}}\left[A(t_0,T)-K|\mathscr{F}_t\right] \\ &= e^{-r(T-t_0)}\frac{S_0}{M_d+1}\sum_{i=0}^{M_d} e^{ir\Delta t}-e^{-r(T-t_0)}K, \end{aligned} \tag{4.76}$$

such that we can price the arithmetic Asian call option in terms of the put option price,

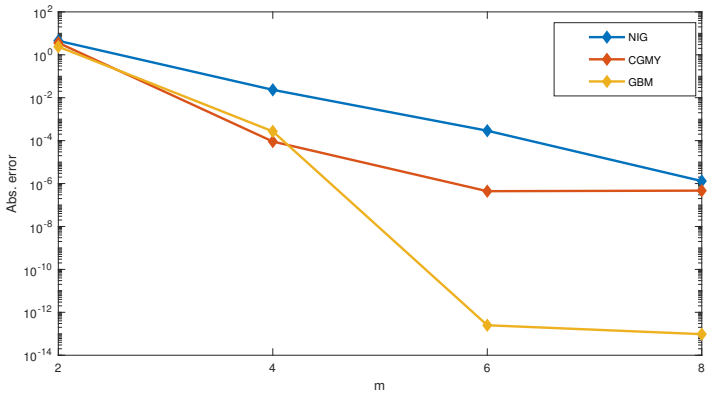


Figure 4.9: Abs. error of the valuation of an arithmetic Asian option under GBM, NIG and CMGY dynamics for various m.

		SWIFT-SIA				
M_d	CPU time and Abs.error	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$
12	Abs error	$9.710e-02$	$5.740e-03$	$2.627e-04$	$1.892e-05$	$2.557e-09$
	CPU time	0.013	0.027	0.077	0.31	1.10

Table 4.10: Absolute error and CPU time (in seconds) of the valuation of an arithmetic Asian option under NIG dynamics. Reference value is obtained by the SWIFT-SIA method with $m=10$, $V_{ref}(S_0,0)=1.013549562718903$.

		SWIFT-SIA				
M_d	CPU time and Abs.error	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$
12	Abs error	$8.664e-05$	$1.415e-08$	$7.305e-09$	$4.585e-09$	$1.012e-09$
	CPU time	0.023	0.059	0.23	0.85	3.14

Table 4.11: Absolute error and CPU time (in seconds) of the valuation of an arithmetic Asian option under CGMY dynamics. Reference value is obtained by the SWIFT-SIA method with $m=9$, $V_{ref}(S_0,0)=11.502766050995911$.

$$\begin{aligned} V_C(S_0,t_0) &= e^{-r(T-t_0)}\mathbb{E}[A(t_0,T)-K] \\ &= e^{-r(T-t_0)}\frac{S_0}{M_d+1}\sum_{i=0}^{M_d} e^{ir\Delta t} - e^{-r(T-t_0)}K + V_P(S_0,t_0). \end{aligned} \tag{4.77}$$

Since the payoff of a call option is unbounded, the truncation of the interval to $[a,b]$ might lead to large errors (see [25] for details). Therefore Zhang et al. [25] advised the use of the put-call parity if the option is deep in-the-money, admits a heavy-tailed underlying density function or if the option has a long time to maturity T. Since the SWIFT method is a highly robust method and because of the local Shannon wavelet basis, the resulting density coefficients compensate for the high payoff coefficients in the tail of the distribution. Figures 4.10 and 4.11 illustrate the density coefficients and the corresponding payoff coefficients of an European style arithmetic Asian call and put option with the same maturity T and strike K. We can conclude that the the payoff coefficients of the arithmetic put option are bounded, whereas the payoff coefficients of the arithmetic call option grow exponentially.

The difference between the accuracy of the direct valuation of an arithmetic Asian call option and application of the put-call parity, will be illustrated by the next numerical example. An European-style arithmetic Asian call option with maturity $T=10$ and strike $K=110$ will be priced by the put-call parity and by the direct method. The underlying price process follows a GBM with $S_0=100$, $r=0.1$ and $\sigma=0.25$. The arithmetic Asian option is discretely monitored with $M_d=100$ at $[0,T]$. Since we want to show the application of the put-call parity for options with a long time to maturity, we consider $T=10$. The reference value is obtained by the SWIFT-SIA method with $m=8$.

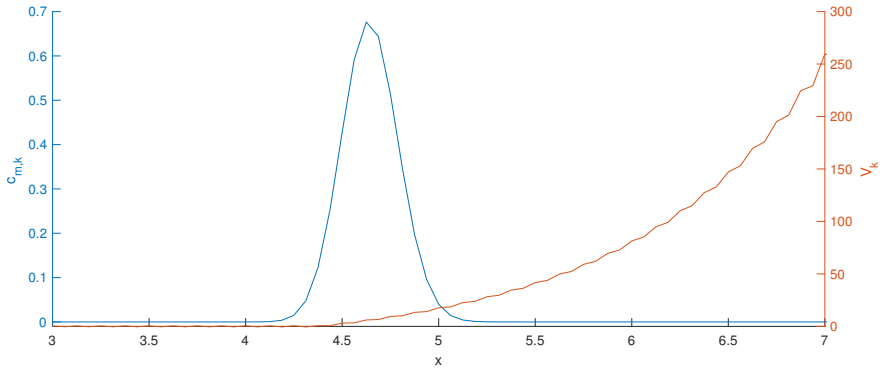


Figure 4.10: The $c_{m,k}$ and $\mathcal{V}_k$ of an arithmetic Asian call option at scale $m = 4$.

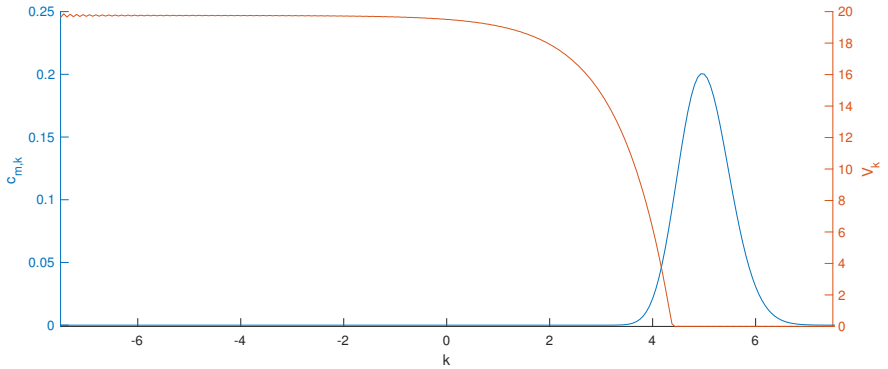


Figure 4.11: The $c_{m,k}$ and $\mathcal{V}_k$ of an arithmetic Asian put option at scale $m = 4$.

	ASCOS		
	CPU time	Abs. error (direct)	Abs. error (put-call parity)
$N = 64$ $N_q = 100$	0.13	$1.694e-02$	$6.648e-07$
$N = 128$ $N_q = 200$	0.83	$8.747e-03$	$1.212e-08$
$N = 256$ $N_q = 400$	5.31	$1.516e-05$	$5.706e-11$
$N = 384$ $N_q = 600$	12.67	$5.749e-08$	$4.023e-11$

Table 4.12: ASCOS: Put-call parity to obtain European-style arithmetic Asian call option values.

	SWIFT-SIA		
	CPU time	Abs. error (direct)	Abs. error (put-call parity)
$m = 2$	0.022	3.162	$1.156e-03$
$m = 3$	0.061	0.132	$5.637e-05$
$m = 4$	0.18	$1.268e-04$	$4.782e-08$
$m = 5$	0.56	$3.344e-11$	$4.026e-11$

Table 4.13: SWIFT-SIA: Put-call parity to obtain European-style arithmetic Asian call option values.

Table 4.12 illustrates the comparison between the put-call parity valuation and the direct valuation of an arithmetic Asian call option. The put-call parity has a positive effect on the accuracy of the obtained call option value. The put-call parity approach by the ASCOS method with $N = 128$ terms in the expansion and $N_q = 200$ integration points results in an option value that is five digits more accurate than the directly obtained option value. Next, the difference between the put-call parity approach and the direct valuation of an arithmetic Asian option by the SWIFT-SIA method will be stated. Table 4.13 shows that the influence of the put-call parity in the SWIFT-SIA method is less, compared to the ASCOS method. Within less than one second (0.56 seconds) the SWIFT-SIA method obtains directly a call option value which is eleven digits accurate, whereas the ASCOS method reaches the same level of accuracy by the valuation of the call option through the put-call parity in more than five seconds. This difference highlights the robustness of the SWIFT-SIA method, since the SWIFT-SIA method is capable of pricing arithmetic Asian call options with a long time to maturity.

5

Pricing European Options under the SABR model with SWIFT

In the pricing of European options under the SABR model the method proposed by Leitao et al. [16] reveals an interesting step in which the application of the SWIFT method could be beneficial in terms of accuracy and CPU time. The work of Leitao et al. proposes a one time-step Monte Carlo method to compute European option prices under SABR dynamics. The approach relies on an accurate approximation of the distribution function of the time-integrated variance, by means of a copula and a Fourier technique. The Fourier technique proposed in this setting is the COS method. As shown in the previous chapter, in some cases it is more efficient to replace the COS method by the SWIFT method (which was the case in the valuation of arithmetic Asian options.).

First, the one time-step SABR model will be defined, after which the SWIFT method will be applied to recover an approximation to the cumulative distribution function of the time integrated variance. The performance of the one time-step Monte Carlo method with SWIFT will be compared to the original method stated in the paper for two test cases, one originating from the literature, [1], and the second test case is a pure Monte Carlo simulation method.

5.1. The SABR model

The *Stochastic Alpha Beta Rho* (SABR) model is a stochastic local volatility model which is often used for modeling interest rates and foreign-exchange rates. The model describes a forward $S(t)$ by means of its volatility $\sigma(t)$. $S(t)$ and $\sigma(t)$ are stochastic variables with the following system of stochastic differential equations (SDEs).

Definition 5.1. The formal definition of the SABR model reads

$$\mathrm{d}S(t) = \sigma(t) S^{\beta}(t) \mathrm{d}W_S(t), \tag{5.1}$$

$$\begin{aligned} \mathrm{d}\sigma(t) &= \alpha\sigma(t)\mathrm{d}W_\sigma(t), \\ S(0) = S_0 \exp(rT) \text{ and } \sigma(0) &= \sigma_0, \end{aligned} \tag{5.2}$$

where $S(t) = \hat{S}(t)\exp(r(T-t))$ denotes the forward value of the underlying $\hat{S}(t)$, r the interest rate, S_0 the price of the underlying at time $t = 0$ and T the time of maturity of the contract. $\sigma(t)$ denotes the stochastic volatility, with $\sigma(0) = \sigma_0$, $W_S(t)$ and $W_\sigma(t)$ are two correlated Brownian motions with

constant correlation ρ, ($W_S W_\sigma = \rho t$). The model parameters are: α the volatility of the volatility (vol-vol), β the elasticity ($0 \leq \beta \leq 1$) and ρ the correlation coefficient.

From equation (5.2) we can see that the local-volatility is of the *constant elasticity of variance* (CEV) form. In the work of Leitao et al. [15] an analytical expression for the cumulative distribution function (cdf) of the SABR conditional process is obtained. With the cdf as in the work of Leitao et al. a one time-step Monte Carlo method was proposed which is based on inverting the conditional SABR cdf. This one time-step Monte Carlo method can be divided in three steps:

1. *Simulation of SABR's volatility.*
2. *Simulation of SABR's time integrated variance, conditional on the terminal value of the volatility.*
3. *Simulation of SABR's forward underlying process.*

From the system of SDEs we can conclude that the volatility follows a lognormal distribution, which yields an analytical solution. Therefore, the simulation of SABR's volatility (Step 1) is exact.

Lemma 5.1. *The analytical solution to the volatility in the SABR model is given by*

$$\sigma(T) = \sigma(0) \exp\left(\alpha W_\sigma(T) - \frac{1}{2}\alpha^2 T\right), \tag{5.3}$$

Proof. Itô integration results for the SDE of the volatility in the SABR model in

$$\begin{aligned} \int_0^t \frac{d\sigma(t)}{\sigma(t)} &= \int_0^t \alpha dW_\sigma(t) \\ &\Leftrightarrow \\ \int_0^t \frac{d\sigma(t)}{\sigma(t)} &= \alpha W_\sigma(t). \end{aligned} \tag{5.4}$$

Itô's formula results in

$$\begin{aligned} d\log(\sigma(t)) &= \frac{d\sigma(t)}{\sigma(t)} - \frac{1}{2}\frac{1}{\sigma(t)^2} d\sigma(t) d\sigma(t) \\ &\Leftrightarrow \\ d\log(\sigma(t)) &= \frac{d\sigma(t)}{\sigma(t)} - \frac{1}{2}\alpha^2 dt, \end{aligned} \tag{5.5}$$

substituting this result in (5.4) yields

$$\int_0^t d\log(\sigma(t)) + \frac{1}{2}\alpha^2 \int_0^t dt = \alpha W_\sigma(t) \Leftrightarrow \log\left(\frac{\sigma(t)}{\sigma_0}\right) = \alpha W_\sigma(t) - \frac{1}{2}\alpha^2 t, \tag{5.6}$$

taking the exponential gives the desired result. □

More involved is Step 2, since the conditional distribution of the time integrated variance given the volatility, $\int_0^T \sigma^2(s) ds | \sigma(T)$, is not known in closed form. To compute this conditional distribution we will follow the method proposed by Leitao et al. in which a copula multi-variate distribution is applied

to simulate $\int_0^T \sigma^2(s)ds|\sigma(T)$. A simulation with a copula technique requires the two marginal distributions, thus the distribution of $\int_0^T \sigma^2(s)ds$ and the distribution of $\sigma(T)$, (5.3). In [15] the approximation of the distribution of the integrated variance, $\int_0^T \sigma^2(s)ds$, is obtained by the COS method, but an alternative approach is the SWIFT method.

The next sections will elaborate on the method proposed in [15], however we will implement the SWIFT method instead of the COS method to approximate the cumulative distribution function and discuss the results in terms of accuracy and speed. The recovery of the distribution of the integrated variance shows great similarity with the recovery of the density of arithmetic Asian options. For example the Carverhill-Clewlow-Hodges factorization scheme will be implemented and the method can also benefit from the sinc integral approximation, as stated in Theorem 4.1. To gain full insight in the way the one time-step Monte Carlo method will be adjusted due to the SWIFT implementation, each step of the method will be briefly explained in the upcoming sections.

5.2. Cumulative distribution function of SABR's time integrated variance

This section will elaborate on the method to recover the CDF of the SABR time integrated variance, i.e. $\int_0^T \sigma^2(s)ds$. First of all, the time integrated variance will be approximated by its discrete analogue with $M \in \mathbb{N}$ time points:

$$\int_0^T \sigma^2(s)ds \approx \sum_{i=1}^{M} \sigma^2(t_i)\Delta t := Y(T), \tag{5.7}$$

where $i = 1,\ldots,M$, $\Delta t = \frac{T}{M}$ and $t_i = i\Delta t$. In order to find the characteristic function of $\log Y(T)$, we will apply the SWIFT method. For the application of SWIFT, a recursive procedure is developed which is similar to the recursive procedure to obtain the characteristic function for the valuation of arithmetic Asian options, as stated in Section 4.4. The definition of a sequence of logarithmic increments is given first,

$$R_{t_i} = \log\left(\frac{\sigma^2(t_i)}{\sigma^2(t_{i-1})}\right) = \log\left(\sigma^2(t_i)\right) - \log\left(\sigma^2(t_{i-1})\right). \tag{5.8}$$

As stated previously, the volatility process is distributed lognormally, therefore the increment process is independent and identically distributed, i.e. $R_{t_i} \overset{d}{=} R$. Furthermore the characteristic function of the increments is known and reads

$$\hat{f}_{R_{t_i}}(\omega) = \hat{f}_R = \exp(-\imath\omega\alpha^2\Delta t - 2u^2\alpha^2\Delta t), \tag{5.9}$$

By equation (5.8) we can write

$$\begin{aligned}
\sigma^2(t_i) =& \sigma^2(t_0)\exp\left(\log\left(\frac{\sigma^2(t_i)}{\sigma^2(t_0)}\right)\right) \\
=& \sigma^2(t_0)\exp\left(\log\left(\frac{\sigma^2(t_1)}{\sigma^2(t_0)}\right) + \log\left(\frac{\sigma^2(t_2)}{\sigma^2(t_1)}\right) + \ldots + \log\left(\frac{\sigma^2(t_i)}{\sigma^2(t_{i-1})}\right)\right) \\
=& \sigma^2(t_0)\exp\left(R_{t_1} + R_{t_2} + \ldots + R_{t_i}\right).
\end{aligned} \tag{5.10}$$

With these results we can set up a recursion in terms of $R(t_i)$ to recover the characteristic function $\hat{f}_{\log Y(T)}$. Let us specify the recursion

$$\begin{aligned} Y_1 &= R_{t_M} \\ Y_i &= R_{t_{M+1-i}} + Z_{i-1}, \end{aligned} \tag{5.11}$$

for $i = 2,\ldots,M$ and $Z_i = \log\big(1+\exp(Y_i)\big)$. As a result, the time integrated variance can be written as

$$Y(T) = \sum_{i=1}^{M} \sigma^2(t_i)\Delta t = \Delta t \sigma_0^2 \exp(Y_M). \tag{5.12}$$

From equation (5.12) the characteristic function can be written as

$$\begin{aligned} \hat{f}_{\log Y(T)}(\omega) &= \mathbb{E}\left[\exp\left(-\imath\omega \log Y(T)\right)\right] \\ &= \mathbb{E}\left[\exp\left(-\imath\omega \log(\Delta t\sigma_0^2) - \imath\omega Y_M\right)\right] \\ &= \mathbb{E}\left[\exp\left(-\imath\omega \log(\Delta t\sigma_0^2)\right)\right]\mathbb{E}\left[exp\left(-\imath\omega Y_M\right)\right] \\ &= \exp\left(-\imath\omega \log(\Delta t\sigma_0^2)\right)\hat{f}_{Y_M}(\omega). \end{aligned} \tag{5.13}$$

The computation of the characteristic function of $\hat{f}_{Y(T)}$ is reduced to the recovery of the characteristic function of Y_M, $\hat{f}_{Y_M}(\omega)$.

5.2.1. Characteristic function

Since Y_M is defined in a recursive way, so is $\hat{f}_{Y_M}$ which results in the following recursion relation,

$$\begin{aligned} \hat{f}_{Y_1}(\omega) &= \hat{f}_{R_{t_M}}(\omega) = \hat{f}_R(\omega) \\ \hat{f}_{Y_i}(\omega) &= \hat{f}_R(\omega)\hat{f}_{Z_{i-1}}(\omega). \end{aligned} \tag{5.14}$$

The characteristic function of the increment process, R, in (5.14) is analytically available, because it is in the class of Lévy processes. Unfortunately, this does not hold for the characteristic function $\hat{f}_{Z_{i-1}}$, which is, again, defined by

$$\hat{f}_{Z_{i-1}}(\omega) = \int_{\mathbb{R}} \big(\exp(x)+1\big)^{-\imath\omega} f_{Y_{i-1}}(x)\,dx. \tag{5.15}$$

Up to this point, the method to obtain the characteristic function coincided with the paper by Leitao et al. In that work the probability density function of Y_i, f_{Y_i} in (5.15), was computed by its Fourier cosine expansion. Instead we will approximate the density function f_{Y_i} by a Shannon wavelet approximation. Recall the SWIFT density approximation $f^*_{Y_i}$ in (3.15), which will be applied to (5.15),

$$f_{Y_{i-1}} \approx f^*_{Y_{i-1}}(x) = \sum_{k=k_1}^{k_2} c_{m,k}^{i-1}\phi_{m,k}. \tag{5.16}$$

Substituting (5.16) in (5.15) we obtain

$$\begin{aligned}\hat{f}_{Z_{i-1}}(\omega) &= \int_{\mathbb{R}} \left(\exp(x)+1\right)^{-\imath\omega} f_{Y_{i-1}}(x)dx \\ &\approx \int_{\mathbb{R}} \left(\exp(x)+1\right)^{-\imath\omega} f^*_{Y_{i-1}}(x)dx \\ &= \sum_{k=k_1}^{k_2} c^{i-1}_{m,k} 2^{\frac{m}{2}} \int_{\mathbb{R}} \left(e^x+1\right)^{-\imath\omega} \operatorname{sinc}\left(2^m x-k\right)dx,\end{aligned} \tag{5.17}$$

which yields

$$\begin{aligned}\hat{f}_{Y_i}(\omega) &= \hat{f}_R(\omega)\hat{f}_{Z_{i-1}}(\omega) \\ &\approx \hat{f}_R(\omega) \sum_{k=k_1}^{k_2} c^{i-1}_{m,k} 2^{\frac{m}{2}} \int_{\mathbb{R}} \left(e^x+1\right)^{-\imath\omega} \operatorname{sinc}\left(2^m x-k\right)dx,\end{aligned} \tag{5.18}$$

where ω is defined by the vector $\omega_n = \frac{2n+1}{2^N}\pi 2^m$ and the density coefficients are defined as in equation (3.26),

$$c^i_{m,k} \approx \frac{2^{\frac{m}{2}}}{2^{N-1}} \sum_{n=0}^{2^N-1} \text{Real}\left\{\hat{f}_{Y_i}\left(\frac{2n+1}{2^N}\pi 2^m\right) e^{\frac{\imath k\pi(2n+1)}{2^N}}\right\}. \tag{5.19}$$

Since the recursion relation is necessary to compute the density coefficients $c^i_{m,k}$, for $i = 1,\ldots,M$, we recall Algorithm 4.1. This algorithm will be used to compute the density coefficients and to decrease the CPU time required for this recursive procedure, a matrix-vector multiplication will be used. Let us recall the matrix defined in equation (4.48) and the resulting matrix-vector multiplication

$$\begin{aligned}M_{(n,k)} &= 2^{\frac{m}{2}} \int_{\mathbb{R}} \left(e^x+1\right)^{-\imath\omega_n} \operatorname{sinc}\left(2^m x-k\right)dx, \\ \hat{f}_{Z_i}(\omega_n) &\approx M_{(n,k)} c^i_{m,k}.\end{aligned}$$

Again the computation of the matrix $M_{(n,k)}$ by numerical integration is rather expensive and in [15] the integration is done by the Clenshaw-Curtis quadrature rule. However, in this case (equivalent to the arithmetic Asian option valuation) we can apply Theorem 4.1, and for the computation of the matrix $M_{(n,k)}$ we refer the reader to Section 4.6. With these results we can conclude the computation of the characteristic function $\hat{f}_{Y(T)}$ of the discrete analogue of the integrated variance $Y(T)$.

5.3. Simulation of $\int_0^T \sigma^2(s)ds|\sigma(T)$

In this section the copula approach to simulate $Y(T)|\sigma(T)$ will be explained. A copula is used to describe the dependency between random variables. In our case we deal with two random variables, $\log Y(T)$ and $\sigma(T)$. By Sklar's Theorem [21] we know that a multivariate joint distribution can be written in terms of univariate marginal distribution functions together with a copula that describes the dependence between the random variables. Copulas are divided into three categories:

- *Fundamental copulas*: represent perfect positive dependence, perfect negative dependence or independence,
- *Implicit copulas*: copulas based on multivariate distribution without a closed form expression, for example the Gaussian copula and the Student-t copula,

- *Archimedean copulas*: copulas with closed form expressions, for example the Gumbel copula, Frank copula and Clayton copula.

According to Sklar's Theorem, for every univariate marginal distribution of the random variables $\int_0^T \sigma^2(s)ds$ and $\sigma(T)$ there exists a copula $C_{\mathscr{R}}(\cdot,\cdot)$, such that

$$F_{\int_0^T \sigma^2(s)ds,\sigma(T)}(x_1,x_2) = C_{\mathscr{R}}\left(F_{\int_0^T \sigma^2(s)ds}(x_1), F_{\sigma(T)}(x_2)\right), \tag{5.20}$$

where $\mathscr{R}$ is the covariance matrix between the two random variables $\int_0^T \sigma^2(s)ds$ and $\sigma(T)$. In order to apply a copula model, we thus need to approximate the correlation between the random variables. In this case we use the Pearson correlation coefficient, since it is directly applicable to the Gaussian copula and there exists a relation between the Pearson correlation and Kendall's τ, which is applicable to the Gumbel copula.

Definition 5.2. The **Pearson correlation coefficient** of two random variables X and Y, $\rho_{X,Y}$, is given by

$$\rho_{X,Y} = \text{corr}[X,Y] = \frac{\text{cov}[X,Y]}{\sigma(X)\cdot\sigma(Y)} = \frac{\text{cov}[X,Y]}{\sqrt{\text{var}[X]\cdot\text{var}[Y]}}. \tag{5.21}$$

In this section, we aim to compute the multivariate joint distribution of $Y(T) = \log\int_0^T \sigma^2(s)ds$ and $\log\sigma(T)$, therefore we wish the compute the following Pearson correlation coefficient

$$\rho_{Y(T),\log\sigma(T)} = \frac{\text{cov}\left[\log\int_0^T \sigma^2(s)ds, \log\sigma(T)\right]}{\sqrt{\text{var}\left[\log\int_0^T \sigma^2(s)ds\right]\cdot\text{var}\left[\log\sigma(T)\right]}}. \tag{5.22}$$

Leitao et al. [15] propose an elegant approximation of the Pearson correlation coefficient. In the following section we will step-by-step derive the same approximation as in [15].

First of all, by Jensen's inequality we know that

$$Y(T) = \log\int_0^T \sigma^2(s)ds \geq \int_0^T \log\sigma^2(s)ds. \tag{5.23}$$

We can use the following approximation,

$$Y(T) = \log\int_0^T \sigma^2(s)ds \approx \int_0^T \log\sigma^2(s)ds = 2\int_0^T \log\sigma(s)ds. \tag{5.24}$$

If we substitute this approximation in equation (5.22), we obtain

$$\begin{aligned}\rho_{Y(T),\log\sigma(T)} &= \frac{\operatorname{cov}\left[\log\int_0^T \sigma^2(s)ds, \log\sigma(T)\right]}{\sqrt{\operatorname{var}\left[\log\int_0^T \sigma^2(s)ds\right]\cdot\operatorname{var}\left[\log\sigma(T)\right]}} \\ &\approx \frac{\operatorname{cov}\left[2\int_0^T \log\sigma(s)ds, \log\sigma(T)\right]}{\sqrt{\operatorname{var}\left[2\int_0^T \log\sigma(s)ds\right]\cdot\operatorname{var}\left[\log\sigma(T)\right]}} \\ &= \frac{\operatorname{cov}\left[\int_0^T \log\sigma(s)ds, \log\sigma(T)\right]}{\sqrt{\operatorname{var}\left[\int_0^T \log\sigma(s)ds\right]\cdot\operatorname{var}\left[\log\sigma(T)\right]}}\end{aligned} \tag{5.25}$$

Since the distribution of $\log\sigma(T)$ is available in closed form, (5.3), we know $\operatorname{var}\left[\log\sigma(T)\right]$. What remains are two other quantities in equation (5.25). First, we will derive the covariance. From the definition of covariance it follows that

$$\operatorname{cov}\left[\int_0^T \log\sigma(s)ds, \log\sigma(T)\right] = \mathbb{E}\left[\log\sigma(T)\cdot\left(\int_0^T \log\sigma(s)ds\right)\right] - \mathbb{E}\left[\int_0^T \log\sigma(s)ds\right]\cdot\mathbb{E}\left[\log\sigma(T)\right]. \tag{5.26}$$

Recall the distribution of $\log\sigma(T)$ in equation (5.3),

$$\sigma(T) = \sigma(0)\exp\left(\alpha W_\sigma(T) - \frac{1}{2}\alpha^2 T\right), \tag{5.27}$$

which gives

$$\log\sigma(T) = \log\sigma(0) + \alpha W_\sigma(T) - \frac{1}{2}\alpha^2 T. \tag{5.28}$$

The expectation of $\log\sigma(T)$ can be computed as

$$\begin{aligned}\mathbb{E}\left[\log\sigma(T)\right] &= \mathbb{E}\left[\log\sigma(0) + \alpha W_\sigma(T) - \frac{1}{2}\alpha^2 T\right] \\ &= \log\sigma(0) - \frac{1}{2}\alpha^2 T + \alpha\mathbb{E}\left[W_\sigma(T)\right] \\ &= \log\sigma(0) - \frac{1}{2}\alpha^2 T,\end{aligned} \tag{5.29}$$

and by Itô's formula we obtain for $Y(T) = \int_0^T \log\sigma(s)ds$,

$$\begin{aligned}\int_0^T \log\sigma(s)ds &= \int_0^T \left(\log\sigma(0) + \alpha W_\sigma(T) - \frac{1}{2}\alpha^2 T\right)ds \\ &= T\log\sigma(0) - \frac{1}{4}\alpha^2 T^2 + \alpha\int_0^T W(s)ds \\ &= T\log\sigma(0) - \frac{1}{4}\alpha^2 T^2 + \alpha\int_0^T (T-s)dW(s).\end{aligned} \tag{5.30}$$

By Itô's formula the last step in equation (5.30) follows. The expectation yields

$$\begin{aligned}\mathbb{E}\left[\int_0^T \log\sigma(s)ds\right] =& \mathbb{E}\left[T\log\sigma(0) - \frac{1}{4}\alpha^2T^2 + \alpha\int_0^T (T-s)dW(s)\right] \\ =& T\log\sigma(0) - \frac{1}{4}\alpha^2T^2.\end{aligned} \tag{5.31}$$

Finally, the expectation of the product of the two variables is computed,

$$\begin{aligned}&\mathbb{E}\left[\log\sigma(T)\left(\int_0^T \log\sigma(s)ds\right)\right] = \mathbb{E}\left[\left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2 + \alpha\int_0^T (T-s)dW(s)\right)\left(\log\sigma(0) + \alpha\hat{W}_\sigma(T) - \frac{1}{2}\alpha^2T\right)\right] \\ &= \left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)\left(\log\sigma(0) - \frac{1}{2}\alpha^2T\right) + \alpha\left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)\mathbb{E}[W(T)] \\ &+\alpha\left(\log\sigma(0) - \frac{1}{2}\alpha^2T\right)\mathbb{E}\left[\int_0^T (T-s)dW(s)\right] + \alpha^2\mathbb{E}\left[\int_0^T (T-s)dW(s)\int_0^T 1dW(s)\right] \\ &= \left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)\left(\log\sigma(0) - \frac{1}{2}\alpha^2T\right) + \alpha^2\mathbb{E}\left[\int_0^T (T-s)dW(s)\int_0^T 1dW(s)\right] \\ &= \left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)\left(\log\sigma(0) - \frac{1}{2}\alpha^2T\right) + \alpha^2\mathbb{E}\left[\int_0^T (T-s)ds\right] \\ &= \left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)\left(\log\sigma(0) - \frac{1}{2}\alpha^2T\right) + \frac{1}{2}\alpha^2T^2.\end{aligned} \tag{5.32}$$

The final step is to compute the variance of $\int_0^T \log\sigma(s)ds$, for which we compute

$$\begin{aligned}\mathbb{E}\left[\left(\int_0^T \log\sigma(s)ds\right)^2\right] =& \mathbb{E}\left[\left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2 + \alpha\int_0^T (T-s)dW(s)\right)^2\right] \\ =& \left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)^2 + \alpha\left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)\mathbb{E}\left[\int_0^T (T-s)dW(s)\right] \\ &+\alpha^2\mathbb{E}\left[\left(\int_0^T (T-s)dW(s)\right)^2\right] \\ =& \left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)^2 + \alpha\left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)\mathbb{E}\left[\int_0^T (T-s)dW(s)\right] \\ &+\alpha^2\mathbb{E}\left[\int_0^T (T-s)^2ds\right] \\ =& \left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)^2 + \frac{1}{3}\alpha^2T^3,\end{aligned} \tag{5.33}$$

so that the variance reads

$$\begin{aligned}\text{var}\left[\int_0^T \log\sigma(s)ds\right] =& \mathbb{E}\left[\left(\int_0^T \log\sigma(s)ds\right)^2\right] - \mathbb{E}\left[\int_0^T \log\sigma(s)ds\right]^2 \\ =& \left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)^2 + \frac{1}{3}\alpha^2T^3 - \left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)^2 \\ =& \frac{1}{3}\alpha^2T^3.\end{aligned} \tag{5.34}$$

At this point we have all the ingredients to compute the Pearson correlation coefficient:

$$\begin{aligned}\rho_{Y(T),\log\sigma(T)} &\approx \frac{\mathbb{E}\left[\log\sigma(T)\cdot\left(\int_0^T \log\sigma(s)ds\right)\right] - \mathbb{E}\left[\int_0^T \log\sigma(s)ds\right]\cdot\mathbb{E}\left[\log\sigma(T)\right]}{\sqrt{\mathrm{var}\left[\int_0^T \log\sigma(s)ds\right]\cdot\mathrm{var}\left[\log\sigma(T)\right]}} \\ &= \frac{\left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)\left(\log\sigma(0) - \frac{1}{2}\alpha^2T\right) + \frac{1}{2}\alpha^2T^2 - \left(T\log\sigma(0) - \frac{1}{4}\alpha^2T^2\right)\left(\log\sigma(0) - \frac{1}{2}\alpha^2T\right)}{\sqrt{\left(\frac{1}{3}\alpha^2T^3\right)\left(\alpha^2T\right)}} \\ &= \frac{\frac{1}{2}\alpha^2T^2}{\sqrt{\frac{1}{3}\alpha^4T^4}} \\ &= \frac{\sqrt{3}}{2}.\end{aligned} \tag{5.35}$$

5.3.1. Sampling $Y(T)|\sigma(T)$ by means of a bivariate copula

The copula approach to sample the conditional distribution of $Y(T)|\sigma(T)$ consists of the following steps:

1. Computation of the marginal distributions $F_{\log Y(T)}$ and $F_{\log\sigma(T)}$. The first marginal distribution can be attained as explained in Section 5.2, whereas the marginal distribution of $\log\sigma(T)$ can be obtained analytically (5.3).
2. Define the bivariate copula by means of the two marginal distributions and the covariance matrix $\mathscr{R}$.
$$\mathscr{R} = \begin{bmatrix} \mathrm{var}\left[\log\sigma(T)\right] & \rho_{Y(T),\log\sigma(T)} \\ \rho_{Y(T),\log\sigma(T)} & \mathrm{var}\left[\log Y(T)\right] \end{bmatrix} \approx \begin{bmatrix} \alpha^2T & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{3}\alpha^2T^3 \end{bmatrix} \tag{5.36}$$
3. With the previously defined copula, compute correlated uniform samples $U_{\log\sigma(T)}$ and $U_{\log Y(T)}$.
4. Invert the original marginal distributions $F_{\log Y(T)}$ and $F_{\log\sigma(T)}$ and take the exponential to obtain the samples of $Y(T)|\sigma(T)$.

As Leitao et al. have stated, the Gumbel copula is the most robust copula to use. However if short maturities are considered, the Gaussian copula would be a good choice. In [15] a goodness-of-fit (GOF) test was proposed to determine which copula was the most appropriate to describe the model. In the numerical tests, we will only consider the Gumbel and the Gaussian copulas to determine European option prices.

5.3.2. Simulation of the forward asset price $S(T)$ conditional on S_0

The final step in the one time-step Monte Carlo method is the simulation of the forward asset process $S(T)$ conditional on the asset process at $t = 0$. Chen et al. [8] proposes a forward asset simulation method based on the combination of moment matching and a direct inversion method. This method has also been applied by Leitao et al.in [15].

5.4. Numerical results

Having looked at all the components that are required to perform the already existing one time-step Monte Carlo simulation of the SABR model as well as the method adjusted for SWIFT-SIA, it is time

to compare the two models in an option pricing problem. We will do so by the valuation of European call options under the SABR dynamics. The payoff function of a European call option is given by

$$V(S(T),T) = \max\{S(T)-K,0\}, \tag{5.37}$$

with expiry T, asset price at expiry $S(T)$ and the strike K. The strike values K_i are chosen corresponding the work by Leitao et al., which represent both in-the-money, out-of-the-money and at-the-money European options,

$$\begin{aligned} K_i(T) &= S_0 \exp(0.1\,T\partial_i), \\ \partial_i &= -1.5, 0, 1.5. \end{aligned} \tag{5.38}$$

In this section we will compare three methods to obtain European options under SABR dynamics. The first method is the one time-step Monte Carlo simulation proposed by Leitao et al., the second method is a pure Monte Carlo simulation based on the Milstein discretization scheme and the third is the method proposed by Leitao et al, which has been adjusted for the application of SWIFT-SIA method.

Since each method involves a Monte Carlo simulation, for each path the resulting payoff is calculated, after which averaging over the number of paths leads to the option price. Table 5.1 lists the two test cases we will use to compare the results of the three methods. These particular test cases are considered since they cover different aspects of the SABR model. For *Set 1* the reference value is obtained with help of Antonov et al.[1], in which an analytical solution is obtained for the zero-correlation SABR model, $\rho = 0$. In the numerical experiments we can show convergence in m of the SWIFT method, since we can compute the absolute error of the SWIFT method compared to the analytical solution obtained by the method proposed in [1]. The second parameter set represents a SABR model with a high volatility of the volatility (vol-vol,i.e. α). Unfortunately, there is no analytical solution available for the second test case, therefore we will show convergence of the three methods to the reference value which has been obtained by the one time-step Monte Carlo method with the SWIFT-SIA method implemented for a high value of the scale of approximation ($m = 8$).

	S_0	σ_0	α	β	ρ	T
Set 1	1.0	0.5	0.4	0.7	0.0	2
Set 2	0.04	0.4	0.8	1.0	−0.5	2

Table 5.1: Data sets for the valuation of European option prices under SABR dynamics.

In the previous chapters we have shown that the SWIFT wavelet approximation converges in the scale of approximation m. This would mean that theoretically if we choose a higher scale of approximation, the accuracy of the option value should increase simultaneously. In order to show that our method converges, we choose a reference value by the SWIFT-SIA method with a high scale of approximation $m = 8$ and we will show convergence of the pure Monte Carlo method and the one time-step Monte Carlo method with SWIFT-SIA implemented towards this reference value.

Convergence in N_{MC}, Milstein discretization scheme

First let us show convergence of the Monte Carlo method. The theoretical convergence of a Monte Carlo method is of order $\mathcal{O}\left(\frac{1}{\sqrt{N_{MC}}}\right)$. From Table 5.2 we can confirm the convergence numerically. The 95% confidence interval of the option value converges by $\mathcal{O}\left(\frac{1}{\sqrt{N_{MC}}}\right)$.

	$V_{MC}(S_0, t_0)$	95% Confidence Interval	Interval size
$N_{MC} = 100$	0.021258956891641	$[1.397e-02, 2.854e-02]$	$1.456e-02$
$N_{MC} = 1e03$	0.016245321001685	$[1.468e-02, 1.780e-02]$	$3.123e-03$
$N_{MC} = 1e04$	0.015535134875164	$[1.507e-02, 1.599e-02]$	$9.247e-04$
$N_{MC} = 1e05$	0.015699727704500	$[1.554e-02, 1.585e-02]$	$3.032e-04$
$N_{MC} = 1e06$	0.015776985248290	$[1.572e-02, 1.582e-02]$	$9.636e-05$

Table 5.2: Convergence in N_{MC}, the mean option value $V_{MC}(S_0, t_0)$ and its 95% confidence interval.

5.4.1. Set 1: zero-correlation SABR model

The second numerical example is based on the zero-correlation method proposed by Antonov et al. For the expression of the analytical solution, the reader is refered to Appendix B.1. In this numerical example we will show that both the SWIFT-SIA method and the ASCOS method applied to the one time-step Monte Carlo method converge tot the exact reference value. The number of monitoring dates M is chosen to be fixed to $M = 1000$. In the original method proposed by Leitao et al., the value M did not have a significant impact on the CPU time, thus choosing a large value could only result in a more accurate option value (which resembles a continuously monitored European option value). The number of Monte Carlo paths is also fixed to $N_{MC} = 1e06$, to assure that the resulting option value is accurate.

- Parameter Set 1 (Antonov et al.) reference values:
  - $\partial_1 = -1.5$: $V_{ref} = 0.405630070518358$,
  - $\partial_3 = 1.5$: $V_{ref} = 0.172476427794756$.

Table 5.3 shows the accuracy of the option value of the one time-step Monte Carlo method with the SWIFT-SIA method implemented. The results show convergence in the scale of approximation m for the in-the-money and out-of-the-money option. At $m = 7$ we obtain an option value accurate up to basis points in 48.23 seconds. If we compare these results to the results of the original method (Table 5.4), we see that the SWIFT-SIA approach cannot compete with that method for the zero-correlation SABR model.

In the previous chapters where we have compared the SWIFT-SIA approach to the COS method, we have shown that the SWIFT-SIA method is comparable to the COS method. Unfortunately, this is not the case in the SABR zero-correlation model. The increase in m does result in highly accurate option values, but CPU time increases tremendously.

	CPU time	$\partial_1 = -1.5$	$\partial_3 = 1.5$
$m = 2$	1.48	0.1184	$1.562e-02$
$m = 3$	1.56	$5.855e-02$	$7.124e-02$
$m = 4$	2.44	$3.113e-03$	$2.826e-03$
$m = 5$	4.62	$1.135e-03$	$2.292e-04$
$m = 6$	13.42	$1.667e-04$	$9.037e-04$
$m = 7$	48.23	$1.036e-05$	$1.390e-03$

Table 5.3: Absolute error and the CPU time of the SWIFT-SIA method to price an European Option under the SABR model, parameter Set 1, reference values obtained by Antonov et al.

5.4.2. Set 2: high vol-vol SABR model

In this second numerical example there is no analytical solution available, but since we have shown convergence of the SWIFT-SIA method in the previous chapters, the reference value is obtained by the

	CPU time	$\partial_1 = -1.5$	$\partial_3 = 1.5$
$N = 50$	1.43	$8.206e-04$	$2.117e-03$
$N = 75$	1.61	$6.371e-04$	$2.062e-03$
$N = 100$	1.64	$3.748e-04$	$1.582e-03$
$N = 125$	1.69	$1.108e-03$	$2.685e-03$
$N = 150$	2.03	$5.692e-04$	$1.719e-03$

Table 5.4: Absolute error and the CPU time of the ASCOS method to price an European Option under the SABR model, parameter Set 1, reference values obtained by Antonov et al.

one time-step Monte Carlo method with the SWIFT-SIA method implemented. The reference values are computed with $m = 8, M = 500$ and $N_{MC} = 1e06$. Three strike prices will be considered, resulting in the valuation of an in-the-money, an out-of-the money and an at-the-money European option.

- Parameter Set 2, SWIFT-SIA method ($m = 8$, $M = 500$, $N_{MC} = 1e06$) reference values:
  - $\partial_1 = -1.5$: $V_{ref} = 0.014512272992185$,
  - $\partial_2 = 0$: $V_{ref} = 0.008411970261185$,
  - $\partial_3 = 1.5$: $V_{ref} = 0.003656664712094$.

With the results listed in Table 5.5 and Table 5.6 we can compare the SWIFT-SIA implementation to the original method in [15]. For both methods we obtain option values with basis point accuracy in less than a second ($m = 2$ in SWIFT-SIA and $N = 50$ in the original method). The increase in the number of terms in the Fourier cosine expansion in the original method does not result in more accurate option values, as shown in Table 5.6. However, the increase of the scale of approximation in the SWIFT-SIA implementation does improve the accuracy of the resulting option values. Unfortunately, the CPU time required is also affected by the increase of m.

Nonetheless, the SWIFT-SIA implementation is a competitive approach in the one time-step Monte Carlo method, since the SWIFT-SIA recovery of the integrated variance is a highly robust method, due to the single parameter m that needs to be chosen in advance.

	CPU time	$\partial_1 = -1.5$	$\partial_2 = 0$	$\partial_3 = 1.5$
$m = 2$	0.55	$6.1848e-05$	$7.656e-05$	$4.162e-05$
$m = 3$	0.91	$9.917e-05$	$1.766e-04$	$1.777e-04$
$m = 4$	1.92	$5.7049e-05$	$7.232e-05$	$7.149e-05$
$m = 5$	5.50	$3.035e-05$	$2.851e-05$	$3.069e-05$
$m = 6$	18.89	$6.3510e-06$	$8.129e-06$	$3.768e-06$
$m = 7$	77.29	$2.3089e-06$	$3.049e-06$	$1.564e-05$

Table 5.5: Accuracy and CPU time of the SWIFT-SIA method to price an European Option under the SABR model, parameter Set 2, reference values obtained by the SWIFT-SIA method with $m = 8$.

	CPU time	$\partial_1 = -1.5$	$\partial_2 = 0$	$\partial_3 = 1.5$
$N = 50$	0.58	$2.327e-06$	$1.050e-05$	$1.435e-05$
$N = 75$	0.62	$1.0619e-05$	$1.971e-05$	$2.339e-05$
$N = 100$	0.69	$1.593e-05$	$3.158e-05$	$4.724e-05$
$N = 125$	0.84	$1.304e-05$	$3.174e-05$	$4.228e-05$
$N = 150$	1.00	$9.960e-06$	$2.399e-05$	$3.842e-05$

Table 5.6: Absolute error and the CPU time of the ASCOS method to price an European Option under the SABR model, parameter Set 1, reference values obtained by the SWIFT-SIA method with $m = 8$.

6

Conclusion

In this thesis we extended the SWIFT method to two option pricing problems in which the sum of lognormals and sum of increments occur. The sum of lognormals has appeared in the valuation of arithmetic Asian options under GBM dynamics and in the one time-step Monte Carlo method under SABR dynamics. We referred to the sum of increments in the valuation of arithmetic Asian options under NIG and CGMY dynamics.

The SWIFT method relies on a wavelet approximation of the probability density function by means of its characteristic function. Another Fourier method which has a characteristic function approach is the COS method. The COS framework was applied to Asian option valuation, which resulted in the ASCOS method and was used as a reference value to which the comparison between SWIFT and ASCOS could be made. The COS framework has also been used to obtain reference values for the one time-step Monte Carlo method under SABR dynamics, where the COS method was applied to find an approximation to the distribution function of the SABR variance process.

The valuation of arithmetic Asian options was based on the Carverhill-Clewlow-Hodges factorization scheme, in which an iterative procedure was formed to evaluate the price of the underlying at maturity T. If the underlying is in the class of Lévy processes, the asset price increments are independent and identically distributed, such that the recovery of the characteristic function of the increment process resulted in the computation of the characteristic function of the sum of lognormals. The density coefficients were obtained recursively for each monitoring date $t_i = 0, \ldots, M_d$, which could be done efficiently by a matrix-vector product (with matrix $M_{(n,k)}$) and could benefit from the FFT transform.

We proposed two ways to compute the matrix $M_{(n,k)}$. First the matrix has been computed by a numerical integration technique: Clenshaw-Curtis quadrature rule. This approach has resulted in the SWIFT-CC method. The method has shown great results in terms of accuracy and convergence compared to the state-of-the-art ASCOS method, however the computational speed of the SWIFT-CC method was disappointing due to the complexity of the Clenshaw-Curtis quadrature rule.

The SWIFT-SIA (*SWIFT Sinc Integral Approximation*) method differed from the SWIFT-CC method in the way the matrix $M_{(n,k)}$ was recovered. Since the cardinal sine function was implemented as the wavelet basis, the method could benefit from the properties regarding bandlimited functions and the cardinal sine (sinc) function. The function g present in the matrix formulation has been numerically shown to be bandlimited and satisfied the assumption made in the theory from [24], such that the approximation stated in Theorem 4.1 of matrix $M_{(n,k)}$ converged exponentially in the wavelet scale m. The SWIFT-SIA method has shown great results in the valuation of arithmetic Asian options in terms of both the accuracy and the speed compared to the ASCOS method. Tests were carried out for three underlying Lévy processes: the GBM process, the NIG process and the CGMY process. Al three of them admitted highly satisfactory results for both short and long-maturity options.

Therefore we will draw the conclusion that the SWIFT-SIA method is a highly efficient method for the valuation of geometric and arithmetic Asian option under Lévy price processes. Furthermore, the SWIFT-SIA method appears more robust compared to the ASCOS method, since the only parameter to fix in advance is the scale of approximation m. Options with long maturities can be evaluated very efficiently by means of the SWIFT method and there is no need to apply the put-call parity to price arithmetic Asian call options with long-maturity.

The second option pricing problem in which a sum of lognormals occurred is the valuation of European options under the SABR model. In the method called *the one time-step Monte Carlo simulation of the SABR model* [15] the recovery of the distribution of the time-integrated variance in the SABR model showed similarities with the ASCOS method. The density of the integrated variance was approximated by the COS method, but in our modifications to the method the SWIFT-SIA method was applied to the computation of the density of the integrated variance. The modification of the existing method with the SWIFT approach, led to satisfactory results in terms of accuracy of the method. We have shown that the the SWIFT-SIA approach has led to highly accurate option values. The SWIFT-SIA method applied to the SABR model needed approximately as much time as the original method to compute the option price at the same level of accuracy and was more robust compared to the COS approach, since the parameter m is the only parameter needed for the density approximation.

6.1. Future Work

- In this thesis we did not compute the Greeks of the resulting option values, this is interesting and could be a starting point for further research.
- The thesis provided two applications of the SWIFT method in the presence of a sum of lognormals. The sum of lognormals occur in different settings, for example in the recovery of the density of the discounted compound Poisson process. Appendix C gives an idea of the problem in which SWIFT could be applied to recover the density.
- The valuation of European-style Asian options have been studied in this thesis, but different types of Asian options could be interesting topics for further research. The SWIFT-SIA method could be extended to the valuation of conditional Asian options or the class of American-style Asian options.
- The one time-step Monte Carlo method has recently been extended to a multiple time-step Monte Carlo method (see [16] for details). The extension to multiple time-steps results in more accurate long maturity option values. Since the SWIFT-SIA method has shown to result in a highly competitive method, it might be interesting to apply the SWIFT-SIA method in each time-step in the method discussed in [16].

A

Analytic Pricing Formula for the Geometric Asian Option under GBM Dynamics

Let us define the discrete geometric average stock price, G, as

$$G := \frac{1}{M_d} \sum_{i=1}^{M_d} \log(S(t_i)), \tag{A.1}$$

with $M_d \in \mathbb{N}$ monitoring dates of size $\Delta t = \frac{T}{M_d}$. Furthermore under GBM dynamics, the stock process is defined by

$$S(t_i) = S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right) t_i + \sigma W(t_i,)\right) \tag{A.2}$$

with W a standard Brownian Motion. Then G can be written as,

$$\begin{aligned}
G &= \frac{1}{M-d} \sum_{i=1}^{M_d} \log(S(t_i)) \\
&= \frac{1}{M_d} \sum_{i=1}^{M_d} \log\left(S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right) t_i + \sigma W(t_i)\right)\right) \\
&= \log(S_0) + \left(r - \frac{1}{2}\sigma^2\right) \frac{\Delta t}{M_d} \sum_{i=1}^{M_d} i + \frac{\sigma\sqrt{\Delta t}}{M_d} \sum_{i=1}^{M_d} \sum_{j=1}^{i} Z_j \\
&= \log(S_0) + \left(r - \frac{1}{2}\sigma^2\right) \frac{\Delta t}{M_d} \frac{M_d(M_d+1)}{2} + \frac{\sigma\sqrt{\Delta t}}{M_d} \sum_{i=1}^{M_d} \sum_{j=1}^{i} Z_j \\
&= \log(S_0) + \left(r - \frac{1}{2}\sigma^2\right) T \frac{(M_d+1)}{2M_d} + \frac{\sigma\sqrt{\Delta t}}{M_d} \sum_{i=1}^{M_d} \sum_{j=1}^{i} Z_j,
\end{aligned} \tag{A.3}$$

with $Z_j \sim \mathcal{N}(0,1)$. The double summation (A) can be written as

$$\sum_{i=1}^{M_d}\sum_{j=1}^{i} Z_j = M_d Z_1 + (M_d-1)Z_2 + \ldots + Z_{M_d}, \tag{A.4}$$

such that we can find the mean of the variable G,

$$\begin{aligned}\mathbb{E}[G] =& \mathbb{E}\left[\log(S_0) + \left(r-\frac{1}{2}\sigma^2\right)T\frac{(M_d+1)}{2M_d} + \frac{\sigma\sqrt{\Delta t}}{M_d}\sum_{i=1}^{M_d}\sum_{j=1}^{i} Z_j\right] \\ =& \log(S_0) + \left(r-\frac{1}{2}\sigma^2\right)T\frac{(M_d+1)}{2M_d},\end{aligned} \tag{A.5}$$

and the variance is defined by

$$\begin{aligned}\text{var}(G) =& \mathbb{E}\left[G^2\right] - \mathbb{E}[G]^2 \\ =& \mathbb{E}\left[\left(\log(S_0) + \left(r-\frac{1}{2}\sigma^2\right)T\frac{(M_d+1)}{2M_d} + \frac{\sigma\sqrt{\Delta t}}{M_d}\sum_{i=1}^{M_d}\sum_{j=1}^{i} Z_j\right)^2\right] \\ & -\mathbb{E}\left[\log(S_0) + \left(r-\frac{1}{2}\sigma^2\right)T\frac{(M_d+1)}{2M_d} + \frac{\sigma\sqrt{\Delta t}}{M_d}\sum_{i=1}^{M_d}\sum_{j=1}^{i} Z_j\right]^2 \\ =& \mathbb{E}\left[\left(\frac{\sigma\sqrt{\Delta t}}{M_d}\sum_{i=1}^{M_d}\sum_{j=1}^{i} Z_j\right)^2\right] \\ =& \mathbb{E}\left[\frac{\sigma^2\Delta t}{M_d^2}\left(M_d Z_1 + (M_d-1)Z_2 + \ldots + Z_{M_d}\right)^2\right] \\ =& \frac{\sigma^2\Delta t}{M_d^2}\left(M_d^2 + (M_d-1)^2 + \ldots + 1^2\right) \\ =& \sigma^2 T\frac{(M_d+1)(2M_d+1)}{6M_d^2}.\end{aligned} \tag{A.6}$$

As a result G can be written as

$$G = S_0\exp\left(\left(r-\frac{1}{2}\sigma^2\right)T\frac{(M_d+1)}{2M_d} + \sigma\sqrt{T}\sqrt{\frac{(M_d+1)(2M_d+1)}{6M_d^2}}Z\right), \tag{A.7}$$

where $Z \sim \mathcal{N}(0,1)$. Thus G follows a lognormal distribution,

$$\log G \sim \mathcal{N}\left(\log(S_0) + \left(r-\frac{1}{2}\sigma^2\right)T\frac{(M_d+1)}{2M_d}, \sqrt{\sigma^2 T\frac{(M_d+1)(2M_d+1)}{6M_d^2}}\right)$$

A.1. Adjustment to the Black-Scholes Pricing formula

By the well-known Black-Scholes formula the price European options under GBM dynamics, given $S(t_0) = S_0$ and t_0, is defined by

$$\begin{aligned} V(S_0,t_0) &= S_0\Phi(d_1) - Ke^{-r(T-t_0)}\Phi(d_2), \\ d_1 &= \frac{\log\left(\frac{S_0}{K}\right) + \left(r+\frac{1}{2}\sigma^2\right)(T-t_0)}{\sigma\sqrt{T-t_0}} \\ d_2 &= d_1 - \sigma\sqrt{T-t_0}, \end{aligned} \tag{A.8}$$

where $\Phi(\cdot)$ defines the standard normal cumulative distribution function.

This formula can be adjusted to price a geometric Asian option, since the geometric average follows a lognormal distribution. Let us define the mean and the variance of G by

$$\begin{aligned} \mu_G &= \log(S_0) + \left(r - \frac{1}{2}\sigma^2\right)T\frac{(M_d+1)}{2M_d} \\ \sigma_G &= \sqrt{\sigma^2 T\frac{(M_d+1)(2M_d+1)}{6M_d^2}} \end{aligned} \tag{A.9}$$

Then the pricing formula for a geometric Asian option under GBM dynamics is given by

$$\begin{aligned} C_G(S_0,t_0) &= S_0 e^{(\mu_G - r)(T-t_0)\Phi(d_1^G)} - Ke^{-r(T-t_0)}\Phi(d_2^G), \\ d_1^G &= \frac{\log\left(\frac{S_0}{K}\right)}{\sigma_G\sqrt{T-t_0}} + \left(\frac{\mu_G}{\sigma_G} + \frac{\sigma_G}{2}\right)\sqrt{T-t_0} \\ d_2^G &= d_1^G - \sigma_G\sqrt{T-t_0}. \end{aligned} \tag{A.10}$$

B

Reference values computation of European options under SABR dynamics

B.1. Set 1: SABR spreads its wings

The price of an European Call option under the SABR dynamics with zero correlated Brownian motions ($\rho = 0$) is given by

$$C(t,K) - [S_0 - K]^+ = \frac{2}{\pi}\sqrt{KS_0}\left\{\int_{s_-}^{s_+} \frac{\sin\left(\eta\phi(s)\right)}{\sinh s} G\left(t\alpha^2, s\right) ds + \sin(\eta\pi)\int_{s_+}^{\infty} \frac{e^{-\eta\psi(s)}}{\sinh s} G(t\alpha^2, s) ds\right\}, \quad \text{(B.1)}$$

with $G(t,s)$ defined as:

$$G(t,s) = 2\sqrt{2}\frac{e^{-\frac{t}{8}}}{t\sqrt{2\pi t}}\int_s^{\infty} u e^{-\frac{u^2}{2t}} \sqrt{\cosh u - \cosh s}\, du \quad \text{(B.2)}$$

$$\begin{aligned} \eta &= \left|\frac{1}{2\left(\beta - 1\right)}\right| \\ \phi(s) &= 2\arctan\sqrt{\frac{\sinh^2 s - \sinh^2 s_-}{\sinh^2 s_+ - \sinh^2 s}} \\ \psi(s) &= 2\operatorname{arctanh}\sqrt{\frac{\sinh^2 s - \sinh^2 s_+}{\sinh^2 s - \sinh^2 s_-}} \end{aligned} \quad \text{(B.3)}$$

$$\begin{aligned} s_- &= \operatorname{arcsinh}\left(\frac{\alpha\left|q - q_0\right|}{\sigma_0}\right) \\ s_+ &= \operatorname{arcsinh}\left(\frac{\alpha\left(q + q_0\right)}{\sigma_0}\right) \end{aligned} \quad \text{(B.4)}$$

with q and q_0 defined as

$$q = \frac{K^{1-\beta}}{1-\beta} \text{ and } q_0 = \frac{S_0^{1-\beta}}{1-\beta}.$$

B.2. Set 2: Monte Carlo method to price European options under the SABR model

Another way to compute European Option prices is by a pure Monte Carlo simulation. Monte Carlo simulation thanks his name to the famous casino in Monaco. A Monte Carlo simulation is an algorithm in which repeated random sampling leads to an broad spectrum of possible outcomes of a variable. Averaging over the set of random outcomes results in an estimate for that variable. In order to price an European option with an underlying asset price process, the goal is to simulate N_{MC} sample paths of the price process, with start value S_0, such that the estimated price of the asset at maturity, $S(T)$, is the average of all the N_{MC} possible outcomes.

The SABR model is the underlying price process in this thesis, with the system of stochastic differential equations defined in (5.1) and (5.2), i.e.,

$$\begin{aligned} \mathrm{d}S(t) &= \sigma(t)S^{\beta}(t)\mathrm{d}W_S(t), \\ \mathrm{d}\sigma(t) &= \alpha\sigma(t)\mathrm{d}W_\sigma(t), \\ S(0) = S_0 \exp(rT) &\text{ and } \sigma(0) = \sigma_0. \end{aligned} \tag{B.5}$$

Discretization of the price and volatility paths is done by the Milstein and Log-Euler discretization schemes. Let us define Z_1 and Z_2 two correlated standard normal random variables, with $\mathbb{E}[Z_1 Z_2] = \rho dt$, then the Milstein scheme is defined by,

$$\begin{aligned} S_\Delta^{\text{Milstein}} &= S_0 + \sigma_0 S_0^{\beta} Z_1 \sqrt{\Delta} + \frac{1}{2}\beta\sigma_0^2 S_0^{2\beta-1}\left(Z_1^2\Delta - \Delta\right), \\ \sigma_\Delta^{\text{Milstein}} &= \sigma_0 \exp\left(-\frac{1}{2}\alpha^2\Delta + \alpha Z_2\sqrt{\Delta}\right) \end{aligned} \tag{B.6}$$

The Log-Euler discretization is defined by,

$$\begin{aligned} S_\Delta^{\text{Log-Euler}} &= S_0 \exp\left(-\frac{1}{2}\sigma_0^2 S_0^{2\beta-1}\Delta + \sigma_0 S_0^{\beta-1} Z_1\sqrt{\Delta}\right) \\ \sigma_\Delta^{\text{Log-Euler}} &= \sigma_0 \exp\left(-\frac{1}{2}\alpha^2\Delta + \alpha Z_2\sqrt{\Delta}\right). \end{aligned} \tag{B.7}$$

The step-size Δ is defined by $\Delta = \frac{T}{M_d}$, with M_d the number of monitoring dates of the price process.

C

Compound Poisson Process with Discounting

As shown in this work, the SWIFT method is a suitable approach for the computation of the probability density function of a sum of lognormals. The application of SWIFT to Asian option valuation and the SABR model, resulted in more accurate and fast option prices. Since the Poisson process is a Lévy process, the question arose if the SWIFT method could be a good way to recover the probability density function of the compound Poisson process. This appendix gives an idea how the SWIFT method could be applied to the approximation of the probability density function of such a process. First of all, let us define the compound Poisson process.

C.1. Problem definition

Let us define the random variable $Z(t)$ by

$$\begin{aligned} Z(t) &= \sum_{k=1}^{N(t)} B(T_k) \exp\left(\int_{T_k}^{t} r(s)ds\right) \\ &= \sum_{k=1}^{N(t)} B(T_k) \frac{M(t)}{M(T_k)} \end{aligned} \tag{C.1}$$

with

- $\{\tau_k = T_k - T_{k-1} : k \in \mathbb{N}\}$ is a sequence of the inter-arrival times between the claims T_k and T_{k-1}.
- $\{T_k : k \in \mathbb{N}\}$ is a sequence of the arrival time of the k-th claim and thus are random variables with $T_0 = 0$.
- $\{N(t) : t \geq 0\}$ is a Poisson process, non-homogeneous and corresponds to the number of claims present in $[t_0, t]$.
- $\{B(T_k) : k \in \mathbb{N}\}$ is a sequence of iid random variables independent of $N(t)$ and $M(t)$ and represents the amount of the claim.
- $\{M(t) : t \geq 0\}$ represents the money-savings account at time $t \geq 0$, such that

$$\frac{M(t)}{M(T_k)} = \exp\left(\int_{T_k}^{t} r(s)ds\right).$$

- $\{r(s) : s \geq 0\}$ is the underlying interest rate process, in our case we consider the Ho-Lee interest rate model:

$$dr(s) = \theta(s)ds + \sigma dW(s),$$

with constant volatility $\sigma > 0$ and deterministic $\theta(\cdot)$.

The money-savings account $M(t)$ is a log-normal process, thus the random variable $Z(t)$ is a sum of lognormals, which gives the opportunity to use the same approach for recovering the distribution function as we did in the valuation of arithmetic Asian options, Chapter 4.

The aim is to find a way to recover the probability density function of the random variable $Z(t)$ in an efficient way by the SWIFT method. Since the random variable is a sum of lognormals, it is possible to define a recursion relation and approximate the density function by an algorithm which avoids an expensive quadrature rule (which coincides with the approximation of the probability density function of the arithmetic Asian option under a Lévy asset price process).

Let us define the characteristic function as following and let us apply the tower property by conditioning on the number of claims, $N(t) = n$, that occurred in $[t_0, t]$. We obtain

$$\begin{aligned}
\hat{f}_{Z(t)}(\omega) =& \mathbb{E}\left[e^{\mathrm{i}\omega Z(t)}\right] \\
=& \mathbb{E}\left[\mathbb{E}\left[e^{\mathrm{i}\omega Z(t)}|N(t) = n\right]\right] \\
=& \mathbb{E}\left[\mathbb{E}\left[e^{\mathrm{i}\omega \sum_{k=1}^{N(t)} B(T_k)\frac{M(t)}{M(T_k)}}|N(t) = n\right]\right] \\
=& \sum_{n\geq 0} \mathscr{P}(N(t) = n)\mathbb{E}\left[e^{\mathrm{i}\omega \sum_{k=1}^{N(t)} B(T_k)\frac{M(t)}{M(T_k)}}|N(t) = n\right] \\
=& \sum_{n\geq 0} \frac{e^{-\lambda t}(\lambda t)^n}{n!}\mathbb{E}\left[e^{\mathrm{i}\omega \sum_{k=1}^{N(t)} B(T_k)\frac{M(t)}{M(T_k)}}|N(t) = n\right].
\end{aligned} \tag{C.2}$$

Let us now assume that $\forall k : B(T_k) = B \in \mathbb{R}_{>0}$, then if we define $X(T_k) = \frac{M(t)}{M(T_k)}$ it follows that

$$\begin{aligned}
Z(t) =& \sum_{k=1}^{N(t)} B(T_k)\exp\left(\int_{T_k}^{t} r(s)ds\right) \\
=& B\sum_{k=1}^{N(t)} \frac{M(t)}{M(T_k)} \\
=& B\sum_{k=1}^{N(t)} X(T_k).
\end{aligned} \tag{C.3}$$

Since it is a sum of $N(t)$ lognormals $X(T_k)$ we can apply the Carverhill-Clewlow-Hodges factorization. Let us define $R_k = \log\left(\frac{X(T_k)}{X(T_{k-1})}\right)$, such that $Z(t)$ can be written as

$$\begin{aligned}
Z(t) =& BX(T_0)\left(1 + \frac{X(T_1)}{X(T_0)}\left(1 + \frac{X(T_2)}{X(T_1)}\left(\ldots\frac{X(T_{N(t)-1})}{X(T_{N(t)-2})}\left(1 + \frac{X(T_{N(t)})}{X(T_{N(t)-1})}\right)\right)\right)\right) \\
=& BX(T_0)\left(1 + e^{R_1}\left(1 + e^{R_2}\left(\ldots e^{R_{N(t)-1}}\left(1 + e^{R_{N(t)}}\right)\right)\right)\right).
\end{aligned} \tag{C.4}$$

With this relation we can define the recursion scheme $\{Y_k : k \in \mathbb{N}\}$, i.e.,

$$\begin{aligned} Y_1 &= R_{N(t)} \\ Y_k &= R_{N(t)+1-k} + \log\left(1+\exp\left(Y_{k-1}\right)\right) \\ &= R_{N(t)+1-k} + H_{k-1}. \end{aligned} \tag{C.5}$$

with $H_k = \log\left(1+\exp\left(Y_k\right)\right)$. The returns of the money-savings account are independent and identically distributed, with the Ho-Lee process as underlying interest rate process, thus we can compute the characteristic function of $Y_{N(t)}$ for all $k = 2,\ldots,N(t)$

$$\begin{aligned} \hat{f}_{Y_1}(\omega) &= \hat{f}_{R_1}(\omega) \\ \hat{f}_{Y_k}(\omega) &= \hat{f}_{R_{N(t)+1-k}}(\omega)\cdot\hat{f}_{H_{k-1}}(\omega) \\ &= \hat{f}_{R_1}(\omega)\cdot\hat{f}_{H_{k-1}}(\omega). \end{aligned} \tag{C.6}$$

Thus, if we have recovered the characteristic function of $Y_{N(t)}$, we can compute the density coefficients by the SWIFT method and approximate the probability density function by SWIFT. In the next section we will define the dynamics of the Ho-Lee interest rate process.

C.2. Ho-Lee interest rate process

The Ho-Lee interest rate process reads

$$dr(t) = \theta(t)dt + \sigma dW(t), \tag{C.7}$$

with $W(t)$ a standard Brownian Motion, $\sigma > 0$ and $\theta(t)$ a deterministic function. We have

$$\begin{aligned} \int_0^t dr(u) &= \int_0^t \theta(u)du + \sigma\int_0^t dW(u) \\ &\Leftrightarrow \\ r(t) - r(0) &= \int_0^t \theta(u)du + \sigma\int_0^t dW(u). \end{aligned} \tag{C.8}$$

Let us compute the expectation and the variance of this interest rate process. The expectation is given by

$$\begin{aligned} \mathbb{E}\left[r(t)\right] &= \mathbb{E}\left[r(0) + \int_0^t \theta(u)du + \sigma\int_0^t dW(u)\right] \\ &= r(0) + \int_0^t \theta(u)du, \end{aligned} \tag{C.9}$$

and the variance is given by,

$$\begin{aligned}\mathrm{Var}(r(t)) =& \mathbb{E}[r(t) - \mathbb{E}[r(t)]]^2 \\ =& \mathbb{E}\left[r(0) + \int_0^t \theta(u)du + \sigma\int_0^t dW(u) - \left(r(0) + \int_0^t \theta(u)du\right)\right]^2 \\ =& \mathbb{E}\left[\sigma\int_0^t dW(u)\right]^2 \\ =& \sigma^2\int_0^t du \\ =& \sigma^2 t.\end{aligned} \tag{C.10}$$

Since the interest rate is contained in the discount factor, we will determine the expectation and the variance of the integrated Ho-Lee process $\int_t^T r(s)ds$.

$$\begin{aligned}\int_t^T r(s)ds =& \int_t^T \left(r(0) + \int_0^s \theta(u)du + \sigma\int_0^s dW(u)\right)ds \\ =& \int_t^T r(0)ds + \int_t^T\int_0^s \theta(u)duds + \int_t^T \sigma\int_0^s dW(u)ds \\ =& r(0)(T-t) + \int_0^t\int_t^T \theta(u)dsdu + \int_t^T\int_u^T \theta(u)dsdu + \int_0^t\int_t^T \sigma dsdW(u) + \int_t^T\int_u^T \sigma dsdW(u) \\ =& r(0)(T-t) + \int_0^t \theta(u)(T-t)du + \int_t^T \theta(u)(T-u)du + \int_0^t \sigma(T-t)dW(u) + \int_t^T \sigma(T-u)dW(u) \\ =& (T-t)\left(r(0) + \int_0^t \theta(u)du + \sigma\int_0^t dW(u)\right) + \int_t^T \theta(u)(T-u)du + \int_t^T \sigma(T-u)dW(u) \\ =& (T-t)\,r(t) + \int_t^T \theta(u)(T-u)du + \int_t^T \sigma(T-u)dW(u)\end{aligned} \tag{C.11}$$

Since the discount factor is defined as $\exp\left(-\int_t^T r(s)ds\right)$, we will compute the expectation and the variance of $-\int_t^T r(s)ds$. It follows that

$$\begin{aligned}\mathbb{E}\left[-\int_t^T r(s)ds\right] =& \mathbb{E}\left[-(T-t)\,r(t) - \int_t^T \theta(u)(T-u)du - \int_t^T \sigma(T-u)dW(u)\right] \\ =& -(T-t)r(t) - \int_t^T \theta(u)(T-u)du\end{aligned} \tag{C.12}$$

$$\begin{aligned}\mathrm{Var}\left(-\int_t^T r(s)ds\right) =& \mathbb{E}\left[-\int_t^T r(s)ds - \mathbb{E}\left[-\int_t^T r(s)ds\right]\right]^2 \\ =& \mathbb{E}\left[-\left((T-t)r(t) + \int_t^T \theta(u)(T-u)du + \int_t^T \sigma(T-u)dW(u)\right) - \left(-(T-t)r(t) - \int_t^T \theta(u)(T-t)du\right)\right]^2 \\ =& \mathbb{E}\left[-\int_t^T \sigma(T-u)dW(u)\right]^2 \\ =& \sigma^2\int_t^T (T-u)^2 du \\ =& \frac{\sigma^2}{3}(T-t)^3\end{aligned} \tag{C.13}$$

Bibliography

[1] Alexander Antonov, Michael Konikov, and Michael Spector. SABR spreads its wings. *Risk*, 26(8): 58, 2013.

[2] Ole E Barndorff-Nielsen. Normal inverse Gaussian distributions and stochastic volatility modelling. *Scandinavian Journal of statistics*, 24(1):1–13, 1997.

[3] Norman C Beaulieu, Adnan A Abu-Dayya, and Peter J McLane. Estimating the distribution of a sum of independent lognormal random variables. *IEEE Transactions on Communications*, 43 (12):2869, 1995.

[4] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[5] Peter Carr and Dilip Madan. Option valuation using the fast Fourier transform. *Journal of Computational Finance*, 2(4):61–73, 1999.

[6] Peter Carr, Hélyette Geman, Dilip B Madan, and Marc Yor. The fine structure of asset returns: An empirical investigation. *The Journal of Business*, 75(2):305–332, 2002.

[7] Andrew Carverhill and Les Clewlow. Flexible convolution. *Risk*, 3(4):25–29, 1990.

[8] Bin Chen, Cornelis W Oosterlee, and Hans Van Der Weide. Efficient unbiased simulation scheme for the SABR stochastic volatility model. *Preprint, http://ta. twi. tudelft. nl/mf/users/oosterle/oosterlee/SABRMC. pdf*, 2011.

[9] Gemma Colldeforns-Papiol and Luis Ortiz-Gracia. Computation of market risk measures with stochastic liquidity horizon. *Preprint*, 2016.

[10] Ingrid Daubechies. *Ten lectures on wavelets*. SIAM, 1992.

[11] Ernst Eberlein and Antonis Papapantoleon. Equivalence of floating and fixed strike Asian and lookback options. *Stochastic Processes and their Applications*, 115(1):31 – 40, 2005.

[12] Fang Fang and Cornelis W Oosterlee. A novel pricing method for European options based on Fourier cosine series expansions. *SIAM Journal on Scientific Computing*, 31(2):826–848, 2008.

[13] William B Gearhart and HS Schultz. The function sin (x) x. *The College Mathematics Journal*, 2 (2):90–99, 1990.

[14] Patrick Hagan, Andrew Lesniewski, and Diana Woodward. Probability distribution in the SABR model of stochastic volatility. In *Large deviations and asymptotic methods in finance*, pages 1–35. Springer, 2015.

[15] Alvaro Leitao, Lech A Grzelak, and Cornelis W Oosterlee. On a one time-step SABR simulation approach: Application to European options. *Submitted to Applied Mathematics and Computation*, 2016.

[16] Alvaro Leitao, Lech A Grzelak, and Cornelis W Oosterlee. On an efficient multiple time step Monte Carlo simulation of the SABR model. *Quantitative Finance*, pages 1–17, 2017.

[17] Stefanus C Maree, Luis Ortiz-Gracia, and Cornelis W Oosterlee. Pricing early-exercise and discrete barrier options by Shannon wavelet expansions. *Numerische Mathematik*, 136(4):1035–1070, 2017.

[18] Luis Ortiz-Gracia. *Haar wavelets-based methods for credit risk portfolio modeling*. PhD thesis, Universitat Politècnica de Catalunya, 2011.

[19] Luis Ortiz-Gracia and Cornelis W Oosterlee. Robust pricing of European options with wavelets and the characteristic function. *SIAM Journal on Scientific Computing*, 35(5):B1055–B1084, 2013.

[20] Luis Ortiz-Gracia and Cornelis W Oosterlee. A highly efficient Shannon wavelet inverse Fourier technique for pricing European options. *SIAM Journal on Scientific Computing*, 38(1):B118–B143, 2016.

[21] Ludger Rüschendorf. On the distributional transform, Sklar's theorem, and the empirical copula process. *Journal of Statistical Planning and Inference*, 139(11):3921–3927, 2009.

[22] Ken-iti Sato. *Lévy processes and infinitely divisible distributions*. Cambridge University Press, 1999.

[23] Ben S Slimane. Bounds on the distribution of a sum of independent lognormal random variables. *IEEE Transactions on Communications*, 49(6):975–978, 2001.

[24] Frank Stenger. *Handbook of Sinc numerical methods*. CRC Press, 2016.

[25] Bowen Zhang and Cornelis W Oosterlee. Efficient pricing of European-style Asian options under exponential Lévy processes based on Fourier cosine expansions. *SIAM Journal on Financial Mathematics*, 4(1):399–426, 2013.