Imperial College London

Imperial College London

Department of Mathematics

FX Barrier Option Pricing

Author: Valentin BLANDA (CID: 02293988)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2022-2023

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Signed: Valentin Blanda

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Damiano Brigo, at Imperial College London for his support and guidance throughout the project.

I also wish to extend my heartfelt appreciation to Massimiliano Saba and Alessandro Piccininno at Barclays for their generous assistance and for providing me with the invaluable opportunity to work on my thesis at Barclays. I thank the whole CCR team for their support and their warm welcome.

Finally, I would like to thank my family for their unwavering support and encouragement throughout my years of study.

Abstract

In today's ever-evolving financial landscape, options play a pivotal role, serving diverse purposes for financial institutions and corporations alike. This paper zooms in on a specific category of options known as barrier options, which have gained significant prominence in over-the-counter (OTC) markets. Among these, barrier options in the foreign exchange (FX) market are particularly noteworthy, given their dual role in risk management and speculative trading. This paper is dedicated to thoroughly exploring the pricing techniques for FX barrier options, focusing on their accuracy and adaptability.

Contents

List of Figures

List of Tables

Introduction

In today's dynamic financial landscape, options play a pivotal role for both financial institutions and corporations. These versatile financial instruments serve a multitude of purposes, including hedging against risk, speculating on market movements, and crafting innovative solutions for advanced trading strategies. As financial markets continue to evolve, there is a continuous emergence of more sophisticated option types, commonly referred to as exotic options. This vibrant market for exotic options has witnessed significant growth over the past decades, characterized by rising trading volumes and enhanced liquidity. In this paper, our focus centers on a specific category of exotic options known as barrier options.

As barrier options gain increasing traction within the OTC markets, they have garnered heightened attention within the field of derivatives research. In the FX market, barrier options stand out as the most frequently traded type among exotic options. They serve a dual purpose, being integral to risk hedging strategies for corporate cash flows and offering speculative opportunities for traders seeking sophisticated FX spot rate exposure. Consequently, they present significant risk considerations for market makers. Notably, significant developments have unfolded in the literature, particularly concerning their analytical valuation. One of the early contributions in this field can be attributed to Reiner and Rubinstein, as documented in [1]. Their groundbreaking work provided pricing formulas for Single Barrier options. One other major contribution comes from Kunitomo and Ikeda in 1992 in [2]. They introduced a valuation formula for Double Barrier options, building upon a generalization of the Levy formula.

This thesis is dedicated to the comprehensive examination of various techniques employed in the pricing of FX Barrier options. Our primary objectives are to assess their accuracy and to evaluate their adaptability in the context of Barrier option pricing. Through rigorous analysis and empirical testing, we aim to provide an understanding of the strengths and limitations of these pricing techniques. In the first chapter of this thesis, we provide an extensive exposition of Barrier options, delving into their characteristics and offering an overview of the significance and appeal of these exotic financial instruments. Chapter 2 delves into the Garman-Kohlhagen framework, designed for pricing FX options. Additionally, we explore the theoretical formulas that are indispensable for pricing Barrier options, serving as fundamental building blocks for the subsequent chapters of this thesis. The remainder of the thesis shifts its focus towards the diverse methods employed to price these exotic options. Chapter 3 delves into investigations of Monte-Carlo methods, encompassing the testing of various variants within the Monte-Carlo framework. In contrast, Chapter 4 centers its attention on Lattice methods, exploring the utility of Binomial, Trinomial, and an enhanced version of the Binomial Tree. These methods serve as invaluable tools for pricing exotic options and form a significant part of our analytical exploration. Chapter 5 represents a thorough exploration of Finite Difference Methods. This chapter delves into the examination of various schemes and discretization techniques designed to efficiently price Barrier options. This pursuit marks a significant stride forward in the quest for effective numerical pricing methods.

Chapter 1

Barrier Options

A barrier option is a type of financial derivative whose payoff depends on whether the underlying asset's price reaches or crosses some specified barrier level during the lifetime of the option. If the barrier is not crossed, the option remains active until its expiration, and the usual payoff conditions apply. If the barrier is crossed, either the option is knocked out, meaning an immediate termination of the option with no further value or the option is knocked in, that results in the activation of the option. Barrier options are path-dependent options and are composed of four basic forms: down-and-out, down-and-in, up-and-out and up-and-in. The barrier is either set 'up' or 'down' that is to say above or below the asset price at the time the option is created.

1.1 Some elements of context

Barrier options were developed in order to address specific hedging needs and market conditions that could not be accommodated with European or American options. This specific type of options have started to be traded in the OTC market in the 1970s when the Black-Scholes model revolutionized option pricing. Barrier options gained popularity since they provide tailored risk management solutions. They indeed allow investors or institutions to have positions that can provide specific levels of protection. One can then maintain exposure to profits while be protected against unfavorable movements of the market.

In the 1990s as financial markets continued to evolve, barrier options became more standardized and accessible. Financial institutions and derivatives exchanges started offering barrier options on a wider range of underlying assets, including currencies, commodities, and equities. That is the precise context of this work.

The 2008 global financial crisis prompted a renewed interest in risk management and derivatives products. Barrier options remained relevant due to their ability to tailor risk profiles to specific market conditions and individual preferences. Additionally, advancements in computational tools and trading platforms facilitated the trading and pricing of these options, making them more accessible to a broader range of market participants.

1.2 Single Barrier options

As mentioned before, a barrier option is an option whose payoff is either "knocked out" or "knocked in" if the price of the underlying crosses a barrier. The price of the option is reduced since there is a chance that the option may be made worthless. The simplest type of barrier option are Single Barrier options. The four types of Single Barrier call options are the following:

An up-and-out call option with a constant barrier $B > S_0$ has a payoff if the underlying price does not go beyond the barrier value until maturity T:

$$\begin{cases} (S_T - K)^+ & S_t < B, \forall t \\ 0 & \text{otherwise} \end{cases}$$

An up-and-in call option with a constant barrier $B > S_0$ has a payoff if the underlying price go beyond the barrier value until maturity T:

$$\begin{cases} 0 & S_t < B, \forall t \\ (S_T - K)^+ & \text{otherwise} \end{cases}$$

An down-and-out call option with a constant barrier $B < S_0$ has a payoff if the underlying prices stays beyond the barrier value until maturity T:

$$\begin{cases} (S_T - K)^+ & S_t > B, \forall t \\ 0 & \text{otherwise} \end{cases}$$

An down-and-in call option with a constant barrier $B < S_0$ has a payoff if the underlying prices stays bellow the barrier value until maturity T:

$$\begin{cases} 0 & S_t > B, \forall t \\ (S_T - K)^+ & \text{otherwise} \end{cases}$$

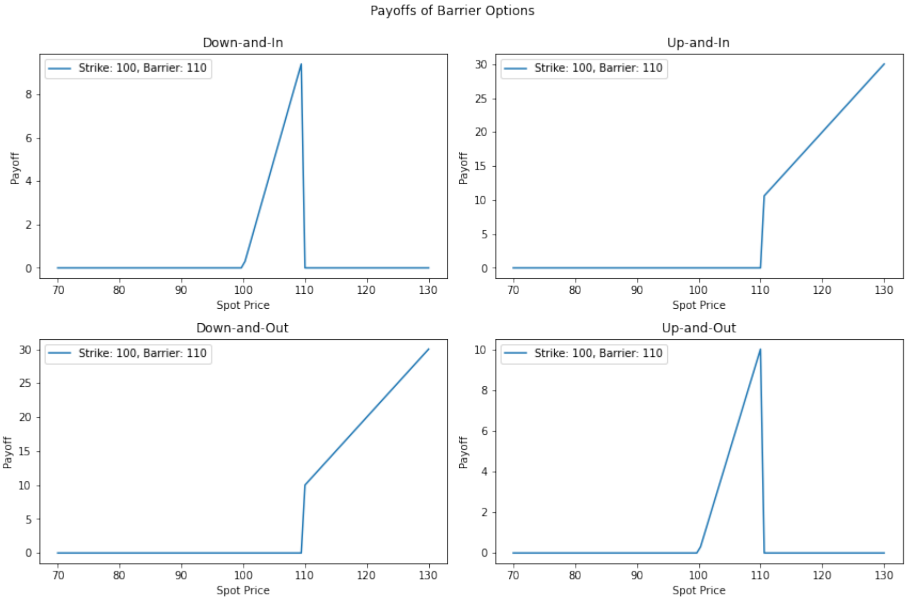

Figure 1.1: Payoffs of Single Barrier options

1.3 Double Barrier options

Double Barrier options are either knocked in or knocked out if the underlying price crosses the lower barrier B_d or the upper one B_u before the expiration of the option. There are two different variations of double barrier options, including:

Knock-In Double Barrier Option only becomes active and valid if the underlying asset's price crosses one of the barriers during its lifetime. If the asset's price does not breach the barriers, the option remains inactive and expires worthless.

Knock-Out Double Barrier Option loses its validity if the underlying asset's price crosses one of the barriers during its lifetime. If the asset's price breaches either barrier, the option becomes null and void.

In Figure 1.2, the underlying does not cross any barrier. Assuming the option is a knocked-out option, the option will not be affected by the barriers and its value will be the one the equivalent vanilla option.

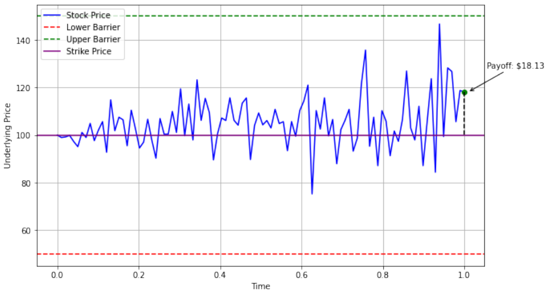

Figure 1.2: Price of the underlying under the condition of two barriers

1.4 European and American Barrier options

As with standard options, we can define American Barrier options. American Barrier options differ from their corresponding European options due to their exercise flexibility. European options can only be exercised at the expiration date, while American options can be exercised at any time before or on the expiration date. The modifications required for pricing and hedging these new Barrier options are similar to those for their Vanilla equivalents. In this work, we will not extensively delve into American Barrier options since their characteristics are either akin to European Barrier options or can be compared to standard American options. However, we can note that Haug developed in [3] closed-form solution for Single American Barrier options.

1.5 Rebates

A rebate feature is a distinctive aspect that can be added to barrier options, enhancing their flexibility and attractiveness to investors. In the context of barrier options, a rebate refers to a fixed cash payment that is granted to the option holder when the option is knocked out or becomes void due to a barrier breach. This rebate acts as a consolation prize for the option holder, compensating them for the potential loss of profit resulting from the barrier being hit.

Rebate features can be particularly appealing to investors who are interested in barrier options but want a degree of protection against barrier breaches. Rebates can make these options more attractive by adding an element of potential income, regardless of whether the option achieves its full profit potential. Additionally, rebates can be used as a tool for managing risk and enhancing risk-reward profiles in trading and investment strategies. In this thesis, we are considering zero-rebate barrier options.

1.6 Window Barrier options

Window Barrier options are similar to standard Barrier options but include an additional feature known as a 'window.' In this case, the Barrier (or Barriers, in the context of Double Barrier options) is active only during a specific period between the trading date and the option's maturity. This concept of a 'window' can be further extended by introducing multiple monitoring windows throughout the option's lifespan, each with its unique barrier levels. This feature allows for a more nuanced and flexible approach to structuring options, accommodating different market conditions and price dynamics during various time intervals.

In Figure 1.3, we can observe that the underlying crosses the barriers multiple times, the first being the upper barrier at $T = 0.55$. If this is a knock-out barrier, the option will be valued zero.

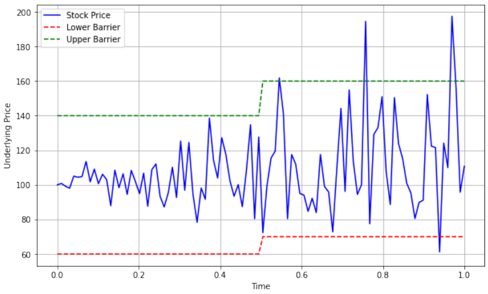

Figure 1.3: Price of the underlying under the condition of two time-dependent barriers

1.7 Discrete monitoring

Another important characteristic of Barrier options is the monitoring method: they can be either continuously monitored or discretely monitored. In the case of continuous monitoring, if the underlying price reaches the barrier at any time before the option's expiry, the barrier will be effectively considered reached. In the second case, discrete monitoring is used, where specific monitoring times are defined (such as daily or weekly checks), and the value of the underlying is only assessed at these predetermined times.

In practice, barriers are monitored only at discrete points in time for equity or commodity options. On top of the differences of implementation, there are financial or legal reasons for this type of monitoring. Broadie, Glasserman and Kou introduced in [4] a continuity correction for discrete barrier options. Their work resulted in the following theorem:

Theorem 1.7.1. *Let V and V_d^m be the respective prices of the continuous barrier option and the discrete barrier option (with m monitoring times). The barrier level is B. We can assume the following approximation:*

$$V_d^m(B) = V(Be^{\pm\beta\sigma\sqrt{\frac{T}{m}}}) + o(\frac{1}{\sqrt{m}}) \tag{1.7.1}$$

+ is designed for 'up' options while - is for 'down' options. $\beta = -\frac{\zeta(0.5)}{\sqrt{2\pi}} \approx 0.5826$
where $\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$ is Riemann zeta function.

Chapter 2

Garman-Kohlhagen model

2.1 A generalised version of the Black-Scholes model

In Black-Scholes model, the underlying is a non-dividend paying stock with returns that follows a geometric Brownian motion. Hence, the model that accounts for domestic risk-free interest rates only is not sufficient to be applied to foreign exchange options. The Garman-Kohlhagen option-pricing model (1983) is suitable with interest rate parity. The partial differential equation is the following:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r_d - r_f)S\frac{\partial V}{\partial S} - r_d V = 0 \tag{2.1.1}$$

The terminal condition is $V(T,S) = g(S)$, the payoff of the option. The variables used in the model are:
$V(S,T)$: price of a call option in domestic units per foreign units
S: spot price
T: time remaining until maturity
K: exercise price of the call option
r_d: domestic interest rate
r_f: foreign interest rate
σ: volatility of spot currency price

The derivation of the PDE is explained in details in the Appendix. As with the Black-Scholes model, the equation above assumes that one will pay proportional dividend. The spot price follows a geometric Brownian motion, option price includes one stochastic volatility and interest rates are constant.

$$dS = \mu S dt + \sigma S dZ \tag{2.1.2}$$

where $\mu = r_d - r_f$ is the drift of spot currency price and Z a standard Wiener process.

The closed-form solutions for the European calls and puts are the following:

$$C(S,T) = e^{-r_f T} S N(d_1) - e^{-r_d T} K N(d_2) \tag{2.1.3}$$

$$P(S,T) = -e^{-r_f T} S N(-d_1) + e^{-r_d T} K N(-d_2) \tag{2.1.4}$$

with d_1 and d_2 defined as follows:

$$d_1 = \frac{\ln(\frac{S}{K}) + (r_d - r_f + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}, d_2 = d_1 - \sigma\sqrt{T} \tag{2.1.5}$$

2.2 Estimation of the Implied volatility using the Garman-Kohlhagen model

In the context of option pricing, estimating the implied volatility is sometimes a crucial step in matching the theoretical option prices to the observed market prices. For the Garman-Kohlhagen model, which is commonly used for pricing currency options, this involves solving for the volatility parameter that equates the calculated theoretical option price to the market price. Two widely used numerical methods for estimating implied volatility are the Newton-Raphson method and the bisection method.

The Newton-Raphson method is an iterative numerical technique that aims to find the root of a function. In the context of implied volatility estimation, we use it to find the volatility σ that satisfies the equation $C_{\text{calculated}} - C_{\text{market}} = 0$, where $C_{\text{calculated}}$ is the option price calculated using Equation (1) and C_{market} is the observed market option price.

The Newton-Raphson iteration formula for updating the estimate of volatility is given by:

$$\sigma_{\text{new}} = \sigma_{\text{old}} - \frac{C_{\text{calculated}} - C_{\text{market}}}{\text{Vega}} \tag{2.2.1}$$

Where:

σ_{new} is the updated estimate of volatility
σ_{old} is the previous estimate of volatility
Vega is the sensitivity of the option price with respect to volatility, defined as the derivative of the option price with respect to volatility

The bisection method is another iterative numerical technique that is particularly useful for functions with a single root within an interval. In the context of implied volatility estimation, the bisection method involves finding the volatility σ that makes the calculated option price $C_{\text{calculated}}$ match the observed market option price C_{market}.

The bisection method is a straightforward numerical technique used for finding the root of a function within a given interval. In the context of estimating implied volatility, the bisection method aims to find the volatility parameter σ that makes the calculated option price $C_{\text{calculated}}$ match the observed market option price C_{market}. The bisection algorithm involves iteratively narrowing down the interval in which the root lies until a sufficiently accurate estimate is obtained. The basic steps of the algorithm are as follows:

1. Choose an initial interval $[\sigma_{\text{low}}, \sigma_{\text{high}}]$ where you believe the root (implied volatility) lies.
2. Calculate the option price $C_{\text{calculated}}$ using the Garman-Kohlhagen pricing formula with the midpoint volatility $\sigma_{\text{mid}} = (\sigma_{\text{low}} + \sigma_{\text{high}})/2$.
3. Calculate the error $C_{\text{error}} = C_{\text{calculated}} - C_{\text{market}}$.
4. If $|C_{\text{error}}|$ is sufficiently small (i.e., convergence criterion met), stop and consider σ_{mid} as the estimated implied volatility.
5. Determine the new interval based on the sign of C_{error}. If $C_{\text{error}} > 0$, set $\sigma_{\text{low}} = \sigma_{\text{mid}}$; if $C_{\text{error}} < 0$, set $\sigma_{\text{high}} = \sigma_{\text{mid}}$.
6. Repeat steps 2 to 5 until the desired level of accuracy is achieved.

The midpoint volatility σ_{mid} is updated in each iteration using the formula:

$$\sigma_{\text{mid}} = \frac{\sigma_{\text{low}} + \sigma_{\text{high}}}{2} \tag{2.2.2}$$

During each iteration, the interval $[\sigma_{\text{low}}, \sigma_{\text{high}}]$ is halved, bringing the algorithm closer to the true implied volatility.

As implied volatility increases, a notable distinction arises between barrier options and plain vanilla options. Unlike plain vanilla options, where vega (the sensitivity of the option price to changes in implied volatility) usually increases uniformly with volatility, barrier options exhibit a unique behavior.

In this context, the impact of volatility on a barrier option's price becomes nonlinear. Higher volatility pushes the option's value toward the barrier, enhancing the likelihood of breaching it. Consequently, the vega of a barrier option "flattens" as volatility increases. This flattening vega is a result of the growing significance of the barrier relative to other factors as volatility rises. This behavior is in stark contrast to plain vanilla options, where vega tends to rise steadily with increasing volatility.

For simplicity in the next chapters, we will impose a fixed volatility. For example, in lattice methods there is a local volatility different from the input volatility and in numerical schemes we can define either a constant volatility or using a deterministic function of two variables.

2.3 Theoretical values

For some types of Barrier options, we have theoretical formulas. Some have been developed and proved in [2] by Kunitomo and Ikeda for more general examples of Double Barrier options. The formulas derived there are for curved boundaries barriers, our use case is a specific example with constant barriers.

2.3.1 Single Barrier options

The formulas for Single Barrier options have been proved by Reiner and Rubinstein in [1]. B represents the value of the barrier, otherwise we kept the notation previously introduced in section 2.1.

$$A_1 = e^{-r_f T} aSN(ax_1) - e^{-r_d T} KN(ax_1 - a\sigma\sqrt{T})$$

$$A_2 = e^{-r_f T} aSN(ax_1) - e^{-r_d T} KN(ax_2 - a\sigma\sqrt{T})$$

$$A_3 = e^{-r_f T} a(\frac{B}{S})^{2(\mu+1)} SN(bx_3) - e^{-r_d T} K(\frac{B}{S})^{2\mu} N(bx_3 - b\sigma\sqrt{T})$$

$$A_4 = e^{-r_f T} a(\frac{B}{S})^{2(\mu+1)} SN(bx_4) - e^{-r_d T} K(\frac{B}{S})^{2\mu} N(bx_4 - b\sigma\sqrt{T})$$

$$A_5 = Ke^{-r_d T}[N(bx_2 - b\sigma\sqrt{T}) - (\frac{B}{S})^{2\mu} N(bx_4 - b\sigma\sqrt{T})]$$

$$A_6 = K[(\frac{B}{S})^{\mu+\lambda} N(bx_5) + (\frac{B}{S})^{\mu-\lambda} N(bx_5 - 2b\lambda\sigma\sqrt{T})]$$

with

$$\begin{cases} a = 1, -1 & \text{call,put} \\ b = 1, -1 & \text{out,in} \end{cases}$$

$$x_1 = \frac{\log(\frac{S}{K})}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}, x_2 = \frac{\log(\frac{S}{B})}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}$$

$$x_3 = \frac{\log(\frac{B^2}{SK})}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}, x_4 = \frac{\log(\frac{B}{S})}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}$$

$$x_5 = \frac{\log(\frac{B}{S})}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}, \mu = \frac{r_d - r_f - \frac{\sigma^2}{2}}{\sigma^2}, \lambda = \sqrt{\mu^2 + \frac{2r_d}{\sigma^2}}$$

Down/Up	In/Out	Call/Put	Payoff (K $\leq B$)	Payoff (K $\geq B$)
Down	In	Call	$A_1 - A_2 + A_4 + A_5$	$A_3 + A_5$
Up	In	Call	$A_2 - A_3 + A_4 + A_5$	$A_1 + A_5$
Down	In	Put	$A_1 + A_5$	$A_2 - A_3 + A_4 + A_5$
Up	In	Put	$A_3 + A_5$	$A_1 - A_2 + A_4 + A_5$
Down	Out	Call	$A_2 - A_4 + A_6$	$A_1 - A_3 + A_6$
Up	Out	Call	$A_1 - A_2 + A_3 - A_4 + A_6$	A_6
Down	Out	Put	A_6	$A_1 - A_2 + A_3 - A_4 + A_6$
Up	Out	Put	$A_1 - A_3 + A_6$	$A_2 - A_4 + A_6$

Table 2.1: Theoretical Values of Single Barrier options

2.3.2 Double Barrier options

The formulas for Double Barrier options are more complex, for simplification we will only refer to Double Knock-Out (KOKO) call options. With parity conditions and other symmetries the other types of options can be derived.B_u and B_d represent respectively the upper and the lower barrier of the considered option.

$$C = Se^{-r_f T} \sum_{n=-\infty}^{\infty} (\frac{B_u^n}{B_d^n})^{\mu}[N(d_1) - N(d_2)] - (\frac{B_d^{n+1}}{B_u^n S})^{\mu}[N(d_3) - N(d_4)] \tag{2.3.1}$$

$$-Ke^{-r_d T} \sum_{n=-\infty}^{\infty} (\frac{B_u^n}{B_d^n})^{\mu-2}[N(d_1 - \sigma\sqrt{T}) - N(d_2 - \sigma\sqrt{T})] - (\frac{B_d^{n+1}}{B_u^n S})^{\mu-2}[N(d_3 - \sigma\sqrt{T}) - N(d_4 - \sigma\sqrt{T})]$$

with

$$d_1 = \frac{\log(\frac{SB_u^{2n}}{KB_d^{2n}}) + (r_d - r_f + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}, d_2 = \frac{\log(\frac{SB_u^{2n-1}}{B_d^{2n}}) + (r_d - r_f + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}$$

$$d_3 = \frac{\log(\frac{B_d^{2n+2}}{KSB_u^{2n}}) + (r_d - r_f + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}, d_4 = \frac{\log(\frac{B_d^{2n+2}}{SB_u^{2n+1}}) + (r_d - r_f + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}$$

$$\mu = \frac{2(r_d - r_f)}{\sigma^2}$$

Double Barrier option prices are represented using infinite series. However, the convergence of these series can be achieved with only a limited number of terms. When the two barriers are closely positioned, adding more terms is necessary to ensure the accuracy of the computed price.

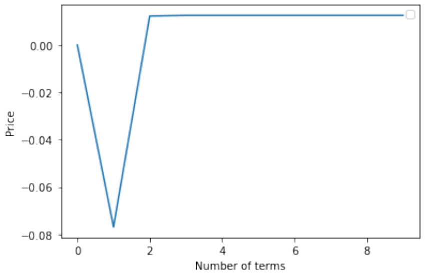

Figure 2.1: Convergence of the option price with S=147,B$_u = 150, B_d = 145, K = 147$

2.3.3 In-Out parity

Owning both a Knock-Out option and a Knock-In option with identical features is essentially equivalent to holding a comparable vanilla option, regardless of how the underlying asset behaves concerning the barrier level. Barriers effectively cancel each other out. If the underlying asset's price touches the barrier level (which would activate the Knock-In option), it also crosses the barrier level (which would terminate the Knock-Out option). If the underlying asset's price does not touch the barrier level, neither the Knock-In nor the Knock-Out option is triggered. In both cases, the final outcome is equal to the vanilla price.

Chapter 3

Monte Carlo methods

Simulation techniques are invaluable tools in managing complex financial scenarios involving path-dependent payoffs or distributions that lack analytical expressions. One prominent application is in pricing derivative securities, estimating value at risk, and emulating hedging strategies. Monte-Carlo simulation, in particular, proves highly effective for such tasks, offering a blend of simplicity and adaptability as its chief advantages. However, the method's drawback lies in its considerable computational demands, which can be mitigated to some extent. Our approach employs Monte-Carlo simulation to generate simulated option price values, which are then juxtaposed against outcomes derived from analytical formulas. In this chapter, we will delve into the fundamental tenets of Monte Carlo simulation, expounding on its core concepts and providing pertinent formulas for a comprehensive understanding.

3.1 Some basic definitions

The idea of Monte Carlo method is to estimate $\theta = E(f(X))$ with $f(X)$ is an arbitrary function. For a sequence of independent and identically distributed random variables $X_1, X_2 ... X_n$. We can estimate θ with the sample mean defined by:

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} f(X_i) \tag{3.1.1}$$

As n grows to infinity, we have by the law of large number that $\hat{\theta} \longrightarrow E(f(X)) = \theta$

Furthermore, the Central Limit Theorem underscores that regardless of the distribution of individual samples, when the sample size is sufficiently large, the distribution of the sample means approximates a normal distribution, that is to say:

$$\frac{\hat{\theta} - \theta}{\sqrt{\frac{\sigma^2}{n}}} \longrightarrow N(0,1)$$

with

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(f(X_i) - \hat{\theta})^2$$

This is equivalent to the following probabilistic approach:

$$P(\hat{\theta} - \eta_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}} < \theta < \hat{\theta} + \eta_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}) = 1 - \alpha \tag{3.1.2}$$

The accuracy of the estimation can be measured using the term $\frac{\sigma}{\sqrt{n}}$, which is the standard error. Additionally, when feasible, we will analyze the percent error relative to theoretical values.

In order to simulate barrier options using Monte Carlo, we first have to simulate sample paths of the underlying price. For vanilla options, this path generation is not useful since only the price of the underlying at maturity is concerned in the calculations. However, for Barrier options, this is mandatory since we need to know whether the price hits the barrier before expiry. So, we have to simulate the entire evolution. We consider equation (2.1.2) to describe the price of the underlying asset.

The Euler scheme gives us the following equation:

$$S_{t+dt} = (1 + \mu dt)S_t + \sigma S_t \sqrt{dt} Z$$

with Z is a standard normal random variable. Using the properties of Wiener process, we have:

$$S_{t+dt} = S_t e^{(\mu - \frac{\sigma^2}{2})dt + \sigma\sqrt{dt}Z}$$

In Table 3.1, we present the results of Monte Carlo simulations for various Down-and-Out call options. We have chosen options with different Barrier values as well as varying stock values. These results will be compared with those obtained through alternative variations of Monte Carlo methods. The errors listed in the tables represent the percent errors from the analytical values. The standard errors of the estimations are also written under the parenthesis. We started with simple Barrier options to initially assess the efficiency of the method.

	Price	MC_{100}	err_{100}	MC_{1000}	err_{1000}	MC_{5000}	err_{5000}	MC_{10000}	err_{10000}
$\mathrm{S}_0 = 100$ B=95	6.7447	9.1516 (1.3131)	0.3569	7.4283 (0.4712)	0.1014	7.0233 (0.2107)	0.0413	6.7394 (0.1468)	$7.8.10^{-4}$
$\mathrm{S}_0 = 100$ B=90	10.9501	12.2386 (1.5597)	0.1176	11.3816 (0.4885)	0.0394	11.1965 (0.2284)	0.0316	11.0001 (0.1578)	$4.56.10^{-3}$
$\mathrm{S}_0 = 110$ B=100	15.0164	13.7898 (2.3698)	0.0817	14.9260 (0.6672)	0.0060	15.4914 (0.2987)	0.0316	15.2934 (0.2146)	0.0184
$\mathrm{S}_0 = 110$ B=90	21.1631	19.1583 (1.7933)	0.0947	21.8267 (0.6145)	0.0314	21.4614 (0.2743)	0.0141	21.2005 (0.1961)	$1.8.10^{-3}$
$\mathrm{S}_0 = 120$ B=100	28.0566	28.8587 (2.3466)	0.0286	28.5134 (0.7669)	0.0163	28.2530 (0.3471)	$7.0.10^{-3}$	28.0295 (0.2410)	$1.0.10^{-3}$
$\mathrm{S}_0 = 120$ B=110	18.4403	19.2732 (2.7871)	0.0452	19.1378 (0.8465)	0.0378	18.7095 (0.3773)	0.0146	18.1083 (0.2664)	0.0180

Table 3.1: (MC)Down-and-out call option with $r_d = 0.08$,$r_f = 0.04$,$T = 0.5$,$\sigma = 0.25$, $K = 90$

Monte Carlo estimations give accurate prices compared to the analytical formulas for Single Barrier Options. We worked on simulations for $n = 100, 1000, 5000$ and 10000 samples. Results are very promising, for $n = 1000$, we get for example on average a percent error of order 10^{-2}. With the table we confirm one main aspect of Monte Carlo theory: increasing the number of simulations leads to an improved accuracy for the computation.

We then transition to Double Barrier options, which involve both a lower and an upper barrier. The key algorithmic modification here is the addition of the upper barrier, along with monitoring to ensure that our estimates fall within the defined lower and upper barrier levels. Monte Carlo computations offer the flexibility required to easily adapt to the unique characteristics of our exotic options. Table 3.2 provides the results for various Double Barrier options.

	Price	MC_{100}	err_{100}	MC_{1000}	err_{1000}	MC_{5000}	err_{5000}	MC_{10000}	err_{10000}
$\mathrm{B}_d = 50$ $\mathrm{B}_u = 150$	5.4636	5.8933 (0.9468)	0.0786	5.6280 (0.2546)	0.0301	5.6248 (0.1133)	0.0295	5.4615 (0.0811)	$4.0.10^{-4}$
$\mathrm{B}_d = 60$ $\mathrm{B}_u = 140$	5.2200	5.6990 (0.8387)	0.0918	5.3711 0.2453	0.0289	5.2930 (0.1088)	0.0140	5.2386 (0.0774)	$3.6.10^{-3}$
$\mathrm{B}_d = 70$ $\mathrm{B}_u = 130$	4.3806	4.1325 (0.7008)	0.0566	4.2668 (0.1999)	0.0260	4.4774 (0.0941)	0.0221	4.3846 (0.0658)	$9.0.10^{-4}$
$\mathrm{B}_d = 80$ $\mathrm{B}_u = 120$	2.4642	2.6122 (0.4752)	0.0601	2.6041 (0.1446)	0.0568	2.4168 (0.0628)	0.0192	2.4346 (0.0438)	0.0120
$\mathrm{B}_d = 90$ $\mathrm{B}_u = 110$	0.3003	0.3474 (0.1524)	0.1568	0.3149 (0.0401)	0.0486	0.3140 (0.0181)	0.0456	0.3015 (0.0125)	$4.0.10^{-3}$

Table 3.2: (MC)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 0.25$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$

The results show that Monte-Carlo techniques are well-adapted to price Barrier options of diverse types. For most of the cases, we obtain a percent error inferior to 0.1 with less than 1000 simulations. The last case is relevant to analyze since we have close upper and lower barrier (namely $B_d = 90$ and $B_u = 110$). However with an increasing number of simulations, the results even for this kind of close up and down barriers appear to be accurate. Nevertheless, it is important to acknowledge that Monte Carlo methods do have some limitations, including the computational time required for estimations and the potential width of confidence intervals. The next section will introduce techniques aimed at mitigating variance, addressing this second challenge.

One element to notice is the computational time of Monte Carlo simulations. On average we have:

	MC_{100}	MC_{1000}	MC_{5000}	MC_{10000}
Time(s)	0.017	1.589	39.24	158.9

Table 3.3: Average computation time for Monte-Carlo

Figure 3.1 highlights the second issue raised above, that is to say the size of the confidence intervals. We can observe that it reduces as the number of simulations increase, as introduced in the probabilistic approach of Monte-Carlo in (3.1.2). One goal in the next section is to analyse estimations conduced with Variance reduction techniques.

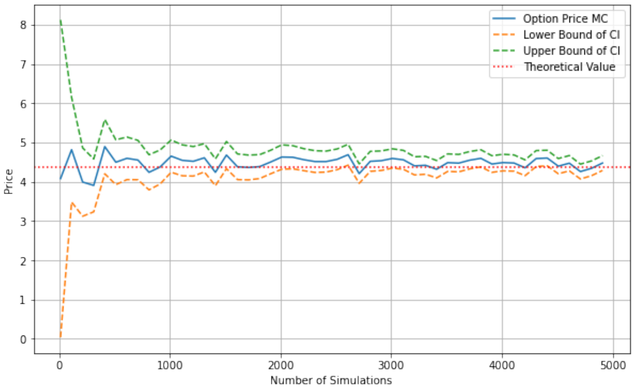

Figure 3.1: Comparision between Monte-Carlo methods for S=100, K=90,$B_d = 90, B_u = 110, r_d = 0.1, r_f = 0.05, T = 0.25, \sigma = 0.25$

3.2 Variance reduction techniques

Variance reduction techniques play a crucial role in enhancing the efficiency and accuracy of Monte-Carlo simulations, particularly in the context of option pricing. Monte-Carlo methods rely on generating random samples to approximate complex mathematical calculations, such as option prices in finance. Variance, representing the dispersion of these sampled outcomes around their mean, is a key factor in determining the reliability and precision of these estimates. Variance reduction techniques aim to minimize this dispersion, thereby increasing the reliability of the Monte Carlo estimates.

3.2.1 Control Variates

In option pricing, one widely used variance reduction technique is the control variate method. This technique involves identifying a correlated variable with known expected value and using it to create an adjusted payoff structure. By subtracting this correlated variable's expected value from the simulated payoffs, the variance of the resulting estimates is significantly reduced.

Let us define the control variate $g(X)$, such that its expectation $G = E[g(X)]$ is known. We can suppose correlation between $f(X)$ and $g(X)$:

$$Cov(f, g) = E[(f - \hat{\theta})(g - G)]$$

The new random variable $Y = f(X) - \alpha(g(X) - G)$ has less variance than $f(X)$. Therefore, we can replace the formula of the estimator for the following:

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^{n} (f(X_i) - \alpha g(X_i)) + \alpha G \tag{3.2.1}$$

α is a fixed parameter that needs to be chosen carefully, it has to minimize the variance

$$\sigma_Y^2 = \sigma_f^2 - 2\alpha Cov(f, g) + \alpha^2 \sigma_g^2$$

With the optimal $\alpha^* = \frac{Cov(f,g)}{\sigma_g^2}$, we get:

$$\sigma_Y^2 = \sigma_f^2 - \frac{Cov(f, g)^2}{\sigma_g^2}$$

Correlation between f and g can be defined as

$$\rho_{fg} = \frac{Cov(f, g)}{\sigma_f \sigma_g}$$

Hence, we can write the dependency between the control variate and the correlation of f and g.

$$\sigma_Y^2 = \sigma_f^2 (1 - \rho_{fg}^2)$$

The fact is that knowing α^* is not very likely to happen, so we estimate it with Monte-Carlo information. Let us build this new estimator $\widetilde{\theta}$:

$$\widetilde{\theta} = \hat{\theta} - \widetilde{\alpha^*} \frac{1}{n} \sum_{i=1}^{n} (g(X_i) - G) \tag{3.2.2}$$

with

$$\widetilde{\alpha^*} = \frac{\widetilde{Cov(f, g)}}{\widetilde{\sigma_g^2}} = \frac{\frac{1}{n} \sum_{i=1}^{n} (f(X_i) - \hat{\theta})(g(X_i) - G)}{\frac{1}{n} \sum_{i=1}^{n} (g(X_i) - G)^2}$$

The generalisation is to use a set of control variates $g_1(X), g_2(X)....g_n(X)$. The new random variable formed is $Y = f(X) - \sum_{i=1}^{n} \alpha_i g_i(X)$. As in the case with one variate, we need to minimize the variance of Y.

For $\alpha_1, ...\alpha_n$ to be optimal, it means that $f(X)$ and $\sum_{i=1}^{n} \alpha_i g_i(X)$ are uncorrelated. The two terms being independent, we can write:

$$Var(f(X)) = Var(Y) + Var(\sum_{i=1}^{n} \alpha_i g_i(X))$$

The method allows for more accurate and efficient estimation of option prices while maintaining Monte-Carlo framework's flexibility and simplicity. The numerical results will illustrate the improvements made. The control variate technique demonstrates how a clever choice of auxiliary variables can substantially improve the quality of Monte-Carlo estimates.

	Price	MC_{100}	err_{100}	MC_{1000}	err_{1000}	MC_{5000}	err_{5000}	MC_{10000}	err_{10000}
$\mathrm{S}_0 = 100$ B=95	6.7447	8.6713 (1.0458)	0.2856	7.3981 (0.2145)	0.0969	7.0076 (0.1000)	0.0390	6.7402 (0.0725)	$7.0.10^{-4}$
$\mathrm{S}_0 = 100$ B=90	10.9501	12.1457 (0.6727)	0.1092	11.3214 (0.1122)	0.0339	11.1765 (0.0516)	0.0207	10.9899 (0.0352)	$3.6.10^{-3}$
$\mathrm{S}_0 = 110$ B=100	15.0164	13.8975 (0.9800)	0.0745	14.7765 (0.2389)	0.0160	15.3567 (0.1045)	0.0227	15.2251 (0.0762)	0.0139
$\mathrm{S}_0 = 110$ B=90	21.1631	19.3758 (0.2956)	0.0845	21.8189 (0.03687)	0.0310	21.4701 (0.0226)	0.0145	21.1974 (0.0140)	$1.6.10^{-3}$
$\mathrm{S}_0 = 120$ B=100	28.0566	28.7554 (0.4304)	0.0249	28.5563 (0.1250)	0.0178	28.2365 (0.0588)	$6.4.10^{-3}$	28.0265 (0.0409)	$1.1.10^{-3}$
$\mathrm{S}_0 = 120$ B=110	18.4403	19.1341 (1.3179)	0.0376	19.0007 (0.3925)	0.0304	18.6541 (0.1685)	0.0116	18.5567 (0.1192)	$6.3.10^{-3}$

Table 3.4: (CV-MC)Down-and-out call option with $r_d = 0.08$,$r_f = 0.04$,$T = 0.5$,$\sigma = 0.25$, $K = 100$

The use of this first Variance reduction technique helped with improving the percent as well as the standard error for most of the cases as shown in Table 3.3. For each number of simulations tested, we obtained better results and smaller confidence intervals. We tested as before the new technique with Double Barrier options.

	Price	MC_{100}	err_{100}	MC_{1000}	err_{1000}	MC_{5000}	err_{5000}	MC_{10000}	err_{10000}
$\mathrm{B}_d = 50$ $\mathrm{B}_u = 150$	5.4636	5.8576 (0.5935)	0.0721	5.6135 (0.1953)	0.0274	5.4897 (0.0845)	$4.8.10^{-3}$	5.4617 (0.0562)	$3.0.10^{-4}$
$\mathrm{B}_d = 60$ $\mathrm{B}_u = 140$	5.2200	5.5567 (0.5163)	0.0645	5.3352 (0.1853)	0.0221	5.2769 (0.0748)	0.0109	5.2298 (0.0578)	$1.9.10^{-3}$
$\mathrm{B}_d = 70$ $\mathrm{B}_u = 130$	4.3806	4.1457 (0.4339)	0.0536	4.2758 (0.1586)	0.0239	4.3985 (0.0854)	$4.1.10^{-3}$	4.3833 (0.0315)	$1.1.10^{-3}$
$\mathrm{B}_d = 80$ $\mathrm{B}_u = 120$	2.4642	2.6088 (0.2956)	0.0587	2.5957 (0.0956)	0.0534	2.4844 (0.0498)	$8.2.10^{-3}$	2.4839 (0.0416)	$8.0.10^{-3}$
$\mathrm{B}_d = 90$ $\mathrm{B}_u = 110$	0.3003	0.3485 (0.0997)	0.1605	0.3120 (0.0317)	0.0390	0.3102 (0.0113)	0.0330	0.3016 (0.0092)	$4.3.10^{-3}$

Table 3.5: (CV-MC)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 0.25$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$

The outcomes for Double Barrier options exhibit varying degrees of contrast when Control Variates are employed. We continue to obtain reasonable estimations, especially with a substantial number of simulations, and in some cases, we observe improvements compared to the standard Monte-Carlo algorithm. Variance reduction techniques have proven effective, resulting in progressively narrower confidence intervals for each option price. However, it is important to note that the issue raised in the previous section concerning closely positioned upper and lower barriers remains unsolved with Control Variates. Over time, the estimations for this type of Double Barrier option tend to resemble the previous results obtained through the standard Monte-Carlo approach.”

3.2.2 Antithetic Variates

Antithetic Variates is another powerful variance reduction technique used in option pricing simulations to improve accuracy and efficiency. The method aims at leveraging negative correlation between the pairs of random variables in order to reduce the entire variance of the estimates. In the context of option pricing, antithetic variates involves generating paired random paths, where one path follows the original model and the other path follows the mirrored model. Let us consider a set of n copies of the random variable X. Now, we just suppose they are identically distributed but not independent. The antithetic variates method relies on the following equality:

$$\hat{\theta} = \frac{1}{2n} \sum_{i=1}^{2n} f(X_i) = \frac{1}{n} \sum_{i=1}^{n} Y_i = Y \tag{3.2.3}$$

with

$$Y_1 = \frac{f(X_1) + f(X_2)}{2}, ..., Y_n = \frac{f(X_{n-1}) + f(X_n)}{2}$$

The previous estimators are identical and we have by linearity that $E[Y_i] = E[f(X)] = \theta$. By taking the example of Y_1, we compute the variance.

$$Var(Y_1) = \frac{1}{2}(\sigma^2 + Cov(f(X_1), f(X_2)) \tag{3.2.4}$$

Usually, we make the assumption of independent variables, so we get $Var(Y) = \frac{\sigma^2}{n}$. However, we will build Y such that $f(X_i)$ and $f(X_{i+1})$ are negatively correlated for each pair but with independent pairs so that Y_i are independent and identical. It enables us to use Central Limit Theorem and we get : $Var(Y) < \frac{\sigma^2}{n}$.

In our application, we need to construct standard normal random variables Z_i, $-Z_i$ are also standard normal and correlation between them is $\rho = -1$. Then, we can use the following theorem to estimate the expected payoff of the options.

Theorem 3.2.1. *Consider a monotone function f, then $Y_1 = f(Z_1, ..., Z_n)$ and $Y_2 = f(-Z_1, ..., -Z_n)$ with Z_i iid $N(0,1)$ are negatively correlated: $Cov(Y_1, Y_2) < 0$*

	Price	MC$_{100}$	err$_{100}$	MC$_{1000}$	err$_{1000}$	MC$_{5000}$	err$_{5000}$	MC$_{10000}$	err$_{10000}$
S$_0$ = 100 K=95	6.7447	7.8821 (1.0322)	0.1686	7.3114 (0.3210)	0.0840	6.5826 (0.1421)	0.0240	6.7348 (0.1057)	1.5.10^{-3}
S$_0$ = 100 K=90	10.9501	10.7399 (1.1215)	0.0192	11.1422 (0.3597)	0.0175	11.0503 (0.1581)	9.2.10^{-3}	11.0127 (0.1117)	5.7.10^{-3}
S$_0$ = 110 K=100	15.0164	16.9545 (1.4472)	0.1291	15.6856 (0.4881)	0.0446	15.3699 (0.2144)	0.0235	14.9484 (0.1493)	4.5.10^{-3}
S$_0$ = 110 K=90	21.1631	20.9043 (1.4217)	0.0122	21.2639 (0.4346)	4.8.10^{-3}	21.1341 (0.1938)	1.4.10^{-3}	21.1393 (0.1361)	1.1.10^{-3}
S$_0$ = 120 K=100	28.0566	28.4078 (1.6323)	0.0125	28.2745 (0.5354)	7.8.10^{-3}	28.2103 (0.2450)	5.5.10^{-3}	28.0639 (0.1729)	3.0.10^{-4}
S$_0$ = 120 K=110	18.4403	20.1230 (1.7805)	0.0913	18.3566 (0.6010)	4.5.10^{-3}	18.3785 (0.2637)	3.4.10^{-3}	18.4617 (0.1879)	1.2.10^{-3}

Table 3.6: (A-MC)Down-and-out call option with $r_d = 0.08$,$r_f = 0.04$,$T = 0.5$,$\sigma = 0.25$, $B = 95$

Antithetic Variates produced highly promising results for Single Barrier options, notably reducing the standard error as indicated in Table 3.6 (values within parentheses). The percent errors achieved with Antithetic Variates are comparable to those of Control Variates for sample sizes of $n = 100$ and $n = 1000$. However, as the sample size increases beyond this point, the new technique significantly reduces the percent errors.

Results for Double Barrier options presented in Table 3.7 suggest that using Antithetic Variates are a reasonable technique to use to price Double Barrier options using Monte-Carlo methods. The technique made it possible to reduce variance significantly, similarly to the Control Variates technique. Furthermore, the results are more accurate with a percent error of order 10^{-4} for $n = 10000$ simulations. It seems to be a good compromise since the computation is similar to standard Monte-Carlo while enhancing greatly the different types of errors.

	Price	MC_{100}	err_{100}	MC_{1000}	err_{1000}	MC_{5000}	err_{5000}	MC_{10000}	err_{10000}
$\text{B}_d = 50$ $\text{B}_u = 150$	5.4636	5.5535 (0.5884)	0.0165	5.5564 (0.1896)	0.0170	5.4704 (0.0823)	$1.2.10^{-3}$	5.4676 (0.0576)	$7.0.10^{-4}$
$\text{B}_d = 60$ $\text{B}_u = 140$	5.2200	5.5554 (0.5013)	0.0643	5.1176 (0.1616)	0.0196	5.2315 (0.0758)	$2.2.10^{-3}$	5.2233 (0.0550)	$6.0.10^{-4}$
$\text{B}_d = 70$ $\text{B}_u = 130$	4.3806	4.5404 (0.4272)	0.0365	4.1528 (0.1517)	0.0520	4.3647 (0.0660)	$3.6.10^{-3}$	4.3766 (0.0467)	$9.0.10^{-4}$
$\text{B}_d = 80$ $\text{B}_u = 120$	2.4642	2.8276 (0.2974)	0.1475	2.4379 (0.0983)	0.0107	2.4553 (0.0432)	$3.6.10^{-3}$	2.4839 (0.0312)	$8.0.10^{-3}$
$\text{B}_d = 90$ $\text{B}_u = 110$	0.3003	0.3869 (0.0905)	0.2884	0.2637 (0.0293)	0.1219	0.3034 (0.0123)	0.0103	0.3005 (0.0087)	$7.0.10^{-4}$

Table 3.7: (A-MC)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 0.25$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$

3.2.3 Conditional Monte-Carlo

Conditional Monte-Carlo is a specialized simulation technique that offers a refined approach to valuing barrier options. Traditional Monte-Carlo simulations may suffer from inefficiencies when pricing barrier options, as many simulated paths may not cross the barrier. Conditional Monte-Carlo addresses this challenge by focusing only on paths that trigger the barrier condition. Rather than simulating the entire range of possible paths, the technique selectively simulates only those paths that lead to barrier crossing and then use the Garman-Kohlhagen formula to compute the price of the option until expiry. This targeted approach significantly reduces computation time and enhances the precision of option price estimates.

For simplicity and clarity of the explanation, we consider a down-and-out call option. We write the discreet stock prices $S = \{S_{t_0}, S_{t_1}, ..., S_{t_n}\}$. The price of the option at time 0 is the discounted expected payoff:

$$C_{down-out} = C_{GK} - e^{-r_d T} E[\mathbb{1}_{t_c<T}(S_{t_n} - K, 0)^+] \tag{3.2.5}$$

where t_c is the time the stock crosses the down-barrier and C_{GK} is the vanilla call price using the Garman-Kohlhagen model.

Considering $t_c < T$, we can write:

$$E[(S_{t_n} - K, 0)^+] = E[E[(S_{t_n} - K, 0)^+ | c, S_{t_c}]] \tag{3.2.6}$$

Since S follows a log-normal process, we can deduce that:

$$\begin{aligned} E[(S_{t_n} - K, 0)^+ | c = k, S_{t_c = t_k}] &= e^{r_d(T-t_k)} E[e^{-r_d(T-t_k)}(S_{t_n} - K)^+ | S_{t_k} = S] \\ &= S\phi(d_1(T-t_k)) - Ke^{-r_d(T-t_k)}\phi(d_2(T-t_k)) \end{aligned}$$

By taking the expectation, we get the new expression of equation (3.2.6). Conditional Monte-Carlo estimator simulates n paths and average the results:

$$\begin{cases} e^{r_d(T-t_k)} S\phi(d_1(T-t_k)) - Ke^{-r_d(T-t_k)}\phi(d_2(T-t_k)) & t_k < T \\ 0 & t_k \geq T \end{cases} \tag{3.2.7}$$

Then we discount by $e^{-r_d T}$ to get the second term of (3.2.5). The same method can be detailed for other types of Barrier options such as Double Barrier options.

This technique captures the essential features of barrier options while eliminating the need to simulate irrelevant paths. The result might be a more efficient and accurate valuation process, particularly suited for complex barrier option structures and higher-dimensional models.

	Price	MC$_{100}$	err$_{100}$	MC$_{1000}$	err$_{1000}$	MC$_{5000}$	err$_{5000}$	MC$_{10000}$	err$_{10000}$
S$_0$ = 100 K=95	6.7447	8.5435 (1.0155)	0.2667	7.0187 (0.3658)	0.0406	6.9803 (0.1562)	0.0349	6.7985 (0.0993)	8.0.10^{-3}
S$_0$ = 100 K=90	10.9501	12.2243 (1.1142)	0.1164	11.1414 (0.3349)	0.0175	11.0078 (0.1447)	5.3.10^{-3}	11.0097 (0.0992)	5.4.10^{-3}
S$_0$ = 110 K=100	15.0164	14.0856 (1.4175)	0.0620	15.9356 (0.4684)	0.0612	15.4888 (0.2035)	0.0315	15.3756 (0.1394)	0.0239
S$_0$ = 110 K=90	21.1631	19.3004 (1.2265)	0.0880	21.7564 (0.4458)	0.0280	21.4475 (0.1915)	0.0134	21.1995 (0.1203)	1.7.10^{-3}
S$_0$ = 120 K=100	28.0566	28.8653 (1.5126)	0.0288	28.5553 (0.5503)	0.0178	28.2319 (0.2180)	6.2.10^{-3}	28.0304 (0.1583)	9.0.10^{-4}
S$_0$ = 120 K=110	18.4403	19.2856 (1.6277)	0.0458	19.1358 (0.5713)	0.0377	18.6574 (0.2451)	0.0118	18.1059 (0.1572)	0.0181

Table 3.8: (C-MC)Down-and-out call option with $r_d = 0.08$,$r_f = 0.04$,$T = 0.5$,$\sigma = 0.25$, $B = 95$

Conditional Monte Carlo is a powerful technique to reduce variance as shown in the results of Table 3.8. Standard errors is reduced by 30% to 60% for each value, which is similar to the improvements made with Antithetic Variates. However, this improvement is not changing much the percent errors with the theoretical prices of Single Barrier options. Indeed, we get similar or worse errors than standard Monte-Carlo and the other Variance Reduction techniques.

	Price	MC$_{100}$	err$_{100}$	MC$_{1000}$	err$_{1000}$	MC$_{5000}$	err$_{5000}$	MC$_{10000}$	err$_{10000}$
B$_d$ = 50 B$_u$ = 150	5.4636	5.9598 (0.5773)	0.0908	5.6198 (0.1772)	0.0286	5.6321 (0.0729)	0.0308	5.4756 (0.0615)	2.2.10^{-3}
B$_d$ = 60 B$_u$ = 140	5.2200	5.7065 (0.5118)	0.0932	5.3532 (0.1818)	0.0255	5.2763 (0.0735)	0.0108	5.2380 (0.0530)	3.4.10^{-3}
B$_d$ = 70 B$_u$ = 130	4.3806	4.1145 (0.4281)	0.0607	4.2764 (0.1503)	0.0238	4.4751 (0.0672)	0.0216	4.3839 (0.0458)	8.0.10^{-4}
B$_d$ = 80 B$_u$ = 120	2.4642	2.1907 (0.3003)	0.1110	2.2136 (0.0793)	0.1017	2.4106 (0.0503)	0.0218	2.4451 (0.0274)	7.8.10^{-3}
B$_d$ = 90 B$_u$ = 110	0.3003	0.3975 (0.0974)	0.3237	0.3675 (0.0262)	0.2238	0.3354 (0.0155)	0.1169	0.3146 (0.0093)	0.0476

Table 3.9: (C-MC)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 0.25$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$

The same conclusions can be made for Double Barrier options. In the analysis, it was found that when applying Conditional Monte-Carlo, there was a decrease in the standard error. This means that the estimation precision improved when using Conditional Monte-Carlo, which is a positive result indicating that the technique has potential in reducing uncertainty in the estimated option values. However, despite the reduction in standard error, the percent error did not see a significant improvement when compared to other previously employed techniques. The percent error is a crucial metric because it provides insights into the accuracy of the estimations relative to the true values. The fact that the percent error did not decrease substantially suggests that while Conditional Monte-Carlo may reduce uncertainty, it may not necessarily provide more accurate estimates for Double Barrier options compared to other methods.

3.2.4 Importance Sampling

Importance sampling is a sophisticated technique widely used in Monte-Carlo simulations to enhance the accuracy and efficiency of estimating rare or unlikely events, such as extreme outcomes in financial models. In traditional Monte-Carlo methods, each simulated scenario contributes equally to the final estimate, even if some scenarios are relatively insignificant or rarely occur. Importance sampling addresses this inefficiency by intelligently biasing the simulation towards scenarios that have a higher impact on the desired estimation.

The key idea behind importance sampling is to choose a different probability distribution, known as the "importance distribution," that guides the simulation towards relevant scenarios. By sampling more frequently from regions where the outcome of interest is more likely to occur, importance sampling reduces the variance and increases the precision of Monte-Carlo estimates. This technique is particularly beneficial in financial modeling, where rare events, like extreme market movements, are essential to capture accurately. In out context, we aim at changing the probability distribution of the stock to increase the probability of crossing the barrier.

In the Monte-Carlo framework, we want to estimate $\theta = E_X[f(X)] = \int f(x)p_X(x)dx$ for some function f with density p_X. If we know another random variable Y with density p_Y, then:

$$\begin{aligned}\theta = E_X[f(X)] &= \int f(x)p_X(x)\,dx \\ &= \int f(x)\frac{p_X(x)}{p_Y(x)}p_Y(x)\,dx \\ &= E_Y[f(x)\frac{p_X(x)}{p_Y(x)}p_Y(x)] \\ &= E_Y[f^*(X)]\end{aligned} \tag{3.2.8}$$

The expected value is taken with respect to the other random variable.
$\frac{p_X(x)}{p_Y(x)}$ is called the likelihood ratio and variance reduction depends on the choice of the density $p_Y(x)$. The variance is given by:

$$\begin{aligned}Var_Y[f^*(X)] &= E_Y[(f^*(X))^2] - \theta^2 \\ &= \int f(x)^2\frac{p_X(x)}{p_Y(x)}p_X(x)\,dx - \theta^2 \\ &= E_X[f(X)^2\frac{p_X(X)}{p_Y(X)}] - \theta^2\end{aligned} \tag{3.2.9}$$

As the objective is to reduce the variance, let us evaluate the difference betwenn the variances:

$$Var_X[f(X)] - Var_Y[f^*(x)] = \int f(x)^2[1 - \frac{p_X(x)}{p_Y(x)}]p_X(x)\,dx \tag{3.2.10}$$

Variance reduction is ensured if the integral defined before is positive, so we need to choose Y such that:

$$\begin{cases} p_X(x) > p_Y(x) & \text{if } f(x)^2p_X(x) \text{ small} \\ p_X(x) < p_Y(x) & \text{if } f(x)^2p_X(x) \text{ large} \end{cases}$$

3.2.5 Combination of methods

Ross and Shanthikumar developed in [5] the idea that Importance Sampling and Conditional Monte Carlo could be used together. This is particularly relevant for barrier options. Importance Sampling can be used until the price crosses the barrier, after that we apply conditional estimator.

Again, we assume lognormal model for the price process. The mean is $\mu = (r - \frac{\sigma^2}{2})dt$ and variance $\sigma_X = \sigma^2 dt$. The joint density function of the multivariate normal formula is:

$$f(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}e^{-\frac{(x-\mu)^T\Sigma^{-1}(x-\mu)}{2}} \tag{3.2.11}$$

with Σ is the covariance matrix.

We also select a parameter β to diminish the mean in order to make the crossing barrier a more likely event, we name this new joint density g.

We rewrite the same conditional expectation as in the conditional Monte Carlo with these new density functions.

$$\begin{aligned} E_g[\frac{f(X)}{g(X)}\mathbb{1}_{t_c<T}(S_{t_n}-K)^+|c,S_{t_c}] &= \frac{f(x_1,...,x_c)}{g(x_1,...,x_c)}E_g[\frac{f(X_{c+1},...,X_n)}{g(X_{c+1},...,X_n)}\mathbb{1}_{t_c<T}(S_{t_n}-K)^+|c,S_{t_c}] \\ &= \frac{f(x_1,...,x_c)}{g(x_1,...,x_c)}E_f[\mathbb{1}_{t_c<T}(S_{t_n}-K)^+|c,S_{t_c}] \\ &= \frac{f(x_1,...,x_c)}{g(x_1,...,x_c)}e^{r(T-t_c)}C_{GK}(S_{t_c},K,T-t_c) \end{aligned} \tag{3.2.12}$$

with

$$\begin{aligned} \frac{f(x_1,...,x_c)}{g(x_1,...,x_c)} &= exp\{-\frac{1}{2\sigma^2 dt}\sum_{i=1}^{c}[(x_i-\mu)^2-(x_i-\mu+b)^2]\} \\ &= exp\{\frac{1}{2\sigma^2 dt}\sum_{i=1}^{c}[2(x_i-\mu)b+b^2]\} \\ &= exp\{\frac{b}{\sigma^2 dt}\sum_{i=1}^{c}x_i+\frac{cb^2}{2\sigma^2 dt}-\frac{cb}{\sigma^2}(r-\frac{\sigma^2}{2})\} \end{aligned}$$

	Price	MC$_{100}$	err$_{100}$	MC$_{1000}$	err$_{1000}$	MC$_{5000}$	err$_{5000}$	MC$_{10000}$	err$_{10000}$
S$_0 = 100$ K=95	6.7447	8.6578 (1.0317)	0.2836	7.2135 (0.2541)	0.0695	6.9766 (0.1132)	0.0344	6.7997 (0.0853)	8.2.10^{-3}
S$_0 = 100$ K=90	10.9501	12.2179 (0.8827)	0.1158	11.2351 (0.0935)	0.0260	11.1756 (0.0671)	0.0206	10.9928 (0.0479)	3.9.10^{-3}
S$_0 = 110$ K=100	15.0164	13.9876 (0.9913)	0.0685	15.2313 (0.3546)	0.0143	15.3798 (0.1679)	0.0242	15.2354 (0.0994)	0.0146
S$_0 = 110$ K=90	21.1631	19.1563 (0.5428)	0.0948	21.7734 (0.1455)	0.0288	21.4461 (0.0891)	0.0134	21.1999 (0.0507)	1.7.10^{-3}
S$_0 = 120$ K=100	28.0566	28.8453 (0.6315)	0.0281	28.5102 (0.3418)	0.0162	28.1796 (0.1002)	4.4.10^{-3}	28.0305 (0.0672)	9.0.10^{-4}
S$_0 = 120$ K=110	18.4403	19.2728 (1.4427)	0.0451	19.1334 (0.4495)	0.0376	18.6613 (0.2049)	0.0120	18.1023 (0.1321)	0.0183

Table 3.10: (IS-MC)Down-and-out call option with $r_d = 0.08$,$r_f = 0.04$,$T = 0.5$,$\sigma = 0.25$, $B = 95$

Importance sampling has demonstrated enhancements to Conditional Monte Carlo compared to the preceding section. Notably, the resulting confidence intervals have decreased in size, indicating a higher level of precision in our estimates. Additionally, the percent errors, while showing improvement and tending toward zero, have yet to rival the performance achieved by techniques like Antithetic Variates, for instance.

	Price	MC_{100}	err_{100}	MC_{1000}	err_{1000}	MC_{5000}	err_{5000}	MC_{10000}	err_{10000}
$\mathrm{B}_d = 50$ $\mathrm{B}_u = 150$	5.4636	5.8532 (0.6002)	0.0713	5.6112 (0.1773)	0.0270	5.6299 (0.0665)	0.0304	5.4602 (0.0458)	$6.0.10^{-4}$
$\mathrm{B}_d = 60$ $\mathrm{B}_u = 140$	5.2200	5.7123 (0.4883)	0.0943	5.3423 (0.1655)	0.0234	5.2754 (0.0643)	0.0106	5.2345 (0.0410)	$2.8.10^{-3}$
$\mathrm{B}_d = 70$ $\mathrm{B}_u = 130$	4.3806	4.1401 (0.4014)	0.0549	4.2891 (0.1318)	0.0209	4.4561 (0.0945)	0.0172	4.3845 (0.0395)	$9.0.10^{-4}$
$\mathrm{B}_d = 80$ $\mathrm{B}_u = 120$	2.4642	2.7194 (0.2761)	0.1036	2.5891 (0.1045)	0.0507	2.4173 (0.0382)	0.0190	2.4387 (0.0277)	0.0103
$\mathrm{B}_d = 90$ $\mathrm{B}_u = 110$	0.3003	0.3546 (0.0965)	0.1808	0.3241 (0.0314)	0.0793	0.3129 (0.0115)	0.0420	0.3087 (0.0061)	0.0280

Table 3.11: (IS-MC)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 0.25$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$

For Double Barrier options, combining Conditional Monte Carlo with Importance Sampling improved on average the accuracy. However, the overall conclusions closely align with those of Single Barrier options. This suggests that while the combined technique shows promise for enhanced precision, it still faces similar challenges compared to Single Barrier options.

3.3 Window Barrier options

Previously, we analyzed the results of Monte Carlo methods for always-active barriers. Now, let us shift our focus to Window Barrier options. Since closed-form solutions are not available in the literature for these options, Monte Carlo simulations can provide valuable approximations and confidence intervals for pricing. We will employ both standard Monte Carlo simulations and the Antithetic Variates technique for our estimations. Antithetic Variates can be chosen since it appears to be the most efficient method used for standard Barrier options in the previous sections.

Let us examine a KOKO option with fixed parameters, except for the time window during which the barriers are active. Initially, we assumed that both barriers were active in the same window, but we can generalize this by considering specific time windows for each barrier.

Start	End	Price	MC_{5000}	MC^A_{5000}
0	0	11.7343	11.7213 (0.2738)	11.7438 (0.1957)
0.45	0.55		7.0135 (0.2783)	6.9928 (0.1955)
0.25	0.75		4.4329 (0.2686)	4.3761 (0.1966)
0.25	1		2.2469 (0.2722)	2.2259 (0.1984)
0	0.75		4.2096 (0.0792)	4.2013 (0.0554)
0	1	2.1018	2.2677 (0.0795)	2.1290 (0.0546)

Table 3.12: (MC)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 1$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$, $B_u = 130$,$B_d = 70$

In Table 3.12, we have compiled various results for increasing time windows. In the 'Price' column, you will find the theoretical prices for either the standard call option when the barrier is never active or the standard Double Barrier option when it is always active. It becomes apparent that Monte Carlo simulations may tend to overprice the options, particularly when the window is small. This is evident as the prices exceed those of vanilla options, a situation that is typically not observed in practice.

Chapter 4

Lattice methods

Lattice methods are computational techniques used to value options and other derivatives by constructing a lattice, which is essentially a grid-like representation of the underlying asset's possible price movements over discrete time intervals. The two most common types of lattice methods are the Binomial Tree and the Trinomial Tree.

In a lattice, each node represents a possible price level of the underlying asset at a specific point in time. Starting from the initial price, the lattice evolves over time by considering possible upward and downward movements based on the asset's volatility. The key concept is to calculate the option's value at each node by working backward through the lattice, determining the expected payoff at expiration and discounting it to the present.

Lattice methods are flexible and can accommodate various option types, including European and American options, as well as more complex derivatives. They offer a balance between accuracy and computational efficiency, making them suitable for a wide range of financial instruments.

Pricing barrier options can be challenging due to the added complexity introduced by the barrier conditions. Lattice methods, like the Trinomial Tree, are often used to model the price dynamics of barrier options. In a lattice, the barrier levels can be incorporated into the calculations by adjusting the probabilities of price movements, it will analyze with the Bino-Trinomial model.

4.1 Binomial Tree

The binomial method was first introduced by Cox, Ross and Rubinstein in [6]. They showed the construction of a recombining tree to discretize the geometric Brownian motion. Considering an important number of time steps, binomial tree is equivalent to Black-Scholes formula for the case of European options pricing. The method is particularly relevant for American options, or exotic options, where no closed-form solution exists.

At each step of the tree, the asset price can over the time step Δt either move up by a fixed amount u with a probability p_u or decrease by a fixed amount d with a probability $p_d = 1 - p_u$. We assume there are n time steps in the tree. Figure 4.1 depicts a two step Binomial Tree.

We define i and j as the numbers of price and time steps until the considered node (j, i). As a consequence, the number of paths that lead to the node is $\binom{j}{i}$ with probability $\binom{j}{i} p_u^i p_d^{j-i}$. At each time point j, the random variable S_n (describing the stock price) takes value on $\{S_0 u^j, S_0 u^{j-1} d ..., S_0 d^j\}$. The discrete stock price process follows $S_n = S_0 \prod_{i=1}^{n} \xi_i$ where ξ_i are iid random variables defined as follows:

$$\xi_i = \begin{cases} u & \text{with probability } q_u \\ d & \text{with probability } q_d \end{cases}$$

A simple approach is to use backward induction to price the option with a binomial tree.

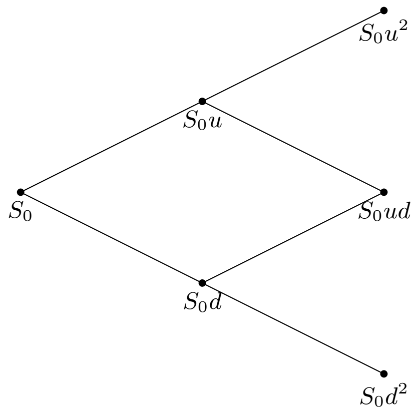

Figure 4.1: A two-step Binomial Tree

Theorem 4.1.1. *Consider $V^j = e^{-r_d(n-j)\Delta t}E[g(S_n)|\mathcal{F}_j]$ the time-j fair value of the European call option. We have the following recursive algorithm:*

$$V^j = \begin{cases} g(S_n) & j = n \\ e^{-r_d\Delta t}E[V^{j+1}|\mathcal{F}_j] & j \leq n-1 \end{cases} \tag{4.1.1}$$

The up and down factors u and d and their probabilities must match the first two moments of the price distribution.There are only two equations and three parameters to determine (knowing one probability p_u makes it possible to know also $p_d = 1 - p_u$. Cox-Ross-Rubinstein (CRR) set the additional equation to $ud = 1$. The CRR model has parameters:

$$\begin{aligned} u &= e^{\sigma\sqrt{\Delta t}} \\ d &= e^{-\sigma\sqrt{\Delta t}} \\ p_u &= \frac{e^{(r_d - r_f)\Delta t} - d}{u - d} \end{aligned} \tag{4.1.2}$$

Under this configuration and with the previous theorem, we define the following recursive formula:

$$V_i^j = e^{-r_d\Delta t}[p_u V_i^{j+1} + p_d V_{i+1}^{j+1}] \tag{4.1.3}$$

At the end of the recursive algorithm, we obtain the time-zero option value.

In the standard CRR tree, the local volatility is constant for each time step and converges to the input volatility as n becomes large:

$$\sigma_i^j = \frac{1}{\sqrt{\Delta t}}\sqrt{p_u p_d}\ln u^2 \tag{4.1.4}$$

Theorem 4.1.2. *The price of a Vanilla option for an $n \in \mathbb{N}$ step tree is given by,*

$$V = \mathrm{e}^{-r_d)n\Delta t}\left[\sum_{j=0}^{n}\binom{n}{j}p_u^j p_d^{n-j}V^j\right]. \tag{4.1.5}$$

One remark to make with that CRR model is that in some cases where $\sigma < |(r_d - r_f)\sqrt{\Delta t}|$, negative probabilities can occur (see formulas (4.1.2)). First, it is inconsistent as far as probability theory is concerned, but on top of that it could prevent the model from covering relevant events.

4.1.1 Single Barrier Options

In this section, we present several results for Barrier options using the Binomial Tree lattice. First, we tested on Single Barrier to estimate accuracy based on analytical formulas presented before.

	Price	BT_{100}	err_{100}	BT_{1000}	err_{1000}	BT_{5000}	err_{5000}	BT_{10000}	err_{10000}
$\mathrm{S}_0 = 100$ B=95	6.7447	6.9232	0.0265	7.2117	0.0692	6.8689	0.0184	6.9234	0.0265
$\mathrm{S}_0 = 100$ B=90	10.9501	10.9894	$3.6.10^{-3}$	10.9961	$4.2.10^{-3}$	11.0644	0.0104	10.9881	$3.5.10^{-3}$
$\mathrm{S}_0 = 110$ B=100	15.0164	16.0780	0.0707	15.5487	0.0354	15.2386	0.0148	15.0318	$1.0.10^{-3}$
$\mathrm{S}_0 = 110$ B=90	21.1631	21.4149	0.0119	21.1770	$7.0.10^{-4}$	21.2066	$2.1.10^{-3}$	21.1835	$1.0.10^{-3}$
$\mathrm{S}_0 = 120$ B=100	28.0566	28.7007	0.0230	28.1782	$4.3.10^{-3}$	28.0668	$4.0.10^{-4}$	28.1432	$3.1.10^{-3}$
$\mathrm{S}_0 = 120$ B=110	18.4403	18.6470	0.0112	18.8119	0.0202	18.5149	$4.0.10^{-4}$	18.6502	0.0114

Table 4.1: (BT)Down-and-out call option with $r_d = 0.08$,$r_f = 0.04$,$T = 0.5$,$\sigma = 0.25$, $K = 90$

We can observe that the calculated prices closely align with the theoretical values. Remarkably, the Binomial Tree model achieved an error of less than 0.1 with fewer than 1000 steps. Nevertheless, a systematic error becomes evident, especially when the barrier closely approaches the stock price. To delve deeper into this matter, let's focus on a specific trade.

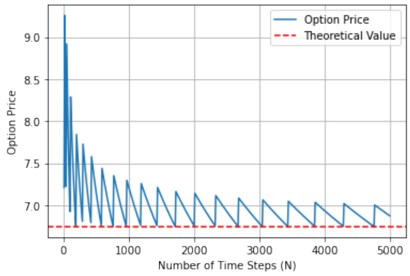

Figure 4.2: Convergence of Single Barrier Option Price with increasing Tree Steps in the Binomial Tree for $\mathrm{S}_0 = 100, K = 90, B = 95, r_d = 0.08, r_f = 0.04, T = 0.5, \sigma = 0.25$

These results can be also obtained and generalized with Double Barrier options which is our main interest in the thesis. As observed in Figure 4.2, the error between the obtained price and the theoretical value is not linear in the number of steps. However, the shape of the curve is rapidly decreasing before $n = 1000$. Jhihrong Lin and Ken Palmer proved in [7] that the binomial tree is $\mathcal{O}(\frac{1}{\sqrt{n}})$ convergent.

Regarding computational time, which can become an issue as the number of nodes increases, we have the following values:

	T_{100}	T_{1000}	T_{5000}	T_{10000}
Time(s)	0.010	0.96	24.7	97.5

Table 4.2: Average computation time for Lattice methods

As an initial point of comparison, these computational times are notably shorter than those observed with standard Monte Carlo simulations. This efficiency underscores the advantages of the method in terms of speed and computational resource utilization.

4.1.2 Double Barrier Options

In this section, we will consider more complex barrier options. First, we can work on standard Double Barrier options, where we have theoretical values.

	Price	BT$_{100}$	err$_{100}$	BT$_{1000}$	err$_{1000}$	BT$_{5000}$	err$_{5000}$	BT$_{10000}$	err$_{10000}$
B$_d = 50$ B$_u = 150$	5.4636	5.4671	$6.0.10^{-4}$	5.4656	$4.0.10^{-4}$	5.4653	$3.0.10^{-4}$	5.4647	$2.0.10^{-4}$
B$_d = 60$ B$_u = 140$	5.2200	5.2270	$1.3.10^{-3}$	5.2422	$4.3.10^{-3}$	5.2272	$1.4.10^{-3}$	5.2264	$1.2.10^{-3}$
B$_d = 70$ B$_u = 130$	4.3806	4.3879	$1.7.10^{-3}$	4.4236	$9.8.10^{-3}$	4.3986	$4.1.10^{-3}$	4.3833	$6.0.10^{-4}$
B$_d = 80$ B$_u = 120$	2.4642	2.6089	0.0587	2.5622	0.0398	2.5083	0.0179	2.4696	$2.2.10^{-3}$
B$_d = 90$ B$_u = 110$	0.3003	0.3974	0.3233	0.3484	0.1602	0.3044	0.0137	0.3144	0.0470

Table 4.3: (BT)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 0.25$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$

In Table 4.3, we encounter an intriguing insight into the behavior of lattice methods when applied to barrier options. Specifically, when the lower and upper barriers are situated in close proximity to each other, a discernible pattern emerges. Take, for instance, the case where $B_d = 90$ and $B_u = 110$. Here, we observe that the price estimations are not as accurate as in other scenarios.

This phenomenon can be attributed to the nature of barrier options. When the barriers are nearly adjacent, the underlying asset's price often hovers around this region, making it more challenging for the lattice method to accurately capture the complex dynamics. Consequently, the estimations exhibit a degree of imprecision in such cases.

In contrast, when the barriers are sufficiently separated, a remarkable transformation occurs. The lattice method rapidly converges to highly accurate estimations. The errors encountered in these scenarios are typically of the order of 10^{-3}, which signifies a remarkable level of precision.

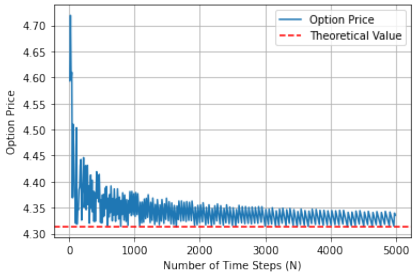

Figure 4.3: Convergence of Double Barrier Option Price with increasing Tree Steps in the Binomial Tree S$_0 = 100, K = 100, B_u = 130, B_l = 70, , r_d = 0.1, r_f = 0.05, T = 0.25, \sigma = 0.25$

In Figure 4.3, we illustrate the price evolution of a Double Barrier option. The reduction in error observed here appears to follow a similar trend to the one depicted in Figure 4.2. This consistent pattern highlights the effectiveness of the method in reducing errors and enhancing the precision of option pricing.

4.2 Trinomial Tree

Trinomial trees, a concept similar to that of binomial trees, were introduced by Boyle in [8]. In contrast to Binomial Trees, where the stock can take two paths at any node, Trinomial trees offer three possible paths. This unique characteristic allows us to achieve the same level of accuracy while reducing the number of time steps. The central branch, where the asset price remains unchanged, is commonly denoted as m. This m branch represents a stable price movement within the Trinomial Tree framework.

The discrete stock price process follows $S_n = S_0 \prod_{i=1}^{n} \xi_i$ where ξ_i are iid random variables defined as follows:

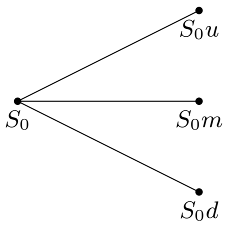

Figure 4.4: A one step trinomial tree

$$\xi_i = \begin{cases} u & \text{with probability } p_u \\ m & \text{with probability } p_m \\ d & \text{with probability } p_d \end{cases}$$

We have a variety of models at our disposal, and for our analysis, we opted for the Kamrad-Ritchken parametrization, which was developed in [9].This parametrization offers a robust framework for our analysis.

$$u = e^{\lambda\sigma\sqrt{\Delta t}}, m = 1, d = e^{-\lambda\sigma\sqrt{\Delta t}}$$

$$p_u = \frac{1}{2\lambda^2} + \frac{(r_d - r_f - \frac{\sigma^2}{2})\sqrt{\Delta t}}{2\lambda\sigma}, p_m = 1 - \frac{1}{\lambda^2}, p_d = \frac{1}{2\lambda^2} - \frac{(r_d - r_f - \frac{\sigma^2}{2})\sqrt{\Delta t}}{2\lambda\sigma}$$

where $\lambda \geq 1$ is a free parameter.

4.2.1 Single Barrier options

The selection of the parameter λ is a crucial aspect that warrants thorough investigation. Edward Omberg, as outlined in [10], conducted a comprehensive analysis focused on enhancing the Trinomial process using what he referred to as a 'sharpened' approach. This particular model addresses the question of discontinuity in the first derivative of the option's pricing function, and the corresponding parameter, λ, is set to a specific value, namely $\lambda = \sqrt{\frac{\pi}{2}}$. What makes this particularly interesting is the striking similarity between Boyle and Omberg's models. In fact, for very small values of $\sqrt{\Delta t}$ and the choice of λ as described above, the Trinomial Tree method exhibits uniform probabilities. Boyle's work, discussed in [8] , demonstrated that the Trinomial Tree model is most effective when uniform probabilities are employed.

This convergence towards uniform probabilities, as observed in both Boyle and Omberg's research, highlights the efficiency and effectiveness of the Trinomial Tree model, particularly when the chosen parameters align with the goal of achieving uniformity in probability distributions.

	Price	TT_{100}	err_{100}	TT_{1000}	err_{1000}	TT_{5000}	err_{5000}	TT_{10000}	err_{10000}
$\text{S}_0 = 100$ B=95	6.7447	7.0599	0.0467	6.7735	$4.3.10^{-3}$	6.9024	0.0234	6.8737	0.0191
$\text{S}_0 = 100$ B=90	10.9501	10.9292	$1.9.10^{-3}$	10.9357	$1.3.10^{-3}$	10.9709	$1.9.10^{-3}$	10.9330	$1.6.10^{-3}$
$\text{S}_0 = 110$ B=100	15.0164	15.3094	0.0195	15.2302	0.0142	15.2262	0.0140	15.0204	$3.0.10^{-4}$
$\text{S}_0 = 110$ B=90	21.1631	21.5106	0.0164	21.2918	$6.1.10^{-3}$	21.2051	$2.0.10^{-3}$	21.1740	$5.0.10^{-4}$
$\text{S}_0 = 120$ B=100	28.0566	28.9394	0.0315	28.1167	$2.1.10^{-3}$	28.0797	$8.0.10^{-4}$	28.0958	$1.4.10^{-3}$
$\text{S}_0 = 120$ B=110	18.4403	18.4953	$3.0.10^{-3}$	18.5427	$5.6.10^{-3}$	18.4579	$1.0.10^{-3}$	18.6102	$9.2.10^{-3}$

Table 4.4: (TT)Down-and-out call option with $r_d = 0.08$,$r_f = 0.04$,$T = 0.5$,$\sigma = 0.25$, $K = 90$

The results presented in Table 4.4, specifically for Down-and-Out call options, exhibit striking similarities to those obtained using the Binomial Tree method. Notably, this new lattice method effectively reduces errors, offering the advantage of achieving highly accurate results with a reduced number of computational steps.

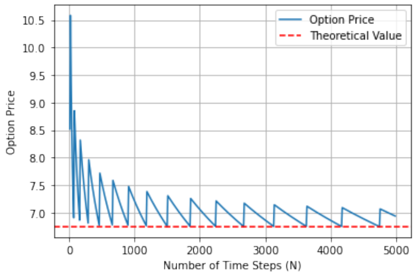

Figure 4.5: Convergence of Single Barrier Option Price with increasing Tree Steps in the Trinomial Tree for $\mathrm{S}_0 = 100, K = 90, B = 95, r_d = 0.08, r_f = 0.04, T = 0.5, \sigma = 0.25$

When analyzing a specific trade, a similar dynamic unfolds as seen with the Binomial Tree method. Rather than a smooth reduction, we discern a discernible pattern of error reduction. Interestingly, this pattern is not characterized by a constant period, but as the number of simulations increases, it tends to stabilize and display reduced variability.

4.2.2 Double Barrier Options

Similar to our exploration with the Binomial Tree model, we have conducted tests on advanced Barrier options. Our focus is on standard Double Barrier options.

	Price	TT_{100}	err_{100}	TT_{1000}	err_{1000}	TT_{5000}	err_{5000}	TT_{10000}	err_{10000}
$\mathrm{B}_d = 50$ $\mathrm{B}_u = 150$	5.4636	5.4687	$9.0.10^{-4}$	5.4650	$3.0.10^{-4}$	5.4671	$6.0.10^{-4}$	5.4640	$1.0.10^{-4}$
$\mathrm{B}_d = 60$ $\mathrm{B}_u = 140$	5.2200	5.2782	0.0111	5.2237	$7.0.10^{-4}$	5.2220	$6.0.10^{-4}$	5.2222	$4.0.10^{-4}$
$\mathrm{B}_d = 70$ $\mathrm{B}_u = 130$	4.3806	4.4711	0.0207	4.3865	$1.3.10^{-3}$	4.4033	$5.2.10^{-3}$	4.3952	$3.3.10^{-3}$
$\mathrm{B}_d = 80$ $\mathrm{B}_u = 120$	2.4642	2.6496	0.0752	2.4946	0.0123	2.5093	0.0183	2.4921	0.0113
$\mathrm{B}_d = 90$ $\mathrm{B}_u = 110$	0.3003	0.5321	0.7719	0.3601	0.1991	0.3305	0.1006	0.3068	0.0216

Table 4.5: (TT)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 0.25$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$

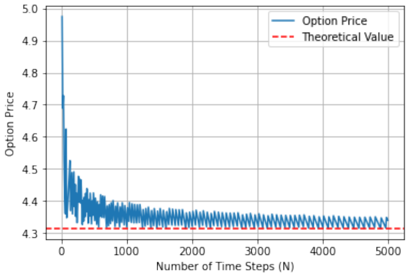

Figure 4.6: Convergence of Double Barrier Option Price with increasing Tree Steps in the Trinomial Tree $\mathrm{S}_0 = 100, K = 100, B_u = 130, B_l = 70, , r_d = 0.1, r_f = 0.05, T = 0.25, \sigma = 0.25$

The results for Double Barrier options are more contrasting when comparing to Binomial Tree results. While we do find similarities between the results, there are no substantial improvements in terms of accuracy and convergence speed. Surprisingly, in certain cases, we even encounter less accurate pricing outcomes.

4.3 Improved Tree for Barrier options

As evident from the preceding sections, both Binomial and Trinomial trees exhibit a notable bias in valuing barrier options. This bias stems from the fact that barrier values are not genuine 'levels' within these trees, leading to erratic convergence in their pricing. One idea presented by Dai-Lyuu in [11] is the Bino-Trinomial Tree. In this model, the barrier is adjusted to be at a correct price level.

Considering a standard Double Barrie option with lower barrier B_d and upper barrier B_u. We take the logarithm of each barrier defining $l = \log(\frac{B_d}{S_0})$ and $b = \log(\frac{B_u}{S_0})$.

Hence, one must define another time step as well as another space step:

$$\Delta T = (\frac{b-l}{2k\sigma})^2$$

with k an integer such that $k = \lceil \frac{b-l}{2\sqrt{\Delta T}\sigma} \rceil$. The corresponding space step is: $\sigma\sqrt{\Delta T}$

The modified CCR model has now for parameters:

$$\begin{aligned} u &= e^{\sigma\sqrt{\Delta T}} \\ d &= e^{-\sigma\sqrt{\Delta T}} \\ p_u &= \frac{e^{(r_d - r_f)\Delta t} - d}{u-d} \end{aligned} \tag{4.3.1}$$

The layers of the grid are now matching the barrier values and there are $\lfloor \frac{T}{\Delta T} \rfloor$ steps . At the end of the tree we get three nodes N_1, N_2 and N_3 and a remaining amount of time $\Delta T'$ such as $\Delta T \leq \Delta T' < 2\Delta T$ and:

$$\Delta T' = T - (\lfloor \frac{T}{\Delta T} \rfloor - 1)\Delta T$$

After that, we need to calculate the three branching probabilities to them. The variance Var and the mean μ of the stock price needs to be matched and we need to define three probabilities p_1, p_2 and p_3.

$$\begin{aligned} \mu &= (r_d - r_f - \frac{\sigma^2}{2})\Delta T' \\ Var &= \sigma^2 \Delta T' \end{aligned} \tag{4.3.2}$$

In Figure 4.7, we observe that the middle node N_2 exhibits the closest logarithmic return to μ, denoted as $\hat{\mu}$, when compared to the other two nodes. This figure provides detailed insights into the initial time step of the new model.

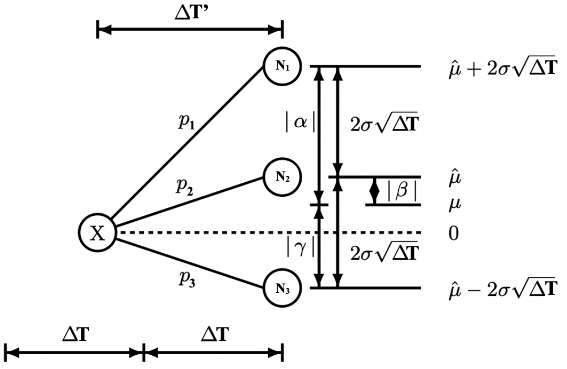

Figure 4.7: Close-up to the first time step of the Bino-Trinomial Tree

In Figure 4.7, α, β, and γ represent the differences in the mean logarithmic returns among each node. By properties of the three, we have the following formulas:

$$\begin{cases} \alpha & = \hat{\mu} + 2\sigma\sqrt{\Delta T} - \mu = \beta + 2\sigma\sqrt{\Delta T} \\ \beta & = \hat{\mu} - \mu \\ \gamma & = \hat{\mu} - 2\sigma\sqrt{\Delta T} - \mu = \beta - 2\sigma\sqrt{\Delta T} \end{cases} \tag{4.3.3}$$

To compute the price X in Figure 4.7, it is imperative to determine the probabilities p_1, p_2, and p_3. We solve this system of equations to match the requirement of a Tree model:

$$\begin{cases} p_1\alpha + p_2\beta + p_3\gamma & = 0 \\ p_1\alpha^2 + p_2\beta^2 + p_3\gamma^2 & = \sigma^2\Delta T' \\ p_1 + p_2 + p_3 & = 1 \end{cases} \tag{4.3.4}$$

	Price	BTT$_{100}$	err$_{100}$	BTT$_{1000}$	err$_{1000}$	BTT$_{5000}$	err$_{5000}$	BTT$_{10000}$	err$_{10000}$
B$_d = 50$ B$_u = 150$	5.4636	5.4673	$7.0.10^{-4}$	5.4623	$2.0.10^{-4}$	5.4641	$1.0.10^{-4}$	5.4639	$1.0.10^{-4}$
B$_d = 60$ B$_u = 140$	5.2200	5.2713	$9.8.10^{-3}$	5.2224	$5.0.10^{-4}$	5.2221	$4.0.10^{-4}$	5.2215	$3.0.10^{-4}$
B$_d = 70$ B$_u = 130$	4.3806	4.4652	0.0193	4.3881	$1.7.10^{-3}$	4.4059	0.0058	4.3980	$4.0.10^{-3}$
B$_d = 80$ B$_u = 120$	2.4642	2.6344	0.0691	2.4875	$9.5.10^{-3}$	2.4938	0.0120	2.4688	$1.9.10^{-3}$
B$_d = 90$ B$_u = 110$	0.3003	0.4120	0.3720	0.3255	0.0839	0.3029	$8.7.10^{-3}$	0.3018	$5.0.10^{-3}$

Table 4.6: (BTT)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 0.25$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$

The results obtained from the Bino-Trinomial Tree model strongly suggest that it represents a compelling improvement over the standard Binomial Tree. This improvement is achieved through subtle node modifications. Although errors are more pronounced for fewer than 100 time steps, given a sufficiently large number of simulations, these errors appear to diminish significantly. In cases where the barriers are in close proximity, such as $B_d = 90$ and $B_u = 110$, we previously encountered challenges in substantially reducing errors. However, with the implementation of the Bino-Trinomial Tree, this issue is effectively addressed, resulting in an approximate tenfold reduction in errors for the same number of time steps.

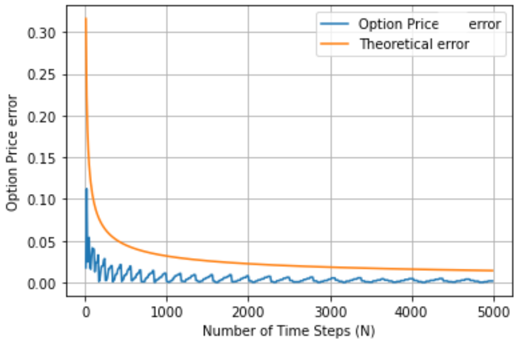

Figure 4.8: Evolution of the percent error for the BTT Tree model S$_0 = 100, K = 100, B_u = 130, B_d = 70, r_d = 0.1, r_f = 0.05, T = 0.25, \sigma = 0.25$

In Figure 4.8, both the percent error of the Bino-Trinomial Tree and the theoretical error of the Binomial Tree are depicted. The data unequivocally demonstrates the superior performance of the improved Tree, effectively reducing errors.

One significant improvement introduced by the BTT model, as previously mentioned, is its ability to address a bias that commonly occurred in the pricing of options when dealing with barriers close to the stock price. In standard Tree methods, there was a tendency for the nodes of the trees to inaccurately capture the barrier levels. Figure 4.9 provides a detailed analysis of the last example from Table 4.6, comparing all tree methods up to 1000 time steps. It is evident that the enhancements introduced by the Dai-Lyuu model have a noticeable impact on the pattern of error reduction, effectively mitigating the limitations observed in standard tree methods.

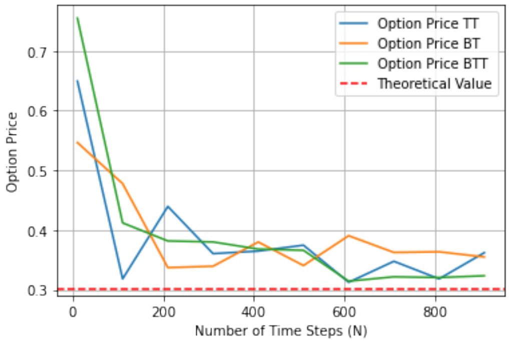

Figure 4.9: Comparision between Tree methods for S=100, K=90,B$_d = 90, B_u = 110, r_d = 0.1, r_f = 0.05, T = 0.25, \sigma = 0.25$

4.4 Window Barrier options

In the preceding sections of this chapter, we conducted an analysis of the estimations provided by Lattice methods for always-active barriers. As in the Monte-Carlo estimation chapter, our current objective is to derive price approximations for Window Barrier options using all three methods. Specifically, we focus on the KOKO call option and vary the time window during which the barriers are active, comparing the results with theoretical values whenever applicable

Start	End	Price	BT$_{5000}$	TT$_{5000}$	BTT$_{5000}$
0	0	11.7343	11.7338	11.7342	11.7321
0.45	0.55		6.9628	6.9823	6.9249
0.25	0.75		4.2788	4.3001	4.2368
0.25	1		2.1828	2.2010	2.1514
0	0.75		4.2467	4.2684	4.2041
0	1	2.1018	2.1681	2.1465	2.1366

Table 4.7: (Tree)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 1$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$, $B_u = 130$,$B_d = 70$

The results we have obtained exhibit a remarkable degree of similarity when compared to the outcomes derived from Monte-Carlo simulations. This alignment between our lattice-based calculations and the results generated through the computationally intensive Monte-Carlo simulations serves as a compelling endorsement of the accuracy and effectiveness of our pricing model for these particular options.

Chapter 5

Finite Difference Methods

Finite Difference Methods are essential numerical techniques employed in option pricing to solve partial differential equations (PDEs). These methods discretize the PDE into a grid of points in both time and space, allowing us to approximate option prices efficiently. By breaking down complex derivatives pricing problems into manageable steps, Finite Difference Methods play an important role in the valuation of exotic options.

5.1 Garman-Kohlhagen model

Let us recall the model and its partial differential equation that we will be working on in this chapter. The Garman-Kohlhagen option-pricing model (1983) is suitable with interest rate parity. Its partial differential equation is the following:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r_d - r_f) S \frac{\partial V}{\partial S} - r_d V = 0 \tag{5.1.1}$$

The terminal condition is $V(T, S) = g(S)$, the payoff of the option. The variables used in the model are:
$V(S,T)$: price of a call option in domestic units per foreign units
S: spot price
T: time remaining until maturity
K: exercise price of the call option
r_d: domestic interest rate
r_f: foreign interest rate
σ: volatility of spot currency price

The derivation of the PDE is explained in details in the Appendix. The spot price follows a geometric Brownian motion, option price includes one stochastic volatility and interest rates are constant.

$$dS = \mu S dt + \sigma S dZ \tag{5.1.2}$$

where μ is the drift of spot currency price and Z a standard Wiener process.

5.2 General framework for Finite Difference Methods

First, we reformulate the PDE as an initial condition problem. We consider $\tau = T - t$ in order to reformulate the problem.

$$\begin{cases} -\dfrac{\partial V}{\partial \tau} + \frac{\sigma^2 S^2}{2} \dfrac{\partial^2 V}{\partial S^2} + (r_d - r_f) s \dfrac{\partial V}{\partial S} - r_d V = 0 & \tau > 0 \\ V(0, S) = g(S) & \tau = 0 \end{cases} \tag{5.2.1}$$

Generally we work on the truncated domain $D = [0,T]x[s_{min}, s_{max}]$. We construct a uniform grid over D with N+1 points along the time dimension and M+1 points along the space dimension. That is to say:

$$\Delta t = \frac{T}{N}, \Delta x = \frac{S_{max} - S_{min}}{M}$$

The grid points are given by the following expressions:

$$\begin{cases} t_n = n\Delta t & \text{n=0,...,N} \\ s_k = s_{min} + k\Delta x & \text{k=0,...,M} \end{cases}$$

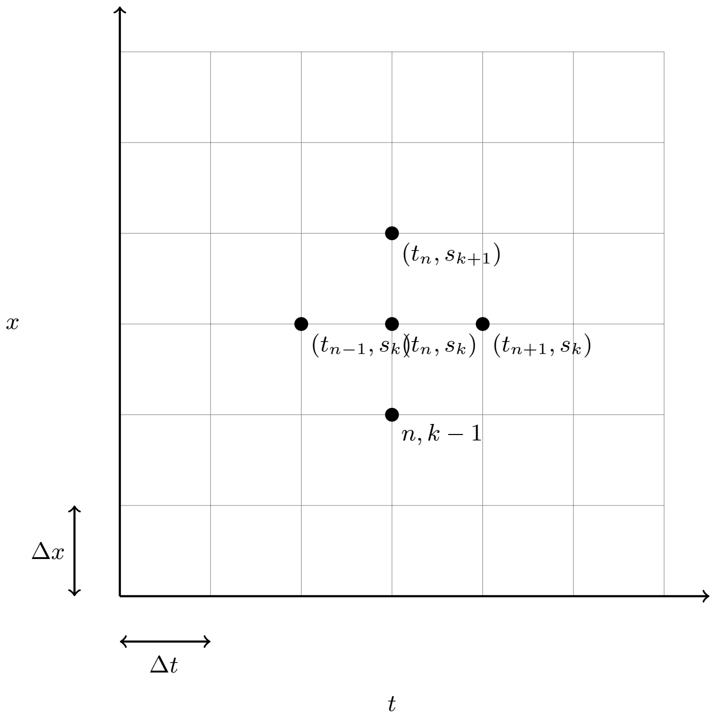

Figure 5.1: The partitioning of the t, x plane for a two dimensional finite difference method

Finite Difference Methods rely on Taylor series expansions. Considering V a function of two variables s and t that is twice differentiable. With the subdivisions we can define $V_{n,k} = V(t_n, s_k) = V(t, s)$. Using Taylor series, we have the following approximation of order Δx^2:

$$\frac{\partial^2 V}{\partial s^2} = \frac{V(t_n, s_{k+1}) - 2V(t_n, s_k) + V(t_n, s_{k-1})}{\Delta x^2} + \mathcal{O}(\Delta x^2)$$
$$\frac{\partial^2 V}{\partial s^2} = \frac{V^n_{k+1} - 2V^n_k + V^n_{k-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2)$$

We can get similar approximations for the other time derivatives that are used in the PDE, respectively the forward and the backward difference approximation

$$\frac{\partial V}{\partial t} = \frac{V^{n+1}_k - V^n_k}{\Delta t} + \mathcal{O}(\Delta t),$$
$$\frac{\partial V}{\partial t} = \frac{V^n_k - V^{n-1}_k}{\Delta t} + \mathcal{O}(\Delta t).$$

5.2.1 Explicit scheme

In light of the given PDE, we apply the forward difference approximation to derive the equations. Additionally, it is worth noting that the scheme presented here accounts for a local volatility model, but it can be adapted to incorporate a uniform volatility throughout the entire grid if needed. This represents an adaptation of the FX model from the Black-Scholes Explicit scheme, as derived in [12].

$$\begin{aligned}
\frac{V_k^{n+1}-V_k^n}{\triangle t} &= \frac{\sigma(s_k,t_n)^2 s_k^2}{2}\frac{V_{k+1}^n-2V_k^n+V_{k-1}^n}{\triangle x^2}+(r_d-r_f)s_k\frac{V_{k+1}^n-V_{k-1}^n}{2\triangle x}-r_dV_k^n \\
\text{Hence}\quad V_k^{n+1} &= \left(\frac{\triangle t}{\triangle x^2}\frac{\sigma(s_k,t_n)^2 s_k^2}{2}-\frac{\triangle t}{2\triangle x}(r_d-r_f)s_k\right)V_{k-1}^n+\left(1-\frac{\triangle t}{\triangle x^2}\sigma(s_k,t_n)^2s_k^2-\triangle tr_d\right)V_k^n \\
&\quad+\left(\frac{\triangle t}{\triangle x^2}\frac{\sigma(s_k,t_n)^2 s_k^2}{2}+\frac{\triangle t}{2\triangle x}(r_d-r_f)s_k\right)V_{k+1}^n \\
&= A_k^nV_{k-1}^n+(1+B_k^n)V_k^n+C_k^nV_{k+1}^n
\end{aligned}$$

for $k=1,2,...,M-1$ and $n=1,2,...,N-1$ where

$$A_k^n := \frac{\triangle t}{\triangle x^2}\frac{\sigma(s_k,t_n)^2 s_k^2}{2}-\frac{\triangle t}{2\triangle x}(r_d-r_f)s_k, \qquad B_k^n := -\frac{\triangle t}{\triangle x^2}\sigma(s_k,t_n)^2-\triangle tr_d,$$

$$C_k^n := \frac{\triangle t}{\triangle x^2}\frac{\sigma(s_k,t_n)^2 s_k^2}{2}+\frac{\triangle t}{2\triangle x}(r_d-r_f)s_k$$

The recursive formula presented above, coupled with the boundary conditions V_0^{n+1} and V_M^{n+1}, provides the foundation for computing prices through a forward induction algorithm.
Without employing the reformulation outlined before, and under the assumption that in the explicit method, the spatial derivatives of V are equivalent for both (n,k) and $(n+1,k)$, we can reconfigure the scheme as follows:

$$\frac{V_k^{n+1}-V_k^n}{\Delta t}+(r_d-r_f)s_k\frac{V_{k+1}^{n+1}-V_{k-1}^{n+1}}{2\triangle x}+\frac{1}{2}\sigma(s_k,t_{n+1})^2s_k^2\frac{V_{k+1}^{n+1}-2V_k^{n+1}+V_{k-1}^{n+1}}{\triangle x^2}=r_dV_k^n.$$

The equation can be reformulated as:

$$V_k^n = p_dV_{k-1}^{n+1}+p_mV_k^{n+1}+p_uV_{k+1}^{n+1}$$

where,

$$\begin{aligned}
p_d &= \frac{1}{1+r_d\Delta t}\left(-\frac{1}{2}(r_d-r_f)s_k\Delta t+\frac{1}{2}\sigma^2s_k^2\Delta t\right), \\
p_m &= \frac{1}{1+r_d\Delta t}\left(1-\sigma^2s_k^2\Delta t\right), \\
p_u &= \frac{1}{1+r_d\Delta t}\left(\frac{1}{2}(r_d-r_f)j\Delta t+\frac{1}{2}\sigma^2s_k^2\Delta t\right).
\end{aligned}$$

The notation is not arbitrary, it serves a purpose. By drawing a parallel between the explicit scheme and the Trinomial Tree method developed in Chapter 4, we establish a meaningful connection. However, the probabilities dependent on the second variable which is not true in a Trinomial model. A change of variable, specifically $z = \ln(S)$, facilitates the elimination of this dependency and aligns the scheme more closely with the Trinomial model. Details are provided in the corresponding Appendix.

For this reason, as well as the others elaborated upon in subsequent sections, we will not delve extensively into the Explicit scheme for our estimations.

5.2.2 Fully implicit scheme

In the case of the Fully Implicit scheme, we employ the backward difference method and derive the following estimations as in [12] for the Black-Scholes case:

$$\frac{V_k^n - V_k^{n-1}}{\triangle t} = \frac{\sigma(s_k,t_n)^2 s_k^2}{2}\frac{V_{k+1}^n - 2V_k^n + V_{k-1}^n}{\triangle x^2} + (r_d - r_f)s_k\frac{V_{k+1}^n - V_{k-1}^n}{2\triangle x} - r_d V_k^n$$

$$\begin{aligned}\text{Hence}\quad V_k^{n-1} &= -\left(\frac{\triangle t}{\triangle x^2}\frac{\sigma(s_k,t_n)^2 s_k^2}{2} - \frac{\triangle t}{2\triangle x}(r_d - r_f)s_k\right)V_{k-1}^n + \left(1 + \frac{\triangle t}{\triangle x^2}\sigma(s_k,t_n)^2 s_k^2 + \triangle t r_d\right)V_k^n \\ &\quad - \left(\frac{\triangle t}{\triangle x^2}\frac{\sigma(s_k,t_n)^2 s_k^2}{2} + \frac{\triangle t}{2\triangle x}(r_d - r_f)s_k\right)V_{k+1}^n \\ &= -A_k^n V_{k-1}^n + (1 - B_k^n)V_k^n - C_k^n V_{k+1}^n\end{aligned}$$

for $k = 1, 2, ..., M-1$ and $n = 1, 2, ..., N-1$ where

$$A_k^n := \frac{\triangle t}{\triangle x^2}\frac{\sigma(s_k,t_n)^2 s_k^2}{2} - \frac{\triangle t}{2\triangle x}(r_d - r_f)s_k, \qquad B_k^n := -\frac{\triangle t}{\triangle x^2}\sigma(s_k,t_n)^2 - \triangle t r_d,$$

$$C_k^n := \frac{\triangle t}{\triangle x^2}\frac{\sigma(s_k,t_n)^2 s_k^2}{2} + \frac{\triangle t}{2\triangle x}(r_d - r_f)s_k$$

The structure of the equation is similar with the Explicit scheme and we also have to couple it with the boundary conditions to use the recursive algorithm and obtain prices.

5.2.3 Crank-Nicolson scheme

John Crank and Phyllis Nicolson are credited with the development of the Crank-Nicolson method, a numerical solution technique designed to address the challenges posed by PDEs arising from heat-conduction problems in [13] . This method was introduced with the dual objectives of mitigating instability while simultaneously enhancing the efficiency and accuracy of both implicit and explicit numerical methods. The Crank-Nicolson scheme can indeed be viewed as a midpoint between the implicit and explicit schemes. To implement this scheme, we blend elements from both the implicit and explicit methods. We construct the scheme by adopting the structure of the implicit scheme and then incorporate the missing terms according to the following formula. This approach combines the stability of the implicit scheme with the computational efficiency of the explicit scheme, resulting in a well-balanced numerical solution. This has been also analyzed in [14].

To define the scheme, we can introduce a generalized θ-scheme, which operates as a weighted scheme with θ serving as the weight.

$$\begin{aligned}\frac{V_k^n - V_k^{n-1}}{\triangle t} &= \theta\left\{\frac{\sigma(s_k,t_n)^2 s_k^2}{2}\frac{V_{k+1}^n - 2V_k^n + V_{k-1}^n}{\triangle x^2} + (r_d - r_f)s_k\frac{V_{k+1}^n - V_{k-1}^n}{2\triangle x} - r_d V_k^n\right\} \\ &\quad + (1-\theta)\left\{\frac{\sigma(s_k,t_{n-1})^2 s_k^2}{2}\frac{V_{k+1}^{n-1} - 2V_k^{n-1} + V_{k-1}^{n-1}}{\triangle x^2} + (r_d - r_f)s_k\frac{V_{k+1}^{n-1} - V_{k-1}^{n-1}}{2\triangle x} - r_d V_k^{n-1}\right\}\end{aligned} \tag{5.2.2}$$

With $\theta = 0.5$, the scheme is the Crank-Nicolson scheme. We use the same boundary conditions as for the other schemes and compute the price using a similar recursive algorithm.

For all three numerical schemes, we arrive at a tridiagonal system, which we solve at each time step using LU decomposition.

Definition 5.2.1. LU decomposition
The principle of LU decomposition is to represent any matrix M as the product of a lower triangular matrix L and an upper triangular matrix U.This decomposition, denoted as $M = LU$ is called the LU decomposition of M.

The primary reasons for employing LU decomposition lie in its advantages, particularly its efficiency in computing the inverses of triangular matrices. This characteristic proves invaluable when dealing with the resolution of the linear systems integral to our work.

When we impose the number of spatial points and increment the number of time steps, we observe the following computational time trends:

	CN_{100}	CN_{1000}	CN_{5000}	CN_{10000}
Time(s)	0.010	0.10	0.54	1.05

Table 5.1: Average Computational Time for Finite Difference Methods with a Fixed Spatial Grid

PDE-based computations offer faster processing speeds. In the upcoming sections, we will delve into whether this increased speed comes at the expense of accuracy.

5.2.4 Study of consistency, stability and convergence

In this section, we will present a series of theorems and propositions that pertain to numerical schemes, particularly focusing on their stability, consistency and convergence. These aspects are of paramount importance in the context of pricing options, especially when dealing with grids that are imposed by the specific characteristics of our financial products. The analysis is exhaustively done in [15].

Definition 5.2.2. A numerical scheme is considered consistent when its discrete operator, implemented using finite differences, converges toward the continuous operator of the underlying partial differential equation (PDE) for $\Delta t, \Delta x \to 0$. (the local truncation error L_k^n vanishes)

For instance, the local truncation error of the Explicit scheme is:

$$L_k^n = \frac{V_k^{n+1} - V_k^n}{\triangle t} - \frac{\sigma^2 s_k^2}{2}\frac{V_{k+1}^n - 2V_k^n + V_{k-1}^n}{\triangle x^2} - (r_d - r_f)s_k\frac{V_{k+1}^n - V_{k-1}^n}{2\triangle x} + r_d V_k^n$$

Measuring how well the values of V_k^n at discrete points satisfy the original PDE is challenging due to their discrete nature. However, we can assess the accuracy of the solution to the PDE when it is evaluated at these grid points, determining how effectively it aligns with the finite difference scheme.

Theorem 5.2.3. *Explicit and Implicit schemes are consistent and their truncation errors are such that:* $L_k^n = O(\Delta t) + \mathcal{O}(\Delta x^2)$

Theorem 5.2.4. *Crank-Nicolson schemes is consistent and its truncation error is such that:* $L_k^n = O(\Delta t^2) + \mathcal{O}(\Delta x^2)$

Proofs for both of these theorems are straightforward using the local truncation error and Taylor expansions of the derivatives. The final theorem serves as an explanation for the widespread adoption of the Crank-Nicolson scheme. Consistency, while necessary, is not alone sufficient to ensure convergence. The reason behind this lies in the potential accumulation of small local errors introduced at each time step during recursive calculations, a challenge commonly referred to as the stability issue.

Definition 5.2.5. Let us consider a finite difference scheme such as $X^{n+1} = DX^n$ with D an operator applied to each of the X^i . The scheme is stable if for every n, $\|D^n X\| \leq C_T\|X\|$ with C_T constant

Theorem 5.2.6. *Lax-Richtmyer Equivalence Theorem*
A finite different scheme is convergence if and only if it is stable and consistent.

Now that we have introduced the concept of stability, it becomes crucial to determine whether our schemes possess this property.

Theorem 5.2.7. *Implicit and Crank-Nicolson schemes for Garman-Kohlhagen PDE are unconditionally stable while Explicit scheme is stable if* $\Delta t \leq \frac{\Delta x^2}{2}$

The unconditional stability exhibited by the Implicit scheme and Crank-Nicolson scheme, in contrast to the conditional stability of the Explicit scheme, further elucidates why the latter is not the preferred choice when solving PDEs.

5.2.5 Numerical results

In this section, we will present the results obtained using the Crank-Nicolson scheme.

	Price	CN$^{100}_{100}$	err$^{100}_{100}$	CN$^{100}_{1000}$	err$^{100}_{1000}$	CN$^{500}_{1000}$	err$^{500}_{1000}$	CN$^{1000}_{1000}$	err$^{1000}_{1000}$
$\mathrm{S}_0 = 100$ B=95	6.7447	6.7357	$1.3.10^{-3}$	6.7358	$1.3.10^{-3}$	6.7444	$4.4.10^{-5}$	6.7446	$1.5.10^{-5}$
$\mathrm{S}_0 = 100$ B=90	10.9501	10.9459	$3.8.10^{-4}$	10.9459	$3.8.10^{-4}$	10.9499	$1.8.10^{-5}$	10.9501	$< 10^{-5}$
$\mathrm{S}_0 = 110$ B=100	15.0164	15.0159	$3.3.10^{-5}$	15.0159	$3.3.10^{-5}$	15.0163	$< 10^{-5}$	15.0164	$< 10^{-5}$
$\mathrm{S}_0 = 110$ B=90	21.1631	21.1603	$1.3.10^{-4}$	21.1603	$1.3.10^{-4}$	21.1629	$< 10^{-5}$	21.1630	$< 10^{-5}$
$\mathrm{S}_0 = 120$ B=100	28.0566	28.0539	0.0319	28.0540	$9.3.10^{-5}$	28.0564	$< 10^{-5}$	28.0565	$< 10^{-5}$
$\mathrm{S}_0 = 120$ B=110	18.4403	18.4304	$< 10^{-5}$	18.4305	$< 10^{-5}$	18.4398	$< 10^{-5}$	18.4401	$< 10^{-5}$

Table 5.2: (CN)Down-and-out call option with $r_d = 0.08$,$r_f = 0.04$,$T = 0.5$,$\sigma = 0.25$, $K = 90$

The results and estimations presented here exhibit significant differences from those in the previous chapter. Notably, the convergence of the price occurs much more rapidly, with prices stabilizing after only 1000 time steps for each trade. As a result, we conducted separate modifications of the spatial steps to assess the impact of this parameter on the computed prices. The numbers in the exponent correspond to the number of spatial steps, and the indices correspond to the number of time steps. It is noteworthy that the Finite Difference method proves to be highly efficient for pricing Single Barrier options. With just 100 steps, we manage to achieve an error of less than 10^{-3} for most cases. Furthermore, increasing the number of spatial points acts as a grid refinement, leading to even more precise price estimations. In particular, with 1000 time steps and spatial points, we consistently achieve errors on the order of 10^{-5} or lower. This level of accuracy far surpasses what was attainable with the previous methods employed.

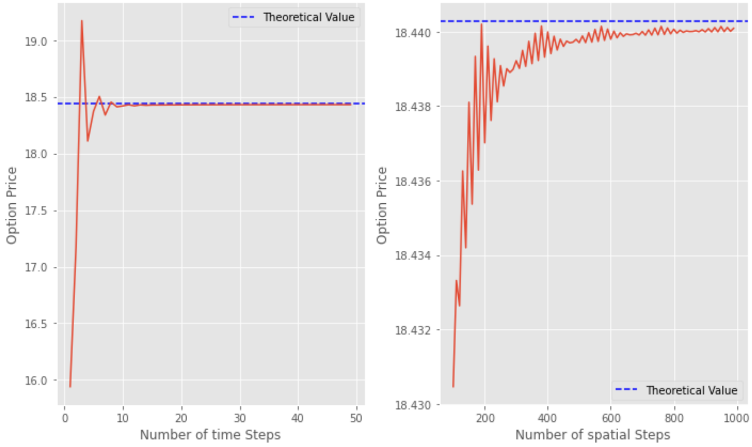

Figure 5.2: Convergence of the CN scheme for $\mathrm{S}_0 = 120, K = 90, B = 110, r_d = 0.08, r_f = 0.04, T = 0.5, \sigma = 0.25$

In Figure 5.2, we depict the convergence of prices for a specific trade, offering valuable insights into the behaviour of the Crank-Nicolson scheme. In the left plot, we set the spatial step to $M = 100$ and observe how the price evolves with an increasing number of time steps. Notably, we observe that time convergence is exceptionally rapid with the Crank-Nicolson scheme. In fact, with fewer than 20 time steps, we achieve a remarkably accurate approximation of the price. Conversely, in the right plot, we reverse the scenario by fixing $N = 100$ time steps and scrutinising the convergence as we vary the number of spatial steps. Here, we encounter a different pattern of convergence. Firstly, we require a higher number of spatial steps, and the convergence rate is comparatively slower. Beyond $M = 700$, we observe that the price undergoes minimal changes, indicating that convergence has effectively been reached.

We performed the similar computations for Double Barrier options. As for the Single Barrier options, computations did not need much time steps to converge to a value, so we worked on time and spatial grid points.

	Price	CNN_{100}^{100}	err_{100}^{100}	CNN_{1000}^{100}	err_{1000}^{100}	CNN_{1000}^{500}	err_{1000}^{500}	CN_{1000}^{1000}	err_{1000}^{1000}
$\text{B}_d = 50$ $\text{B}_u = 150$	5.4636	5.4640	$7.3.10^{-5}$	5.4639	$5.5.10^{-5}$	5.4637	$1.8.10^{-5}$	5.4636	$< 10^{-5}$
$\text{B}_d = 60$ $\text{B}_u = 140$	5.2200	5.2211	$2.1.10^{-4}$	5.2211	$2.1.10^{-4}$	5.2202	$3.8.10^{-5}$	5.2200	$< 10^{-5}$
$\text{B}_d = 70$ $\text{B}_u = 130$	4.3806	4.3850	$1.0.10^{-3}$	4.3849	$9.8.10^{-4}$	4.3805	$2.3.10^{-5}$	4.3805	$2.3.10^{-5}$
$\text{B}_d = 80$ $\text{B}_u = 120$	2.4642	2.4660	$7.3.10^{-4}$	2.4661	$7.7.10^{-4}$	2.4645	$1.2.10^{-4}$	2.4644	$8.1.10^{-5}$
$\text{B}_d = 90$ $\text{B}_u = 110$	0.3003	0.3014	$3.6.10^{-3}$	0.3013	$3.3.10^{-3}$	0.3005	$6.6.10^{-3}$	0.3004	$3.3.10^{-4}$

Table 5.3: (CN)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 0.25$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$

We observe that the convergence dynamics for Double Barrier options when using the Crank-Nicolson scheme closely resemble those for Single Barrier options. In most cases, we achieved errors below the threshold of 10^{-3} within fewer than 100 computational steps. However, when dealing with Double Barrier options featuring extremely tight barriers, achieving such convergence proved more challenging. Nonetheless, even in this specific case, the Crank-Nicolson scheme demonstrated significantly improved convergence performance when compared to alternative numerical techniques discussed in previous chapters. The overall results suggest that this numerical scheme is highly suitable for accurately pricing always-active Double Barrier options.

5.2.6 Window Barrier options

We have completed our investigation of the Crank-Nicolson scheme, including the computation of Window Barrier option prices. Throughout the preceding chapters, we observed consistent results between Monte-Carlo estimations and Lattice methods.

Start	End	Price	CN_{5000}^{500}
0	0	11.7343	11.7131
0.45	0.55		6.9486
0.25	0.75		4.2748
0.25	1		2.1815
0	0.75		4.2748
0	1	2.1018	2.1674

Table 5.4: (CN)KOKO call option with $r_d = 0.1$,$r_f = 0.05$,$T = 1$,$\sigma = 0.25$, $K = 100$, $S_0 = 100$, $B_u = 130$,$B_d = 70$

The results presented in Table 5.4 align closely with the outcomes produced by Lattice methods and Monte-Carlo estimations. This consistency provides further evidence of the correctness and accuracy of our pricing estimations, bolstering our confidence in the reliability of the methods employed in this study.

5.3 Improved scheme for Discrete Barrier options

As elucidated by Daniel J. Duffy in [16], while Crank-Nicolson remains a widely favored and highly accurate numerical technique in various scenarios, it can exhibit inaccuracies and erratic behavior, particularly when applied to Barrier options. This observation underscores the importance of carefully considering the choice of numerical methods, especially for complex financial instruments like Barrier options.

In the previous sections, our focus was primarily on Continuously monitored Barrier options, which are often suitable in an FX context. However, it is essential to acknowledge that in many real-world scenarios, barriers are applied and monitored discretely. This discrete monitoring can have a profound impact on the convergence of the Crank-Nicolson method. Unlike Monte-Carlo and even Lattice methods, Crank-Nicolson may exhibit spurious oscillations when confronted with discrete barriers. Over time, these oscillations may persist and, in some cases, fail to converge, leading to substantial inaccuracies in pricing and Greek estimations. Consequently, the accurate pricing of such options can be severely compromised. This highlights the need for alternative numerical methods or adaptations when dealing with discrete barrier options.

An interesting solution for this issue is suggested in [17]. The method consists in using a modified grid to compute the prices, a "point-distributed finite volume scheme" as defined by the authors of the article.

The discretization of the Garman-Kohlhagen PDE follows a similar approach as described in Section 5.2.2 with the key difference being that the spatial step, denoted as Δx , may not be uniformly distributed across the grid. This non-uniform spatial discretization is employed to accurately capture the behavior of barriers. They also suggest that Implicit Method ($\theta = 0$ in our case)produces better results than Crank-Nicolson. The oscillations are indeed responsible for inaccurate prices.

As an example, we decide to price Down-and-Out call options with a barrier monitored five times a day. As explained in [17], we applied the Implicit scheme to compute the required values. The modified algorithm generates a non-uniform grid tailored to different types of barrier options by adjusting the grid density around specific barrier levels (in this case it is around a "Down" barrier). It ensures that the grid captures the dynamics of the underlying asset's price accurately, taking into account the discrete monitoring of barrier levels. The distribution of the grid points is customized using a coefficient, for the following trades we set the ratio of grid points around the barrier to 85%.

	Continuous Price	Discrete $\mathrm{IM}_{1000}^{1000}$
$S_0 = 100$ B=95	6.7447	7.5324
$S_0 = 100$ B=90	10.9501	11.1591
$S_0 = 110$ B=100	15.0164	15.4032
$S_0 = 110$ B=90	21.1631	21.3947
$S_0 = 120$ B=100	28.0566	28.9748
$S_0 = 120$ B=110	18.4403	18.8456

Table 5.5: (IM)Down-and-out call option with $r_d = 0.08$,$r_f = 0.04$,$T = 0.5$,$\sigma = 0.25$, $K = 90$

The observation that discrete barrier options with discrete monitoring are more expensive than continuously monitored ones aligns with intuition. This price difference can be attributed to the reduced probability of the underlying asset crossing the barrier when monitoring is less frequent. In discrete monitoring, the barrier can only be crossed during the specific monitoring times, and if it is not breached on those times, it will not impact the option price. Therefore, the discrete monitoring reduces the effective barrier-crossing probability, making the Knock-Out option more expensive compared to continuously monitored options where the barrier can be crossed at any time. This highlights the significance of monitoring frequency in the pricing of barrier options.

Conclusion

This thesis aimed to delve into various methodologies for pricing barrier options. Barrier options encompass a highly diverse category of exotic options, and in this paper, we focus on a select subset of these, including Single Barrier options, Double Barrier options, Window Barrier options, and, in Chapter 5, some Discrete Barrier options using Finite Difference methods. This expansive nature of Barrier options renders them both captivating and pertinent in numerous scenarios.

Our work was based on FX options, so Our research focused on FX options, and as such, we introduced the Garman-Kohlhagen model as the foundation for pricing the options under study. With the valuable contributions of researchers such as Reiner and Rubinstein [1] or Kunitomo and Ikeda [2], among others, we were able to calculate theoretical values for specific categories of Barrier options.

Our primary objective was to obtain precise and robust estimates for Barrier options, with a particular focus on extending our investigation to Window Barrier options. We initiated our analysis using Monte Carlo simulations, which offer the advantage of being computationally straightforward and providing reasonably accurate estimates, provided that a sufficient number of simulations are conducted. Nevertheless, they come with their limitations, notably in terms of computational time and the width of the resulting confidence intervals. To address the latter issue, we explored various Variance reduction techniques. Among these techniques, the Antithetic Variates method yielded the most promising results in our context. However, despite our efforts in variance reduction, we still encountered challenges in reducing computational time significantly. One potential avenue for improvement could involve delving into parallel programming as a means to expedite the computations.

We then turned our attention to lattice methods, specifically employing Binomial and Trinomial Trees, as well as an Improved Tree method. We observed that these lattice methods generally exhibited faster convergence compared to Monte Carlo simulations and provided more accurate price estimates with fewer time steps.The CRR Binomial model [6] emerged as an efficient approach, notably faster than Monte Carlo simulations. Trinomial Trees, with the addition of an extra node at each time step, allowed for convergence in fewer steps as a general trend. However, these methods sometimes faced challenges when the barrier levels did not align precisely with the layers of nodes in the trees.To address this issue, we explored a solution based on adapting the grid to the barrier levels, drawing inspiration from the work of Dai-Lyuu [11]. This approach enabled us to tackle specific cases where barriers were not correctly handled, further enhancing the accuracy and robustness of our pricing methods.

Finally, we introduced Finite Difference methods for pricing Barrier options. We defined the primary schemes utilized and elaborated on why the Crank-Nicolson scheme stands out as the most widely employed method within the financial industry. Its unconditional stability and superior consistency rendered it the optimal choice, allowing us to price Barrier options with exceptional accuracy and efficiency compared to other methods. However, it is important to acknowledge that the model is not without its imperfections, as we aim to elucidate in the concluding section. Notably, it can exhibit severe oscillations when dealing with discontinuities, as was the case with Discrete Barrier options. These oscillations are typically not a significant concern when focusing solely on pricing continuously monitored barriers (our primary objective), but they become problematic when computing sensitivities. To address this issue, we explored one solution. Our approach involved implementing a non-uniform grid to better capture the dynamics around the barrier [17]. Additionally, the Rannacher stepping method, which combines several steps of the Implicit-Scheme with Crank-Nicolson, emerged as a popular alternative. This method provides all the advantages of the Crank-Nicolson scheme while mitigating the undesirable oscillations around barrier levels.

Appendix A

Garman-Kohlhagen PDE

In this section, we will delve on the Garman-Kohlhagen PDE and how to derive its PDE. First, let us redefine all the variables involved in the proof:
$V(S,T)$: price of a call option in domestic units per foreign units
S: spot price
T: time remaining until maturity
K: exercise price of the call option
r_d: domestic interest rate
r_f: foreign interest rate
σ: volatility of spot currency price

Three other variables are useful for the computation and need to be defined as follows:
α: expected rate of return on the security
β: standard deviation of the security rate of return
γ: drift of spot price

Let us consider a contigent claim with some payoff function $\Phi(\chi(T))$ based on the exchange rate χ at time T and a pricing function $f(t, \chi(t)) = \Pi(t) = \Pi$ that depends on time t and exchange rate. The notation Π aims at simplifying the notation and does not change the proof. The exchange rate is supposed to be domestic to foreign.

$$d\chi(t) = \alpha\chi(t)dt + \sigma\chi(t)dZ(t) \tag{A.0.1}$$

where α is the drift of the exchange rate and σ is the volatility. Let B^f and B^d be respectively the foreign and the domestic bank accounts:

$$dB^f(t) = r_f B^f(t)dt, dB^d(t) = r_d B^d(t)dt$$

Since we must work in domestic currency, we have to define a new bank account:

$$\tilde{B^f}(t) = \chi(t)B^f(t)$$

By Itô product rule, we have

$$d\tilde{B^f}(t) = \chi(t)dB^f(t) + d\chi(t)B^f(t) + dB^f(t)d\chi(t)$$

$$d\tilde{B^f}(t) = \chi(t)r_f B^f(t)dt + (\alpha\chi(t)dt + \sigma\chi(t)dZ(t))B^f(t) + r_f B^f(t)dt(\alpha\chi(t)dt + \sigma\chi(t)dZ(t))$$

$$d\tilde{B^f}(t) = \chi(t)B^f(t)(\alpha + r_f)dt + \sigma\chi(t)B^f(t)dZ(t)$$
$$d\tilde{B^f}(t) = \tilde{B^f}(t)(\alpha + r_f)dt + \sigma\tilde{B^f}(t)dZ(t)$$

Supposing V is the value process of the portfolio:

$$dV(t) = V(w^\chi \frac{d\tilde{B^f}(t)}{\tilde{B^f}(t)} + w^\Pi \frac{d\Pi}{\Pi})$$

where w^χ is the weight of the underlying and w^Π the weight of our claim

By applying Itô rule to f we get:

$$df = f_t dt + f_\chi d\chi + \frac{1}{2} f_{\chi\chi}(d\chi)^2$$

$$df = f_t dt + f_\chi(\alpha\chi(t)dt + \sigma\chi(t)dZ(t)) + \frac{1}{2} f_{\chi\chi}\sigma^2\chi^2(t)dt$$

$$df = (f_t + f_\chi\alpha\chi(t) + \frac{1}{2} f_{\chi\chi}\sigma^2\chi^2(t))dt + f_\chi\sigma\chi(t)dZ(t)$$

Let us define α_f and σ_f such as:

$$\alpha_f = \frac{f_t + f_\chi\alpha\chi(t) + \frac{1}{2} f_{\chi\chi}\sigma^2\chi^2(t)}{f}, \sigma_f = \frac{f_\chi\sigma\chi(t)}{f}$$

$$df = f\alpha_f dt + f\sigma_f dZ(t), \Pi = f$$

$$\frac{d\Pi}{\Pi} = \alpha_f dt + \sigma_f dZ(t)$$

$$\frac{d\tilde{B^f}(t)}{\tilde{B^f}(t)} = \frac{\tilde{B^f}(t)(\alpha + r^f)dt + \sigma\tilde{B^f}(t)dZ(t)}{\tilde{B^f}(t)}$$

$$\frac{d\tilde{B^f}(t)}{\tilde{B^f}(t)} = (\alpha + r^f)dt + (t)$$

$$dV(t) = V(w^\chi[(\alpha + r_f)dt + (t)] + w^\Pi[\alpha_f dt + \sigma_f dZ(t)])$$

$$dV(t) = V(w^\chi(\alpha + r_f) + w^\Pi\alpha_f)dt + V(w^\chi\sigma + w^\Pi\sigma_f)dZ(t)$$

We can deduce the following system of three equations:

$$\begin{cases} r_d = w^\chi(\alpha + r_f) + w^\Pi\alpha_f & \text{no arbitrage (1)} \\ w^\chi\sigma + w^\Pi\sigma_f = 0 & \text{portfolio with no risk (2)} \\ w^\chi + w^\Pi = 1 & \text{portfolio with no risk (3)} \end{cases} \tag{A.0.2}$$

Using (2) and (3), we have:

$$w^\Pi = \frac{\sigma}{\sigma - \sigma_f}$$

And with (3):

$$w^\chi = \frac{\sigma_f}{\sigma_f - \sigma}$$

Finally, with (1)

$$\sigma_f(\alpha + r_f - r_d) = \sigma(\alpha_f - r_d)$$

Hence, the market price risk equivalency between the underlying and the claim

$$\frac{\alpha + r_f - r_d}{\sigma} = \frac{\alpha_f - r_d}{\sigma_f} = \lambda \tag{A.0.3}$$

so

$$\lambda\sigma_f = \alpha_f - r_d$$

$$\lambda\frac{f_\chi\sigma\chi(t)}{f} = \frac{f_t + f_\chi\alpha\chi(t) + \frac{1}{2} f_{\chi\chi}\sigma^2\chi^2(t)}{f} - r_d$$

$$f_t + f_\chi\alpha\chi(t) - \lambda f_\chi\sigma\chi(t) + \frac{1}{2} f_{\chi\chi}\sigma^2\chi^2(t) - r_d f = 0$$

$$f_t + f_\chi(\alpha - \lambda\sigma)\chi(t) + \frac{1}{2} f_{\chi\chi}\sigma^2\chi^2(t) - r_d f = 0$$

using (A.1.3) we get the PDE

$$f(T, \chi) = \Phi(\chi)$$

Appendix B

Lattice methods

B.1 Proof of Theorem 3.1.1: Backward Induction Algorithm

For $j = n$,

$$V^j = E[g(S_n)|\mathcal{F}_j] = g(S_n)$$

with $g(S_n)$ is $\mathcal{F}_n$ measurable.

For $j < n$, we get by using Tower's property:

$$\begin{aligned} V^j &= e^{-r_d(n-j)\Delta t} E[g(S_n)|\mathcal{F}_j] \\ &= e^{-r_d(n-j)\Delta t} E[E[g(S_n)|\mathcal{F}_{j+1}]|\mathcal{F}_j] \\ &= e^{-r_d\Delta t} E[e^{-r_d(n-j-1)\Delta t} E[g(S_n)|\mathcal{F}_{j+1}]|\mathcal{F}_j] \\ &=^{-r_d\Delta t} E[v^{j+1}|\mathcal{F}_j] \end{aligned}$$

B.2 Proof of Theorem 4.1.2 : Price of a Vanilla option

We prove this by induction. Clearly for $n = 1$ we have,

$$\begin{aligned} V &= \mathrm{e}^{-r_d\Delta t}\left[\sum_{j=0}^{1}\binom{1}{j} p_u^j p_d^{1-j} V_{u^j,d^{1-j}}\right] \\ &= \mathrm{e}^{-r_d\Delta t}\left[\binom{1}{0} p_u^0 p_d^1 V_{u^0,d^1} + \binom{1}{1} p_u^1 p_d^0 V_{u^1,d^0}\right] \\ &= \mathrm{e}^{-r_d\Delta t}[p_u V_u + p_d V_d]. \end{aligned}$$

Which is exactly the kind of formulas we compute in practice in the numerical implementation for each step . We can also check this for $n = 2$. We now perform our inductive step. Assume that the equation hold for some $k \in \mathbb{N}$. Then note that,

$$\begin{aligned} V &= \mathrm{e}^{-r_d k\Delta t}\left[\sum_{j=0}^{k}\binom{n}{j} p_u^j p_d^{k-j} V_{u^j,d^{k-j}}\right] \\ &= \mathrm{e}^{-r_d k\Delta t}\left[\binom{k}{0} p_d^k V_{d^k} + \binom{k}{1} p_u p_d^{k-1} V_{u,d^{k-1}} + \dots \right. \\ &\qquad \left. + \binom{k}{k-1} p_u^{k-1} p_d V_{u,d^{k-1}} + \binom{k}{k} p_u^k V_{u^k}\right]. \end{aligned}$$

Now we can expand this formula,

$$\begin{aligned}V = \mathrm{e}^{-r_d k\Delta t}\Bigg[\binom{k}{0}p_d^k[p_u V_{u,d^k} + p_d V_{d^{k+1}}]\mathrm{e}^{-r_d\Delta t} \\ + \binom{k}{1}p_u p_d^{k-1}[p_u V_{u^2,d^{k-1}} + p_d V_{u,d^k}]\mathrm{e}^{-r_d\Delta t} + \dots \\ + \binom{k}{k}p_u^k[p_u V_{u^{k+1}} + p_d V_{u^k,d}]\mathrm{e}^{-r\Delta t}\Bigg].\end{aligned}$$

For a term $V_{u^j,d^{k-j+1}}$ for $j = 1, 2, \dots, k+1$ the contributing factors are $V_{u^j,d^{k-j+1}}$ and $V_{u^{j+1},d^{k-j}}$. These contributing factors are $\binom{k}{j-1}V^j p_d^{k-j+1}$ and $\binom{k}{j}p_u^{j-1}p_u p_d^{k-j}p_d$ respectively. Hence we have that the coefficient of $V_{u^j,d^{k-j+1}}$ is given by,

$$\begin{aligned}p_d^{k-j+1}p_u^j\binom{k}{j-1} + \binom{k}{j}p_d^{k-j+1}p_u^j &= p_d^{k-j+1}p_u^j\left(\binom{k}{j} + \binom{k}{j-1}\right) \\ &= (p_d^{k-j+1}p_u^j\binom{k+1}{j}.\end{aligned}$$

For the next step, we have now that

$$\begin{aligned}V &= \mathrm{e}^{-r_d k\Delta t}\sum_{j=1}^{k+1}\mathrm{e}^{-r_d\Delta t}\binom{k+1}{j}p_d^{k-j+1}p_u^j V_{u^j,d^{k-j+1}} \\ &= \mathrm{e}^{-r_d(k+1)\Delta t}\sum_{j=1}^{k+1}\binom{k}{j}p_d^{k-j}p_u^j f_{u^j,d^{k-j}}.\end{aligned}$$

Then if it is true for k it is also true for $k + 1$. By induction, the theorem is proved.

Appendix C

Finite difference methods

C.1 Explicit scheme and Trinomial Tree

In this section, we will prove that we can define probabilities p_d^*, p_m^* and p_u^* for the Explicit scheme that are independent of s_k.

First, one must reformulate Garman-Kohlhagen PDE defined as :

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r_d - r_f) S \frac{\partial V}{\partial S} - r_d V = 0$$

By using the change of variable $z = ln(S)$, the PDE transforms into:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 V}{\partial z^2} + (r_d - r_f - \frac{\sigma^2}{2}) \frac{\partial V}{\partial z} - r_d V = 0$$

The explicit difference methods gives the new estimation:

$$\frac{V_k^{n+1} - V_k^n}{\triangle t} = \frac{\sigma^2}{2} \frac{V_{k+1}^{n+1} - 2V_k^{n+1} + V_{k-1}^{n+1}}{\triangle z^2} + (r_d - r_f - \frac{\sigma^2}{2}) \frac{V_{k+1}^{n+1} - V_{k-1}^{n+1}}{2\triangle z} - r_d V_k^n$$

Hence

$$V_k^n = p_{dk-1}^{*n+1} + p_m^* V_k^{n+1} + p_u^* V_{k+1}^{n+1}$$

where,

$$\begin{aligned} p_d^* &= \frac{1}{1 + r_d \Delta t} \left(-\frac{1}{2}(r_d - r_f - \frac{\sigma^2}{2}) \frac{\Delta t}{\Delta z} + \frac{1}{2}\sigma^2 \frac{\Delta t}{2\Delta z^2} \right), \\ p_m^* &= \frac{1}{1 + r_d \Delta t} \left(1 - \sigma^2 \frac{\Delta t}{2\Delta z^2} \right), \\ p_u^* &= \frac{1}{1 + r_d \Delta t} \left(\frac{\Delta t}{2\Delta z}(r_d - r_f) j \Delta t + \frac{1}{2}\sigma^2 \frac{\Delta t}{2\Delta z^2} \right). \end{aligned}$$

As claimed in Section 5.2.1, the probabilities obtained are no more dependent on k (assuming the fact that the volatility is constant).

C.2 Stability of the Explicit Scheme

Garman-Kohlhagen PDE can be reinterpreted as as the heat equation of the form

$$\frac{\partial u}{\partial t^*} = \frac{\partial^2 u}{\partial x^2}$$

We consider a set of transformations given by:

$$\begin{aligned} s &= Ke^x \\ t &= T - \frac{2t^*}{\sigma^2} \\ V &= Kv(t*, x). \end{aligned}$$

Then we define $u(t*, x) = e^{\alpha x + \beta t} v(t*, x)$ with some constants α and β. These transformations are employed to derive the Heat equation PDE. We omit the detailed computation steps in this section, as our primary focus here is to analyze the stability of the numerical schemes used.

Von Neumann stability analysis is a technique used to analyze the stability of numerical methods, particularly finite-difference methods. We assume that the solution can be written in the form of a Fourier series:

$$\begin{aligned} u_j^n &= Ae^{iwj\Delta x} e^{\lambda^n \Delta t} \\ &= R^n e^{iwj\Delta x} \end{aligned}$$

The scheme is stable if $|R| \leq 1$. We will determine the condition of stability for the Explicit scheme as an example.

$$\begin{aligned} \frac{u_j^{n+1} - u_j^n}{\triangle t^*} &= \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\triangle x^2} \\ u_j^{n+1} &= u_j^n + \frac{\Delta t^*}{\Delta x^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) \end{aligned}$$

By substituting with the formula of u_j^n and simplification, we have that:

$$\begin{aligned} R &= 1 + \frac{\Delta t^*}{\Delta x^2}(e^{iwj\Delta x} + e^{-iwj\Delta x}) \\ &= 1 + 2\frac{\Delta t^*}{\Delta x^2}(cos(w\Delta x) - 1) \\ &= 1 - 4\frac{\Delta t^*}{\Delta x^2} sin^2(\frac{w\Delta x}{2}) \end{aligned}$$

We need to have $|R| \leq 1$, that is to say $1 - 4\frac{\Delta t^*}{\Delta x^2} \geq -1$. Hence the Explicit scheme is stable if and only if $\frac{\Delta t^*}{\Delta x^2} \leq \frac{1}{2}$.

Bibliography

[1] Reiner and Rubinstein. Breaking down the barriers. *Risk Magazine*, 4(8), 1991.

[2] Naoto Kunitomo and Masayuki Ikeda. Pricing options with curved boundaries. *Mathematical Finance*, 2(4):275–298, 1992.

[3] E.G Haug and J.Haug. Resetting strikes, barriers and time. *Wilmott Magazine*, 2001.

[4] Paul Glasserman Mark Broadie and Steven Kou. A continuity correction for discrete barrier options. *Mathematical Finance*, 7(4):325–348, 1997.

[5] J.G. Shanthikumar S.M. Ross. Pricing exotic options: Monotonicity in volatility and efficient simulation. *Probability in the Engineering and Informational Sciences*, 14(14):317–326, 2000.

[6] S.A. Ross Cox J.C and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–263, 1979.

[7] Ken Palmer Jhihrong Lin. Convergence of barrier option prices in the binomial model. *Mathematical Finance*, 23(2):318–338, 2013.

[8] Boyle P.P. Option valuation using a three jump process. *International Options Journal*, 3:7–12, 1986.

[9] B. Kamrad and P. Ritchken. Multinomial approximating models for option with k states variables. *Management Sciences*, 37(23):1640–1652, 1991.

[10] Edward Omberg. Efficient discrete time jump process models in option pricing. *Journal of Financial and Quantitiatve Analysis*, 23(2):161–173, 1998.

[11] Dai Y. Dai, T. The bino-trinomial tree: A simple model for efficient and accurate option pricing. *The Journal of Derivatives*, 17(4):7–24, 2010.

[12] Christopher Salvi. *Numerical Methods for Finance: Lecture Notes.* Imperial College London, 2023.

[13] J. Crank and E Nicotson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. *Advances in Computational Mathematics*, 6:207–226, 1996.

[14] Mayers D. F. Morton, K. W. *Numerical Solution of Partial Differential Equations: an Introduction.* 2005.

[15] John C. Strikwerda. *Difference schemes and partial differential equations.* Philadelphia, PA : Society for Industrial and Applied Mathematic, 2004.

[16] Daniel J.Duffy. A critique of the crank nicolson scheme strengths and weaknesses for financial instrument pricing. *Wilmott magazine*, pages 68–76, 2004.

[17] P.A. Forsyth R. Zvan, K.R. Vetzal. Pde methods for pricing barrier options. *Journal of Economic Dynamics Control*, 24:1563–1590, 2000.