Volatility Models in Option Pricing

Miguel Ângelo Maia Ribeiro

Thesis to obtain the Master of Science Degree in

Engineering Physics

Supervisors: Prof. Cláudia Rita Ribeiro Coelho Nunes Philippart
Prof. Rui Manuel Agostinho Dilão

Examination Committee

Chairperson: Prof. Full Name
Supervisor: Prof. Full Name 1 (or 2)
Member of the Committee: Prof. Full Name 3

Month Year

To my parents and sister

Acknowledgments

A few words about the university, financial support, research advisor, dissertation readers, faculty or other professors, lab mates, other friends and family...

Resumo

Inserir o resumo em Português aqui com o máximo de 250 palavras e acompanhado de 4 a 6 palavras-chave...

Palavras-chave: palavra-chave1, palavra-chave2,...

Abstract

Insert your abstract here with a maximum of 250 words, followed by 4 to 6 keywords...

Keywords: keyword1, keyword2,...

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

α	Angle of attack.
β	Angle of side-slip.
κ	Thermal conductivity coefficient.
μ	Molecular viscosity coefficient.
ρ	Density.

Roman symbols

C_D	Coefficient of drag.
C_L	Coefficient of lift.
C_M	Coefficient of moment.
p	Pressure.
$\mathbf{u}$	Velocity vector.
u, v, w	Velocity Cartesian components.

Subscripts

∞	Free-stream condition.
i, j, k	Computational indexes.
n	Normal component.
x, y, z	Cartesian components.
ref	Reference condition.

Superscripts

*	Adjoint.
T	Transpose.

Glossary

CFD Computational Fluid Dynamics is a branch of fluid mechanics that uses numerical methods and algorithms to solve problems that involve fluid flows.

CSM Computational Structural Mechanics is a branch of structure mechanics that uses numerical methods and algorithms to perform the analysis of structures and its components.

MDO Multi-Disciplinar Optimization is an engineering technique that uses optimization methods to solve design problems incorporating two or more disciplines.

Chapter 1

Introduction

1.1 Mathematical Finance

Mathematical finance, also known as quantitative finance, is a field of applied mathematics focused on the modeling of financial instruments. It is rather difficult to overestimate its importance since it is heavily used by investors and investment banks in everyday transactions. In recent decades, this field suffered a complete paradigm shift, following developments in computer science and new theoretical results that enabled investors to better understand the mechanics of financial markets.

With the colossal sums traded daily in financial markets around the world, mathematical finance has become increasingly important and many resources are invested in the research and development of new and better theories and algorithms.

1.2 Derivatives

Derivatives are currently one of the subjects most studied by financial mathematicians. In finance, a *derivative* is simply a contract whose value depends on other simpler financial instruments, known as *underlying assets*, such as stock prices or interest rates. They can virtually take any form desirable, so long as there are two parties interested in signing it and all government regulations are met.

The importance of derivatives has grown greatly in recent years. In fact, as of June 2017, derivatives were responsible for over $542 trillion worth of trades, in the Over-the-Counter (OTC) market alone [1], as can be seen in Figure 1.1 (the OTC market refers to all deals signed outside of exchanges). This growth peaked in 2008 but stalled after the global financial crisis due to new government regulations, implemented because of the role of derivatives in market crashes [2]. It is easy to see that mishandling derivatives can have disastrous consequences. However, when handled appropriately, derivatives prove to be very powerful tools to investors, as we will see shortly.

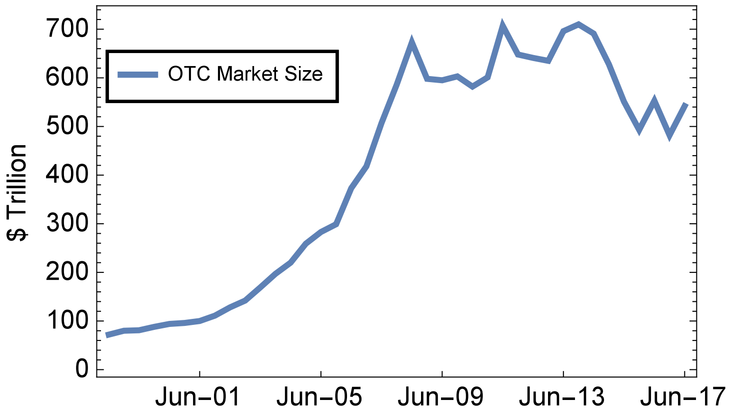


Figure 1.1: Size of OTC derivatives market since May 1996.

1.3 Options

Of all classes of derivatives, in this master thesis we will focus particularly on the most traded type [3]: *options*.

As the name implies, an *option* contract grants its buyer the *option* to buy (in the case of a *call* type option) or sell (for *put* options) its underlying asset at a future date, known as the *maturity*, for a fixed price, known as the *strike price*. In other words, when signing an option, buyers choose a price at which they want to buy/sell (call/put) some asset and a future date to do this transaction. When this date arrives, if the transaction is favorable to the buyers, they exercise their right to execute it.

The description above pertains only to the most traded type of option - *European* options. In this thesis, unless otherwise stated, all options will be assumed European. There exist, however, other option types that enable exercising at dates other than maturity. The most well known such example is *American* options, contracts that enable their buyers to exercise their right to buy/sell the underlying asset at any point in time *until* the maturity date. Other types, commonly known as *Exotic* options, will be studied in more detail in the following sections.

It's important to emphasize the fact that an option grants its buyer the right to do something. If *exercising* the option would lead to losses, the buyer can simply decide to let the maturity date pass, allowing the option to expire without further costs. This is indeed the most attractive characteristic of options.

1.3.1 Why Options are Important

Options are very useful tools to all types of investors.

To hedgers (i.e. investors that want to limit their exposure to risk), options provide safety by fixing a minimum future price on their underlying assets - e.g. if hedgers want to protect themselves against a potential future price crash affecting one of their assets, they can buy put type options on that asset. With these, even if the asset's value does crash, their losses will always be contained because they can exercise the options and sell the asset at the option's higher strike price.

Options are also very useful to speculators (i.e. investors that try to predict future market movements). The lower price of options (when compared to their underlying assets) grants this type of investors great leveraging capabilities and, with them, access to much higher profits if their predictions prove true. The opposite is also true and a wrong prediction can equally lead to much greater losses.

Due to all their advantages, and unlike some other types of derivatives, options have a price. Finding the ideal price for an option is a fundamental concern to investors, because knowing their appropriate value can give them a chance to take advantage of under or overpriced options. Finding this price can be very difficult for some option types, however, and though a lot of research has been done towards this goal, a great deal more is still required.

Chapter 2

Background

2.1 Call and Put Options

show payoff functions of other option types? As stated before, call and put options enable their buyers to respectively buy and sell the underlying stock at the maturity for the fixed strike price. In the case of a call option, if at the maturity the market price of its underlying asset is greater than the strike, investors can exercise the option and buy the asset for the fixed lower strike price. They can then immediately go to the market and sell the asset for its higher value. Thus, in this case, the payoff of the option would be the difference between the asset's price and the option's strike price. On the other hand, if at the maturity the price of the asset decreases past the strike, the investor should let the option expire, since the asset is available in the market for a lower price. In this case, the payoff of the option would be zero. The same reasoning can be made for put type options. The payoff function of these two types of options can then simply be deduced as

$$\begin{aligned} \text{Payoff}_{\text{call}}(K,T) &= \max\left(S(T)-K,0\right); \\ \text{Payoff}_{\text{put}}(K,T) &= \max\left(K-S(T),0\right), \end{aligned} \tag{2.1}$$

where K is the option's strike price and $S(T)$ is the asset's price, $S(t)$, at the maturity, T. These functions are represented in Figure 2.1. is this image really necessary?

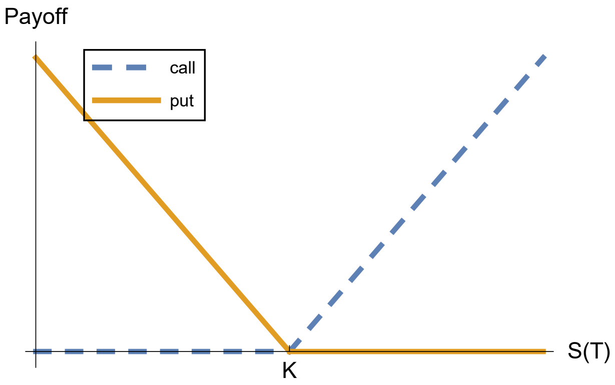


Figure 2.1: Payoff functions of *call* and *put options*.

With eqs.(2.1) in mind, it should be clear that the value of these two types of options corresponds to their expected future payoff discounted back to the present, which gives

$$\begin{aligned} C(K,T) &= e^{-rT}\mathbb{E}\left[\max\left(S(T)-K,0\right)\right] = e^{-rT}\mathbb{E}\left[\left(S(T)-K\right)\mathbb{1}_{\{S(T)>K\}}\right]; \\ P(K,T) &= e^{-rT}\mathbb{E}\left[\max\left(K-S(T),0\right)\right] = e^{-rT}\mathbb{E}\left[\left(K-S(T)\right)\mathbb{1}_{\{S(T)<K\}}\right], \end{aligned} \tag{2.2}$$

with $C(K,T)$ and $P(K,T)$ being the values of call and put options, respectively, and $\mathbb{E}[\cdot]$, $\mathbb{1}_{\{\cdot\}}$ corresponding to the expected value and indicator functions, respectively. The parameter r denotes the risk-free interest rate, which we will describe in the next subsection.

2.2 Black-Scholes-Merton Formulae

Due to their high importance, options have been studied in great detail in the past. Probably the most important result in this field came from Fischer Black, Myron Scholes and Robert Merton, who developed a mathematical model to price options - the famous Black-Scholes-Merton model [4] - still in use in present days [5].

This model states that the price of an European call or put option, whose underlying asset is a *stock*, follows the partial differential equation (PDE)

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \tag{2.3}$$

where V is the price of the option, S is the price of the underlying stock, r is the risk-free interest rate and σ is the stock price volatility.

underlying asset=stocks

The risk-free interest rate, r, is the interest an investor would receive from any risk-free investment (e.g. treasury bills). No investor should ever invest in risky products whose expected return is lower than this interest (e.g. the lottery), since there's the alternative of obtaining a higher (expected) payoff without the disadvantage of taking risks. In general, this rate changes slightly with time and is unknown, but Black *et al.*, in their original model (eq.(2.3)), assumed that it remains constant throughout the option's duration and that it is known. Some authors have suggested solutions to deal with this shortcoming, providing models to replicate the behavior of interest rates [6], but because option prices do not significantly depend on this value [5], in the remainder of this thesis we shall make the same assumptions as Black *et al.*, i.e. constant and known interest rate.

As for the stock price volatility, σ, because we will explore it deeply in the next sections, suffice it to say for now that it is a measure of the future stock price movement's uncertainty.

One important assumption of the Black-Scholes-Merton model is that stock prices follow a stochastic process, known as Geometric Brownian Motion, which can be defined as

$$dS(t) = rS(t)dt + \sigma S(t)dW(t), \tag{2.4}$$

with $\{W(t),\ t > 0\}$ defining a one-dimensional Brownian motion. put image of GBM here

With this result, pricing options is fairly straightforward - we simply need to solve the PDE in eq.(2.3) in a similar fashion to the initial value problem for the diffusion equation [7]. The results published originally by Black *et al.* state that, at time t, call and put options can be valued as

$$\begin{aligned} C(S(t),t) &= N(d_1)S(t) - N(d_2)Ke^{-r(T-t)}; \\ P(S(t),t) &= -N(-d_1)S(t) + N(-d_2)Ke^{-r(T-t)}, \end{aligned} \tag{2.5}$$

where $N(\cdot)$ is the cumulative distribution function of the standard normal distribution and where d_1, d_2 are given by

$$\begin{aligned} d_1 &= \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right]; \\ d_2 &= d_1 - \sigma\sqrt{T-t}. \end{aligned} \tag{2.6}$$

From eq.(2.5) we can derive the relationship between $C(S,t)$ and $P(S,t)$, known as the *put-call parity*

$$C(S(t),t) = S(t) - Ke^{-r(T-t)} + P(S(t),t). \tag{2.7}$$

Because of this duality, we can always obtain the prices of put options from the prices of call options with the same underlying asset, maturity and strike. For this reason, providing results for both option types is redundant and unless otherwise stated, all options will be assumed European calls in the following sections.

2.3 Volatility

Volatility is a measure of the uncertainty of future stock price movements. In other words, a high volatility will lead to great future fluctuations in the stock price, whereas a stock with low volatility is more stable.

put image of varying volatilities here

Of all the parameters in the Black-Scholes formula (eq.(2.3)), volatility is the only one we can't easily measure from market data. Furthermore, unlike the interest rate, volatility has a great impact on the behavior of stock prices and, consequently, on the price of options [5]. These two factors make volatility one of the most important subjects in all of mathematical finance and thus the focus of much research.

It should be noted that there are several types of volatility, depending on what is being measured. Some of these types will be approached in the next subsections.

2.3.1 Implied Volatility

Implied volatility can be described as the value of stock price volatility that, when input into the Black-Scholes pricer in eq.(2.5), outputs a value equal to the market price of a given option. In other words, it would be the stock price volatility that the seller/buyer of the option assumed when pricing it (given that the Black-Scholes model was used).

Because eq.(2.5) is not invertible, we need to use some numerical method (e.g. Newton's method) to find the value of implied volatility that matches the market and model prices i.e. we must find, numerically, the solution to the equation

$$C(\sigma_{imp}, S(t), t) - \overline{C} = 0, \tag{2.8}$$

where $C(\sigma_{imp}, \cdot)$ corresponds to the result of eq.(2.5) using the (implied) volatility σ_{imp} and $\overline{C}$ is the price of the option observed in the market.

Because eq.(2.5) is monotonic in the volatility, we can obtain the implied volatility of an option from its price and vice versa. This duality is so fundamental that investors often disclose options by providing their implied volatility instead of their price [8].

One interesting property of implied volatility is that, in the real-world, it depends on the strike price and the maturity, which should not occur in the "Black-Scholes world". If investors really used the this model to price their options, two options with the same underlying asset should have the same implied volatility, regardless of their strike prices or maturities (i.e. the same underlying asset doesn't have two different volatilities at the same time). However, when observing real market data, this is indeed what is observed.

The implied volatilities' dependence on the strike price is a function that can take one of two forms, known as *smile* and *skew*. An implied volatility smile shows greater values of σ_{imp} for options with strikes farther from the current stock price. The minimum occurs where the strike equals the current stock price. A skew, on the other hand, only shows greater σ_{imp} in one of the directions (i.e. for strikes either bigger or smaller than the current stock price). Both phenomena are represented in Figure 2.2.

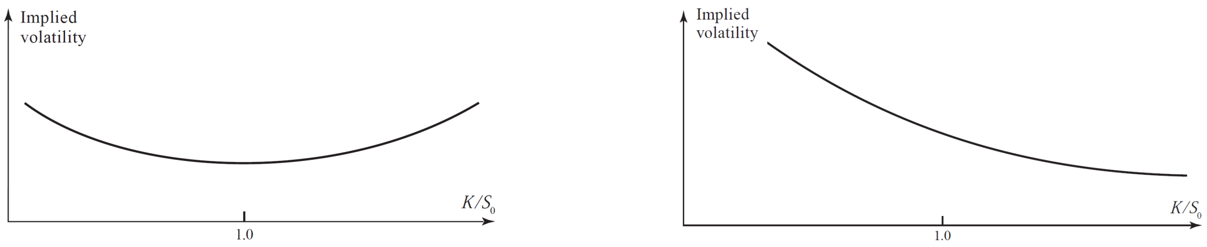
Because of their higher implied volatility, we can conclude that options with strikes different from the current stock price are overpriced. The reason behind this odd market behavior is related to the simple demand-supply rule [5]. On the one hand, some investors are risk-averse and want to hedge their losses in case of a market crash (as explained in subsection 1.3.1). They don't mind paying a higher price for an option if this means they would be relatively safe from potentially devastating market crashes. For this reason, the prices of (call) options with lower strikes increase, driving their implied volatility up. On the other hand, other investors are risk-seekers and want to take advantage of possible sudden price increases, buying the stocks for the lower strike prices. They don't mind paying higher prices for the options and this drives the prices of high strike options (and, consequently, their implied volatility) up. This fear-greed duality gives rise to the observed volatility smile.

The dependence of the implied volatility on the maturity date is more complex, but in general it decreases with T.

It can also be shown that the implied volatility is the same for calls and puts [3].

2.3.2 Local Volatility

In their original work, Black *et al.* assumed that volatility is constant throughout the whole contract. From market data, it can be clearly seen that this is not the case. There may be times where new information



(a) Smile

(b) Skew

Figure 2.2: Example of an implied volatility smile (left) and skew (right).source=Hull

reaches the market and trading increases, driving volatility up. On the other hand, if investors are waiting for some new information to reach the market (e.g. the results of an election) trading may stall, and volatilities go down.

The model in eq.(2.5) is therefore not enough to completely grasp real-world trading. We should have a model where volatility is dynamic, measuring the uncertainty on the stock price at any point in time. However, as we saw in subsection 2.3.1 for implied volatility, the market's view of volatility also changes with the strike price. A volatility model that merely depends on time is thus insufficient.

The volatility should therefore be a function of both time and strike price: $\sigma(K, t)$. We call this model *local volatility* and the geometric Brownian motion from eq. 2.4 is transformed into

$$dS(t) = rS(t)dt + \sigma(K, t)S(t)dW(t), \tag{2.9}$$

where $\sigma(K, t)$ is some function of K and t.

Finding the local volatility function is not very important when pricing typical European options, because we can simply find the implied volatility from similar options in the market and assume a constant volatility throughout the option's duration. However, for other contracts, such as American, Asian or Barrier options (among others) where the option's value depends on the intermediate stock prices, this function is indeed very useful.

Because we can't directly measure the local volatility of a stock from market data, we need some models to find it. The best known of these is Dupire's formula.

Dupire's Formula

notation problems... T=T', K=K' One of the most famous results in the modelling of the local volatility function was obtained by Dupire [9]. In his article, the author derives a theoretical formula for $\sigma(K, t)$, given by

$$\sigma(K, T) = \sqrt{\frac{\dfrac{\partial C}{\partial T} + rK\dfrac{\partial C}{\partial K}}{\dfrac{1}{2}K^2\dfrac{\partial^2 C}{\partial K^2}}}, \tag{2.10}$$

where $\sigma(S, t)$ is the local volatility function for stock prices S at time t and $C = C(K, T)$ is the price of an European call option with strike price K and maturity T. A brief demonstration of this formula can be found in Appendix A.

As can be seen, we need to differentiate the option prices with respect to their strikes and maturities. To achieve this, we need to gather from the market a large number of prices for options with different maturities and strikes, and perform some interpolation on them to obtain an option price surface (with K and T as variables). We then calculate the gradients of this interpolated surface and input them into eq. 2.10 to obtain the local volatility surface.

Even before implementation, four potential sources of error can be found:

- First, it should be noted that markets only trade options with very specific maturities (e.g. 1, 2, 4 and 6-months maturity). For this reason, our data will be very sparse with respect to maturity and the interpolation obtained may not perform as required.

- Furthermore, it can be shown that, for options with strikes very different from the current stock price, the option's price is approximately linear with respect to the strike. The second derivative in these regions would therefore be very close or even equal to zero. Because this second derivative is in the denominator of eq.(2.10), our volatilities may explode for very large or very small strikes.

- There is also the problem of noise. Because we are interpolating very sparse data, even small fluctuations in the option market price will cause variations in the option price interpolation. This can be very problematic in regions where the second derivative is small, because the result is very sensitive to this value.

- Finally, some problems arise from the market itself. While most investors use some theoretical basis in their trades, the market is still governed by the demand-supply rule. If too many investors want to buy the option and few want to sell it, the option price will increase, even if it means that the option will be overpriced, and vice versa. It may also happen that the market is not liquid enough (i.e. very few trades or even no trades at all occur for some options with very large maturities or very large/small strikes or for some options on relatively unknown stocks) causing the option prices to not truly follow the market's perception of future price movements.

All these problems must be taken into account when applying Dupire's model.

Besides, it can be shown that if the local volatility surface truly matched reality, it should remain unchanged (i.e. the local volatility surface measured today and in one month's time should, in theory, be the same). However, by studying market data, we can see that this is really not the case [5]. We can therefore conclude that the model doesn't completely correspond to reality and for that reason it shouldn't be used blindly.

Some authors have also pointed out that the volatility smile obtained from Dupire's local-volatility model doesn't follow real market dynamics [10]. They demonstrate that when the price of the stock either increases or decreases, the volatility smile predicted by the model shifts in the opposite direction. The minimum of the volatility smile would therefore be offset and no longer correspond to the local stock price. The volatility smile dynamics obtained from the local-volatility model would then be actually even worse than if we assumed a constant volatility.

Dupire also developed an alternative local-volatility formula based on the implied volatility surface rather than the option price's, as seen in eq.2.10. The relation obtained is as follows

$$\sigma(K,T) = \sqrt{\frac{\sigma_{imp}^2 + 2(T-t)\sigma_{imp}\frac{\partial \sigma_{imp}}{\partial T} + 2rK(T-t)\sigma_{imp}\frac{\partial \sigma_{imp}}{\partial K}}{\left(1 + Kd_1\sqrt{T-t}\frac{\partial \sigma_{imp}}{\partial K}\right)^2 + K^2(T-t)\sigma_{imp}\left(\frac{\partial^2 \sigma_{imp}}{\partial K^2} - d_1\left(\frac{\partial \sigma_{imp}}{\partial K}\right)^2\sqrt{T-t}\right)}}, \tag{2.11}$$

where d_1 is given by

$$d_1 = \frac{\log(S(t)/K) + \left(r + \frac{1}{2}\sigma_{imp}^2\right)(T-t)}{\sigma_{imp}\sqrt{T-t}}, \tag{2.12}$$

with t representing the current time (usually $t = 0$), T being the time at which the local volatility is being measured, and $S(t)$ the stock price at time t. We assume that $\sigma_{imp} = \sigma_{imp}(K,T)$ is the interpolated surface of the implied volatilities evaluated at time T, and price K. This relation might be more useful than eq.(2.10) because the interpolated implied volatility surface might be smoother than the option price's. We will compare the results from both formulas in the next chapters.

Despite all of its shortcomings, Dupire's formula is nonetheless very much used by traders to find the local volatility surfaces of assets and with them price exotic options afterwards.

2.3.3 Stochastic Volatility

As stated before, the volatility is not constant, is not observable and is not predictable, despite our attempts to model it. This seems to indicate that volatility is itself a stochastic process. Some research has been done into this hypothesis, and many models have been developed to replicate real-world volatilities.

As before, we assume that the stock price follows a geometric Brownian motion

$$dS(t) = rS(t)dt + \sigma(t)S(t)dW_1(t), \tag{2.13}$$

but we further hypothesize that the volatility follows

$$d\sigma(t) = p(S,\sigma,t)dt + q(S,\sigma,t)dW_2(t), \tag{2.14}$$

where $p(S,\sigma,t)$ and $q(S,\sigma,t)$ are some functions of the stock price S, time t and of the volatility σ itself. We also assume that W_1 and W_2 are two Brownian motion processes with a correlation of ρ, which gives

$$dW_1 dW_2 = \rho dt. \tag{2.15}$$

This correlation factor ρ can be explained by the relationship between prices and volatilities. Usually, when prices decrease, trade goes up and with it, rises the volatility. The inverse is true when prices increase. This seems to indicate a negative correlation between stock prices and volatilities, but positive correlations are also possible.

Choosing the right functions $p(S,\sigma,t)$ and $q(S,\sigma,t)$ is very important since the whole evolution of the stock price depends on them. All stochastic volatility models present a different version of these functions, and each may be more adequate for some types of assets. Furthermore, these functions have some parameters that we have to calibrate in order to best fit our model to market data.

We now present two of the most used stochastic volatility models - *Heston* and *SABR*.

Heston Model

One of the most popular stochastic volatility models is known as *Heston model*. It was developed in 1993 by Steven Heston [11] and it states that stock prices satisfy the relations

$$dS = rSdt + \sqrt{\nu}SdW_1, \tag{2.16}$$

$$d\nu = \kappa(\overline{\nu} - \nu)dt + \eta\sqrt{\nu}dW_2, \tag{2.17}$$

with ν corresponding to the stock price variance (i.e. $\nu = \sigma^2$) and where W_1 and W_2 have a correlation of ρ. The original model used a drift parameter μ instead of the usual risk-free measure drift r, but a measure transformation, using Gisarnov's theorem, can be easily implemented [12].

The parameters κ, $\overline{\nu}$ and η are, respectively, the mean-reversion rate (i.e. how fast the volatility converges to its mean value), the long-term variance (i.e. the mean value of variance) and the volatility of volatility (i.e. how erratic is the volatility evolution).

Feller condition

One of the main advantages that make the Heston model so popular is that there is a closed-form solution for the prices of options priced under this model, which is given by

$$\begin{aligned} C_H(\theta;K,T) &= e^{-rT}\mathbb{E}\left[\left(S(T)-K\right)\mathbb{1}_{\{S(T)>K\}}\right] \\ &= e^{-rT}\left(\mathbb{E}\left[S(T)\mathbb{1}_{\{S(T)>K\}}\right] - K\mathbb{E}\left[\mathbb{1}_{\{S(T)>K\}}\right]\right) \\ &= S(0)P_1 - e^{-rT}KP_2, \end{aligned} \tag{2.18}$$

where $C_H(\theta;K,T)$ corresponds to the theoretical European call option price under the Heston model, assuming a set of parameters θ, strike K and maturity T. The variables P_1 and P_2 are given by

$$P_1 = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathrm{Re}\left(\frac{e^{-iu\log K}}{iuf}\phi(\theta;u-i,T)\right)du, \tag{2.19}$$

$$P_2 = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathrm{Re}\left(\frac{e^{-iu\log K}}{iu}\phi(\theta;u,T)\right)du, \tag{2.20}$$

where i is the imaginary unit, f is the forward price (i.e. $f = S(0)e^{rT}$) and $\phi(\theta;u,t)$ is the characteristic function of the logarithm of the stock price process. The characteristic function of a random variable is the Fourier transform of the probability density function of that variable.

Now we just have to find the appropriate characteristic function $\phi(\theta;u,t)$ in order to evaluate the integrals in eqs.(2.19) and (2.20) and find the option price with eq.(2.18). In the original article, Heston derived this very characteristic function [11] but it presented some discontinuities for large maturities and wasn't easily derivable. These shortcomings led some other authors to propose several modified versions of this function. Most recently, Cui *et al.* [13] presented a characteristic function that not only doesn't have the previously mentioned discontinuities but is also easily derivable, given by

$$\phi(\theta;u,t)=\exp\left\{iu\left(\log S(0)+rt\right)-\frac{t\kappa\overline{\nu}\rho iu}{\eta}-\nu_0 A+\frac{2\kappa\overline{\nu}}{\eta^2}D\right\}, \tag{2.21}$$

with A and D given by

$$A=\frac{A_1}{A_2}, \tag{2.22}$$

$$D=\log d+\frac{\kappa t}{2}-\log A_2, \tag{2.23}$$

where we introduce the variables A_1, A_2, d and ξ

$$A_1=(u^2+iu)\sinh\frac{dt}{2}, \tag{2.24}$$

$$A_2=e^{dt/2}\left(\frac{d+\xi}{2}+\frac{d-\xi}{2}e^{-dt}\right), \tag{2.25}$$

$$d=\sqrt{\xi^2+\eta^2(u^2+iu)}, \tag{2.26}$$

$$\xi=\kappa-\eta\rho iu. \tag{2.27}$$

With this result, the calibration of the Heston model can be easily processed. We simply have to find the model parameters θ that minimize the difference between the model generated option prices, $C_H(\theta;K,T)$, and the market prices, $\overline{C}$. This calibration will be explored in detail in the next sections.

SABR Model

One other very famous model for stochastic volatility was developed by Hagan *et al.* [10] and is known as *SABR*. It stands for "*stochastic-*$\alpha\beta\rho$" and in this model it is assumed that the option prices and volatilities follow

$$dF=\sigma F^{\beta}dW_1, \tag{2.28}$$

$$d\sigma=\nu\sigma dW_2, \tag{2.29}$$

with F being the forward price (related to the spot price S by $F(t)=S(t)e^{r(T-t)}$). We further define $\alpha=\sigma(0)$ as the starting volatility, $f=F(0)=S(0)e^{rT}$ as the starting forward price. Finally, as before, the two Brownian motion processes W_1 and W_2 have a correlation of ρ.

SABR is not mean-reverting

In the original article, the authors claim that β can be fitted from historical market data, but usually investors choose this value arbitrarily, depending on the type of assets traded. Typical values used are $\beta=1$ (stochastic lognormal model), usually for foreign exchange options, $\beta=0$ (stochastic normal model), typical for interest rate options where forwards $F(t)$ can be negative, and $\beta=0.5$ (stochastic

CIR model), also common for interest rate options.

One of the main reasons why SABR is so popular is due to the quasi-closed-form solutions that greatly simplify its calibration. With the corrections done by Oblój [14] on Hagan's original formula, it can be shown that the implied volatilities of options priced under the SABR model are given by

$$\begin{aligned}\sigma_{SABR}(K,f,T) \approx & \frac{1}{\left[1+\frac{(1-\beta)^2}{24}\log^2\left(\frac{f}{K}\right)+\frac{(1-\beta)^4}{1920}\log^4\left(\frac{f}{K}\right)\right]}\cdot\left(\frac{\nu\log\left(f/K\right)}{x(z)}\right) \\ & \cdot\left\{1+T\left[\frac{(1-\beta)^2}{24}\frac{\alpha^2}{(Kf)^{1-\beta}}+\frac{1}{4}\frac{\rho\beta\nu\alpha}{(Kf)^{(1-\beta)/2}}+\frac{2-3\rho^2}{24}\nu^2\right]\right\},\end{aligned} \tag{2.30}$$

with z and $x(z)$ given by

$$z=\frac{\nu\left(f^{1-\beta}-K^{1-\beta}\right)}{\alpha(1-\beta)}, \tag{2.31}$$

$$x(z)=\log\left\{\frac{\sqrt{1-2\rho z+z^2}+z-\rho}{1-\rho}\right\}. \tag{2.32}$$

With this result, calibrating the SABR model is fairly straightforward. We simply need to find the parameters that minimize the difference between the implied volatilities obtained from the model and those obtained from market data.

One of the main setbacks of the SABR model is the fact that this calibration only works for a set of options with the same maturity. The model behaves badly when we try to fit options with different maturities [10]. To solve this problem, Hagan *et al.* suggest a similar model known as *Dynamic SABR* [10]. It follows the same processes presented in eqs.(2.28) and (2.29) but with time-dependent parameters $\rho(t)$ and $\nu(t)$. These processes become

$$dF=\sigma F^{\beta}dW_1, \tag{2.33}$$

$$d\sigma=\nu(t)\sigma dW_2, \tag{2.34}$$

with the correlation between W_1 and W_2 now given by $\rho(t)$.

To calibrate this new model, Hagan *et al.* derived again a quasi-closed-form solution for the implied volatility. Osajima later simplified this expression with asymptotic expansions [15]. The resulting formula is given by

$$\sigma_{DynSABR}(K,f,T)=\frac{1}{\omega}\left(1+A_1(T)\log\left(\frac{K}{f}\right)+A_2(T)\log^2\left(\frac{K}{f}\right)+B(T)T\right), \tag{2.35}$$

where $\omega=f^{1-\beta}/\alpha$ and $A_1(T)$, $A_2(T)$ and $B(T)$ are given by

$$A_1(T)=\frac{\beta-1}{2}+\frac{\eta_1(T)\omega}{2}, \tag{2.36}$$

$$A_2(T)=\frac{(1-\beta)^2}{12}+\frac{1-\beta-\eta_1(T)\omega}{4}+\frac{4\nu_1^2(T)+3(\eta_2^2(T)-3\eta_1^2(T))}{24}\omega^2, \tag{2.37}$$

$$B(T)=\frac{1}{\omega^2}\left(\frac{(1-\beta)^2}{24}+\frac{\omega\beta\eta_1(T)}{4}+\frac{2\nu_2^2(T)-3\eta_2^2(T)}{24}\omega^2\right), \tag{2.38}$$

with $\nu_1^2(T)$, $\nu_2^2(T)$, $\eta_1(T)$ and $\eta_2^2(T)$ defined as

$$\nu_1^2(T) = \frac{3}{T^3}\int_0^T (T-t)^2\nu^2(t)dt, \tag{2.39}$$

$$\nu_2^2(T) = \frac{6}{T^3}\int_0^T (T-t)t\nu^2(t)dt, \tag{2.40}$$

$$\eta_1(T) = \frac{2}{T^2}\int_0^T (T-t)\nu(t)\rho(t)dt, \tag{2.41}$$

$$\eta_2^2(T) = \frac{12}{T^4}\int_0^T \int_0^t \left(\int_0^s \nu(u)\rho(u)du\right)^2 dsdt, \tag{2.42}$$

where $\rho(t)$ and $\nu(t)$ are the functions chosen to model the time dependent parameters. It might happen that, for some chosen functions, one or more of the integrals in eqs.(2.39)-(2.42) is not analytically solvable. In these cases, the integral would have to be evaluated numerically. On the other hand, if the integral does have an analytical solution, the calibration of the Dynamic SABR is greatly simplified.

One classical choice for the functions [16] corresponds to

$$\rho(t) = \rho_0 e^{-at}, \tag{2.43}$$

$$\nu(t) = \nu_0 e^{-bt}, \tag{2.44}$$

with $\rho_0 \in [-1, 1]$, $\nu_0 > 0$, $a > 0$ and $b > 0$. In this particular case, $\nu_1^2(T)$, $\nu_2^2(T)$, $\eta_1(T)$ and $\eta_2^2(T)$ can be exactly derived as

$$\nu_1^2(T) = \frac{6\nu_0^2}{(2bT)^3}\left[\left(\frac{(2bT)^2}{2} - 2bT + 1\right) - e^{-2bT}\right], \tag{2.45}$$

$$\nu_2^2(T) = \frac{12\nu_0^2}{(2bT)^3}\left[e^{-2bT}(1+bT) + bT - 1\right], \tag{2.46}$$

$$\eta_1(T) = \frac{2\nu_0\rho_0}{T^2(a+b)^2}\left[(a+b)T + e^{-(a+b)T} - 1\right], \tag{2.47}$$

$$\eta_2^2(T) = \frac{3\nu_0^2\rho_0^2}{T^4(a+b)^4}\left[e^{-2(a+b)T} - 8e^{-(a+b)T} + (7 + 2T(a+b)(-3+(a+b)T))\right]. \tag{2.48}$$

Because we have more parameters to fit, the calibration of the Dynamic SABR will be slower than the original SABR. In the next sections we shall compare both models.

Chapter 3

Implementation

3.1 Option Pricing

The theoretical models presented in Chapter 2 attempt to model the movements of real-world stock prices. With these predictions, we should be able to better replicate real option prices than if we assumed a simple constant volatility.

Currently, the two most used methods to computationally price options are known as *finite differences* [3] and *Monte Carlo* [17].

Finite differences is an extremely fast procedure when used to price either European or American-type options, making it very appealing in these circumstances. However, when used to price other option types whose value depends on the stock prices until maturity (e.g. Asian options), the algorithm becomes very slow, rendering it almost useless. The implementation of both Heston and SABR models (presented before) using finite differences can be found in deGraaf [18].

With the Monte Carlo algorithm, we begin by simulating a very large number of stock price paths (e.g. 100,000 simulations). The option's payoff is then calculated for each of these simulated paths and averaged, providing a fair estimate of the option's value. This algorithm can also be easily adapted to price exotic options, making it very attractive in such cases. In the past, simulating all the stock price paths took prohibitively long computation times and this method was often discarded for this reason. However, with the recent advancements in computer hardware and new algorithmic developments, such as GPU implementation, this method has become quite popular. For these reasons, the Monte Carlo method will be used for the analysis of the models presented before.

3.1.1 Simulating stock prices

As stated, to implement the Monte Carlo algorithm, one needs to simulate stock price paths. However, by analyzing eq.(2.4), we can see that the stock prices depend on a Brownian motion process (also known as a Wiener process) which, due to its self-similarity, is not differentiable [19]. It follows that stock price paths can never be exactly simulated. Though this exact simulation is impossible, we can approximate the movement of stock price paths by discretizing the Brownian motion process in time, thus solving its self-similarity problem. The two most common discretization procedures are presented below.

Euler–Maruyama discretization

put this in background section? One of the most well known discretization methods is known as *Euler–Maruyama discretization*, and can be applied to stochastic differential equations of the type

$$dX(t) = a(X(t))dt + b(X(t))dW(t), \tag{3.1}$$

where $a(X(t))$ and $b(X(t))$ are some given functions of $X(t)$ and $\{W(t),\ t > 0\}$ defines a one-dimensional Brownian motion process. To apply this discretization, we begin by partitioning the simulation interval $[0, T]$ into N subintervals of width $\Delta t = T/N$ and then recursively define

$$X_{n+1} = X_n + a(X_n)\Delta t + b(X_n)\Delta W_n, \tag{3.2}$$

for $n = 1, \ldots, N$ where $\Delta W_n = W_{t+\Delta t} - W_t$. Using the known properties of Brownian motion processes, we can produce $\Delta W_n \sim \sqrt{\Delta t} Z$, where $Z \sim N(0, 1)$ defines a standard normal distribution.

Applying this discretization to the Geometric Brownian motion followed by stock price paths, as seen in eq.(2.4), we arrive at

$$S(t + \Delta t) = S(t) + rS(t)\Delta t + \sigma(S(t), t)S(t)\sqrt{\Delta t} Z. \tag{3.3}$$

Due to its simplicity, the Euler–Maruyama discretization method is the most common in the simulation of stock price paths.

Milstein Discretization

For stochastic volatility models, such as Heston and SABR, where the volatility itself follows a stochastic differential equation, such as eq.(3.1), the Euler–Maruyama discretization may not be sufficiently accurate. In these cases, we can apply the more precise Milstein method [20], defined as

$$X_{n+1} = X_n + a(X_n)\Delta t + b(X_n)\Delta W_n + \frac{1}{2} b(X_n)b'(X_n)((\Delta W_n)^2 - \Delta t), \tag{3.4}$$

where $b'(X_n)$ denotes the derivative of $b(X_n)$ w.r.t. X_n. Note that when $b'(X_n) = 0$, the Milstein method collapses to the simpler Euler–Maruyama discretization.

Applying this discretization to the Heston model, we arrive at

$$S(t + \Delta t) = S(t) + rS(t)\Delta t + S(t)\sqrt{\nu(t)}\sqrt{\Delta t} Z_1 + \frac{1}{2}\nu(t)S(t)\Delta t(Z_1^2 - 1), \tag{3.5}$$

$$\nu(t + \Delta t) = \nu(t) + \kappa(\overline{\nu} - \nu(t))\Delta t + \eta\sqrt{\nu(t)\Delta t} Z_2 + \frac{\eta^2}{4}\Delta t(Z_2^2 - 1), \tag{3.6}$$

where Z_1 and Z_2 are two normal random variables with a correlation of ρ.

Applying the Milstein discretization to the SABR model results in

$$F(t+\Delta t) = F(t) + \sigma(t)F^{\beta}(t)\sqrt{\Delta t}Z_1 + \frac{\beta}{2}\sigma^2(t)F^{2\beta-1}(t)\Delta t(Z_1^2 - 1), \tag{3.7}$$

$$\sigma(t+\Delta t) = \sigma(t) + \nu\sigma(t)\sqrt{\Delta t}Z_2 + \frac{\nu^2}{2}\sigma(t)\Delta t(Z_2^2 - 1), \tag{3.8}$$

where again Z_1 and Z_2 are two normal random variables with a correlation of ρ.

In both models we need to generate the two correlated normal variables, Z_1 and Z_2, which we can generate from

$$\begin{aligned} &Z_1 \sim N(0,1); \\ &Z_2 = \rho Z_1 + \sqrt{1-\rho^2}Y, \end{aligned} \tag{3.9}$$

where $Y \sim N(0,1)$ is uncorrelated with Z_1.

Because it is more precise, the Milstein method will be used in the implementation of both Heston and SABR stochastic volatility models. The simpler Euler–Maruyama discretization will be assumed for both constant and Dupire's local volatility.

3.1.2 Pricing options from simulations

this should come before the discretization methods? To price options, we generate M paths by recursively calculating $\{S_i(t),\ \ i = 1,\dots,M\}$ (or $F_i(t)$ in the case of SABR), using either of the discretization methods presented before.

When the stock price at the maturity, $S_i(T)$ (or $F_i(T) = S_i(T)$), is obtained, the option's payoff for each path is calculated from eq.(2.1). We then average all these results and discount them to the present, obtaining the (call) option's value

$$C(K,T) = e^{-rT}\frac{1}{M}\sum_{i=1}^{M}\max\left(S_i(T) - K, 0\right). \tag{3.10}$$

It is important to note that, the smaller our subintervals Δt are, the better is the approximation done when discretizing the Brownian motion process. However, by decreasing Δt we increase the number of intervals and with it the number of calculations needed to obtain each $S_i(T)$. The compromise between computation time and precision must be handled appropriately. put some image here to exemplify the different time steps dt

3.2 Model Calibration

Both SABR and Heston stochastic volatility models contain variables that need to be calibrated in order to appropriately replicate market option prices.

Calibrating the models' parameters means finding the optimal values for these parameters such that the difference between the prices of market options and options priced under the models' assumptions is minimized. This difference should be measured with a cost function such as

$$\mathrm{Cost}(\theta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{C_{\mathrm{market}}(T_i, K_j) - C_{\mathrm{model}}(\theta, T_i, K_j)}{C_{\mathrm{market}}(T_i, K_j)}\right)^2, \tag{3.11}$$

where we denote θ as the model's parameter set and $C_{\mathrm{model}}(\cdot)$ and $C_{\mathrm{market}}(\cdot)$ correspond to the model and market option prices, respectively, for maturities $T_i, (i = 1, \ldots, n)$ and strikes $K_j, (j = 1, \ldots, m)$.

To obtain the value of the cost function for a given set of parameters, we need to calculate $m \times n$ model option prices. We could achieve this by applying the Monte Carlo method along with the discretization procedures described before. Because we want to calibrate the model's parameters, a large number of instances of the cost function will have to be called for our optimization algorithms to converge to an optimal solution. Thus, it can be seen that a very great number of Monte Carlo pricers will have to be executed. Even with GPU implementation and expensive hardware, using Monte Carlo to calibrate the model's parameters will become prohibitively slow. Furthermore, the Monte Carlo method is very noisy (i.e. two Monte Carlo pricers with the same initial conditions will produce slightly different outputs), making the whole convergence even more difficult. An algorithm that takes several hours to converge does not meet market demands.

replace ”Monte Carlo” by ”MC”?

The main reason why Heston and SABR models are so popular is the fact that both of them have closed-form solutions, shown in eqs. (2.18), (2.30) and (2.35), that we can use to directly price the options under each model's assumptions, without the need to run the slow Monte Carlo pricer. The optimization algorithms should then converge much faster to the optimal solution for the model's parameters.

3.2.1 Optimization Algorithms

There are several possible algorithms to find the minimum value for the cost function shown in eq.(3.11). Our main concern when choosing the best algorithms for calibration is the nonlinearity of the cost function. This is problematic because several local minima might exist and an unsuitable algorithm may get stuck in these points, causing the globally optimal solution to not be found.

With this issue in mind, we selected two powerful algorithms known as *Multi-Start* [21] and *CMA-ES* [22] (short for Covariance Matrix Adaptation Evolution Strategy), which we will summarize below. It should be noted that we will only provide a general idea of how each optimizer works. For detailed descriptions, the original sources should be consulted.

how to deal with parameter boundaries? (All these algorithms enable the use of bounds which are required for our model parameters (e.g. the correlation parameter, ρ, in both Heston and SABR is obviously contained between -1 and 1). Furthermore, they all require an initial guess at the values of the parameters, θ_0, from which they will try to converge.)

Multi-Start Optimizer

The Multi-Start optimizer is actually a combination of multiple simple (and fast) optimizers.

The algorithm starts by generating a set of N different starting points. These points can be generated at random (i.e. by sampling from a uniform distribution) or using some complex meta-heuristic such as scatter search [21].

we will implement randomized sampling

It then applies a weak optimizer to each of these starting points, finding a (local) minimum for all of them. Examples of such simple optimizers are the *Active Set Method*, *Sequential Quadratic Programming*, among others. These optimizers are weak because they are only expected to converge to the (local) optimum closest to their starting point. They are, however, very fast to compute.

After a local minimum is found for each of the selected starting points, the minimum where the cost function is minimized is chosen as optimal solution.

This procedure is depicted in Algorithm 1.

Algorithm 1: Multi-Start Optimizer

```
Generate $\theta_{0,i}, \quad i = 1, \ldots, N$                /* Multiple starting points */
for $i = 1, \ldots, N$ do
    Run weak optimization algorithm with starting point $\theta_{0,i}$
    Calculate $\mathrm{Cost}(\theta'_i)$ for the minimum found, $\theta'_i$
end
Optimal parameters: $\theta^* = \arg\min_{\theta'_i} \{\mathrm{Cost}(\theta'_i)\}$
```

One of the advantages of the Multi-Start is the fact that, because the weak algorithms are independent of one another (assuming no meta-heuristics are used), we can run them in parallel, further increasing calibration speed. As a disadvantage, we can point out the fact that, for highly nonlinear functions, a large number of starting points may be required, decreasing the calibration speed.

As a sidenote, we should point out that, for a large enough starting sample set, N, the global minimum will be found with probability 1, even for highly nonlinear objective functions. Though the proof is trivial, this remark is important, because we need a compromise between a large starting set that will take too long to compute and a small data set that might become stuck at some local minimum.

this function is in MATLAB

CMA-ES Optimizer

The CMA-ES optimizer belongs to the class of evolutionary algorithms. These methods are based on the principle of biological evolution: at each iteration (generation), new candidate solutions (individuals) are generated from a given random distribution (mutation) obtained using the data (genes) of the previous solutions (parents). Of these newly generated solutions (individuals), we select the ones where the cost function is minimized (with the best fitness) to generate the candidate solutions of the next iterations (to become the parents of the next generation) and reject the others.

As for the CMA-ES in particular, the algorithm takes λ samples from a multivariate normal distribution in the sample space

$$N(\mathbf{x};\mathbf{m},\mathbf{C}) = \frac{1}{\sqrt{(2\pi)^D|\det\mathbf{C}|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{m})\right), \tag{3.12}$$

where $\mathbf{m}$ and $\mathbf{C}$ correspond to the distribution's mean vector and covariance matrix and D is the number of dimensions of the distribution (i.e. number of parameters). These λ samples will be our candidate solutions.

We classify each of these points according to their fitness (i.e. the cost function's value for a given point). We then select the μ samples with the lowest cost, discarding the others. These new points will be the parents of the next generation.

The number of candidate solutions generated at each step, λ, and the ones that remain after classification, μ, can be chosen at will, but an appropriate heuristic is to choose $\lambda = \lfloor 4 + 3\log D\rfloor + 1$ and $\mu = \lfloor \lambda/2\rfloor + 1$.

With these μ points, we generate a new mean and covariance matrix for the multivariate normal distribution. We then repeat this process until a stopping criterion is met, such as a given number of iterations are executed or the error decreases past some value.

At each iteration, the new mean is produced from a weighted average of the points, with the weights proportional to each point's fitness. The method for the covariance matrix generation is rather complex and depends not only on the μ best samples but also on the values of previous covariance matrices. All the basic equations required for the implementation of this optimizer can be found in Appendix B. For a more detailed explanation, as well as other aspects of the algorithm, see Hansen [23].

This method is summarized in Algorithm 2.

Algorithm 2: CMA-ES Optimizer

```
Define mean vector $\mathbf{m} = \theta_0$                    /* Initial guess */
Define covariance matrix $\mathbf{C} = I$
while Termination criterion not met do
    Sample $\lambda$ points from multivariate normal distribution $N(\mathbf{x};\mathbf{m},\mathbf{C})$
    Calculate the cost for all generated points and keep the $\mu$ best, discarding the rest
    Update the mean vector and covariance matrix with eqs.(B.5) and (B.9)
end
Optimal parameters: $\theta^* = \mathbf{m}$
```

The complexity of the covariance matrix adaptation process makes CMA-ES a very robust optimization algorithm, enabling it to find the global optimum of highly nonlinear functions [24]. Furthermore, the CMA-ES is almost parameter free. It simply requires an initial guess, to generate the starting mean vector, and the algorithm should converge. As for disadvantages, there may be cases where the convergence of this algorithm is slow, and the Multi-Start may be used as an alternative when a fast convergence is required.

Chapter 4

Results

show data and mention where it came from. confidentiality, etc

Chapter 5

Conclusions

Implement importance sampling

Implement antithetic paths

we tried several algorithms but CMA and multi-start were better

Bibliography

[1] Bank for International Settlements. Semiannual otc derivatives statistics. `http://stats.bis.org/statx/srs/table/d5.1`, November 2017.

[2] Financial Times. Otc derivatives shrink to lowest level since financial crisis. `https://www.ft.com/content/dbc08ae2-1247-11e6-91da-096d89bd2173`, May 2016.

[3] J. Hull. *Options, Futures, and Other Derivatives*. Boston: Prentice Hall, 2012.

[4] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.

[5] P. Wilmott. *Paul Wilmott on Quantitative Finance*. The Wiley Finance Series. Wiley, 2006.

[6] D. Heath, R. Jarrow, and A. Morton. Bond pricing and the term structure of interest rates: A new methodology for contingent claims valuation. *Econometrica*, 60(1):77–105, 1992.

[7] R. Dilão, J. A. de Matos, and B. Ferreira. On the value of european options on a stock paying a discrete dividend. *Journal of Modelling in Management*, 4(3):235–248, 2009.

[8] P. Wilmott. *Paul Wilmott Introduces Quantitative Finance*. The Wiley Finance Series. Wiley, 2013.

[9] B. Dupire. Pricing with a smile. *Risk Magazine*, pages 18–20, 1994.

[10] P. Hagan et al. Managing smile risk. 1:84–108, 01 2002.

[11] S. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. 6:327–43, 02 1993.

[12] R. Crisóstomo. An analysis of the heston stochastic volatility model: Implementation and calibration using matlab. 2015.

[13] Y. Cui et al. Full and fast calibration of the heston stochastic volatility model. *European Journal of Operational Research*, 263(2):625 – 638, 2017.

[14] J. Obloj. Fine-tune your smile: Correction to hagan et al. Mar 2008.

[15] Y. Osajima. The asymptotic expansion formula of implied volatility for dynamic sabr model and fx hybrid model. 2007.

[16] J. Fernández et al. Static and dynamic sabr stochastic volatility models: Calibration and option pricing using gpus. *Mathematics and Computers in Simulation*, 94:55 – 75, 2013.

[17] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2004.

[18] C. de Graaf. Finite difference methods in derivatives pricing under stochastic volatility models.

[19] T. Mikosch. *Elementary Stochastic Calculus with Finance in View*. Advanced Series on Statistical Science and Applied Probability. 1998.

[20] G. N. Milstein. Approximate integration of stochastic differential equations. *Theory of Probability and Its Applications*, 19(3):557–562, 1975.

[21] Z. Ugray et al. Scatter search and local nlp solvers: A multistart framework for global optimization. *INFORMS Journal on Computing*, 19(3):328–340, 2007.

[22] N. Hansen. The cma evolution strategy: a comparing review. In *Towards a new evolutionary computation*, pages 75–102. Springer, 2006.

[23] N. Hansen. The cma evolution strategy: A tutorial.

[24] R. Dilão and D. Muraro. A parallel multi-objective optimization algorithm for the calibration of mathematical models. *Swarm and Evolutionary Computation*, 8:13 – 25, 2013.

Appendix A

Dupire's Formula Derivation

Here is presented a brief demonstration of Dupire's formula, as shown in eq. (2.10).

In his article, Dupire begins by assuming that the stock price S follows a dynamic transition probability density function $p(S(t), t, S'(t'), t')$. In other words, integrating this function would result in the probability of the stock price reaching a price S' at a time t' having started at S at time t.

The present value of a call option, $C(S, t, K, T)$, can be deduced as its expected future payoff, discounted backwards in time, which results in

$$\begin{aligned} C(K,T) = e^{-r(T-t)}\mathbb{E}\left[\max\left(S'-K,0\right)\right] &= e^{-r(T-t)}\int_0^\infty \max\left(S'-K,0\right)p(S,t,S',T)dS' \\ &= e^{-r(T-t)}\int_K^\infty (S'-K)p(S,t,S',T)dS'. \end{aligned} \tag{A.1}$$

Deriving this result once with respect to the strike price K, we obtain

$$\frac{\partial C}{\partial K} = -e^{-r(T-t)}\int_K^\infty p(S,t,S',T)dS'. \tag{A.2}$$

Deriving again with respect to the same variable results in

$$\frac{\partial^2 C}{\partial K^2} = e^{-r(T-t)}p(S,t,S',T). \tag{A.3}$$

Due to its stochastic nature, the transition probability density function follows the Fokker-Planck equation, given by

$$\frac{\partial p}{\partial T} = \frac{1}{2}\sigma^2\frac{\partial^2(S^2p)}{\partial S^2} - r\frac{\partial(Sp)}{\partial S}. \tag{A.4}$$

with σ our, still unknown, function of S and t, evaluated at $t = T$.

From eq. A.1 we can easily derive

$$\frac{\partial C}{\partial T} = -rC + e^{-r(T-t)}\int_K^\infty (S'-K)\frac{\partial p}{\partial T}dS'. \tag{A.5}$$

Using eq. A.4, we can transform this relation into

$$\frac{\partial C}{\partial T} = -rC + e^{-r(T-t)} \int_K^\infty (S' - K) \left(\frac{1}{2}\sigma^2 \frac{\partial^2 (S'^2 p)}{\partial S'^2} - r\frac{\partial (S' p)}{\partial S'} \right) dS'. \tag{A.6}$$

Integrating twice by parts and collecting all terms, we get

$$\frac{\partial C}{\partial T} = \frac{1}{2}\sigma^2 K^2 \frac{\partial^2 C}{\partial K^2} - rK\frac{\partial C}{\partial K}. \tag{A.7}$$

Rearranging all terms, we are left with the Dupire's formula

$$\sigma = \sqrt{\frac{\frac{\partial C}{\partial T} + rK\frac{\partial C}{\partial K}}{\frac{1}{2}K^2 \frac{\partial^2 C}{\partial K^2}}}. \tag{A.8}$$

Appendix B

CMA-ES Algorithm Formulas

Here we present the formulas required for the calculation of the mean vector, $\mathbf{m}$, and the covariance matrix, $\mathbf{C}$, to be used, at each iteration of the CMA-ES optimization algorithm, on the multivariate normal distribution

$$N(\mathbf{x};\mathbf{m},\mathbf{C}) = \frac{1}{\sqrt{(2\pi)^D|\mathrm{det}\mathbf{C}|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{m})\right). \tag{B.1}$$

B.1 The Optimization Algorithm

B.1.1 Initialization

We initialize the algorithm by setting the first mean vector, $\mathbf{m}^{(0)}$, to some initial guess and the covariance matrix to the unit matrix, $\mathbf{C}^{(0)} = \mathbf{I}$.

B.1.2 Sampling

We sample λ points, $\mathbf{y}_i^{(1)}$, $i = 1, \ldots, \lambda$, from the multivariate normal distribution $N(\mathbf{x};\mathbf{0},\mathbf{C}^{(0)})$, generating the first candidate solutions

$$\mathbf{x}_i^{(1)} = \mathbf{m}^{(0)} + \sigma^{(0)}\mathbf{y}_i^{(1)}, \qquad i = 1, \ldots, \lambda, \tag{B.2}$$

where $\sigma^{(0)} = 1$.

B.1.3 Classification

The candidate solutions are ordered based on their cost function, such that we denote $\mathbf{x}_{i:\lambda}^{(1)}$ as the i-th best classified point from the set $\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_\lambda^{(1)}$. In other words, $f(\mathbf{x}_{1:\lambda}^{(1)}) \leq f(\mathbf{x}_{2:\lambda}^{(1)}) \leq \ldots \leq f(\mathbf{x}_{\lambda:\lambda}^{(1)})$ (for simplicity, we refer here to the cost function as $f(\cdot)$).

B.1.4 Selection

From the ordered set $\mathbf{x}_{i:\lambda}^{(1)}$ we choose the first μ data points (with the lowest cost) and discard the others. We then define the weights ω_i as

$$\omega_i = \frac{(\log(\mu + 1/2) - \log(i))}{\sum_{i=1}^{\mu} (\log(\mu + 1/2) - \log(i))}, \qquad i = 1, \ldots, \mu. \tag{B.3}$$

As an alternative we can also use $\omega_i = 1/\mu$.

B.1.5 Adaptation

We are finally able to calculate the new mean vector and covariance matrix using

$$\left\langle \mathbf{y}^{(k)} \right\rangle_w = \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda}^{(k)}, \tag{B.4}$$

$$\mathbf{m}^{(k)} = \mathbf{m}^{(k-1)} + \sigma^{(k-1)} \left\langle \mathbf{y}^{(k)} \right\rangle_w = \sum_{i=1}^{\mu} \omega_i \mathbf{x}_{i:\lambda}^{(k)}, \tag{B.5}$$

$$\mathbf{p}_\sigma^{(k)} = (1 - c_\sigma)\mathbf{p}_\sigma^{(k-1)} + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}} \left(\mathbf{C}^{(k-1)}\right)^{-1/2} \left\langle \mathbf{y}^{(k)} \right\rangle_w, \tag{B.6}$$

$$\sigma^{(k)} = \sigma^{(k-1)} \text{exp} \left(\frac{c_\sigma}{d_\sigma} \left(\frac{\|\mathbf{p}_\sigma^{(k)}\|}{E^*} - 1 \right) \right), \tag{B.7}$$

$$\mathbf{p}_c^{(k)} = (1 - c_c)\mathbf{p}_c^{(k-1)} + h_\sigma^{(k)} \sqrt{c_c(2 - c_c)\mu_{\text{eff}}} \left\langle \mathbf{y}^{(k)} \right\rangle_w, \tag{B.8}$$

$$\mathbf{C}^{(k)} = (1 - c_1 - c_\mu)\, \mathbf{C}^{(k-1)} + c_1 \left(\mathbf{p}_c^{(k)} \left(\mathbf{p}_c^{(k)}\right)^T + \delta\left(h_\sigma^{(k)}\right) \mathbf{C}^{(k-1)} \right) + c_\mu \sum_{i=1}^{\mu} \omega_i \mathbf{y}_{i:\lambda}^{(k)} \left(\mathbf{y}_{i:\lambda}^{(k)}\right)^T, \tag{B.9}$$

where we iterate k until the termination criterion is met and using

$$\mu_{\text{eff}} = \left(\sum_{i=1}^{\mu} \omega_i^2 \right)^{-1}, \tag{B.10}$$

$$c_c = \frac{4 + \mu_{\text{eff}}/D}{D + 4 + 2\mu_{\text{eff}}/D}, \tag{B.11}$$

$$c_\sigma = \frac{\mu_{\text{eff}} + 2}{D + \mu_{\text{eff}} + 5}, \tag{B.12}$$

$$d_\sigma = 1 + 2\max\left(0,\ \sqrt{\frac{\mu_{\text{eff}} - 1}{D + 1}} - 1\right) + c_\sigma, \tag{B.13}$$

$$c_1 = \frac{2}{(D + 1.3)^2 + \mu_{\text{eff}}}, \tag{B.14}$$

$$c_\mu = \min\left(1 - c_1,\ 2\frac{\mu_{\text{eff}} - 2 + 1/\mu_{\text{eff}}}{(D + 2)^2 + \mu_{\text{eff}}}\right), \tag{B.15}$$

$$E^* = \frac{\sqrt{2}\Gamma\left(\frac{D+1}{2}\right)}{\Gamma\left(\frac{D}{2}\right)}, \tag{B.16}$$

$$h_\sigma^{(k)} = \begin{cases} 1, & \text{if} \frac{\|\mathbf{p}_\sigma^{(k)}\|}{\sqrt{1-(1-c_\sigma)^{2(k+1)}}} < \left(1.4 + \frac{2}{D+1}\right) E^* \\ 0, & \text{otherwise} \end{cases}, \tag{B.17}$$

$$\delta\left(h_\sigma^{(k)}\right) = \left(1 - h_\sigma^{(k)}\right) c_c \left(2 - c_c\right), \tag{B.18}$$

$$\left(\mathbf{C}^{(k)}\right)^{-1/2} = \mathbf{B}\left(\mathbf{D}^{(k)}\right)^{-1} \mathbf{B}^T, \tag{B.19}$$

where D corresponds to the number of parameters of the model (i.e. the dimensions of the search space) and we define $\mathbf{p}_\sigma^{(0)} = \mathbf{p}_c^{(0)} = 0$.

Appendix C

Placeholder

C.1 Topic Overview

Provide an overview of the topic to be studied...

C.2 Objectives

Explicitly state the objectives set to be achieved with this thesis...

C.3 Thesis Outline

Briefly explain the contents of the different chapters...

C.4 Theoretical Overview

Some overview of the underlying theory about the topic...

C.5 Theoretical Model 1

Multiple citations are compressed when using the `sort&compress` option when loading the `natbib` package as `\usepackage[numbers,sort&compress]{natbib}` in file `Thesis_Preamble.tex`, resulting in citations like [3, 8].

C.6 Theoretical Model 2

Other models...

Insert your chapter material here...

C.7 Numerical Model

Description of the numerical implementation of the models explained in Chapter 2...

C.8 Verification and Validation

Basic test cases to compare the implemented model against other numerical tools (verification) and experimental data (validation)...

Insert your chapter material here...

C.9 Problem Description

Description of the baseline problem...

C.10 Baseline Solution

Analysis of the baseline solution...

C.11 Enhanced Solution

Quest for the optimal solution...

C.11.1 Figures

Insert your section material and possibly a few figures...

Make sure all figures presented are referenced in the text!

Images

Make reference to Figures.

By default, the supported file types are *.png,.pdf,.jpg,.mps,.jpeg,.PNG,.PDF,.JPG,.JPEG*.

See `http://mactex-wiki.tug.org/wiki/index.php/Graphics_inclusion` for adding support to other extensions.

Drawings

Insert your subsection material and for instance a few drawings...

The schematic illustrated in Fig.can represent some sort of algorithm.

C.11.2 Equations

Equations can be inserted in different ways.

The simplest way is in a separate line like this

$$\frac{\mathrm{d}q_{ijk}}{\mathrm{d}t} + \mathcal{R}_{ijk}(\mathbf{q}) = 0\,. \tag{C.1}$$

If the equation is to be embedded in the text. One can do it like this $\partial\mathcal{R}/\partial\mathbf{q} = 0$.

It may also be split in different lines like this

C.11.3 Tables

Insert your subsection material and for instance a few tables...

Make sure all tables presented are referenced in the text!

Follow some guidelines when making tables:

C.11.4 Mixing

If necessary, a figure and a table can be put side-by-side as in Fig.

Insert your chapter material here...

C.12 Achievements

The major achievements of the present work...

C.13 Future Work

A few ideas for future work...

In case an appendix if deemed necessary, the document cannot exceed a total of 100 pages...

Some definitions and vector identities are listed in the section below.

C.14 Vector identities

$$\nabla \times (\nabla\phi) = 0 \tag{C.2}$$

$$\nabla \cdot (\nabla \times \mathbf{u}) = 0 \tag{C.3}$$