

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Master in Advanced Mathematics and Mathematical Engineering
Master's thesis

**Extension Of The SWIFT
Option Pricing Scheme
For European Options Calibration
Under Heston Stochastic Volatility Model**
Eudald Romo Grau

Supervised by Luis Ortiz-Gracia

June, 2020

This master's thesis would not have been possible without the support of a number of amazing individuals and organizations.

I wish to express my deepest gratitude to Luis Ortiz-Gracia for introducing me to the SWIFT method and for providing guidance and support when I needed it.

I would also like to extend my gratitude to Xanadu Trading Limited for initiating me in high-frequency derivatives training. In particular, to Alex Alvarez Ruiz and Florian Burkart for their role in my initial training and for sharing their knowledge through the several mathematical modelling projects I have worked on at Xanadu.

Many thanks to Universitat Politecnica de Catalunya, and all the professors I studied under through my bachelor and master studies for providing me with the motivation and knowledge that ultimately lead to this project.

Special thanks to Centre de Formació Interdisciplinària Superior (CFIS) for believing in me and providing guidance and financial support through all my bachelor experience.

I gratefully thank my closest for their support. Their role in helping me get to this point cannot be overestimated.

Finally, my deep gratitude to my family and partner, whose love, guidance, and patience never wavered and from whom I keep learning every day. Without them I would not be writing these lines.

Abstract

A Heston model calibration technique is presented for European options under the Heston model. The novel Shannon Wavelets Inverse Fourier Technique (SWIFT) is extended for European option price calibration (previously it was used only for pricing European, Asian, barrier, and Bermudan options). This method has different expressions and speed-up techniques, adequate to different set-ups. These are discussed and new expressions and properties are presented for the gradient computation and option calibration.

The Heston characteristic function expression recently proposed by [9] is used in the SWIFT implementation due to its analytic gradient and its continuity properties.

The time performance, robustness, and convergence under set-ups representative of real markets is studied for different implementations of the SWIFT technique and compared with the option calibration scheme presented by [9]

The SWIFT implementations are coded in C++ and uploaded to a public GitHub repository. The library implements several of the different SWIFT expressions for GBM and Heston European options.

Keywords

Options Trading, Heston, SWIFT, Calibration, Inverse Fourier Option Valuation

Contents

1	Motivation	4
2	Option valuation	5
2.1	No-arbitrage assumption and risk-neutral markets	6
2.2	European option valuation	8
3	Underlying asset models	10
3.1	Lévy processes	10
3.2	Black-Scholes-Merton model	11
3.3	Implied volatility smile and surface	11
3.4	The Heston model	13
3.4.1	Calibration difficulties	15
3.4.2	Alternative expressions	16
4	European option valuation through SWIFT	18
4.1	Multi-resolution analysis	18
4.2	Shannon wavelets	19
4.3	SWIFT	20
4.3.1	Sinc integral and density coefficients approximation	22
4.3.2	Payoff coefficients approximation	24
4.3.3	Fast computation of the density and payoff coefficients	25
4.3.4	Alternative method for multiple strikes	26
4.3.5	Option price gradient	28
4.3.6	SWIFT parameter choice	29
5	Option calibration	30

5.1	Objective function	30
5.2	Gradient descent step	31
6	Numerical results	33
6.1	Stress Tests	34
6.2	Speed Tests	36
6.3	Realistic Convergence Tests	37
7	Conclusions	38
7.1	Further Research	39
A	Code	43
B	Complimentary Formulas	43
B.1	Cui Gradient Complimentary Formulas	43
B.2	Strike and Maturity Test Sets	43

1. Motivation

Option pricing has an important role in contract trading, both as a form of derivative trading in itself, as well as a way to hedge other stock or derivative portfolios.

Pricing these options can have a broad range of difficulty, mainly due to two factors:

1. **Contract type:** These options can take a variety of forms, from the relatively simple European options to more complex American, Bermudan, or Asian options (see Chapter 2).
2. **Underlying Model:** The price of the derivative depends on the future price of its underlying. The evolution of this price can be modeled using simple models like Black & Scholes (BS), which has an analytic closed form for European option pricing but fails to capture some key properties of real markets (see Section 3.2), or using more complex models like the Heston model (Section 3.4) or Lévy models other than BS (Section 3.1).

Closed option price formulas only exist for European options under the BS model. Either using more complex option contracts or models generally means one must fall back to numerical schemes to obtain option prices, which typically means less robustness to numerical errors and more computational time.

Some of these pricing techniques rely on the fact that most of the well-known models have a characteristic function with a closed form and use that characteristic function to price options through risk-neutral option valuation techniques (see Section 2.1 for an introduction to risk-neutral markets and Section 2.2 for its applications toward European option valuation). Recently one of these characteristic-function-based pricing methods, called Shannon Wavelets Inverse Fourier Technique (SWIFT) [33] has been shown to successfully price European, Asian [1], barrier, and Bermudan options [29] with models with closed-form characteristic function.

One does not have the choice of always pricing options as European contracts, as the type of contract is decided by the Contract Exchanges that trade derivative contracts, and even if one were to trade only European options, the study of more complex underlying asset models is relevant, as BS fails to capture well-known properties of real markets, which is reflected in the implied volatility surface (see Section 3.3), which option traders will take into account if they use the BS model. This motivated the development of other models, like the stochastic volatility Heston model, which will be the main model considered for option price calibration in this master's thesis.

Calibration refers to finding values for the underlying model intrinsic parameters that minimize the difference between real market option prices and the prices obtained through the model being calibrated (see Chapter 5 for the specific optimization problem considered). As the real market option prices are constantly changing it is of special relevance to find calibration methods that are highly time-efficient to allow real-time model updates.

In particular, the Heston model calibration is a difficult optimization problem to solve efficiently, as it is a multidimensional optimization problem with a fairly unknown structure (it is not known whether it is a convex problem or not nor whether it generally has a single solution or not) which until recently did not have simple analytic expressions for its characteristic function gradient [9] (the implications of not

knowing the problem structure and of not having simple analytic expressions for the gradient are discussed in Section 3.4.1.

The goal of this master's thesis is to extend the SWIFT method to calibrate complex option underlying models. In particular, the base European option pricing SWIFT method will be extended for European options under the Heston model expressions provided by [9]. Then the performance of the resulting option calibration scheme will be compared against the calibration technique presented also in [9]. This calibration technique could then be extended to Barrier and Bermudan contracts following Maree et al. [29] but this will not be implemented here.

In order to extend SWIFT for calibration, an expression of the SWIFT price gradient is provided (see Section 4.3.5) which can benefit from several performance speed-up techniques. Some of these techniques were already presented in [33] and [29] and some are new to this master's thesis and are particularly useful for the SWIFT calibration to be competitive with other highly efficient calibration techniques. To ensure this performance competitiveness several speed and robustness tests are presented (see Chapter 6).

The structure of this master's thesis will be as follows: Chapter 2 will introduce the basic concepts of option valuation, which will require an underlying asset model. Several of these models will be discussed in Chapter 3, and Chapter 4 will show how these models can be used to price European options through the SWIFT method. Then, Chapter 5 will discuss how to calibrate an underlying asset model given an option pricing method. Finally, Chapter 6 will present several numerical tests to compare the proposed option calibration method with other highly competitive methods and the test results will be discussed. Chapter 7 will summarize the most important results of this work and will propose some future research paths.

2. Option valuation

Definition 2.1. A European option is a derivative contract on an underlying asset that fixes a transaction price (called strike price), an expiration time (also called maturity), and an underlying asset quantity and gives the holder of the contract the right (but not the obligation) of buying (call option) or selling (put option) the specified quantity of the underlying at the strike price on expiration.

In the next sections we will use the following nomenclature for option parameters: t will refer to the time at which the option is being valued and T will be the expiration time. Given a time-dependent variable V , V_t will refer to the value of that variable at time t (respectively with T). S, K will refer to the underlying and the strike prices respectively.

Slight variations of these terms can give rise to other families of options by facilitating the exercising of the contracts, like the Bermudan and American options (which allow to exercise your buying/selling right at specified instants of time or at any point prior to expiry, respectively).

Other groups of option families are obtained by changing the definition of the option payoff, that is, the amount of money (be it in cash or in the value of the underlying obtained) the option gives on expiration.

Definition 2.2. Payoff of a European call option:

$$V_{EC}(S_T, 0) = (S_T - K)\mathbf{1}_{S_T \geq K} = \begin{cases} S_T - K & \text{if } S_T > K \\ 0 & \text{if } S_T \leq K \end{cases} \quad (1)$$

So, on maturity, a European call option is only profitable if it allows the holder of the contract to buy the underlying asset at a cheaper price than the current asset spot price. In most models any option has a certain positive value if it is not expired, as the underlying price can always theoretically move enough so that a currently non-profitable option ends up with a positive payoff on expiry. Thus, one can always divide the price of an option with positive time to expiry $\tau := T - t$ as $V(S, \tau) = V_1(S) + V_2(S, \tau)$, where $V_1(S) := V(S, 0)$ and is called the implicit value of the option, and $V_2(S, \tau)$ is its time value. The implicit value of an option is typically used to classify options as follows:

Definition 2.3. A European option at time t is called:

- Out of the money (OTM): If it has no implied value and exercising it would result in loss (that is $K > S_t$ for call options).
- At the money (ATM): If its implied value is 0 and exercising it would result in breaking even (that is $K = S_t$).
- In the money (ITM): If it has positive implied value (that is, $K < S_t$ for call options).

Some examples of other families of functions that have different payoffs are binary options (only have two possible payoffs dependent on K and S_T) and Asian options (the payoff depends on the mean price of the underlying since the start of the contract and until maturity).

In order to price an option that expires at some point in the future, one usually first models the temporal evolution of the price of its underlying asset by using some stochastic process. This model will typically be expressed in terms of parameters that will be calibrated to fit the underlying price evolution of each product one wants to value.

When pricing options, a risk-neutrality assumption is usually taken [22]. In order to describe and justify this assumption, let us first look at the no-arbitrage assumption.

2.1 No-arbitrage assumption and risk-neutral markets

Consider an option call contract C with strike K and current price V_t . Let the underlying associated with it have an initial price S_t and maturity τ , and model the underlying price evolution as a 1-step discrete process that follows a Bernoulli distribution, so that the final price of the underlying is:

Definition 2.4.

$$S_T^{Ber} := \begin{cases} u \cdot S_t & \text{with probability } p_u \\ d \cdot S_t & \text{with probability } p_d := 1 - p_u \end{cases} \quad (2)$$

where u and d are parameters following $u > 1$ and $0 \leq d < 1$ and they define the relative movement of the underlying asset when it moves up or down respectively and p_u is the probability that the underlying asset will move up.

Once the underlying moves either up or down and reaches maturity, the price of the option will be determined by the the payoff formula $V(S_T, 0)$ given in Definition 1. The reasoning that will be followed in this section will apply to any type of option contract, so let us simply call these option values V_u and V_d respectively.

Then, one can hold a portfolio with a long position in the underlying and a short position in the options with ratio $n : 1$, so that the value of the portfolio at time T is $nS_T - V(S_T, 0)$, and choose n so that the portfolio has the same value whether the underlying goes up or down by setting:

$$nS_t u - V_u = nS_t d - V_d \rightarrow n = \frac{V_u - V_d}{S_t(u - d)} \quad (3)$$

This portfolio is what is called a risk-free portfolio, as its final value is perfectly defined and, if one assumes there are no arbitrage opportunities, its rate of returns must match the risk-free rate (which will be denoted r). Here, r rate denotes a theoretic rate at which one can earn money with almost no risk (in US markets is usually assumed to be the US-treasury bonds rate of return for the length of the considered invesment), and an arbitrage opportunity is the possibility to invest without risk obtaining a rate of returns higher than r (this can happen, for example, when a share is traded at two different markets, and the offers are different enough between the markets that one can buy a share in one market and sell it immediately in the other for a profit).

It is easy to see that if the rate of returns of the portfolio is different than r , arbitrage opportunities will arise. Let the values of the portfolio at times t and T be P_t and P_T respectively:

- If $P_T > P_t e^{r\tau}$: Anyone can start holding the porfolio or increase the volume they hold until expiry.
- If $P_T < P_t e^{r\tau}$: Anyone who holds shares of the portfolio can sell them and invest the obtained cash without risk.

If no arbitrage opportunities exist, then it must hold $P_T = P_t e^{r\tau}$.

Hence, the no-arbitrage assumption provides a way to value the current price of an option through its expected payoff. One first needs to build a risk-less portfolio and compute its expected payoff under a certain underlying price evolution model. Then one can discount the risk-free rate during the maturity of the option contract to obtain the initial price of the portfolio and, from there, obtain the price of the option. That is:

Lemma 2.5. *The price of a 1-step option under a Bernouilli model is*

$$V(S_t, \tau) := e^{-r\tau} (p_u V_u + p_d V_d) \quad (4)$$

Proof. First, from the portfolio initial value one can obtain $V(S_t, \tau) := nS_t - P_t$, as the portfolio is risk-less, it earns the risk-free rate, so

$$V(S_t, \tau) = nS_t - e^{-r\tau} P_T = nS_t - e^{-r\tau} (nS_t u - V_u) = nS_t(1 - ue^{-r\tau}) + V_u e^{-r\tau} \quad (5)$$

Using the value of n for a risk-less portfolio, one obtains

$$V(S_t, \tau) = \frac{V_u - V_d}{S_t(u - d)} S_t(1 - ue^{-r\tau}) + V_u e^{-r\tau} \quad (6)$$

$$= \frac{V_u(1 - ue^{-r\tau}) - V_d(1 - ue^{-r\tau})}{(u - d)} + V_u e^{-r\tau} \quad (7)$$

$$= \frac{V_u(1 - de^{-r\tau}) + V_d(ue^{-r\tau} - 1)}{(u - d)} \quad (8)$$

$$= e^{-r\tau} (V_u P + V_d(1 - P)) \quad (9)$$

where

$$P = \frac{e^{r\tau} - d}{u - d} \quad (10)$$

□

Note that the price of the option is independent of p_u and p_d . This probability is already taken into account in the price of the underlying and is a key concept of the modelization of real markets as risk-neutral ones. On the other hand, equation 9 can be interpreted as the discounted expected payoff of the option, if its payoff were to follow a Bernouilli distribution with parameter P . This distribution is called the risk-neutral distribution of the option, and is the basis of risk-neutral option valuation.

The assumption of no-arbitrage is usually generalized when pricing options by the concept of risk-neutrality. A risk-neutral world has two main features that simplify the pricing of derivatives:

- The expected return on a stock (or any other investment) is the risk-free rate.
- The discount rate used for the expected payoff on an option is the risk-free rate.

In Chapter 12 of [22] there is an in-depth discussion of the no-arbitrage and risk-neutrality assumptions, as well as several justifications as to why a risk-neutrality assumption is useful to provide prices for real market options.

Remark 2.6. A well-known consequence of the risk-neutrality assumption is the put-call parity equation:

$$V_t^C - V_t^P = S_t - e^{-r\tau} \cdot K \quad (11)$$

where V_t^C and V_t^P are the prices of a European put and call contract (note that this equation is independent of the model used for the time evolution of the underlying). This provides a way to convert any European call pricing scheme presented in this master's thesis to European put ones and vice-versa. Refer to [22] for a proof of Equation 11.

2.2 European option valuation

The payoff of some option contracts, like Bermudan, Asian or American options, depends on the price of the underlying asset at several points in time, or even through all its life. European options payoff, on the

other hand, only depends on the final price of the underlying asset on the moment of expiration of the contract (S_T).

A general pricing formula for this kind of options uses the risk-neutrality assumption to compute the option price as the discounted expectation of the payoff at expiry. Then the pricing problem is equivalent to defining an stochastic process describing S_T conditioned to the initial value S_t . If one considers generic variables x and y that fully describe the stochastic variables S_t and S_T respectively, the general pricing formula becomes:

$$v(x, \tau) = e^{-r\tau} \mathbb{E}^{\mathbb{Q}}(v(y, 0)|x) = e^{-r\tau} \int_{\mathbb{R}} v(y, 0) f(y|x) dy \quad (12)$$

where v denotes the option value, r the risk-neutral interest rate, $\mathbb{E}^{\mathbb{Q}}$ the expectation under the risk-neutral assumption, and $f(y|x)$ is the probability density of y given x .

Any choice of x and y will allow computing the value of the option through a numerical integration scheme, but most of the underlying asset models used for option pricing have simple expressions in the log space, so a common choice when solving this expression is to define:

$$y = \ln\left(\frac{S_T}{K}\right) \quad (13)$$

$$x = \ln\left(\frac{S_t}{K}\right) \quad (14)$$

Given this maturity state variable choice, one can express the payoff of a European put option as

$$v_K(y, 0) := K \cdot (1 - e^y)^+ := K \cdot \max(1 - e^y, 0) \quad (15)$$

This expression has a lineal dependency with the strike of the option, so one can define the strike-free payoff:

$$g(y) := (1 - e^y)^+ = \frac{v_K(y, 0)}{K} \quad (16)$$

Remark 2.7. When solving Equation 12 using numerical methods, one should prefer calculating the price of a call option via the price of a put one with the same strike and apply the put-call parity formula to recover the original call price, as e^y can have arbitrarily large values inside the integration domain which can lead to errors due to floating point arithmetics [29].

There is a broad literature in methods to solve this integration problem for basic underlying asset models. Some of them, as BS, even have closed solutions (as will be seen in Section 3.2) but it is common that, for more complex models, $f(y|x)$ does not have a known expression.

In some cases, this can be circumvented by using numerical integration methods based on Fourier transforms, if there is a known expression for the characteristic function of $f(y|x)$

$$\hat{f}_y(u; x) = \int_{\mathbb{R}} f(y|x) e^{-iux} dy \quad (17)$$

This characteristic function is equivalent to the Fourier transform of the conditional density function $f(y|x)$, so one can apply the Fourier inversion formula:

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(u) e^{iux} du \quad (18)$$

to recover the original density function given its characteristic function.

This is the case of the Lévy model and the Heston model which will be presented in the next sections.

3. Underlying asset models

3.1 Lévy processes

A well-known model for driving the dynamics of the underlying asset is the geometric Brownian motion (GBM) model (known as the Black-Scholes-Merton [5, 30] model). It can be considered inside a more general concept called Lévy process [36].

Definition 3.1. A stochastic process $X = \{X_t : t \geq 0\}$ is considered a Lévy process if:

- $X_0 = 0$ almost surely.
- For any $0 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq \infty$, $X_{t_2} - X_{t_1}, X_{t_3} - X_{t_2}, \dots, X_{t_n} - X_{t_{n-1}}$ are independent.
- $\forall s < t$, $X_t - X_s$ is equal in distribution to X_{t-s}

A famous property of Lévy processes is the Lévy-Khintchine formula:

Lemma 3.2. Let $X = (X_t)_{t \geq 0}$ be a Lévy process. Then its characteristic function $\hat{f}_X(u)$ is:

$$\hat{f}_X(u)(t) := \mathbb{E} \left[e^{-iuX_t} \right] = e^{t \left(-aiu - \frac{\sigma^2 u^2}{2} + \int_{\{0\}^c} (e^{-iux} \mathbf{1}_{|x| < 1}) \Pi(dx) \right)} = e^{\psi_L(w)}$$

where $\psi_L(u)$ is called the characteristic Lévy exponent, and $a \in \mathbb{R}$, $\sigma \geq 0$, and Π is the Lévy measure of X , a σ -finite measure satisfying $\int_{\{0\}^c} (1 \wedge x^2) \Pi(dx) < \infty$

Note that any Lévy process can be completely characterized by the triplet (a, σ, Π) , which represent a linear drift, a Brownian motion, and a Lévy jump process respectively.

Lévy models in option pricing typically consider x and y as in expressions 14 and 13 and let $X_t := x$, $X_T := y$, and model the log returns evolution by a Lévy process. Then $\hat{f}(y|x) = \hat{f}(X_T|X_t)$ can be obtained through formula 3.2:

$$\hat{f}_y(u; x) := \mathbb{E} \left[e^{-iuX_T} \right] = \mathbb{E} \left[e^{-iu(X_t + X_{T-t})} \right] = e^{-iux} e^{-iu\mu_L T + T\psi_L(-u)} := e^{-iux} \hat{f}(u) := \hat{f}_{y-x}(u) \quad (19)$$

where $\mu_L := r - \psi_L(-i)$ is a drift correction term and $\psi_L(u)$ is the characteristic Lévy exponent of the underlying log-returns process.

3.2 Black-Scholes-Merton model

A particular case of Lévy processes is the widely used BS model, originally presented in [5] by using the following stochastic differential equation:

Definition 3.3. BS model price equation:

$$dS_t = \mu S_t + \sigma S_t dW_t \quad (20)$$

where dW_t is a Brownian motion process, σ is called the BS volatility, and μ describes the drift of the underlying log returns and is usually considered $\mu = r$ for dividend-free products.

This model assumes the underlying process S_t follows a log-normal distribution with standard deviation σ and, for European options, Equation 12 can be solved analytically, obtaining the well-known BS valuation formula:

Lemma 3.4. Black-Scholes-Merton valuation formula for European call options:

$$C_K(\sigma; S_t, \tau) = N(d_1)S_t - N(d_2)PV(K) \quad (21)$$

where

$$d_1 = \frac{1}{\sigma\sqrt{\tau}} \left[\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)\tau \right] \quad (22)$$

$$d_2 = d_1 - \sigma\sqrt{\tau} \quad (23)$$

$$PV(K) = Ke^{-r\tau} \quad (24)$$

The part of the characteristic function independent from the initial state $\hat{f}(u)$ can be obtained by setting $\psi_L(u) = -\frac{\sigma^2}{2}u^2$ in Equation 19 resulting in:

Definition 3.5. BS characteristic function:

$$\hat{f}_{BS}(u) = \exp\left(-iu\left(r - q - \frac{1}{2}\sigma^2\right)\tau - \frac{\sigma^2 u^2 \tau}{2}\right) \quad (25)$$

3.3 Implied volatility smile and surface

If the GBM process modeled accurately real market prices, the BS model could be calibrated using several strikes to obtain a single implied volatility value, and that value would be a good estimator of the standard deviation of the historical standard deviation of the log-normal distribution of the underlying returns (which is called the historical BS volatility). This volatility could then be used to price any option of a given strike and maturity.

In real markets, if one chooses a fixed expiry and calibrates a set of M different models using M different strikes $\{K_i\}$, the obtained implied volatilities $\{\sigma_i\}$ will probably differ. Further, if one estimates

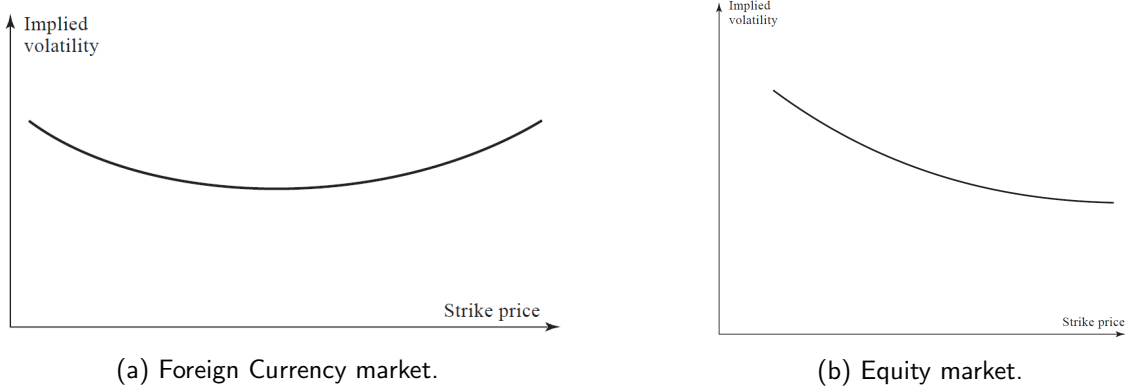


Figure 1: From [22]. Volatility smiles for typical markets.

the historical volatility σ_H from the historical log-returns, the obtained value may not coincide with any of the obtained implied volatilities.

If one were to plot the implied σ_i over K_i , one would obtain a result similar to the ones in Figure 1, but the exact shape would depend on the underlying (and it changes over time). This strike evolution is traditionally called volatility smile (though it can have different names depending on the shape it takes, like volatility skew for equity markets).

There are several justifications as to why such smiles exist. Some of them involve limitations in the GBM description of the underlying stock returns, but some appeared after significant events, like the volatility skew in equity option markets, which only appeared after the stock market crash of 1987 [22]. Their significance is better described with the concept of implied distribution.

If one assumes that the risk-neutrality and no-arbitrage concepts hold true, then the real market option prices should follow from specifying a distribution function $f(y|x)$ and solving Equation 12. One can then estimate the shape of the implied distribution $f_i(y|x)$ that would better match the option prices of a specific product and expiry (there are several techniques to do it, some described in [23, 34]). As can be seen in Figure 2, the upward trend in implied volatility for the deep ITM and OTM options in currency markets is translated into the implied distribution having fatter tails, and the upward trend for small strikes in equity markets translates into a distribution with higher stock market crash probabilities.

A similar effect happens when one fixes a strike price and computes its implied volatility over several expiries. The shape that results from plotting the implied volatility smile over time is called term structure and, when one plots the implied volatility over both strike price and time, one gets the so-called implied volatility surface.

The shape of the implied volatility surface holds certain information about the historical performances of the underlying. In particular, the distribution with fat tails implied by the currency market option prices describes the historical returns of the options underlying better than the GBM [22]. This shows the unsuitability of the GBM model and the need for better theoretic descriptions of the underlying returns.

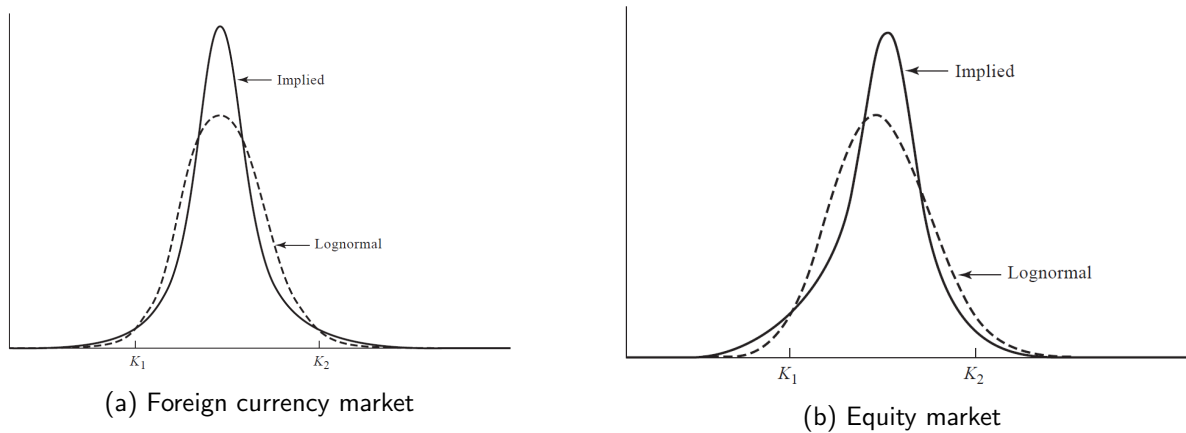


Figure 2: From [22]. GBM (Lognormal) and Implied distributions for typical markets.

3.4 The Heston model

The BS model presented in Section 3.2 fails to capture essential well-known properties of the real world market dynamics of the underlying return distributions, as its high kurtosis, its negative skew, the correlation between the underlying price and its volatility, etc. As well as the risk premium investors give deep ITM or OTM options, which result in the volatility discussed in Section 3.3.

To address that, several variations of this model have been proposed since it was introduced. Some of them, called stochastic volatility (SV) models, considered all the real world dynamics mentioned above and treated both the underlying price and its volatility as (potentially correlated) stochastic processes.

One of the first and most well-known SV models is the Heston model [21], defined by the following system of stochastic differential equations.

Definition 3.6. Heston model price-volatility equations:

$$\begin{aligned}
 dS_t &= \mu S_t dt + \sqrt{\nu_t} S_t dW_t^{(1)} \\
 d\nu_t &= \kappa(\bar{\nu} - \nu_t) dt + \sigma \sqrt{\nu_t} dW_t^{(2)} \\
 dW_t^{(1)} dW_t^{(2)} &= \rho dt
 \end{aligned}$$

where ν_t is the variance of the underlying asset price at time t . The parameters $\kappa, \bar{\nu}, \sigma, \rho$ are respectively called: mean-reversion rate, long-term variance, volatility of volatility, and correlation between the Brownian processes $W_t^{(1)}$ and $W_t^{(2)}$. Then, if one specifies the initial value of the variance ν_0 , the model is properly defined. From now on, $\theta := (\nu_0, \bar{\nu}, \rho, \kappa, \sigma)^T$ will refer to the vector of model parameters.

Several studies have shown the relations between the Heston model parameters and the shape of the implied volatility surface [8, 16, 18, 24] necessary to obtain the same prices with a BS model. These relations can be summarized as follows:

- ν_0 controls the position of the volatility surface.

- ρ controls its skewness.
- κ and σ control the convexity of the surface.
- $\kappa(\nu_0 - \bar{\nu})$ controls the term structure of implied volatility.

An expression of the price of a European option is presented in [21], and it is adapted here in terms of the state variables x and y defined in Section 2.2:

$$v(x, \tau) = e^{-r\tau} \mathbb{E}^{\mathbb{Q}}(v(y, 0)|x) \quad (26)$$

$$= e^{-r\tau} \mathbb{E}^{\mathbb{Q}}(K(e^y - 1)\mathbf{1}_{y \geq 0}) \quad (27)$$

$$= Ke^{-r\tau} (\mathbb{E}^{\mathbb{Q}}(e^y \mathbf{1}_{y \geq 0}) - \mathbb{E}^{\mathbb{Q}}(\mathbf{1}_{y \geq 0})) \quad (28)$$

$$= Ke^x e^{-q\tau} P_1(\theta; x, \tau) - Ke^{-r\tau} P_2(\theta; x, \tau) \quad (29)$$

where P_1 and P_2 are defined as:

$$P_1(\theta; x, \tau) = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left(\frac{e^{iux} \hat{f}(-u+i)}{iu \hat{f}(i)} \right) du \quad (30)$$

$$P_2(\theta; x, \tau) = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left(\frac{e^{iux} \hat{f}(-u)}{iu \hat{f}(-u)} \right) du \quad (31)$$

where $\hat{f}(u)$ is the initial state independent characteristic function of the process followed by the logarithm of price of the underlying asset (the existence of $\hat{f}(\theta; u, \tau)$ means that the Heston model characteristic function can easily be split in initial state dependent and independent terms, like the Lévy process). The expression of this characteristic function is:

$$\hat{f}(u) := \exp \left\{ iur\tau + \frac{\kappa \bar{\nu}}{\sigma^2} \left[(\xi + d)\tau - 2 \log \frac{1 - g_1 e^{d\tau}}{1 - g_1} \right] + \frac{\nu_0}{\sigma^2} (\xi + d) \frac{1 - e^{d\tau}}{1 - g_1 e^{d\tau}} \right\} \quad (32)$$

where

$$\xi := \kappa + \sigma \rho iu \quad (33) \quad d := \sqrt{\xi^2 + \sigma^2 (u^2 - iu)} \quad (34) \quad g_1 := \frac{\xi + d}{\xi - d} \quad (35)$$

Remark 3.7. As in previous sections, the dependence of the characteristic function with time and the model parameters is omitted for readability.

Substituting these values into Equation 29, one can obtain an analytic equation to obtain the price of European call options. Here, the expression given in [9] is adapted to the state variables x and y and provided in a more compact form:

Lemma 3.8. *Heston's pricing method.*

$$V^C(\theta; K, \tau) = K \left[\frac{1}{2} (e^x - e^{-r\tau}) + \frac{e^{-r\tau}}{\pi} \int_0^\infty \operatorname{Re} \left(\frac{\hat{f}(-u+i; x) - \hat{f}(u; x)}{iu} \right) du \right] \quad (36)$$

Proof. Directly adapting equation (9) from [9] to the state variables used in this article (the discount rate q will be assumed to be 0 throughout this work) yields:

$$V^C(\theta; K, \tau) = \frac{1}{2} (S_t - Ke^{-r\tau}) + \frac{e^{-r\tau}}{\pi} \left[S_t \int_0^\infty \operatorname{Re} \left(\frac{e^{iux}}{iu} \hat{f}(-u+i) \right) du - K \int_0^\infty \operatorname{Re} \left(\frac{e^{iux}}{iu} \hat{f}(-u) \right) du \right] \quad (37)$$

. Dividing and multiplying everything by K , and using $S_t/K = e^{\log(S_t/K)} = e^x$, one obtains

$$V^C(\theta; K, \tau) = K \frac{1}{2} (e^x - e^{-r\tau}) + K \frac{e^{-r\tau}}{\pi} \left[\int_0^\infty \operatorname{Re} \left(\frac{e^{i(u-i)x}}{iu} \hat{f}(-u+i) \right) du - \int_0^\infty \operatorname{Re} \left(\frac{e^{iux}}{iu} \hat{f}(u) \right) du \right] \quad (38)$$

Joining the two integrals, and using the Heston characteristic function property $\hat{f}(u; x) = e^{-iux} \hat{f}(u)$ one obtains the expression given in Equation 36 \square

From this expression, one can compute the analytic gradients of the option price in terms of the derivatives of the characteristic function. Then they can be used to optimize an appropriate objective function using gradient-descent based methods (See Chapter 5) but, as will be explained in the next section, the analytic derivatives have not been widely used traditionally to calibrate Heston models prior to [9] due to the complexity of the expressions obtained by derivating the available characteristic function expressions as in Equation 32 [9].

Remark 3.9. Sometimes the characteristic function is defined as $\int_{\mathbb{R}} f(y|x) e^{-iux} dy$ and this leads to a sign difference in the expressions of the characteristic function. One can see that there is a sign difference in all the u -dependent equations and expressions in [9].

3.4.1 Calibration difficulties

As opposed to simpler 1-dimensional models, Heston model calibrations is a multidimensional optimization problem with 5 degrees of freedom given by $\theta := (\nu_0, \bar{\nu}, \rho, \kappa, \sigma)^T$. Furthermore, the structure of this optimization problem is not known.

According to [9] no consensus exists among researchers regarding whether the objective function of this optimization problem is convex or not. Some results point to a non-convex function, as the calibration methods proposed in [4, 31] (which yielded different results for different initial points) and one must use long or short term approximations and rules to provide a sufficient initial guess. Recent research claims to provide methods that reach a unique solution independently of the initial point [17] which, according to that study, indicates some structure that, even if not necessarily convex, tends to lead an initial guess to a stationary result. There is also no consensus on whether the problem always has a single optimum. In particular, it is known that there exist dependencies between the parameters that yield to similar results.

For example, $\lim_{t \rightarrow \infty} \operatorname{Var}(\nu_t) = \frac{\sigma^2 \bar{\nu}}{\kappa}$ (where $\operatorname{Var}(\cdot)$ refers to the variance operator), so large values of κ and σ can provide a model that prices options similarly to one with proportionally smaller values of these

two parameters. The work by [9] defends that this yields to the objective function of the optimization problem being flat close to the optimum.

As said above, there's no guarantee that a gradient-based method converges to the global optimum of the model parameters, but even obtaining a local optimum has been traditionally difficult. A lot of literature uses numerical gradients[17] for these methods when trying to solve the Heston calibration problem (which are less accurate and more computationally consuming), because no simple analytic gradients were available and the ones obtained with symbolic algebra packages from the expressions of the characteristic function were intractable.

Prior to [9], the existing methods could be summarized as:

- **Heuristic based models:** Using the relationships stated in 3.4, some studies reduce the dimension of the optimization problem by assuming some values or relationships between the parameters from the observation of a specific volatility surface. For example [16] sets ν_0 to the short-term ATM implied variance obtained by using a BS model, a heuristic further justified by [4], where the linearity between ν_0 and the BS implied volatility was verified for short maturities (less than 2 months). Other heuristics used in the industry are $\kappa = \frac{2.75}{\tau}$ and setting $\bar{\nu}$ to the BS short term volatility [8].

These assumptions may restrict the optimization problem domain and exclude the optimum.

- **Stochastic methods:** They are usually used in combination with deterministic search methods, as the Nelder and Mead simplex method [26] and would avoid the pitfalls of the gradient-based methods if the optimization problem is not convex. Some examples are Wang-Landau (used in [4]) and differential evolution and particle swarm (used in [19]). These methods are too computationally expensive for real-time use as of now: [13] uses GPU computations to calibrate options using a SV model called SABR, and it took 421.72 seconds to calibrate 12 instruments with a tolerance of 10^{-2} using 2 NVIDIA Geforce GTX470 GPUs.

3.4.2 Alternative expressions

For long-term maturities (i.e. big values of τ), [25] shows that the characteristic function in Equation 32 has discontinuities as u increases, which can lead to numerical problems for many option valuation methods based on the integral expression in Equation 12. They show that these discontinuities arise because of a term in Equation 32 of the form $G^\alpha(u) := e^{\alpha \log G(u)}$ with $G(u) := \frac{1 - g_1 e^{d\tau}}{1 - g_1}$ and $\alpha := \frac{\kappa \bar{\nu}}{\sigma^2}$, for non-integer values of α . This is due to the spiral shape of $G(u)$ which produces a phase jump on $\log(G(u))$ each time $G(u)$ crosses the negative side of the real line.

[2] showed that the dicontinuity arises from taking the principal value of the square root in d (see Equation 34) and it can be avoided if the second value is used. In fact, it was proven that this alternative expression was continuous in the full parameters space [37].

Lemma 3.10.

$$\hat{f}(u) = \exp \left\{ -iur\tau + \frac{\kappa \bar{\nu}}{\sigma^2} \left[(\xi - d)\tau - 2 \log \frac{1 - g_2 e^{-d\tau}}{1 - g_2} \right] + \frac{\nu_0}{\sigma^2} (\xi - d) \frac{1 - e^{-d\tau}}{1 - g_2 e^{-d\tau}} \right\} \quad (39)$$

where $g_2 = \frac{1}{g_1}$.

A more compact version of the characteristic function was later derived in [11] from the moment generating function of the process. This expression also had the benefit of replacing the u -dependent power functions in Equations 32 and 39 by hyperbolic functions, which greatly simplified obtaining analytic expressions of the gradient of the characteristic function.

Lemma 3.11.

$$\hat{f}(u) = \exp\left(-iur\tau + \frac{\kappa\bar{\nu}\rho\tau iu}{\sigma} - A\right) B^{2\kappa\bar{\nu}/\sigma^2} \quad (40)$$

where

$$\begin{aligned} A &:= \frac{A_1}{A_2} \\ A_1 &:= (u^2 - iu) \sinh \frac{d\tau}{2} \\ A_2 &:= \frac{d}{\nu_0} \cosh \frac{d\tau}{2} + \frac{\xi}{\nu_0} \sinh \frac{d\tau}{2} \\ B &:= \frac{de^{\kappa\tau/2}}{\nu_0 A_2} \end{aligned} \quad (41)$$

This last expression also showed discontinuity problems, as the original characteristic function in Equation 32, but [9] provides an equivalent expression, continuous in the full parameters domain, and with an analytic expression for its gradient.

Theorem 3.12. Expression from [9] of the Heston characteristic function.

$$\hat{f}(u) = \exp\left(-iur\tau + \frac{\kappa\bar{\nu}\rho\tau iu}{\sigma} - A + \frac{2\kappa\bar{\nu}}{\sigma^2} D\right) \quad (42)$$

is an algebraically equivalent representation of Equation 32, continuous through all the parameters space, where

$$D := \log \frac{d}{\nu_0} + \frac{(\kappa - d)\tau}{2} - \log \left(\frac{d + \xi}{2\nu_0} + \frac{d - \xi}{2\nu_0} e^{-d\tau} \right) \quad (43)$$

Further, its gradient with respect to the Heston model parameters $\theta = (\nu_0, \bar{\nu}, \sigma, \kappa, \rho)^T$ is given by:

$$\nabla \hat{f}(u) = \mathbf{h}(u) \hat{f}(u) \quad (44)$$

where $\mathbf{h}(u) = [h_{\nu_0}(u), h_{\bar{\nu}}(u), h_{\sigma}(u), h_{\kappa}(u), h_{\rho}(u)]^T$ and:

$$h_{\nu_0}(u) = -\frac{A}{\nu_0} \quad (45)$$

$$h_{\bar{\nu}}(u) = \frac{2\kappa}{\sigma^2} D + \frac{\kappa\rho\tau iu}{\sigma} \quad (46)$$

$$h_{\sigma}(u) = -\frac{\partial A}{\partial \rho} + \frac{2\kappa\bar{\nu}}{\sigma^2 d} \left(\frac{\partial d}{\partial \rho} - \frac{d}{A_2} \frac{\partial A_2}{\partial \rho} \right) + \frac{\kappa\bar{\nu}\tau iu}{\sigma} \quad (47)$$

$$h_{\kappa}(u) = -\frac{1}{\sigma iu} \frac{\partial A}{\partial \rho} + \frac{2\bar{\nu}}{\sigma^2} D + \frac{2\kappa\bar{\nu}}{\sigma^2 B} \frac{\partial B}{\partial \kappa} + \frac{\bar{\nu}\rho\tau iu}{\sigma} \quad (48)$$

$$h_{\rho}(u) = -\frac{\partial A}{\partial \sigma} - \frac{4\kappa\bar{\nu}}{\sigma^3} D + \frac{2\kappa\bar{\nu}}{\sigma^2 d} \left(\frac{\partial d}{\partial \sigma} - \frac{d}{A_2} \frac{\partial A_2}{\partial \sigma} \right) - \frac{\kappa\bar{\nu}\rho\tau iu}{\sigma^2} \quad (49)$$

where the partial derivatives of A , A_2 , B , and d are given in [9] and can be seen in Appendix B.1

4. European option valuation through SWIFT

4.1 Multi-resolution analysis

Multi-resolution analysis (MRA) is a method that ultimately allows to express any function in $L^2(\mathbb{R})$ using a countable orthogonal family of functions. This family can then be truncated into a finite family and the original function can be orthogonally projected into the resulting subspace, obtaining an approximation with a certain level of resolution. Increasing the considered number of elements of the finite family will increase the resolution of the approximation, converging to a perfect representation when all the functions of the original family are used [28].

Given the space $L^2(\mathbb{R}) = \left\{ f : \int_{-\infty}^{\infty} |f(x)|^2 dx < \infty \right\}$, a MRA is defined as a family of nested successive approximation closed spaces:

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

Where the subspaces V_i are complete (they are not redundant and cover $L^2(\mathbb{R})$):

$$\overline{\bigcup_{i \in \mathbb{Z}} V_i} = L^2(\mathbb{R}), \text{ and } \bigcap_{i \in \mathbb{Z}} V_i = 0$$

they have self-similarity in scale (all spaces are geometric scalings of V_0 by powers of 2):

$$f(x) \in V_i \Leftrightarrow f(2x) \in V_{i+1}$$

they have self-similarity in time:

$$f(x) \in V_0 \Rightarrow f(x - k) \in V_0, \forall k \in \mathbb{Z}$$

Note that self-similarity in scale implies that the self-similarity in time translates to all spaces V_i as $f(x) \in V_i \rightarrow f(x - 2^i k) \in V_i$, and the integer shifts of a (or a finite group of) generator function ϕ form an orthogonal basis of V_0 .

In summary, we can define:

Definition 4.1. MRA: Consider $\phi \in L^2(\mathbb{R})$ a function that spans the family $\{\phi_{m,k}\} m, k \in \mathbb{Z}$ defined as the normalized scaled integer shifts of ϕ . That is, $\phi_{m,k} = 2^{m/2} \phi(2^m x - k)$, and let $V_m := \text{closure}_{L^2(\mathbb{R})} \langle \{\phi_{m,k}\}_{k \in \mathbb{Z}} \rangle$

Then, if ϕ and V_m fulfill the conditions above, we say that ϕ is the scaling function or father wavelet of the MRA $\{V_m\}$ (note that the previous definition properly defines $\{V_m\}$ as a sequence of nested subspaces and that ϕ provides an orthonormal basis for each of them).

One of the important implications of obtaining a father wavelet and its MRA is that another wavelet family can be obtained from it, which will be a basis of $L^2(\mathbb{R})$. In order to do that, let us consider the set of subspaces W_m such that $V_{m+1} = V_m \oplus W_m$. Then $L^2(\mathbb{R}) = \sum_m \oplus W_m$ and there exists a function $\psi \in W_0$ (called mother wavelet) that generates an orthonormal basis of $L^2(\mathbb{R})$ [10] by defining the wavelet functions:

$$\psi_{m,k} = 2^{m/2} \psi(2^m x - k)$$

. Note that each $\{\psi_{m,k}\}_{k \in \mathbb{Z}}$ gives an orthonormal basis of W_m , and $\{\psi_{m,k}\}_{m \in [-\infty, m-1], k \in \mathbb{Z}}$ is an orthonormal basis of V_m . So for any $m \in \mathbb{Z}$ we can define

Definition 4.2. Wavelet Projection: $P_m : L^2(\mathbb{R}) \rightarrow V_m$ is the projection from any function $f \in L^2(\mathbb{R})$ into V_m :

$$P_m f(x) = \sum_{j=-\infty}^{m-1} \sum_{k \in \mathbb{Z}} d_{j,k} \psi_{j,k}(x) = \sum_{k \in \mathbb{Z}} D_{m,k} \phi_{m,k}(x) \quad (50)$$

where $d_{j,k} = \langle f, \psi_{j,k} \rangle$, $D_{m,k} = \langle f, \phi_{m,k} \rangle$, $\langle f, g \rangle = \int_{\mathbb{R}} f(x) \overline{g(x)} dx$, and $\bar{\cdot}$ is the complex conjugate operator. Further, this projection converges in the L^2 norm as m tends to infinity [28].

4.2 Shannon wavelets

Claude Shannon introduced the usage of the cardinal sine function for information modeling [38]:

$$\text{sinc}(x) := \begin{cases} \frac{\sin(\pi x)}{\pi x} & \text{for } x \neq 0 \\ 1 & \text{for } x = 0 \end{cases} \quad (51)$$

This function serves as the father wavelet from which we obtain the Shannon Wavelet families $\phi_{m,k}$ and $\psi_{m,k}$:

$$\phi_{m,k}(x) = 2^{m/2} \text{sinc}(\pi(2^m x - k)), \quad k \in \mathbb{Z} \quad (52)$$

$$\psi_{m,k}(x) = 2^{m/2} \left(\text{sinc}(\pi(2^m x - k - 1/2)) - \frac{\sin(2\pi(2^m x - k - 1/2))}{\pi(2^m x - k - 1/2)} \right), \quad k \in \mathbb{Z} \quad (53)$$

One of the interesting properties of Shannon wavelets it that they have both a slow decay in time domain and a simple expression and sharp compact support in the frequency domain.

$$\hat{\phi}_{m,k}(w) = \frac{e^{-ik/2^m w}}{2^{m/2}} \text{rect}\left(\frac{w}{2^{m+1}\pi}\right), \quad k \in \mathbb{Z} \quad (54)$$

$$\hat{\psi}_{m,k}(w) = -\frac{e^{-i\frac{k+1/2}{2^m} w}}{2^{m/2}} \left(\text{rect}\left(\frac{w}{2^m \pi} - \frac{3}{2}\right) + \text{rect}\left(-\frac{w}{2^m \pi} - \frac{3}{2}\right) \right), \quad k \in \mathbb{Z} \quad (55)$$

When using Shannon Wavelets to approximate a function with a truncated wavelet expansion [29] shows a bound for the projection error into V_m , by using concepts of band-limited functions:

Definition 4.3. A function f is called band-limited if $\exists B \in \mathbb{R}^+$, with $B < \infty$ such that

$$f(x) = \frac{1}{2\pi} \int_{-B\pi}^{B\pi} \hat{f}(u) e^{iux} du \quad (56)$$

that is, the support of \hat{f} is contained in the interval $[-B, B]$. The parameter B is referred to as the bandwidth of f .

The nested subspaces of a Shannon MRA can be expressed in terms of band-limited functions because of the sinc Fourier transform rectangular shape, as stated in the following Lemma from [39]

Lemma 4.4. Consider an MRA generated from the Shannon scaling function as defined in Definition 4.1, then each subspace V_m corresponds to the space of functions $f \in L^2(\mathbb{R})$ with bandwidth $B \leq 2^m$

Combining this lemma with the L^2 convergence of the projections P_m of f into V_m yields the following corollary [29]:

Corollary 4.5. The orthogonal projection P_m of a Shannon MRA is equivalent to:

$$P_m f(x) = \frac{1}{2\pi} \int_{-2^m\pi}^{2^m\pi} \hat{f}(u) e^{iux} du \quad (57)$$

Which, in turn, can be used to derive the following bound to the error of the orthogonal projection.

Definition 4.6. Given $f \in L^2(\mathbb{R})$, let $H(\xi)$ be:

$$H(\xi) := \frac{1}{2\pi} \int_{|u|>\xi} |\hat{f}(u)| du \quad (58)$$

, the normalized mass of the two-side tails of \hat{f} defined by ξ .

Lemma 4.7. Let $\epsilon_m(x) := f(x) - P_m f(x)$ (the pointwise approximation error due to the projection of f into V_m). Then $|\epsilon_m(x)| \leq H(2^m\pi)$ [29]

4.3 SWIFT

The Shannon Wavelets Inverse Fourier Technique (SWIFT) is an option pricing method presented in [33]. It combines the Shannon Wavelets and MRA concepts with Equation 12 to provide a fast approximation to the value of a specific option $v(x, t)$.

The overall approximation can be summarized, as in [29], by a set of consecutive approximation steps to that equation, which are described below.

- **Wavelet Projection:** f is replaced by its Shannon wavelet projection at scale $m \in \mathbb{Z}$. Using Definition 4.2 this can be stated as:

$$f(y|x) \approx f_1(y|x) := P_m f(y|x) = \sum_{k \in \mathbb{Z}} D_{m,k}(x) \phi_{m,k}(y) \quad (59)$$

$$v(x, t) \approx v_1(x, t) = e^{-r\tau} \int_{\mathbb{R}} f_1(y|x)v(y, T)dy \quad (60)$$

with $D_{m,k}(x) := \langle f(\cdot|x), \phi_{m,k} \rangle$.

- **Series Truncation:** The set of values of k involved in the sum of Equation 59 is reduced to a finite interval $[k_1, k_2]$

$$f_1(y|x) \approx f_2(y|x) = \sum_{k=k_1}^{k_2} D_{m,k}(x)\phi_{m,k}(y) \quad (61)$$

$$v_1(x, t) \approx v_2(x, t) = e^{-r\tau} \int_{\mathbb{R}} f_2(y|x)v(y, T)dy \quad (62)$$

It is important to notice that the first approximation lets us express:

$$f(2^{-m}k|x) \approx f_1(2^{-m}k|x) = 2^{m/2}D_{m,k}(x) \quad (63)$$

which quickly justifies that, for any given x , the density coefficients vanish as $|k|$ increases, because f vanishes at the queues. It is also worth noting that increasing m will result in this mapping being less favorable. That is, for each k , $D_{m,k}$ will be bounded by a point closer to the center of the density function, potentially requiring to increase the interval $[k_1, k_2]$.

Remark 4.8. From this point onward a symmetric interval $[1 - \eta, \eta]$ will be considered both for convenience and for consistency with the code implementation.

- **Density Coefficients Approximation:** The integral required to compute $D_{m,k}$ is replaced by an approximation $D_{m,k}^*$ as discussed in Section 4.3.1.

$$f_2(y|x) \approx f_3(y|x) = \sum_{k=1-\eta}^{\eta} D_{m,k}^*(x)\phi_{m,k}(y) \quad (64)$$

$$\begin{aligned} v_2(x, t) \approx v_3(x, t) &= e^{-r\tau} \int_{\mathbb{R}} f_3(y|x)v(y, T)dy \\ &= e^{-r\tau} \sum_{k=1-\eta}^{\eta} D_{m,k}^*(x)V_{m,k}dy \end{aligned} \quad (65)$$

where

$$V_{m,k} := \int_{\mathbb{R}} \phi_{m,k}(y)v(y, T)dy \quad (66)$$

- **Payoff Coefficients Approximation:** The integral required to compute $V_{m,k}$ is approximated in an analogous way as the integral to compute the density coefficients.

$$v_3(x, t) \approx v_4(x, t) = e^{-r\tau} \sum_{k=1-\eta}^{\eta} D_{m,k}^*(x)V_{m,k}^* \quad (67)$$

For European put options the payoff is given by Equation 15, and the last equation can be rearranged using the strike-free payoff as

$$v_4(x, t) = Ke^{-r\tau} \sum_{k=1-\eta}^{\eta} D_{m,k}^*(x)U_{m,k}^* \quad (68)$$

where $U_{m,k}^*$ is the approximation of

$$U_{m,k} := \int_{\mathbb{R}} \phi_{m,k}(y)g(y)dy = \int_{-\infty}^0 \phi_{m,k}(y)(1 - e^y)dy \quad (69)$$

These coefficients can be precomputed when initializing the SWIFT procedure and shared across different strikes and maturities, saving computation time.

Remark 4.9. The approximation $D_{m,k}^*$ depends on a parameter J that represents the number of intervals in the sinc approximation quadrature. The parameter used for payoff computation is independent from the one used for the density coefficients, and will be labeled J_p and J_d respectively.

4.3.1 Sinc integral and density coefficients approximation

Shannon MRA and SWIFT require solving integrals of the form $\int_{\mathbb{R}} f(x)g(\text{sinc}(x))dx$. In particular, when one wants to compute the density coefficients $D_{m,k}$ of Definition 4.2, or as will be seen in Section 4.3.2, for the payoff coefficients computation, one needs to solve integrals of the form $\int_{\mathbb{R}} f(x)\overline{\phi(x)}dx$ and $\int_{\mathbb{R}} f(x)\phi(x)dx$ respectively.

The simplest situation is to compute the sinc integral $Si(t) = \int_0^t \text{sinc}(x)dx$, but even for this case there is no closed form for the integral.

There are two commonly used approaches to numerically approximate these integrals:

- **Coefficients via Parseval's Theorem:** This approach, shown in [33] uses the fact that the $\phi_{m,k}$ has a very small support in the frequency domain (as shown in Equation 54) together with Parseval's theorem, $\langle f, g \rangle = \frac{1}{2\pi} \langle \hat{f}, \hat{g} \rangle$. As the coefficients will be real numbers, we can take $\langle f, g \rangle = \text{Re}(\langle f, g \rangle) = \text{Re} \left(\frac{1}{2\pi} \langle \hat{f}, \hat{g} \rangle \right)$. Entering the real part operator inside the integral sign and applying the change of variable $t = \frac{u}{2^{m+1}\pi}$ results in expression:

$$D_{m,k} = 2^{m/2} \int_{-1/2}^{1/2} \text{Re} \left(\hat{f}(2^{m+1}\pi t) e^{i2\pi kt} \right) dt \quad (70)$$

Any quadrature can be applied to numerically compute the integral, but there are known techniques to rearrange the trapezoidal and midpoint quadratures to compute all the coefficients with a single FFT [29, 14].

- **Coefficients via Sinc approximations:** An equivalent expression of the sinc function was used in [33] as a cosine expansion using Vieta's formula:

$$\text{sinc}(x) = \prod_{j=1}^{\infty} \cos \left(\frac{\pi x}{2^j} \right) \quad (71)$$

This cosine expansion was then truncated and converted to a summation via the cosine product-to-sum-identity

$$\prod_{j=1}^{\ell} \cos\left(\frac{\pi x}{2^j}\right) = \frac{1}{2^{\ell-1}} \sum_{j=1}^{2^{\ell-1}} \cos\left(\frac{2j-1}{2^{\ell}} \pi x\right) \quad (72)$$

obtaining the first expression for the Vieta coefficients $D_{m,k}^v$, where

$$D_{m,k} \approx D_{m,k}^v = \frac{2^{m/2}}{2^{\ell-1}} \sum_{j=1}^{2^{\ell-1}} \int_{\mathbb{R}} f(x) \cos\left(\frac{2j-1}{2^{\ell}} \pi (2^m x - k)\right) dx \quad (73)$$

Finally, $f(x)$ would be replaced in the integral by defining $D_{m,k}^v = \text{Re}(D_{m,k}^{vc})$.

$$D_{m,k}^{vc} = \frac{2^{m/2}}{2^{\ell-1}} \sum_{j=1}^{2^{\ell-1}} \int_{\mathbb{R}} f(x) e^{-i \frac{2j-1}{2^{\ell}} \pi (2^m x - k)} dx \quad (74)$$

Notice that when applying the $\text{Re}(\cdot)$ operator to $D_{m,k}^{cv}$, it can be moved freely across multiplications by real numbers and summations and integrals, and expression 72 is ultimately recovered by using $\text{Re}(e^{-i u x}) = \cos(x)$. Finally, splitting the complex exponential into x-dependent and x-independent terms and replacing $\int_{\mathbb{R}} f(x) e^{-x A i}$ by $\hat{f}(A)$ yields:

Definition 4.10. Vieta coefficients approximation

$$D_{m,k}^* = \frac{2^{m/2}}{2^{\ell-1}} \sum_{j=1}^{2^{\ell-1}} \text{Re}\left(\hat{f}(u_j 2^m) e^{i k u_j}\right) \quad (75)$$

where $u_j = \frac{\pi}{2^{\ell}}(2j-1)$

It is worth noting that, while the original construction of Definition 4.10 required a summation of a power of 2 number of terms, it was revisited in [29] and generalized by expressing the sinc in terms of its Fourier transform

$$\text{sinc}(t) = \text{Re}(\text{sinc}(t)) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \text{Re}(e^{-i u x}) du = \frac{1}{\pi} \int_0^{\pi} \cos(ux) du \quad (76)$$

Then, if one uses a mid-point quadrature with J buckets and proceeds as in the initial derivation, the expression in Definition 4.10 is recovered, with the new term J being equivalent to the previous $2^{\ell-1}$ one.

The following approximation error bound was given in [29] for the mid-point quadrature:

Lemma 4.11. Error of the Sinc Function Approximation

$$\text{sinc}^*(x; J) = \frac{1}{J} \sum_{j=1}^J \text{Re}(e^{-i u_j x}) \quad (77)$$

where $u_j = \frac{\pi(2j-1)}{2J}$. Then the approximation error of applying the midpoint quadrature on a finite domain $|x| \leq a \leq \frac{\pi}{2} J$ is bounded by:

$$\max_{|x| \leq a} |\text{sinc}(x) - \text{sinc}^*(x; J)| \leq \frac{(\pi a)^2}{(4J)^2 - (\pi a)^2} \quad (78)$$

Note that this approximation only holds for a bounded interval of the sinc function domain. This implies that, when used to compute a sinc integral, the integral must also be truncated so that $|x| \leq \frac{\pi J}{2}$.

In particular, if one approximates the Shannon wavelet $\phi_{m,k}(x)$ in the domain (a, b) by using the $\text{sinc}^*(x; J_k)$, the sinc function will be evaluated in the domain $(2^m a - k, 2^m b - k)$ and one can obtain a bound on J_k for each wavelet. As will be seen in Section 4.3.3, using a single value J that fulfills the inequality $\max(|2^m a - k|, |2^m b - k|) \leq \frac{\pi J}{2}$ for all k will allow to speed up computations by using a Fast Fourier Transform algorithm. So, assuming a single value J is used and symmetric domains $(-c, c)$ and $[1 - \eta, \eta]$ are considered for x and k respectively, one obtains the following corollary, presented in a slightly different form in [29]:

Corollary 4.12. Error of the Wavelet Approximation

$$\phi_{m,k}^*(x) := 2^{m/2} \text{sinc}^*(2^m x - k; J) \quad (79)$$

Then the error of applying the midpoint quadrature on a finite domain $|x| \leq c$ for all $k \in [k_1, k_2]$ is (assuming a fixed $J \geq \frac{\pi}{2}(2^m c + \eta)$)

$$\max_{|y| \leq c, 1-\eta \leq k \leq \eta} |\phi_{m,k}(y) - \phi_{m,k}^*(y)| \leq 2^{\frac{m}{2}} \frac{(\pi(2^m c + \eta))^2}{(4J)^2 - (\pi(2^m c + \eta))^2} \quad (80)$$

4.3.2 Payoff coefficients approximation

One should note that the same procedure as the approximation $D_{m,k} \approx D_{m,k}^*$ (discussed in Section 4.3.1) can be used for the payoff coefficients approximation. Starting from Equation 69, and applying the same sinc approximation as in the density coefficients, for European puts we obtain the expression:

$$U_{m,k} \approx \frac{2^{m/2}}{J_p} \sum_{j=1}^{J_p} \text{Re} \left(\exp \left\{ i \frac{2j-1}{N_p} \pi k \right\} \int_{\mathbb{R}} (1 - e^y)^+ \exp \left\{ -i \frac{2j-1}{N_p} \pi y \right\} dy \right) \quad (81)$$

The last step of the density coefficients approximation requires the convergence of the Fourier transform of the payoff coefficients which, in general, is not guaranteed. In particular the Fourier transform of the strike-free payoff of a European put option does not converge.

The sinc^* approximation is only defined in a finite domain anyway, so the SWIFT algorithm implementation deal with this situation by initially bounding the integral in Equation 12 to a finite interval. Here, for simplicity a symmetric interval $(-c, c)$ will be considered, reaching expression:

Definition 4.13. Strike-free payoff coefficients approximation

$$U_{m,k} \approx U_{m,k}^*(-c, c) := \frac{2^{m/2}}{J_p} \sum_{j=1}^{J_p} \text{Re} \left(e^{i \frac{2j-1}{N_p} \pi k} I_j(c) \right) \quad (82)$$

where

$$I_j(c) := \int_{|y| \leq c} g(y) e^{-i \frac{2j-1}{N_p} \pi y} dy := \int_{|y| \leq c} g(y) e^{-iu_j y} dy \quad (83)$$

And its analytic expression is given by

$$H(y) := -ie^{iu_j y} \left(\frac{1}{u_j} - \frac{e^y}{-i + u_j} \right) \quad (84)$$

$$I_j(c) = H(c) - H(-c) \quad (85)$$

Remark 4.14. As with the density coefficients, it must hold $J_p \geq \frac{\pi}{2}(2^m c + \eta)$.

One can easily see that the value of $I_j(c)$ will be periodic on c . In general, all the sinc approximations used in the SWIFT method are periodic, which can give rise to boundary issues and undervaluation of option prices when the option strikes approach the boundary of $(-c, c)$. This issue also appears in the COS option pricing method [12], another Fourier-transform-based option pricing method closely related to the SWIFT method, and is discussed by Maree et al. [29]. In their study, they use the independence between the parameters regulating the payoff integral domain (c) and the one regulating the wavelet series truncation (η) to carefully choose a value for c to avoid this problem.

4.3.3 Fast computation of the density and payoff coefficients

Starting with a general expression of the summation term that appears in both the density and payoff coefficients approximation:

$$f_k = \sum_{j=1}^J g_j e^{i \frac{2j-1}{2J} \pi k} \quad (86)$$

One can extend it by defining $g_j = 0$ for $j = 0$ and $J < j < 2J$ and take the j -independent terms outside of the summation, obtaining:

$$f_k = e^{-\frac{i\pi k}{2J}} \sum_{j=0}^{2J-1} g_j e^{i \frac{2j\pi k}{2J}} \quad (87)$$

This last summation expression coincides with the Discrete Fourier Transform (DFT) of length $2J$ of $\{g_j\}$, and the computation of all the values f_k can then be speeded up by using a Fast Fourier Transform (FFT) implementation.

Remark 4.15. Note that computing the density or payoff coefficients imposes a restriction on the wavelet series truncation parameter $2\eta < J$.

In all numerical examples in this article the C library FFTW will be used to compute any FFT [15].

4.3.4 Alternative method for multiple strikes

A key property of the SWIFT method is that, given a level of projection m , the payoff and density coefficient associated to each wavelet $\phi_{m,k}$ can be computed through two FFTs (one for all the density coefficients, and one for all payoff coefficients). Without this property, the SWIFT computation speed would not be competitive with other numerical option pricing methods [14].

In the option calibration problem, one usually needs to consider the option prices of several options at different strikes. In this specific case, if one were to compute the option price of M options at strikes $\mathbf{K} := (K_1, \dots, K_M)^T$, then the formulation proposed in Equation 68 would need to recompute the density and payoffs coefficients for every strike K_i . This involves evaluating the characteristic function $\eta \cdot J_p \cdot M$ times, an operation which, for the Heston model is more costly than evaluating the strike-free payoff function, or its integral. As stated in [33] one can improve the computation time of option pricing for multiple strikes when $\hat{f}(u; \mathbf{x}) = \hat{f}(u)e^{-i u \mathbf{x}}$, a property present in both Lévy and Heston models.

As stated in [29], let us start from Equation 68, and considering the previously mentioned vector of strikes \mathbf{K} , with its associate vector of initial states $\mathbf{x} := (\log(S_0/K_1), \dots, \log(S_0/K_M))^T$. One can then substitute the density coefficient approximation into the option price expression and substitute the two resulting summations, obtaining:

$$v_4(\mathbf{x}, t) := e^{-r\tau} \mathbf{K} \sum_{k=1-\eta}^{\eta} \operatorname{Re} \left\{ \sum_{j=1}^{J_d} \hat{f}(u_j 2^m; \mathbf{x}) e^{i u_j k} U_{m,k}^*(-c, c) \right\} \quad (88)$$

$$= e^{-r\tau} \mathbf{K} \sum_{j=1}^{J_d} \operatorname{Re} \left\{ \hat{f}(u_j 2^m; \mathbf{x}) \left[\sum_{k=1-\eta}^{\eta} U_{m,k}^*(-c, c) e^{i u_j k} \right] \right\} \quad (89)$$

$$= e^{-r\tau} \mathbf{K} \sum_{j=1}^{J_d} \operatorname{Re} \left\{ \hat{f}(u_j 2^m) e^{-i u_j \mathbf{x}} \left[\sum_{k=1-\eta}^{\eta} U_{m,k}^*(-c, c) e^{i u_j k} \right] \right\} \quad (90)$$

$$=: e^{-r\tau} \mathbf{K} \sum_{j=1}^{J_d} \operatorname{Re} \left\{ \hat{f}(u_j 2^m) e^{-i u_j 2^m \mathbf{x}} \tilde{U}_j(-c, c) \right\} \quad (91)$$

The original formulation from Equation 68 requires the following computations:

- For each of the M strikes:
 - 1 FFT of length $2J_d$ to compute 2η density coefficients.
 - J_d evaluations of the characteristic function $\hat{f}(u_j 2^m; \mathbf{x})$
- 1 FFT of length $2J_p$ to compute 2η payoff coefficients.
- J_p evaluations of the strike-free payoff integral $I_j(c)$ defined in Equation 83.

where the payoff dependent computations are independent of the strike price and can be precomputed and reused for all strikes. On the other hand, the alternative formulation provided in Equation 91 requires:

- For each of the M strikes:
 - J_d evaluations of the x -dependent term of the characteristic function e^{-iu_jx} .
- J_d evaluations of the x -independent characteristic function $\hat{f}(u_j2^m)$
- 2 FFT of lengths 2η and $2J_p$ to compute the J_d values of $\tilde{U}_j(-c, c)$.
- J_p evaluations of the strike-free payoff integral $I_j(c)$ defined in Equation 83.

where all the required values of $\hat{f}(u_j2^m)$ and $\tilde{U}_j(-c, c)$ (and the values of $I_j(c)$ required to compute the latter) can be precomputed and reused for all strikes.

Remark 4.16. In general, whenever the x dependency in $\hat{f}(u; x)$ can be easily isolated and is cheap to compute, one can benefit from the alternative formulation proposed in this section.

The computation of the characteristic function tends to be more expensive than the computation of the payoff integral, so a SWIFT implementation through Equation 91 tends to outperform the one through Equation 68 when several strike prices are involved. A discussion on the benefits of using a formulation equivalent to the one provided by Equation 91 for multiple strikes appears in [33] and, there, it is shown that is possible to define $F_j := \hat{f}(u_j2^m)$ and precompute its J_d required values once and reuse them for all strikes.

In this master's thesis, several other important speed properties of this formulation are presented. Two are presented below, and the rest will be presented in Section 4.3.5:

- $\tilde{U}_j(-c, c)$ is independent of the strike, so it can also be precomputed and reused with all strikes K_l .
- Let us define

$$G_j := \begin{cases} F_j \tilde{U}(-c, c) & \text{for } j \leq J_d \\ 0 & \text{for } J_d < j \leq 2J_d \end{cases} \quad (92)$$

then Equation 91 can be rearranged as:

$$v_4(x, t) = e^{-r\tau} \mathbf{KRe} \left\{ e^{\frac{\pi i 2^m x}{J_d} \sum_{j=1}^{2J_d} G_j e^{-\frac{2\pi i j 2^m x}{2J_d}}} \right\} \quad (93)$$

Then, if one chooses carefully the values of K_l so that $2^m x_l$ is an integer number, this computation can be speeded up by the use of a FFT. If one chooses $x_k := \frac{2k - J_d}{2^{m+1}}$, Equation 93 becomes:

$$\begin{aligned} v_4(x, t) &= e^{-r\tau} \mathbf{KRe} \left\{ e^{\frac{\pi i 2^m x}{J_d} \sum_{j=1}^{2J_d} G_j e^{-\frac{2\pi i j k}{2J_d}} e^{\pi i j}} \right\} \\ &= e^{-r\tau} \mathbf{KRe} \left\{ e^{\frac{\pi i 2^m x}{J_d} \sum_{j=1}^{2J_d} \tilde{G}_j e^{-\frac{2\pi i j k}{2J_d}}} \right\} \end{aligned} \quad (94)$$

where

$$\tilde{G}_j := G_j e^{\pi i j} = G_j (-1)^j \quad (95)$$

Remark 4.17. Note that, as with other FFT-based computations presented in this document, this approach imposes a bound $M \leq J_d$ to the number of different strikes that can be computed with the FFT.

If one considers the domain D of $x = \log(S_t/K)$, this approach allows pricing options in a symmetrical boundary $(-\frac{J_d}{2^{m+1}}, \frac{J_d}{2^{m+1}}) \in D$ at J_d uniformly distributed points at distance 2^m .

One cannot usually choose the strike prices at which to price the options, particularly not when calibrating a model through real market data, as only a limited set of strike values are listed on any exchange market, but this method could be used to quickly compute the option prices of an already calibrated model at a grid of points that could be tuned by the choice of m and J_d . Then the option prices at any intermediate strike could be interpolated with the help of a derivative-free spline (or, if the derivative with respect to K of Equation 94 preserves the same speed properties, with the help of any spline method that uses derivatives).

4.3.5 Option price gradient

The SWIFT option pricing method will be used to calibrate different models by means of an optimization problem, as will be described in Chapter 5. For gradient-based optimization methods, the gradient of the pricing formula used will be required. Thus, one needs to compute the gradient of $v_4(\theta; x, \tau)$

Lemma 4.18. *Option Price Gradient*

For each strike K_l with correspondent initial state x_l , the gradient of the option price is given by

$$\nabla_{\theta} v_4(\theta; x_l, \tau) = e^{-r\tau} K_l \sum_{j=1}^{J_d} \operatorname{Re} \left\{ \mathbf{h}(u_j 2^m) \hat{f}(u_j 2^m) e^{-iu_j 2^m x_l} \tilde{U}_j(-c, c) \right\} \quad (96)$$

where $\mathbf{h}(u)$ is defined as in Theorem 3.12.

Proof. Starting from Equation 91, as the gradient operator is linear, it can be exchanged with the scalar multiplications, summation, and real part operator present in the expression. Then, $e^{-iu_j 2^m x_l}$ and $\tilde{U}_j(-c, c)$ are both independent from θ and one obtains:

$$\begin{aligned} \nabla_{\theta} v_4(\theta; x_l, \tau) &= \nabla_{\theta} \left(e^{-r\tau} K_l \sum_{j=1}^{J_d} \operatorname{Re} \left\{ \hat{f}(u_j 2^m) e^{-iu_j 2^m x_l} \tilde{U}_j(-c, c) \right\} \right) \\ &= e^{-r\tau} K_l \sum_{j=1}^{J_d} \operatorname{Re} \left\{ \nabla_{\theta} \left(\hat{f}(u_j 2^m) e^{-iu_j 2^m x_l} \tilde{U}_j(-c, c) \right) \right\} \\ &= e^{-r\tau} K_l \sum_{j=1}^{J_d} \operatorname{Re} \left\{ \nabla_{\theta} \left(\hat{f}(u_j 2^m) \right) e^{-iu_j 2^m x_l} \tilde{U}_j(-c, c) \right\} \\ &= e^{-r\tau} K_l \sum_{j=1}^{J_d} \operatorname{Re} \left\{ \mathbf{h}(u_j 2^m) \hat{f}(u_j 2^m) e^{-iu_j 2^m x_l} \tilde{U}_j(-c, c) \right\} \end{aligned} \quad (97)$$

where the last equality follows directly from Theorem 3.12. \square

Remark 4.19. A similar expression can be reached when computing the gradient of any of the equivalent expressions of $v_4(\boldsymbol{\theta}; x, \tau)$ presented in previous sections.

It is worth noting that this expression holds the same speed properties than Equation 91, in particular, $\hat{f}(u_j 2^m)$ can also be precomputed for all j and l and reused for all strikes to compute both the price and the gradient of the option. One can also obtain three more speed properties:

- The value of $e^{-iu_j 2^m x_l}$ can be reused between the price and the gradient computations.
- If the SWIFT parameters are not changed during the gradient descent used in the calibration problem that will be presented in Chapter 5, then the values of both $U_j(-c, c)$ and $e^{-iu_j 2^m x_l}$ can be reused through all the calibration steps.
- One can reuse the values of $\hat{f}(u_j 2^m)$ from the price computation to compute the gradient.

So combining all the speed properties above, when solving a gradient-based calibration problem, one only needs to first precompute $U_j(-c, c)$ and $e^{-iu_j 2^m x_l}$ and then in each gradient-descent step, one can simultaneously compute both the price and the gradient of all strikes by computing once for each $j \in [1, J_d]$ the values of $\hat{f}(u_j 2^m)$ and $\mathbf{h}(u_j 2^m)$.

4.3.6 SWIFT parameter choice

A bound for $\epsilon_m := f(y|x) - P_m f(y|x)$ is provided in Lema 4.7. An approximation to this bound, which converges exponentially with respect to 2^m is presented in [29] for several Lévy functions which, for the GBM process is given by:

$$\epsilon_m \approx \frac{(2^m \pi)^{-1}}{4\pi\tau} \left(\left| \hat{f}(-2^m; x) \right| + \left| \hat{f}(2^m; x) \right| \right) \quad (98)$$

This provides an iterative method to try increasing values of m until the error ϵ_m falls under a desired threshold.

An equivalent result could be derived for the Heston model by following Heston's decay analysis [21], but is not presented in this work. Instead, the Area Under the Curve (AUC) approach presented in [33] is used, where the value of m is iteratively increased and for each value of m , η is decided by the cumulant test presented below. Then the integral of the density function approximation $f(y|x) := \sum_{k=1-\eta}^{\eta} D_{m,k} \phi_{m,k}(x)$ can be approximated as:

$$A := \frac{1}{2^{m/2}} \left(\frac{D_{m,1-\eta}}{2} + \sum_{k=2-\eta}^{\eta-1} D_{m,k} + \frac{D_{m,\eta}}{2} \right) \quad (99)$$

If it is far from the desired value, m is increased and the coefficients and AUC are recomputed until a desired value is obtained (in the tests run in this work, $m = 3$ was sufficient for long-term Heston options, while between $m = 5$ and $m = 7$ was needed for mid-term options).

The values of η , J_d , and J_p are intimately related to the truncation of the European option valuation integral to a finite interval $(-c, c)$ through the restrictions of the sinc function approximation discussed in Section 4.3.1. There are no results on how to find a value of c to guarantee a certain pricing error, but [33] and [9] propose an initial guess based on the distribution cumulants:

$$c := |c_1| + L\sqrt{c_2 + \sqrt{c_4}} \quad (100)$$

Where c_i is the i -th cumulant of the distribution (the values for the Heston distribution are obtained from [6]) and L is a heuristic value. [33] recommends $L = 10$, while [29] recommends $L = 6$. During the numerical tests $L = 10$ provided a better approximation for Heston options.

When using the multiple strike formulation presented in Section 4.3.4, [29] shows that under-pricing for strikes close to the boundary of $(-c, c)$ (due to the periodic nature of the sinc approximation) can be avoided by transforming the obtained interval $(-c, c)$ to $(x_{min} - c, x_{max} + c)$, where x is the initial state variable associated to a strike K and, if M different strikes are considered, $x_{min} := \min_{i \in [1, M]} x_i$ and $x_{max} := \max_{i \in [1, M]} x_i$.

Then one sets $\eta := \lceil 2^m c \rceil$, computes the vector of density coefficients and runs the AUC test presented above. If close to the desired value, η can be increased and the new density coefficients can be computed, while keeping the previously computed ones. As said above, if it is far from the desired value m needs to be increased and all the coefficients need to be recomputed.

Finally, values of J_d and J_p are chosen that enforce $J \geq \frac{\pi}{2}(2^m \max(|x_{min} - c|, |x_{max} + c|) + \eta)$

5. Option calibration

The goal of calibrating a model using market data is to obtain model parameters that, when used for option valuation with an appropriate option valuation method, yield prices similar to the real market ones.

Completely matching the market for all strikes and maturities is an over-determined problem so, generally, the calibration consists on an error minimization problem, using a set of option market prices at different strikes and maturities.

This section follows closely [9] and it adapts the optimization problem presented there to the the option valuation method presented in Section 4.3.

5.1 Objective function

Let $V^*(x_i, \tau_i)$ be the market price of a European call option and $V(\theta; x_i, \tau_i)$ be the price at the same strikes and maturities obtained by using the Heston analytic formula 36. Let us also assume that we use n different options to calibrate the model, so $i \in [1, n] \subset \mathbb{Z}$. Then:

Definition 5.1. The calibration of the model is defined as the minimization problem

$$\begin{aligned} & \min_{\boldsymbol{\theta} \in \mathbb{R}^5} (f(\boldsymbol{\theta})) \\ f(\boldsymbol{\theta}) & := \frac{1}{2} \|\mathbf{r}(\boldsymbol{\theta})\|^2 = \frac{1}{2} \mathbf{r}^T(\boldsymbol{\theta}) \mathbf{r}(\boldsymbol{\theta}) \end{aligned} \quad (101)$$

where $\mathbf{r}(\boldsymbol{\theta})$ is the n -dimensional vector of the residuals obtained when pricing the options considered for calibration using the model parameters. That is:

$$\mathbf{r}(\boldsymbol{\theta}) := [r_1(\boldsymbol{\theta}), \dots, r_n(\boldsymbol{\theta})]^T, \quad r_i(\boldsymbol{\theta}) := V(\boldsymbol{\theta}; x_i, \tau_i) - V^*(x_i, \tau_i), \quad i = 1, \dots, n \quad (102)$$

It is worth noting that the choice of a specific objective function (and appropriate strikes and maturities to use as calibration points) will impact the obtained model. This can be used when trading to fit to the specific needs of a trader, ultimately leading to different objective functions. One example is using log-likelihood based methods for hedging and market making [7].

One must be cautious, as this choice can lead to consistently giving more weight in the calibration to some set of options over the others. This is the case, according to [9], of a commonly used calibration scheme in industry which consists on a least-squares problem similar to Equation 101. If one computes the vector of implied BS volatilities $\boldsymbol{\sigma}_{\text{BS}}$ corresponding to each option price, and runs a squared differences optimization problem based on implied volatility differences, then the resulting calibration scheme can give more weight to OTM options than ITM ones.

The choice of calibration strikes and maturities can also have a similar impact in the pricing model. For example, as an option gets closer to the expiration date, the variance of the price change until expiration of the underlying asset gets smaller. So, if one were to always use the same strikes across different expiries, then for long-term maturities the model would end up with a lot of calibration points ATM but would lack control points in regions where the density function of the underlying asset is still relevant, which could lead to serious pricing errors for mildly OTM and ITM options. For the short-term maturities, the relevant regions of the density function would have small number of sparse strikes and there would be a lot of strikes in very OTM and ITM regions, where prices are mostly close to 0 and to the discounted payoff respectively, so they are less sensitive to the change in the model parameters and do not help much in calibration (the price derivatives with respect to the model parameters will be almost flat at those expiry-maturity points, so they will not affect much the optimization problem), effectively wasting computational power.

As this master's thesis does not strive for any specific trading strategy, but will compare the performance of the calibration method with respect to [9], their calibration objective function (defined by Equations 101 and 102) will be used as it is, computing the option prices with the scheme provided in Section 4.3.

5.2 Gradient descent step

Given the optimization problem presented in the previous section, one can define the Jacobian and Hessian matrices of \mathbf{r} .

Definition 5.2. Jacobian matrix of the residual vector \mathbf{r} :

$$\mathbf{J} := \nabla_{\boldsymbol{\theta}} \mathbf{r}^T = \nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta}; \mathbf{x}, \tau) \quad (103)$$

where

$$J_{ji} = \left(\frac{\partial r_i}{\partial \theta_j} \right) = \left(\frac{\partial V(\boldsymbol{\theta}; x_i, \tau)}{\partial \theta_j} \right) \quad (104)$$

Definition 5.3. Hessian matrix of the residual element r_i :

$$\mathbf{H}(r_i) := \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}}^T r_i = \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}}^T V(\boldsymbol{\theta}; x_i, \tau) \quad (105)$$

where

$$H_{jk}(r_i) = \left(\frac{\partial^2 r_i}{\partial \theta_j \partial \theta_k} \right) = \left(\frac{\partial^2 V(\boldsymbol{\theta}; x_i, \tau)}{\partial \theta_j \partial \theta_k} \right) \quad (106)$$

Then, the gradient and Hessian of the objective function defined by Equations 101 and 102 are:

$$\nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) = \mathbf{J} \mathbf{r} \quad (107)$$

$$\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}}^T f(\boldsymbol{\theta}) = \mathbf{J} \mathbf{J}^T + \sum_{i=1}^M r_i \mathbf{H}(r_i) \quad (108)$$

The gradient descent will be solved by a Levenberg-Marquard (LM) method [32]. This method uses a modification of the Newton method that lets it dynamically use either a steepest-descent or a Gauss-Newton descent step, depending on whether the current guess is close or far from the optimum. The exact expression of the step is:

$$\Delta \boldsymbol{\theta} = (\mathbf{J} \mathbf{J}^T + \mu \mathbf{I})^{-1} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \quad (109)$$

where \mathbf{I} is the identity matrix and $\mathbf{J} \mathbf{J}^T + \mu \mathbf{I}$ substitutes the Hessian matrix used in the Newton method.

When the current guess is far from the optimum, a large value is given to μ so that

$$\Delta \boldsymbol{\theta} \approx \Delta \boldsymbol{\theta}^{(SD)} = (\mu \mathbf{I})^{-1} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \quad (110)$$

and a small step of a steepest-descent method is taken.

When the current guess is close to the optimum, a small value is given to μ so that

$$\Delta \boldsymbol{\theta} \approx \Delta \boldsymbol{\theta}^{(GN)} = (\mathbf{J} \mathbf{J}^T)^{-1} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \quad (111)$$

and the Hessian usually used in the Newton method is replaced by its Gauss-Newton approximation. This approximation is reliable when either r_i or $\mathbf{H}(r_i)$, and [9] justifies its usage by conjecturing that f is nearly linear, a condition that guarantees that the second requirement holds. One should note that even if f were not linear LM should only use small values of μ when $|\mathbf{r}|$ is small at the current step of the optimization problem.

The LM optimization problem will be solved through the LEVMAR c package [27], which can be built to use the LAPACK linear algebra package [35]. For this work LAPACK will be used.

This implementation of LM has 3 different stopping criteria and stops when any of them is fulfilled:

$$|\mathbf{r}_k| \leq \epsilon_1 \quad (112)$$

$$|\mathbf{J}_k \mathbf{e}| \leq \epsilon_2 \quad (113)$$

$$\frac{|\nabla \theta_k|}{|\theta_k|} \leq \epsilon_3 \quad (114)$$

The first stopping criteria is fulfilled when the objective function defined by Equations 101 and 102 has reached a value closer to zero than the indicated tolerance. It is only when the method stops due to this criteria that this master's thesis will consider that the model has been properly calibrated.

The second criteria corresponds to a flat gradient, and the third corresponds to a stagnating update (this last one has never happened while testing the Heston convergence).

6. Numerical results

In this master's thesis, the SWIFT method will be used to calibrate a Heston model through European call options price data at different strikes and maturities and it will be compared to the pricing and calibration method based on Equation 36 proposed by [9], which for the sake of readability will be denoted Cui Pricer(CP). CP will be implemented using a Gauss-Legendre quadrature with 64 nodes for its numerical integration step. The upper limit of the integral will be truncated, whenever possible, at $\bar{u} = 200$, but will be adjusted if necessary. The calibration process will consist on applying an LM method to the objective function defined in Equation 101.

The SWIFT method will be implemented using the characteristic function expression and its derivatives provided also in [9].

The tests that will be performed can be summarized as:

- **Stress tests:** Both the CP and SWIFT will be tested with several combinations of extreme strikes (ATM and deep ITM and OTM), and both really long-term and short-term maturities to detect any possible limitations or numerical issues of both techniques in a wide usage range.
- **Speed tests:** The option calibration speeds for the regular SWIFT method (defined by Equation 68) and the one devised to quickly compute several option prices with different strike and same maturity (defined by Equation 91 and which will be denoted KSWIFT), will be compared against CP for three different strike and expiry sets to check whether the multiple-strike alternative formulation is necessary to obtain a competitive option calibration method. These scenarios will represent:
 - An extreme situation with a single expiry and multiple strikes.
 - A situation with a fixed number of maturities and a fixed number of strikes per maturity.
 - An extreme situation with different expiries for each strike.

When computing options with more than one different strike, a combination of OTM and ITM options will be used to provide an heterogeneous sample of contracts. Similarly, when more than one maturity is considered, a sample of long and short term expiries will be used.

- **Realistic convergence tests:** As in [9], convergence of the method will be tested for realistic model parametrisations representative of long-dated Foreign Exchange (FX), interest rate, and equity options, as they are relevant and, according to Glasserman and Kim [20], challenging for simulations of the Heston model.

Several sets of Heston parameters will be used for the different numerical tests and are presented in Table 1. The last three sets of parameters are representative of long-term FX, interest rate, and equity options respectively [3].

Name	κ	$\bar{\nu}$	σ	ρ	ν_0
$\theta^{(1)}$	3	0.1	0.25	-0.8	0.08
$\theta^{(2)}$	1.5768	0.0398	0.0175	-0.5711	0.0175
$\theta^{(FX)}$	0.5	0.04	1	-0.9	0.04
$\theta^{(IR)}$	0.3	0.04	0.9	-0.5	0.04
$\theta^{(EQ)}$	1	0.09	1	0.04	0.09

Table 1: Set of Heston parameters used in the numerical tests.

The initial guesses for parameters $\theta^{(1)}$ and $\theta^{(2)}$, used in Sections 6.1 and 6.2 respectively will be $\theta_0^{(1)} = (1.2, 0.2, 0.3, -0.6, 0.2)^T$ and $\theta_0^{(2)} = (1.5768, 0.0398, 0.5751, -0.5711, 0.0175)^T$

The computations were performed on a 64-bit Ubuntu 18.04.4 LTS with a 3.70GHz Intel Core i7-8700K processor and 62.8 gigabytes of ram.

6.1 Stress Tests

Deep ITM and OTM call options are priced together with ATM call options for long and short term maturities using two SWIFT methods (one with $m = 3$ and one with $m = 7$) and a CP method. The time until maturity τ is given in years. Thus, the expiries of 0.04 attempt to simulate a situation of around 2 weeks until expiration of the option contract.

Table 2 presents the pricing results $V_{SWIFT}^{m=3}$ and $V_{SWIFT}^{m=7}$ refer to the prices obtained with the SWIFT methods with $m = 3$ and $m = 7$ respectively and $V_{CP}^{\bar{u}=200}$ refers to the price obtained with the CP method.

Both methods run into numerical issues with either really big or really small expiries if no changes are made:

- CP and SWIFT implemented with $m = 7$ produced Not a Number (nan) when evaluating big expiries. Looking at the option price execution with the integrated debugger of the GDB compiler [40] showed that Equations 40 and 42 run into numerical overflow when the exponent $\frac{d\tau}{2}$ of its

Parameters	S	K	τ	$V_{SWIFT}^{m=3}$	$V_{SWIFT}^{m=7}$	$V_{CP}^{\bar{u}=200}$
$\theta^{(1)}$	100	50	45	65.565	nan	nan
$\theta^{(1)}$	100	100	45	46.911	nan	nan
$\theta^{(1)}$	100	200	45	27.198	nan	nan
$\theta^{(1)}$	100	50	0.04	44.221	50.000	50.000
$\theta^{(1)}$	100	100	0.04	0.380	1.045	1.046
$\theta^{(1)}$	100	200	0.04	0	0	1.079e-3

Table 2: Set of Heston parameters used in the numerical tests.

hyperbolic functions is big enough. In most of the tests above, the overflow could be avoided when carefully setting an appropriate value for the upper bound \bar{u} of integral 36, and by using a smaller value of m . The error can also be avoided by using Schouten's expression 39 of Heston's characteristic function (the obtained prices are denoted $V_{Schouten}$ in Table 3). It should be noted that using this other expression just for the characteristic function will not avoid the overflow issue for calibration, as the same hyperbolic functions appear evaluated at the same point in the expression of $\mathbf{h}(u)$ provided in [9].

- SWIFT with $m = 3$ tends to underprice short expiry options. After checking the swift parameters obtained through the parameter choice method defined in Section 4.3 it was observed that the base guess for η obtained by simply using the cumulant expression proposed resulted in a truncated Shannon wavelet expansion that did not cover a sufficient domain of the density function $f(y|x)$ which could be inferred from the values of the obtained AUC. A dynamical choice of the parameter η , as described in [33] could avoid this issue.

Increasing the value of m also fixes the problem and, as the values of m used for the other sections of this chapter give good AUC values, the dynamical method was not implemented.

- None of the methods can handle the deep OTM option with a short expiry. The expected value should be close to but bigger than 0, as there are only 10 trading days to expiry and the price of the underlying should increase 50% so that the option contract would not expire worthless. CP value seems too high and, in fact, when increasing the value of \bar{u} in the interval [100, 400] the price never clearly converges to a certain value, and it can give higher estimates for $\bar{u} > 200$ than $1.079e - 3$, or even negative values. Changing the characteristic function expression does not fix this issue. SWIFT consistently gives it a price of 0. The contribution that makes the price different than 0 probably lies on the tails of the distribution function, and one would require a really big c so that a point with a positive payoff is even considered in Equation 91.

A method to set an optimal \bar{u} value is not provided in [9], so it has to be manually determined when changing the time to expiry of the options one wants to price. This issue is also present in SWIFT but in the cases studied here, using the iterative SWIFT parameter choice scheme provided in Section 4.3.6 seems to suffice to avoid this issue for reasonable tolerance levels.

Further, the errors arise for big values of $\frac{d\tau}{2}$, which also depend on Heston parameters κ , σ , and ρ . It would be interesting to study the dependence between changes in these parameters and changes in the optimal values of \bar{u} . If significant changes can happen during the calibration scheme itself, one may

Parameters	S	K	τ	\bar{u}	$V_{CP}^{\bar{u}}$	$V_{Schouten}^{\bar{u}=200}$
$\theta^{(1)}$	100	50	45	6	65.565	65.565
$\theta^{(1)}$	100	100	45	6	46.911	46.911
$\theta^{(1)}$	100	200	45	6	27.198	27.198
$\theta^{(1)}$	100	50	0.04	200	50.000	50.000
$\theta^{(1)}$	100	100	0.04	200	1.046	1.046
$\theta^{(1)}$	100	200	0.04	300	-1.174e-5	1.079e-3

Table 3: Results when choosing an appropriate \bar{u} , or when keeping \bar{u} at 200 and Schouten's characteristic function. The column \bar{u} indicates the value at which CP integral is truncated, and the option price obtained is shown in column $V_{CP}^{\bar{u}}$. Column $V_{Schouten}^{\bar{u}=200}$ shows the price obtained when keeping $\bar{u} = 200$ but implementing CP using the characteristic function provided in Equation 39. The last row is an example of the negative values obtained in the deep OTM short-term call.

need to automatically change the value of \bar{u} something which is currently not possible in CP. One can automatically change SWIFT parameters at every calibration step, but assuming these parameters constant simplified the expression of the price gradient in Section 4.3.5. If one decides to change them dynamically while running the gradient descent, one would need to integrate the effect of changing them into the option gradient. That is, one would need to take into account how a change in Heston parameters changes the SWIFT parameters, and how the change in these affects the option price. Some of the parameters change discretely, so it is unclear if this is a feasible approach.

Another possibility would be to bound the error between the SWIFT price gradient approximation $\nabla_{\theta} V_{SWIFT}(\theta; x, \tau)$ and the real Heston option price gradient $\nabla_{\theta} V(\theta; x, \tau)$. If the piece-wise error between the two gradient components is maintained below a tolerance ϵ , then every step of the gradient descent will advance in the direction to the optimum while $\nabla_{\theta} V(\theta; x, \tau)$ is not too flat. In particular, if none of its components has an absolute value smaller than ϵ .

6.2 Speed Tests

The calibration speed has been tested for 3 different sets of strike prices and maturities, which are available in Appendix B.2. Set 3 is specially designed for KSWIFT. It consists on the same strikes and expiries as Set 2, but they are fed as input variables to the code implementation in a way that prevents the algorithm from applying the speed up techniques discussed in Section 4.3.4, so in Table 4 Set 3 contains values only for KSWIFT (as Set 3 is equivalent to Set 2 for the other two calibration methods).

The values for KSWIFT and CP have been averaged over 100 executions of the calibration to provide a good estimate of the required calibration time. It can be seen that regular SWIFT is orders of magnitude slower without averaging the required time over several executions. Hence, the multiple-strike alternative formulation presented in Section 4.3.4 and all the speed-up techniques discussed through Section 4.3 are necessary to provide a competitive method that can be used for real-time model updating.

KSWIFT performance is comparable to CP for Set 2, an order of magnitude faster for Set 1, and an

Strike and maturities Set	Heston Parameters	Method	Time(sec)	I	ϵ_1
Set 1	$\theta^{(2)}$	SWIFT	6.923	10	3.932e-11
Set 1	$\theta^{(2)}$	KSWIFT	4.501e-3	10	3.932e-11
Set 1	$\theta^{(2)}$	CP	4.559e-2	10	3.932e-11
Set 2	$\theta^{(2)}$	SWIFT	35.91	13	1.002e-12
Set 2	$\theta^{(2)}$	KSWIFT	5.011e-2	13	1.002e-12
Set 2	$\theta^{(2)}$	CP	6.314e-2	13	1.002e-12
Set 3	$\theta^{(2)}$	KSWIFT	1.727e-1	13	1.002e-12

Table 4: Iterations and time needed to calibrate each speed scenario and objective function value reached. I refers to the number of iterations LM requires until it stops, and ϵ_1 corresponds to LM first stopping criteria (see Section 5.2), which refers to the objective function final value.

order of magnitude slower for Set 3. It can be argued that both Set 1 and 3 are extreme cases that are not really relevant for real option trading situations: One would rarely use a single strike per expiry to calibrate an option pricing model, and using data from a single expiry only seems reasonable when trading a single option expiry (in this case, one could benefit from the speed properties of KSWIFT on scenarios like Set 1). According to [9], a reasonable calibration scenario consists in using option prices from strikes at with 0%, $\pm 25\%$, and $\pm 50\%$ BS delta (derivative of the option price with respect to the underlying price value. It has a closed analytic expression for European BS options).

The calibration speed of KSWIFT and CP is of the order of $5e - 2$ seconds, which seems sufficient for real-time model updating to provide market information to a human trader. In more computationally demanding trading environments, like high-frequency trading neither KSWIFT nor CP would be competitive enough.

Remark 6.1. All the single expiry tests (the first three tests on Table 4) converged to an approximated value different than $\theta^{(2)}$ but approximated all the option prices properly. Using different initial guesses lead to different approximated values which minimized the objective function. It would be interesting to see whether this is a property of the Heston distribution (that is, it has at least a degree of freedom when defined from option prices in a single expiry) or it is due to the specific scenario being tested.

6.3 Realistic Convergence Tests

Each of the realistic parameter sets proposed is calibrated for 100 different initial states. Each of the components of the components of the initial parameters guess is drawn uniformly at random within $\pm 10\%$ of the optimal value.

According to [9] this is representative of real option calibration as, usually, the initial guess used for a certain calibration problem is the last available parameter estimation. If the calibration is updated fast enough, it is expected that the initial guess will be this close to the optimum.

The maturities used in [9] are not available, so for these tests the strike-expiry Set 2 will be used.

	θ_1	θ_2	θ_3
$ \kappa^a - \kappa^* $	6.640e-4	2.657e-4	1.160e-3
$ \bar{\nu}^a - \bar{\nu}^* $	1.547e-4	1.321e-5	1.746e-5
$ \sigma^a - \sigma^* $	1.978e-3	2.248e-4	3.725e-4
$ \rho^a - \rho^* $	2.649e-4	1.365e-5	8.661e-6
$ \nu_0^a - \nu_0^* $	3.629e-5	4.790e-6	8.339e-6
Iterations	13.85	6.32	6.78
Time	3.341e-1	1.938e-1	2.006e-1
ϵ_1	2.867e-11	2.030e-11	3.643e-11

Table 5: Convergence statistics averaged over 100 calibrations. x^a refers to the calibration problem's estimation of variable a so, for example $|\kappa^a - \kappa^*|$ refers to how close LM approximation of κ was to the real value.

As can be seen in Table 5, even under challenging parameter setups representative of real option trading, KSWIFT is able to provide accurate estimations of the Heston parameters, taking on average a computation time of hundreds of milliseconds. These results, both in terms of speed as in terms of accuracy, are comparable to the tests in [9], so it is concluded that KSWIFT is as efficient as CP for real market scenarios.

7. Conclusions

An extension of the SWIFT method for European options calibration has been provided in Section 4.3 together with novel speed-up techniques, which can radically improve the performance when several of the priced/calibrated options have the same time to maturity.

Some numerical issues with Equation 42 are discussed for long-term expiries when evaluated at big values of u . Following the dynamic SWIFT parameters choice method described in Section 4.3.6 seems to be enough to avoid these problems for a reasonable tolerance level requirement, but CP needs to be adjusted manually.

The proposed speed-up techniques are deemed necessary in order for the new SWIFT calibration method to be competitive, as seen in the numerical speed tests. In particular, it has been shown that the only situation where the proposed calibration is significantly slower than CP is when one calibrates the model with many different maturities with no more than one or two strikes per maturity. As the number of strikes per expiry increases, the relative speed of the SWIFT method increases, and it is 10 times faster than CP when calibrating 40 options with a single maturity. Both extreme situations are not representative of real option trading needs, and for a reasonably real situation of 5 strikes per expiry, the SWIFT technique is slightly faster than CP. A SWIFT implementation without the previously discussed speed-up techniques has also been tested and deemed non-competitive, with calibration times that reached the dozens of seconds.

Further, the proposed calibration strategy passes the realistic calibration test for challenging Heston model parameter set-ups presented in [9].

In summary, the proposed method is deemed suitable for real-time updating of option models used in human-supervised trading schemes. Neither SWIFT nor CP method are suitable for the most demanding algorithmic trading situations, like high-frequency trading.

7.1 Further Research

Through this document several lines of research have appeared that escaped the scope of the currently presented research and will be expanded upon in future studies. Some of these lines of research are:

- Extending the calibration method to barrier, Bermudan and Asian options.
- Most of the calibration tests with a single expiry have run into an optimal value different than the original one. It is to be seen if this is a provable property of the Heston model or if this was due instead to the specific parameter or strike/maturity values being used.
- Some of the speed-up techniques used for same-expiry European Heston option calibration with the SWIFT method could be extended to the option pricer used by [9]. In particular, Equation 36 can use $\hat{f}(u; x) = \hat{f}(u)e^{-iux}$ to precompute $\hat{f}(u)$ only once per calibration step and per required value of u (that is, 2 evaluations per node used in the Gauss-Legendre quadrature) and precompute e^{-iux} once per strike K_I (with associated x_I initial state value) and required value of u , and could then be reused for all the calibration steps.
- It would be interesting to study the properties of the SWIFT implementation proposed for a chosen set of strikes and condensed in Equation 94. In particular, the derivative of the price expression with respect to K could be computed and, if the quick properties of the SWIFT are not lost on the obtained expression, one could interpolate the values at all strikes $K_1 \leq K \leq K_M$ with splines methods that require derivatives, and not only derivative-free ones.

References

- [1] L. Ortiz-Gracia A. Leitao and E.I. Wagner. SWIFT valuation of discretely monitored arithmetic Asian options. *Journal of Computational Science*, (28):120–139, 2018.
- [2] H. Albrecher, P. Mayer, W. Schoutens, and J. Tistaert. The little Heston trap. *Wilmott*, 2007.
- [3] L. Andersen. Simple and efficient simulation of the Heston stochastic volatility model. *The Journal of Computational Finance*, 2008.
- [4] C. Bin. Calibration of the heston model with application in derivative pricing and hedging. *Master's thesis, Department of Mathematics, TU Delft*, 2007.
- [5] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–657, 1973.
- [6] G. Bormetti, V. Cazzola, G. Livan, G. Montagna, and O. Nicosini. A generalized Fourier transform approach to risk measures. *Journal of Statistical Mechanics: Theory and Experiment*, 2010.
- [7] P. Christoffersen and K. Jacobs. The importance of the loss function in option valuation. *Journal of Financial Economics*, 2004.
- [8] I. Clark. Foreign exchange option pricing: A practitioner's guide. *Chichester: Wiley*, 2011.
- [9] Y. Cui, S. del Baño Rollin, and G. Germano. Full and fast calibration of the Heston stochastic volatility model. *European Journal of Operational Research*, 263(2):625–638, dec 2017.
- [10] I. Daubechies. Ten Lectures on Wavelets. *CBMS-NSF Reg. Conf. Ser. Appl. Math.*, 1992.
- [11] S. del Baño Rollin, A. Ferreiro-Castilla, and F. Utzet. On the density of log-spot in the Heston volatility model. *Stochastic Processes and their Applications*, 2010.
- [12] F. Fang and C. 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] J. L. Fernández, A. M. Ferreiro, J. A. García-Rodríguez, A. Leitao, J. G. López-Salas, and C. Vázquez. Static and dynamic SABR stochastic volatility models: Calibration and option pricing using GPUs. *Mathematics and Computers in Simulation*, 94:55–75, 2013.
- [14] F. Le Floc'h. Notes on the SWIFT method based on Shannon Wavelets for Option Pricing. Papers, arXiv.org, may 2020.
- [15] M. Frigo and S. G. Johnson. The design and implementation of FFTW3. In *Proceedings of the IEEE*, 2005.
- [16] J. Gatheral. The volatility surface: A practitioner's guide. *Finance*, 357, 2006.
- [17] F. Gerlich, A. M. Giese, J. H. Maruhn, and E. W. Sachs. Parameter identification in financial market models with a feasible point SQP algorithm. *Computational Optimization and Applications*, 51(3):1137–1161, apr 2012.

- [18] M. Gilli and E. Schumann. Calibrating option pricing models with heuristics. *Studies in Computational Intelligence*, 2011.
- [19] M. Gilli and E. Schumann. Heuristic optimisation in financial modelling. *Annals of Operations Research*, 2012.
- [20] P. Glasserman and K. Kuk Kim. Gamma expansion of the Heston stochastic volatility model. *Finance and Stochastics*, 2011.
- [21] S. L. Heston. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *Review of Financial Studies*, 1993.
- [22] J. Hull. *Futures, Options and Other Derivatives*. 2009.
- [23] J. C. Jackwerth and M. Rubinstein. Recovering probability distributions from option prices. *Journal of Finance*, 51(5):1611–1631, 1996.
- [24] A. Janek, T. Kluge, R. Weron, and U. Wystup. FX smile in the Heston model. In *Statistical Tools for Finance and Insurance*, pages 133–162. Springer Berlin Heidelberg, 2011.
- [25] C. Kahl and P. Jäckel. Not-so-complex logarithms in the Heston model. *Wilmott Magazine*, 2006.
- [26] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 1998.
- [27] M. I. A. Lourakis. levmar: Levenberg–Marquardt nonlinear least squares algorithms in C/C++., 2004.
- [28] S. Mallat. *A Wavelet Tour of Signal Processing*. 2009.
- [29] S. C. Maree, L. Ortiz-Gracia, and C. W. Oosterlee. Pricing early-exercise and discrete barrier options by Shannon wavelet expansions. *Numerische Mathematik*, 136(4):1035–1070, aug 2017.
- [30] R. C. Merton. Theory of rational option pricing. *Bell Journal of Economics and Management Science*, 4(1):141–183, 1973.
- [31] S. Mikhailov and U. Nögel. Heston 's Stochastic Volatility Model Implementation , Calibration and Some Extensions. *Wilmott Magazine*, 4:74–79, 2003.
- [32] J. J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. 1978.
- [33] L. Ortiz-Gracia and C. W. Oosterlee. A highly efficient shannon wavelet inverse fourier technique for pricing European options. *SIAM Journal on Scientific Computing*, 2016.
- [34] C. F. Peake. How Useful Are Implied Distributions? Evidence from Stock Index Options. *CFA Digest*, 2000.
- [35] M. Planitz and E. Anderson. LAPACK Users Guide. *The Mathematical Gazette*, 1995.
- [36] W. Schoutens. *Lévy Processes in Finance*. 2003.
- [37] W. Schoutens, E. Simons, and J. Tistaert. A perfect calibration! now what? *Wilmott*, 2004.
- [38] C. E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

- [39] F. Stenger. *Handbook of sinc numerical methods*. CRC Press, apr 2016.
- [40] F. T. Willmore. Debugging with gdb. In *Introduction to Scientific and Technical Computing*. 2016.

A. Code

The implementation of the SWIFT coded for this work can be consulted in the following Github public repository: <https://github.com/eudaldrq/SWIFTOptionCalibration>

B. Complimentary Formulas

B.1 Cui Gradient Complimentary Formulas

$$\frac{\partial d}{\partial \rho} = \frac{\xi \sigma i u}{d} \quad (115)$$

$$\frac{\partial A_2}{\partial \rho} = \frac{\sigma i u (2 + \xi \tau)}{2 d \nu_0} \left(\xi \cosh \frac{d \tau}{2} + d \sinh \frac{d \tau}{2} \right) \quad (116)$$

$$\frac{\partial B}{\partial \rho} = \frac{e^{\kappa \tau / 2}}{\nu_0} \left(\frac{1}{A_2} \frac{\partial d}{\partial \rho} - \frac{d}{A_2^2} \frac{\partial A_2}{\partial \rho} \right) \quad (117)$$

$$\frac{\partial A_1}{\partial \rho} = \frac{i u (u^2 - i u) \tau \xi \sigma}{2 d} \cosh \frac{d \tau}{2} \quad (118)$$

$$\frac{\partial A}{\partial \rho} = \frac{1}{A_2} \frac{\partial A_1}{\partial \rho} - \frac{A}{A_2} \frac{\partial A_2}{\partial \rho} \quad (119)$$

$$\frac{\partial A}{\partial \kappa} = - \frac{i}{\sigma u} \frac{\partial A}{\partial \rho} \quad (120)$$

$$\frac{\partial B}{\partial \kappa} = - \frac{i}{\sigma u} \frac{\partial B}{\partial \rho} + \frac{B \tau}{2} \quad (121)$$

$$\frac{\partial d}{\partial \sigma} = \left(\frac{d}{\sigma} - \frac{1}{\xi} \right) \frac{\partial d}{\partial \rho} + \frac{\sigma u^2}{d} \quad (122)$$

$$\frac{\partial A_1}{\partial \sigma} = \frac{(u^2 - i u) \tau}{2} \frac{\partial d}{\partial \sigma} \cosh \frac{d \tau}{2} \quad (123)$$

$$\frac{\partial A_2}{\partial \sigma} = \frac{\rho}{\sigma} \frac{\partial A_2}{\partial \rho} + \frac{2 + \tau \xi}{\nu_0 \tau \xi i u} \frac{\partial A_1}{\partial \rho} + \frac{\sigma \tau A_1}{2 \nu_0} \quad (124)$$

$$\frac{\partial A}{\partial \sigma} = \frac{1}{A_2} \frac{\partial A_1}{\partial \sigma} - \frac{A}{A_2} \frac{\partial A_2}{\partial \sigma} \quad (125)$$

B.2 Strike and Maturity Test Sets

Sets 1 and 2 are provided in Tables 6 and 7. The goal of the Set 3 is just to check the behavior of KSWIFT in the worst configuration possible for its speeding up techniques. This scenario is only be applied to KSWIFT, and consists on the same strikes and maturities as Set 2. The code implementation of KSWIFT generated for this master's thesis receives as inputs a vector of expiry-defined-data (EDD). Each element

Expiry	Strike	Strike	Strike	Strike	Strike
0.119047619047619	0.9371	0.9956	1.0427	1.2287	1.3939
	0.8603	0.9868	1.0463	1.2399	1.4102
	0.8112	0.9728	1.0499	1.2485	1.4291
	0.7760	0.9588	1.0530	1.2659	1.4456
	0.7470	0.9464	1.0562	1.2646	1.4603
	0.7216	0.9358	1.0593	1.2715	1.4736
	0.6699	0.9175	1.0663	1.2859	1.5005
	0.6137	0.9025	1.0766	1.3046	1.5328

Table 6: Strike and Expiries Set 1. All the strikes have the same expiry

Expiry	Strike	Strike	Strike	Strike	Strike
0.119047619047619	0.9371	0.9956	1.0427	1.2287	1.3939
0.238095238095238	0.8603	0.9868	1.0463	1.2399	1.4102
0.357142857142857	0.8112	0.9728	1.0499	1.2485	1.4291
0.476190476190476	0.7760	0.9588	1.0530	1.2659	1.4456
0.595238095238095	0.7470	0.9464	1.0562	1.2646	1.4603
0.714285714285714	0.7216	0.9358	1.0593	1.2715	1.4736
1.07142857142857	0.6699	0.9175	1.0663	1.2859	1.5005
1.42857142857143	0.6137	0.9025	1.0766	1.3046	1.5328

Table 7: Strike and Expiries Set 2

of the vector of EDD contains a single expiry and a vector of strikes. So, Set 2 will have all the strikes with the same expiry grouped in a single EDD, and Set 3 will have EDD consisting on a single strike. This will enforce full recomputation of the density and payoff coefficients for each strike.