POLITECNICO DI TORINO

Corso di Laurea
in Ingegneria Matematica

Tesi di Laurea

Pricing of financial derivatives : focus on american put options

Relatori
prof. Patrizia Semeraro
prof. Claudio Mattalia
firma dei relatori

. . . .
. . . .

Candidato
Filippo Scaramozzino

firma del candidato

. . . .

Anno Accademico 2024-2025

†

Summary

The first part introduces key financial derivative concepts, such as forwards, futures, swaps, and options. It provides an overview of basic pricing methods, including parameters like stock price, strike price, volatility, and time to maturity. The section also explains foundational models such as Black-Scholes, put-call parity, and Geometric Brownian Motion (GBM), supported by Python simulations to demonstrate how stock prices evolve. The second part delves into more complex exotic options like Bermuda options, barrier options, and Asian options. These options differ from standard ones in terms of payoff structures and risks. The section also includes Python code for pricing exotic options and simulating stock price behaviors. In addition, it covers portfolio management, focusing on risk measurement, variance, correlation, and optimization strategies for building portfolios with multiple assets, including minimum variance portfolios. The third part explores the use of machine learning for pricing American put options, contrasting it with traditional approaches like Longstaff-Schwartz's LSM method. The section outlines the neural network model used to estimate the continuation value, improving the accuracy of option pricing. The final chapter provides a detailed explanation of the implementation, from data preparation to training and testing the model, with Python code integrated throughout.

Acknowledgements

I would like to express my gratitude to the many people who have supported me throughout my university journey. The achievements of these past years would not have been possible without their help.

First and foremost, I thank my family for their support throughout my time at the Politecnico, they have been a fundamental support to my career. I am especially grateful to Maria Antonietta, who played a pivotal role during my Master's in Mathematical Engineering. Her support extended from coursework and exam preparation to invaluable guidance beyond the classroom, and she was there for me in every circumstance.

I would also like to thank my friends and colleagues, whose encouragement was essential during my studies at the Politecnico. Their support was invaluable in countless situations, and I am particularly grateful to Abdelouahab, Shaoyong, Edoardo, and Rawedi for their assistance whenever I needed it. Their ability to provide technical help and jokes in certain situations was appreciated.

Contents

Part I

Introduction

Chapter 1

General Introduction about Financial Derivatives

1.1 General concepts

To provide an exhaustive introduction about financial derivatives without going too much in the details we will refer to the Hull's book Hull [2021]. Financial derivatives are a fundamental part of financial markets, derivatives are contracts between two individuals or institutions in which one counterpart buys the contract and the other counterpart sells the contract. To cite some of the most important derivatives that we will analyze later we have : forwards, futures, swaps and options. Our major focus in this thesis will be about options and how to price them. Derivatives are bought and sold (traded) on specific markets/exchanges or they can be traded "over-the-counter". In the case of trading on exchanges the most commonly traded derivatives are futures and options, when we refer to "over-the-counter" we mean that the exchange is not done on regulated markets, instead it is done with an agreement between two financial institutions, in this case forwards and swaps are usually agreed. Derivatives are instruments whose value depends on the value and properties of another asset, like the price or the historical progress of the asset, that in case of derivatives is called the underlying. Financial derivatives are just one type of derivatives, but there exist many other types of derivatives in other fields like energy, insurance, credit and real estate. In USA one of the most important exchanges for derivatives contracts is the Chicago Board Of Trade(CBOT) while in Europe one of the most important exchanges for derivatives is the Eurex. This Chapter will closely follow the book of Hull [2021] and Merton [2022].

1.2 Forwards

Forwards contracts are derivatives used to buy or sell a specific amount of an asset at a predetermined price, one counterpart agrees to buy the asset (the asset can be a commodity or currency) at the forward price that is agreed when issuing the contract at a future time instant, the other counterpart that sells the contract agrees to sell at the forward price the asset in the future. As anticipated before these types of contracts are usually traded in the over-the-counter market between financial institutions. The institution that buys the asset is said to have assumed the long position, while the one who sells is said to have assumed the short position. At the initial time when the contract is issued the price to buy or sell a forward contract is equal to zero and there is no exchange of cash-flow between the two counterparts, the exchange of money and asset will be realized only at the future time specified by the contract that is usually referred to as "maturity" or "delivery time". Forwards and derivatives in general are a fundamental tool for hedging financial risks, in particular forwards can be used to hedge against increases in the prices of assets; for example suppose that we want to hedge the risk of increase in the price of a commodity that we will have to buy in the future, the forward allows us to buy it in the future at a predetermined price.

1.3 Futures

Futures contracts, similarly to forwards, are agreements to buy or sell an asset in the future at a fixed price. Differently from forwards, futures are traded on a regulated exchange, and the maturity is not necessarily specified. Since futures are traded on an exchange they reduce the risk of default of one of the two individuals in the contract by using marking-to-market. Some of the main exchanges of futures contracts in Europe are Eurex and ICE Futures. The exchange used to trade futures may specify a range for daily price movement limits, a limit move is a change in the price such that the price reaches the boundaries of the price range either up or down. The daily price limits is used to prevent high variations in prices due to excessive speculations on the futures contracts. For the same reason there also exist position limits that are limits in the number of contracts that each individual can buy or sell. As anticipated before futures reduce the risk of default of one of the two counterparties of the contract, the investor must deposit a certain quantity of money or collateral (it depends on the underlying of the futures contract) in a margin account and the initial amount that must be deposited is known as initial margin. Each day at the end of the day the margin account will be adjusted according to the gain/loss of the investor on the contract computing the gain or loss using the variation of the price of the contract and adding/subtracting this quantity to the margin account : this technique is called marking to market. At the end of each day the theoretical value of a future contract is equal to zero, so the cost to buy a contract is equal to zero. The investor can withdraw money when the value of the account is greater than the initial margin and the maximum quantity of money that can be withdrawn must be the difference between the value of the account and the initial margin. To avoid the risk of default of an individual, the balance of the account must not be negative so maintenance margins are used, these are lower than initial margin and if the balance of the account is lower than the maintenance margin a margin call is done and the investor will be asked to put an amount of money (or collateral) on the account to bring the value of the account at the initial margin, if the investor does not put money in the account his position will be closed. When the delivery month of a futures contract is approaching, the futures price converges to the spot price of the underlying asset. If the price of the futures contract were higher during the delivery month traders would short the futures and buy the asset to perform deliveries, if the price were lower people who need the commodity would buy directly the futures. In both cases the contract price would converge to the spot price as many people exploit this difference in prices. Basically the difference in price opens up the opportunity of arbitrage which means the opportunity of creating a positive payoff without any risk; when this opportunities are exploited prices tend to converge to an equilibrium. If the futures price increases with the maturity this is called normal market, if it decreases it is an inverted market. Hedging with futures means to take a position in a contract whose payoff at maturity can offset the possible loss on another side, short hedge is done if we already own an asset and we expect its price to go down, instead long positions are useful to fix now the price for an asset whose price is expected to increase in the future.

1.3.1 Stock Index Futures

Another hedging instrument to hedge a portfolio against unwanted trends of the market are stock index futures. The Euronext market includes the italian derivatives market, there are different futures contracts on the Ftse MIB index : Futures, Mini Futures and Micro Futures. The fair value of this contracts is computed by multiplying a scalar factor measured in euro to the value of the index, in the case of italian contracts for Futures the scalar is 5, for Mini Futures is 1 and for Micro Futures is 0.2.

1.4 A first measure of risk

One important concept of derivatives is basis risk, the basis is defined as the spot price of the asset minus the price of the corresponding futures contract, at maturity the basis is zero and depending on the asses it may be either positive or negative before maturity, it indicates the risk that the spot price of the asset and the futures price do not converge to the same value. This means that our hedging of the position will not be accurate, and price changes will not be offsetted by our hedging instrument.

At initial time :

$$b_0 = S(0) - F(0, T)$$

while at maturity, in normal conditions of the market :

$$b_T = 0$$

1.5 Interest Rates

A basic understanding of how interest rates work is fundamental to analyze derivatives such as options and swaps, we will try to identify the most important rates for our purposes:

- Treasury rate : is the rate at which the government can borrow money from individuals in its own currency, usually this rate is also called risk free rate
- LIBID/LIBOR : these are the rates at which banks can lend money to other banks (London Interbank Offer Rate LIBOR) or take money to deposit from other banks (London Interbank Bid Rate LIBID) for short periods of time (months), these are not risk-free
- Repo rate : used when one individual can temporarily sell its asset and buy it back later at a higher price determined by the repo rate
- Zero rate: rate at which an amount of money is invested for some years, it is computed annually, the zero curve is a plot of the zero rates versus the years of the investment
- Forward rate : rate at which an amount of money is invested for a certain period starting in the future

We can find the values of the current LIBOR rates on the website of Global-Rates and we can look at the picture of the LIBOR rates for USD over the years, from 1987 to 2024:

Figure 1.1. LIBOR rate for 6 months maturity, from 1987 to 2024

1.5.1 Term Structure of Interest Rates

When we plot the interest rate on the y-axis and the maturity for the investment on the x-axis we obtain a yield curve. The yield curve reflects expectations of the market for long-term investments, the set of interest rates for each maturity compose the term structure. Regarding the term structure, different theories exists about its shape, we try to summarize the most important ones:

- Expectations theory : the long-term interest rates are the expectation in the future of the current short-term interest rates
- Segmentation theory : the short-term, medium-term and long-term rates are not correlated
- Liquidity preference theory : the long-term interest rates should be higher because the risk associated with longer period investments is higher and the investor chooses to reduce his liquidity to perform an investment, the premium to take this risk should reflect in a higher interest rate; this is the most commonly used theory.

1.6 Swaps

Another important derivative that is traded between two financial institutions is the swap. Swaps are not traded by retail investors and they are a modern instrument to hedge different risks, such as interest rate risk. For example swaps can be used by institutions to transform a floating rate liability into a fixed rate loan. A swap is a contract between two individuals to exchange cash flows periodically in the future based on some rules defined when issuing the contract. To enter into a swap contract it costs zero at the beginning. There exist different types of swaps, the most common are:

- Interest rate swaps : based on the exchange of two quantities (only the net difference will be exchanged) computed on a notional principal for several timesteps, one individual will pay a fixed rate on the principal while the other will pay a rate that is chosen at the beginning of each time period (floating rate), for example if we use the LIBOR for 6 months periods as floating rate, at the beginning of each period the LIBOR rate is observed and at the end of each period the exchange of the net difference between the two quantities is done.
- Currency swaps : exchange between two individuals of interest of a notional principal (the principal is chosen to be the same for the two individuals according to their exchange rate for the two currencies) in two different currencies, each individual will pay in one currency and receive payments in another currency, it can be used to hedge currency risk

1.7 Options

Differently from the derivatives seen before, options do not constrain an individual to buy or sell an asset, they give the possibility to buy or sell. We consider two types of options, the call and the put. The person who buys (holder) the call option has the right to buy the underlying (the asset) at a specific price K (strike price), instead the put option gives the right to sell an asset at strike price. An european option can be only exercised at maturity, while an american option can be exercised at (almost) any time during the horizon of investment. The person who buys the option is called *holder* while the person who sells the option is called *writer*. There exist many types of options, in this first part we will analyze "vanilla" options and later we will analyze "exotic" options. Options can be written on stocks, stock indices, foreign currencies and futures contracts. Usually stock options contracts give the holder the right to buy or sell 100 shares. Instead index options can be written for example on S&P500 (SPX), Nasdaq 100 (NDX) and Dow Jones Industrial (DJX), in these cases the option gives the right to buy or sell a specific number of times the value of the index at a strike price. For what concerns foreign currencies, the option gives the right to buy or to sell a specific amount of currency. Futures options are options on futures, so the underlying is a futures contract, for example if the holder exercises a call option he gains from the writer the long position in a futures contract that will mature shortly after the maturity of the option plus a cash amount that is equal to the futures price minus the strike price: $f(t,T) - K$, instead with a put option the holder gains the short position in the futures contract plus an amount of cash equal to $K - f(t,T)$. In USA the CME Group provides a large number of different options on different underlyings: Indexes, Interest rates, Forex, Cryptocurrencies, Energy, Agriculture, Metals. In Italy, Borsa Italiana provides options on the main index, FTSE MIB and stocks. When we trade an option the exchange may declare some information like expiration date (maturity), strike price, dividends and how large is the contract position. Options trade with a specific maturity based on the current month, usually is a couple of months from the current month but LEAPS (Long-Term equity anticipation securities) are options with longer expiration date, up to three years and the expiration is usually in January. Equity LEAPS calls can provide benefit from the growth of companies without having to buy stocks, puts instead can provide a hedge against substantial declines in underlying. Index LEAPS options let an investor take a bullish or bearish position on the entire market. Some of the derivatives discussed above can be traded on the CBOE Chicago Board Options Exchange website. The intrinsic value of an option is the maximum between zero and the payoff if the option were exercised immediately.
If we have a long position either in a call or a put option the payoffs will be:

$$\text{payoff(call)} = max(S(T) - K, 0)$$
$$\text{payoff(put)} = max(K - S(T), 0)$$

instead, if we have a short position:

$$\text{payoff(call)} = -max(S(T) - K, 0)$$
$$\text{payoff(put)} = -max(K - S(T), 0)$$

Before going more in detail, we analyze a real example of option.

1.7.1 Example : Index options on FTSE MIB

Let's look at a practical example, we put ourselves in the position of an investor who wants to trade an option on Borsa Italiana on the index FTSE MIB. In August 2024 the value of the index is around 32.000 , so for simplicity let's analyze call and put options with strike $K = 32.000{,}00$ with maturity in October 2024. We analyze the different positions that an investor can take and simulate possible outcomes of profit/loss. To understand the data, we summarized this setting in a table:
(All the data were taken directly from Borsa Italiana, as today 4th August 2024, market closed)
Note : By now we will not be concerned of how this parameters can be computed and how prices of options can be estimated, we will just provide a real-scenario example to understand how it works.

Type	Price	Strike	Value Underlying	Multiplier(euro)
call	1.410,00	32.000	32.018,82	2,50
put	885,00	32.000	32.018,82	2,50

Table 1.1. Parameters of Options and Underlying, on 4th August 2024 for european options with maturity 18th October 2024

The multiplier is the value in currency (in this case euro) of each index point, in our case each index point will be multiplied by 2,50 to obtain a value in euro. For our example, suppose we will buy only one option on the index and let's analyze the possible outcomes we could have by taking four positions:

- long call
- short call
- long put
- short put

The Python code for the following plots and simulations can be found on my GitHub, it is necessary to download the file and open it with a Python notebook editor. In order to carry out our analysis we used steps of 50 index points, the real value of the index was 32.018,82 and we supposed values that went from 31.000,00 to 33.000,00.

To analyze the results, we will use this notation :

- S: Current value of the index
- K: Strike price of the options
- C: Price of the call option
- P: Price of the put option
- PL: Profit or Loss
- a: Price per index point, it is the multiplier, in our case is 2,50€

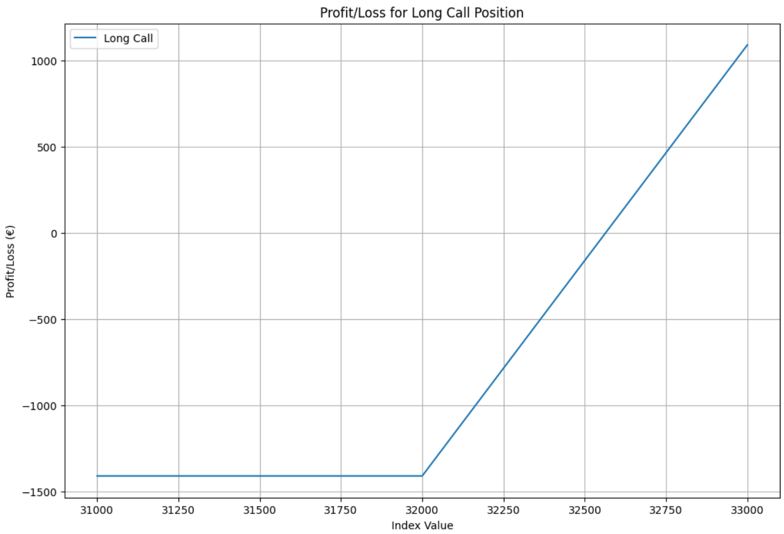

Figure 1.2. Long position in a call

For the long position in the call the profit can be computed as :

$$PL_{\text{Long Call}} = \max(0, a(S - K)) - C$$

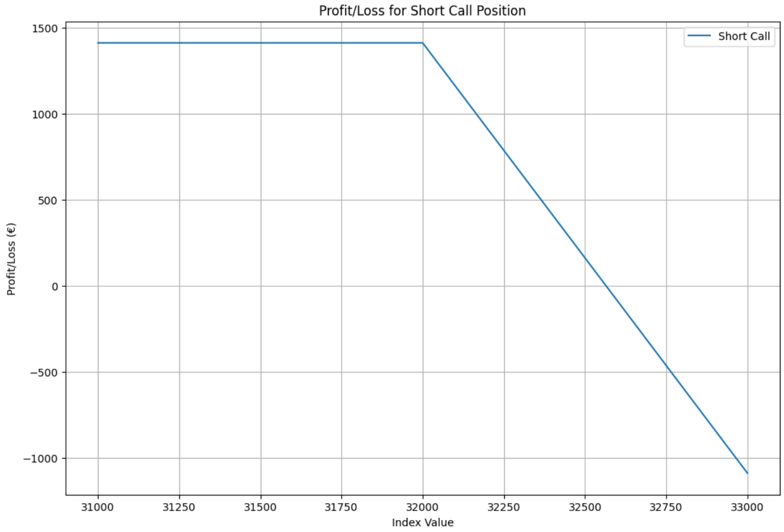

Figure 1.3. Short position in a call

For the short position in the call the profit can be computed as :

$$PL_{\text{Short Call}} = C - \max(0, a(S - K))$$

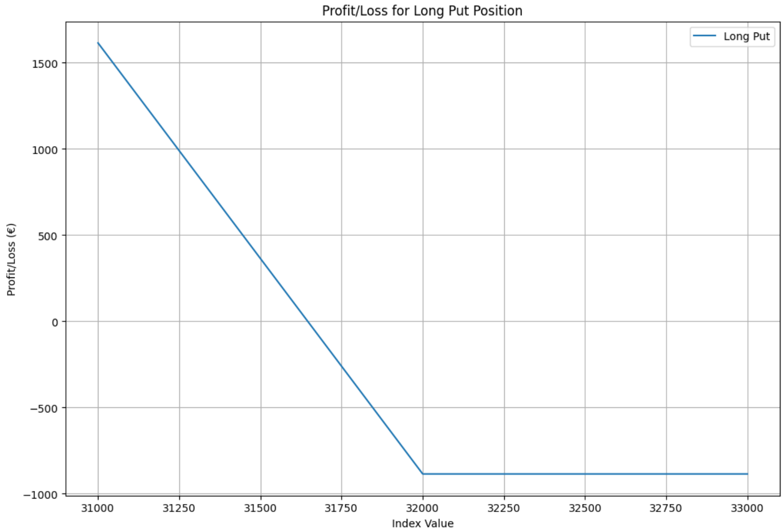

Figure 1.4. Long position in a put

For the long position in the put the profit can be computed as :

$$PL_{\text{Long Put}} = \max(0, a(K - S)) - P$$

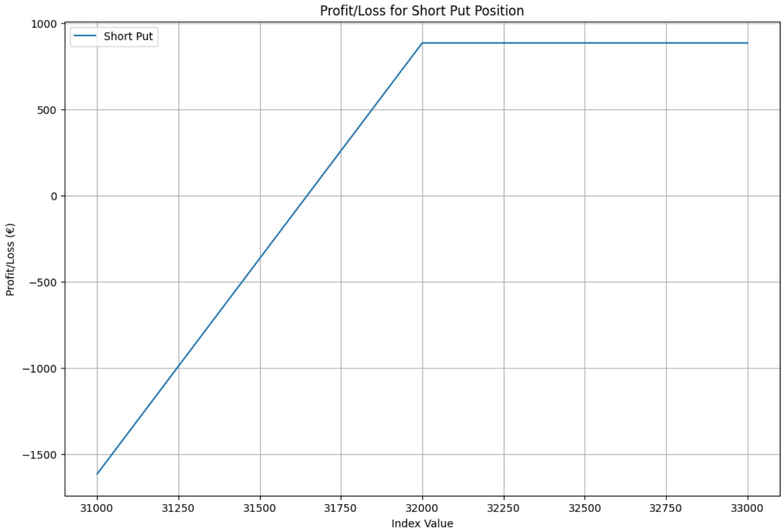

Figure 1.5. Short position in a put

For the short position in the put the profit can be computed as :

$$PL_{\text{Short Put}} = P - \max(0, a(K - S))$$

1.7.2 More details on Vanilla Options

As anticipated before, American options are options that can be exercised when the investor wants to do so. The value of an American in-the-money option must be at least as much as its intrinsic value because the holder can exercise it immediately, even if usually it is better to wait. The total value of an option is the sum of the intrinsic value and the time value, which is the value of movements in the underlying that are well accepted by the holder. The time value is zero in two cases : when the option has reached maturity and when it is optimal to exercise the option immediately. There exist also options where strike prices and expiration dates are different from the ones mentioned before, in this case we call them Flex Options. Exchange-traded options are not adjusted in the strike price when a stock pays dividend but they are adjusted in the contract for stock splits and dividends. Options on stocks are adjusted for stock dividends, that is when the company issues more shares to people who already own shares of the company. Stock dividends have no effect on the asset, like the splits. Stock splits happen when a share splits in more shares, for example 1 share may divide into 3 shares without changing the asset of the owners, in a n-for-m split the stock price goes down by m/n times its previous value. After a n-for-m split, the parameters of the option get adjusted accordingly : the strike price is multiplied by m/n, and the number of shares that represent the underlying of the contract is multiplied by n/m. To be more clear, let's look at this example taken from Hull [2021].

Example 7.1 from Hull [2021] :
A contract of a call option to buy 100 shares is trading for $30 per share, the stock splits in 2-for-1, the contract is adjusted to give the holder the right to purchase 200 shares for $15 per share

Example 7.2 from Hull [2021] :
A put option to sell 100 shares of a company is trading for $15 per share, the stock dividends are 25% , in this case it is equivalent to a 5-for-4 stock split, the contract is adjusted to permit the holder to sell 125 shares at $12 per share

The volume of options is the total number of options exchanged on a specific day on a specific exchange while the option open interest is the number of outstanding options. Market makers quote the bid and offer price for an option, the bid is the price at which the market maker is willing to buy the option while the offer price is the price at which he sells the option, the offer price is always higher than the bid price, the difference between the offer and bid price is called the bid-offer spread. Brokers may include commissions for trading of options, usually with a fixed quote plus a variable one based on the number of contracts or dollars invested. For stocks the commissions are usually computed as a percentage, so if we close out a position on an option by offsetting the position we will pay the option commissions again, while if we exercise the option we will pay the stock commissions. Offsetting is the technique based on closing a position by taking the opposite position on the same option, for example if we are in a long position we can enter a short position with another investor to sell our option, the net result will be that of having no positions. Usually if an option is exercised commissions will be higher than selling it. We may consider an hidden commission cost in the bid price at which we buy the option, whose fair price is a bit less than the one quoted by the market. Usually options cannot be bought on margin and for the writer of the options the broker requires an amount of money to be deposited in the account to be sure that the writer will be able to respect the contract The options clearing corporation is similar to the clearing house for futures, basically in this case it will be responsible to guarantee that the writer of the option will be able to fulfill the contract and it will fund if some default of writers will happen. The holder of option will deposit the price paid for the option in the OCC and the writer will maintain a margin in the broker, if the broker is not a member of the OCC it will have a margin account in the OCC. Other types of options include Warrants and Executive stock options, that are options written by a company on its own stocks, warrants usually come with a bond to increase the value of the bond contract, executive stock options are sold to employees and managers to motivate them to work the best possible way for the company.

Chapter 2

Pricing Options

After the introduction about derivatives, we now introduce some standard models to price options and we try to understand what the prices of options represent; again in this Chapter we will closely follow some of the approaches from Hull [2021], Merton [2022] and Black and Scholes [1973]. We will try to implement some of those approaches to price options in Python.

2.0.1 Assumptions

We will make assumptions regarding the structure of the market and brokers, these include:

- perform trading of derivatives without paying transaction costs when we buy or sell
- we can borrow or invest cash at the risk-free rate
- there are no arbitrage opportunities

2.1 Parameters

The parameters we will use to perform our analysis are :

- $S(t)$ = the stock price where t represents a time instant
- K = the strike price at maturity
- T = the maturity, supposing we start at time $t = 0$ the maturity is
- σ = the standard deviation of the random variable represented by the stock price, it is also called volatility
- r = the risk-free interest rate

2.1.1 Analysis of parameters

The first parameter affecting the price of the option is the stock price. For call options the price of the contract increases if the stock price is increasing, because the expected payoff from exercising the option is greater. Instead, the price of put options will decrease if the stock price increases, because the expected payoff from exercising the option will decrease, and the opposite behavior in the contracts prices is observed when the stock price decreases. For the same reasoning of the expected payoff, when there is an increase in the strike price the call price will decrease, while the put price will increase, the opposite trend is observed when the strike decreases. Longer time to maturity makes the prices of both call and put to increase if the stock pays no dividends, if we have dividends it may depend on the case, note that when dividends are paid the stock price decreases and so call prices may decrease, and put prices may increase. Volatility is a measure of how large the upward and downward movements of the stock price are; for both call and put options high variability of prices may be well accepted because depending on the type of option it may lead to a high payoff if the stock price moves in the direction desired by the investor and lead to a risk-limited position if it moves in the direction that the investor does not desire as we will only lose the premium paid to hold the option, so both call and put prices will increase with volatility. If we assume all the parameters to be constant and we move the interest rate the prices of call options will increase if the rate is increased, instead the prices of put options will decrease. If the interests are higher the expected return on stock prices will be higher, but in reality, the stock prices will decrease so it is not straightforward to understand how prices of options will move.

2.2 Bounds of Options prices

We analyze some basic bounds for options prices, we will use the index 'e' for European options and 'a' for American.

- Upper bounds for European and American call :

$$c_e \leq S_0 \qquad c_a \leq S_0$$

- Upper bounds for European and American put :

$$p_e < Ke^{-rt} \qquad p_a < Ke^{-rt}$$

- Lower bounds for European and American call :

$$c_e \geq S(0) - Ke^{-rt} \qquad c_a \geq S(0) - Ke^{-rt}$$

2.3 Put-call Parity

Assuming we are dealing with options with :

- same underlying
- same strike
- same expiration date

we will try to derive a relationship between the price of European call and put options, called 'Put-Call Parity', as done in the Hull [2021] book. We construct a portfolio consisting of a long position in a call option and a short position in a put option. The total payoff of the portfolio will generate the payoff of a forward contract, that is $S(T) - K$ at time T. We assumed before that there are no arbitrage opportunities, so two portfolios that have the same payoff at time T must have the same value at time $t = 0$. Consider a second portfolio by buying one share $S(0)$ and selling zero-coupon bonds with face value K, the value at time $t = 0$ of the second portfolio is $S(0) - Ke^{-rT}$. We may note that the payoff of the second portfolio is also $S(T) - K$ at time T, like the first portfolio. In conclusion we obtain that :

$$c_e - p_e = S(0) - Ke^{-rT}$$

2.3.1 Implementation

In order to plot the resulting payoff from the first portfolio I used a Python script. The code implemented can be found here. In the Python notebook I used the Black-Scholes formula for European call and put options to create prices based on these fixed parameters, chosen realistically:

- $S(0) = 100$ is the current stock price
- $K = 100$ is the strike price
- $T = 1$ is the time to maturity (measured in years)
- $r = 0.05$ is the risk-free rate
- $\sigma = 0.2$ is the volatility of the stock price

The Black-Scholes formula will be analyzed in detail later, for the moment we only focus on having the prices to verify if the put-call parity holds. In the script, after computing the prices of options and verifying that the put-call parity holds, we plot the payoff of the portfolio and graphically the payoff of a forward can be recognized:

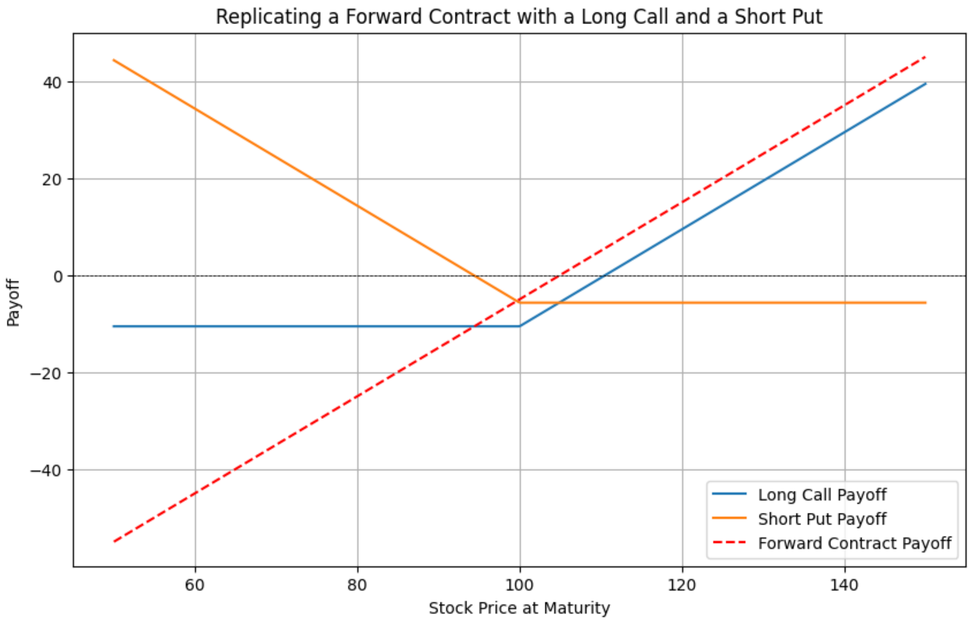

Figure 2.1. Payoff of portfolio in red dashed line

2.3.2 Put-call Parity for American options

For American options, there is no equation relating the prices of call and put but we have that the following inequality holds:

$$S(0) - K \leq c_a - p_a \leq S(0) - Ke^{-rt}$$

2.3.3 Early exercise for American options

It is never optimal to exercise an American call option before maturity if the stock is not paying dividends and if the investor will keep the stock for a time longer than the time to maturity. Instead for a put option, it is optimal to exercise it early if it is in the money. So we have the following inequalities for the prices of American call and put :

$$p_a > p_e \qquad c_a \geq c_e$$

2.3.4 Options on stocks that pay dividends

When we deal with stocks that pay dividends in the future, in the equation of the put-call parity we will subtract from the current price of the stock $S(0)$ the present value of the dividends d_0, as the stock price will decrease by that amount when the dividends will be released.

The put-call parity for European options becomes :

$$c_e + Ke^{-rt} = p + S(0) - d_0$$

and for the American options :

$$S(0) - d_0 - K \leq c_a - p_a \leq S(0) - Ke^{-rt}$$

When dividends are released, the early exercise of an American call option makes sense the day before the dividends are released.

Chapter 3

Stock Pricing

In order to price derivatives with more accuracy, we need to build models that represent the underlying behavior. In our case, when we want to price stock options, it is necessary to have a model that represents the stock's price movements. To build such a model, we may introduce some properties that we assume the stock will follow. In this Chapter will follow Hull [2021] book, and Black and Scholes [1973].

3.0.1 Markov Property

We assume that stock prices are modeled as stochastic processes that follow the Markov property. This implies that the future price of the stock is determined only by its present value, and not by the history of past prices. In other words, all the relevant information about the future is encapsulated in the current price.

To capture the random nature of stock price movements, we will use a continuous-time stochastic process model. These models allow for a realistic representation of asset price dynamics by incorporating both the drift (expected return) and volatility (random fluctuations).

3.0.2 Normal Distribution

For a stock price process where price changes are assumed to follow a normal distribution, we can apply the properties of normal distributions to analyze the behavior of price changes over time. When price changes are measured at different time intervals, the sum of two independent normal distributions remains normally distributed. Thus, over a time horizon T, the price change will follow a normal distribution $N(0, T)$, where the mean is zero and the variance scales linearly with time.

If we use standard deviation as a measure of risk, we observe that the risk (measured as the standard deviation of the random variable that represents the returns) is proportional to the square root of time. This relationship implies that as time increases, the uncertainty or risk associated with the stock's price also increases, but at a slower rate than the passage of time itself. Mathematically, this is observed by the fact that the standard deviation of a normally distributed process with variance T is $\sqrt{T}$.

3.1 Wiener Processes

Wiener processes are Markovian stochastic processes that describe the behavior of a random variable over time. These processes have the Markov property, meaning the future value of the variable depends only on its current value and not on the past.

The change in the value of a variable following a Wiener process, denoted by dz, is a random variable that is normally distributed with a mean of zero and a variance proportional to the time interval dt over which the change is measured, $dz \sim N(0, dt)$. The mean is zero and the variance equals the time increment dt. This property indicates that over very short time intervals, the change is small.

For a process starting at time $t = 0$, the expected value of the variable at any future time T is equal to its current value at time t since it is a zero mean process. In this case indeed we are considering the absence of any drift. However, the variance of the variable increases linearly with time, such that the variance at time T is equal to T. This growing variance captures the property of increasing uncertainty as time progresses.

3.2 Generalized Wiener Processes

A generalized Wiener process is an extension of the standard Wiener process where both drift and volatility components are introduced. The stochastic differential equation for a generalized Wiener process can be written as:

$$dx = a\,dt + b\,dz$$

where:

- dz represents the standard Wiener process (Brownian motion), which follows a normal distribution $N(0, dt)$ with zero mean and variance proportional to the time increment dt
- a represents the drift coefficient, which determines the average rate of change in x over time
- b represents the volatility coefficient, which measures the magnitude of random fluctuations in the process
- dx is the increment over a small time step

For the generalized Wiener process, the increment dx is normally distributed:

$$dx \sim N(a\,dt, b^2\,dt)$$

- The mean of the increment dx is $a\,dt$, indicating that the process tends to drift by $a\,dt$ over each time interval
- The variance of dx is $b^2\,dt$, reflecting that the randomness of the process grows proportionally with time and is scaled by the volatility b

In the generalized Wiener process, the term $a\,dt$ represents the deterministic component (drift), while the term $b\,dz$ introduces randomness via the Wiener process.

Generalized Wiener will be fundamental in the Black-Scholes equation, where they are employed to model the stochastic behavior of asset prices. The drift a represents the expected return, and the volatility b captures the uncertainty or risk associated with the asset.

3.3 Itô Process

An Itô process is an extension of the generalized Wiener process where the drift and volatility coefficients are no longer constants, but functions of both the underlying variable (the asset) x and time t. This allows for greater flexibility in modeling stochastic processes where the dynamics change over time.

The stochastic differential equation for an Itô process is given by:

$$dx = a(x,t)\,dt + b(x,t)\,dz$$

where:

- dz represents the standard Wiener process, which follows a normal distribution $N(0, dt)$ with zero mean and variance proportional to the time increment dt
- $a(x,t)$ is the drift term, a function of both the underlying variable x and time t, representing the deterministic component of the process
- $b(x,t)$ is the volatility term, also a function of x and t, which scales the random fluctuations driven by the Wiener process

For an Itô process, the increment dx over a small time interval dt is normally distributed:

$$dx \sim N\left(a(x,t)\,dt, b^2(x,t)\,dt\right)$$

- The mean of the increment dx is $a(x,t)\,dt$, which represents the expected change in x over the time interval dt, based on both the current state x and the time t
- The variance of dx is $b^2(x,t)\,dt$, showing that the randomness of the process, or volatility, depends on both the state x and the time t

The Itô process introduces a key improvement over the generalized Wiener process by allowing the drift $a(x,t)$ and volatility $b(x,t)$ to vary with both the underlying variable x and time t. One of the most important results involving Itô processes is *Itô's Lemma*, which is used to derive the differential of a function of an Itô process. This is fundamental in derivative pricing, such as the Black-Scholes model, where asset prices follow stochastic dynamics with parameters that depend on time and state.

3.4 Stock Model - Geometric Brownian Motion

Geometric Brownian Motion (GBM) is a model that describes the stochastic behavior of stock prices. It is based on the assumption that the relative change in stock price follows a stochastic differential equation consisting of a deterministic drift and a random component driven by a Wiener process.

The stochastic differential equation for a stock price $S(t)$ under the Geometric Brownian Motion model is given by:

$$\frac{dS}{S} = \mu\, dt + \sigma\, dz$$

where:

- $S(t)$ represents the stock price at time t
- μ is the drift coefficient, representing the expected rate of return of the stock
- σ is the volatility coefficient, representing the degree of randomness or uncertainty in the stock's price movement
- dz is the increment of a standard Wiener process, $dz \sim N(0, dt)$, which introduces randomness into the system

In discrete time intervals, the continuous-time GBM equation can be approximated. Over a small time interval δt, the change in stock price δS can be written as:

$$\frac{\delta S}{S} = \mu\, \delta t + \sigma\, \epsilon\, \sqrt{\delta t}$$

where:

- ϵ is a standard normal random variable, $\epsilon \sim N(0, 1)$
- The term $\sigma\, \epsilon\, \sqrt{\delta t}$ captures the random fluctuations in the stock price due to the volatility σ and the Wiener process

In the discrete-time approximation, the relative change in stock price $\frac{\delta S}{S}$ is normally distributed:

$$\frac{\delta S}{S} \sim N\left(\mu\, \delta t, \sigma^2\, \delta t\right)$$

Analyzing the distribution we may note that:

- The mean of $\frac{\delta S}{S}$ is $\mu\, \delta t$, which represents the expected return over the time interval δt
- The variance of $\frac{\delta S}{S}$ is $\sigma^2\, \delta t$, indicating that the randomness in the stock price scales with the volatility σ and the time interval

The GBM model assumes that stock prices grow exponentially over time and that the random component scales with the square root of time, reflecting the increasing uncertainty over longer time horizons.

3.4.1 Monte Carlo Simulation

Now we will implement a Python code to simulate the behavior of a stock that follows this model, which is just a rewritten form of the equation of the GBM:

$$\delta S = \mu S \delta t + \sigma S \epsilon \sqrt{\delta t}$$

The parameters used in this simulation are:

- $S_0 = 100$ is the initial stock price
- $\mu = 0.05$ for the drift (annual return)
- $\sigma = 0.2$ is the volatility (annualized standard deviation)
- $T = 1$ is the time horizon in years, one year in our case
- $N = 1000$ is the number of time steps

The Python code can be found here, and the result obtained with this simulation is shown below. The simulation was done on ϵ by drawing samples from a standard normal distribution, these samples were used to compute the changes in the stock price with respect to the previous value of the stock, using the equation written above.

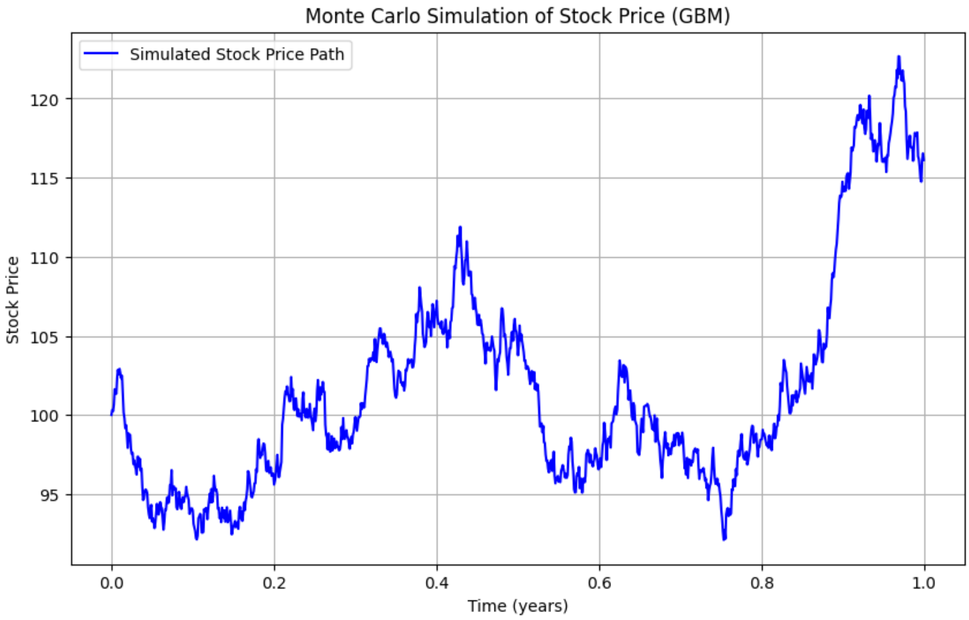

Figure 3.1. Simulation of stock price behavior using a GBM model

3.5 Itô's Lemma

Itô's Lemma provides a way to find the differential of a function $G(x, t)$, where x follows an Itô process, this lemma is used to model the evolution of derivatives. We consider an Itô process for a variable x described by the stochastic differential equation:

$$dx = a(x, t)\, dt + b(x, t)\, dz$$

Now, consider a function $G(x, t)$ that depends on both the stochastic process x and time t. Itô's Lemma provides the differential for $G(x, t)$, which follows the following equation:

$$dG = \left(\frac{\partial G}{\partial x} a + \frac{\partial G}{\partial t} + \frac{1}{2} \frac{\partial^2 G}{\partial x^2} b^2 \right) dt + \frac{\partial G}{\partial x} b\, dz$$

This equation describes the evolution of the function $G(x, t)$ in terms of its partial derivatives with respect to x and t, as well as the drift $a(x, t)$ and volatility $b(x, t)$ of the underlying Itô process for x.

The drift rate (or the expected rate of change) of the function $G(x, t)$ is given by:

$$\frac{\partial G}{\partial x} a + \frac{\partial G}{\partial t} + \frac{1}{2} \frac{\partial^2 G}{\partial x^2} b^2$$

This term represents the deterministic part of the evolution of $G(x, t)$.

The variance rate of $G(x, t)$ is given by:

$$\left(\frac{\partial G}{\partial x} \right)^2 b^2$$

This term represents the randomness in the evolution of $G(x, t)$.

Suppose we apply Itô's Lemma to a function $G(x, t) = \ln(S)$, where S is a stock price that follows a Geometric Brownian Motion. In that case, the lemma helps to prove that the logarithmic change in stock prices over a time interval δt, $\ln(S_t) - \ln(S_0)$ is normally distributed.

The mean of the logarithmic return is:

$$\left(\mu - \frac{\sigma^2}{2} \right) \delta t$$

The variance of the logarithmic return is:

$$\sigma^2 \delta t$$

This result is crucial for understanding the distribution of stock returns in models like the Black-Scholes option pricing model. It implies that over small intervals of time δt, the logarithmic returns of a stock price are normally distributed with the mean and variance described above.

Chapter 4

Black-Scholes-Merton model

The Black-Scholes formula provides a solution for pricing European call and put options. It assumes that the underlying asset follows a Geometric Brownian Motion, and that the price of the option depends on the time to maturity, the underlying stock price, and volatility.

We report an intuitive derivation of the B&S formula, as in Hull [2021] book.

The derivation begins by modeling the price $S(t)$ of a stock using the Geometric Brownian Motion explained before:

$$dS = \mu S \, dt + \sigma S \, dz$$

Let $C(S, t)$ be the price of a European call option as a function of the stock price S and time t. Using Itô's Lemma for a function $C(S, t)$, we get the following stochastic differential equation for the option price:

$$dC = \frac{\partial C}{\partial t} \, dt + \frac{\partial C}{\partial S} \, dS + \frac{1}{2} \frac{\partial^2 C}{\partial S^2} \, dS^2$$

Since $dS = \mu S \, dt + \sigma S \, dz$, we substitute into the above equation and obtain:

$$dS^2 = (\sigma S)^2 dt = \sigma^2 S^2 dt$$

Thus, the equation for dC becomes:

$$dC = \frac{\partial C}{\partial t} \, dt + \frac{\partial C}{\partial S} \, (\mu S \, dt + \sigma S \, dz) + \frac{1}{2} \frac{\partial^2 C}{\partial S^2} \sigma^2 S^2 \, dt$$

Rearranging terms:

$$dC = \left(\frac{\partial C}{\partial t} + \mu S \frac{\partial C}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} \right) dt + \sigma S \frac{\partial C}{\partial S} \, dz$$

Now we construct a risk-free portfolio.

To eliminate risk, we construct a portfolio Π consisting of a long position in the option and a short position in $\Delta = \frac{\partial C}{\partial S}$ units of the stock. The portfolio is:

$$\Pi = C - \Delta S$$

The change in the value of the portfolio $d\Pi$ is:

$$d\Pi = dC - \Delta\, dS$$

Substituting for dC and dS:

$$d\Pi = \left(\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2}\right) dt$$

Note that the dz terms cancel out because $\Delta = \frac{\partial C}{\partial S}$, which eliminates the randomness in the portfolio. Therefore, the portfolio is risk-free, and it must earn the risk-free rate r. Hence, the value of the portfolio must satisfy:

$$d\Pi = r\Pi\, dt$$

Substituting $\Pi = C - \frac{\partial C}{\partial S} S$:

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} = r\left(C - S\frac{\partial C}{\partial S}\right)$$

Expanding the right-hand side:

$$\frac{\partial C}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} = rC - rS\frac{\partial C}{\partial S}$$

Rearranging this equation gives the Black-Scholes partial differential equation (PDE):

$$\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} = rC$$

For a European call option with strike price K and maturity T, the boundary condition is given by the payoff at expiration:

$$C(S, T) = \max(S - K, 0)$$

To solve the Black-Scholes PDE, we make a transformation using the variable $\tau = T - t$ (the time to expiration). After several steps (including change of variables and solving the heat equation), the solution for the price of a European call option is given by the Black-Scholes formula:

$$C(S, t) = SN(d_1) - Ke^{-r(T-t)}N(d_2)$$

where:

- $d_1 = \frac{\ln(S/K)+(r+\sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}$

- $d_2 = d_1 - \sigma\sqrt{T-t}$

- $N(\cdot)$ is the cumulative distribution function of the standard normal distribution

Using put-call parity, we can derive the price of a European put option:

$$P(S,t) = Ke^{-r(T-t)}N(-d_2) - SN(-d_1)$$

4.1 Volatility

Volatility is a measure of the uncertainty associated with the price movements of a stock. Typically, this uncertainty is expressed with a time horizon of one year. To estimate the volatility of a stock, which is equivalent to estimating the standard deviation of its price changes, we proceed with the following assumptions and notations.

Assumptions

- Each measurement is taken at fixed and constant time intervals.

- The stock does not pay dividends.

Notations

- $n+1$: Total number of observations.

- S_i: Stock price at each time instant $i = 0, 1, \ldots, n$.

- u_i: Logarithmic price change between two subsequent instants, computed as:

$$u_i = \ln\left(\frac{S_i}{S_{i-1}}\right), \quad i = 1, 2, \ldots, n$$

- Δt: Time interval between observations.

The quantity u_i represents the log-change in the stock prices. To estimate the volatility of the stock, we compute the standard deviation of the u_i, denoted as s.

Estimation of Volatility

The standard deviation s of the u_i can be estimated using the formula:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (u_i - \bar{u})^2}$$

where $\bar{u}$ is the empirical mean of the u_i:

$$\bar{u} = \frac{1}{n} \sum_{i=1}^{n} u_i$$

As previously defined, the theoretical standard deviation of the u_i is $\sigma\sqrt{\Delta t}$, where σ represents the true volatility of the stock. Therefore, the estimated value of σ can be expressed as:

$$\hat{\sigma} = \frac{s}{\sqrt{\Delta t}}$$

Trade-off in Estimation

Increasing the number of observations generally leads to a more accurate estimate of σ. However, stock volatility tends to vary over time. Thus, it is important to strike a balance between the number of observations and the relevance of recent data. Typically, volatility estimations are based on data from the past 3 to 6 months, using daily closing prices for stocks.

4.1.1 Estimation of Volatility for non-dividend-paying stock

We now try to estimate volatility for a non-dividend-paying stock, in our example it will be the Netflix stock, the data was found on the Nasdaq website and it was free to download and read. The dataset included the closing price for the stock of the past 6 months. The analysis was carried out using Python, the code can be found here. We applied to the data the method explained before for the estimation of the volatility.

In the picture the first rows of the dataset are reported, and a transformation from dollars to non-unit numbers with two decimals was applied to the first column to correctly compute the standard deviation, the total number of observations was 127.

```
First few rows of the dataset before transformation:
         Date    Close   Volume     Open     High        Low
0  08/30/2024  $701.35  3266723  $700.36  $701.86    $688.16
1  08/29/2024  $692.48  2186974  $690.00  $699.80    $686.07
2  08/28/2024  $683.84  2430583  $695.83  $696.67    $677.10
3  08/27/2024  $695.72  3164878  $688.53  $707.89    $686.92
4  08/26/2024  $688.44  1354154  $687.26  $690.59  $681.6376

First few rows of the dataset after transformation:
         Date   Close   Volume     Open     High        Low
0  08/30/2024  701.35  3266723  $700.36  $701.86    $688.16
1  08/29/2024  692.48  2186974  $690.00  $699.80    $686.07
2  08/28/2024  683.84  2430583  $695.83  $696.67    $677.10
3  08/27/2024  695.72  3164878  $688.53  $707.89    $686.92
4  08/26/2024  688.44  1354154  $687.26  $690.59  $681.6376

Total number of observations: 127
```

Figure 4.1. Dataset

The results did show:

- Estimated standard deviation of the logarithmic(daily) returns $= 0.017 \approx 1.7\%$
- Annualized volatility $= 0.27 \approx 27\%$
- Standard error of the annual volatility $= 0.017$

4.2 More on B&S model

The Black-Scholes formulas give the prices at time $t = 0$ for European call and put options on a non-dividend-paying stock. These formulas provide closed-form solutions based on several parameters.

For a European **call** option, the price is given by:

$$c = S_0 N(d_1) - K e^{-rT} N(d_2) \tag{1}$$

For a European **put** option, the price is given by:

$$p = K e^{-rT} N(-d_2) - S_0 N(-d_1) \tag{2}$$

where:

- S_0 is the current stock price
- K is the strike price of the option
- T is the time to maturity (in years)
- r is the risk-free interest rate (continuously compounded)
- σ is the volatility of the stock price (standard deviation of the stock's returns)
- $N(x)$ is the cumulative distribution function (CDF) of the standard normal distribution

The terms d_1 and d_2 are intermediate variables used in the pricing formulas and are defined as:

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right) T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - \frac{\sigma^2}{2}\right) T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

The function $N(x)$ represents the probability that a normally distributed random variable with mean zero and variance one is less than or equal to x. In other words, $N(x)$ is the cumulative distribution function of the standard normal distribution.

4.2.1 Properties of American and European Options

- For non-dividend-paying stocks, the price of an American call option is equal to the price of a European call option. This is because there is no advantage in exercising an American call option early when no dividends are paid.
- On the other hand, there is no closed-form analytical formula for pricing American put options. These options are more complex to price due to the possibility of early exercise, which must be handled using numerical methods, such as finite difference methods, binomial trees, or Monte Carlo simulations.

4.3 Implied Volatility

In the context of option pricing, the implied volatility is a key concept. It refers to the value of the volatility parameter σ that, when substituted into the Black-Scholes (B&S) pricing formula, yields the market-observed price of an option. In other words, the implied volatility represents the market's view on the future volatility of the stock price, inferred from current option prices.

Mathematically, implied volatility (σ_{impl}) is the solution to the following equation:

$$\text{Option price} = f(S_0, K, T, r, \sigma_{\text{impl}})$$

where:

- S_0 is the current stock price.
- K is the strike price.
- T is the time to maturity.
- r is the risk-free interest rate.
- σ_{impl} is the implied volatility.
- $f(\cdot)$ represents the Black-Scholes formula for either a call or a put option price.

4.3.1 Finding Implied Volatility

Since the Black-Scholes formulas cannot be explicitly inverted to solve for σ, iterative numerical methods are required to determine the implied volatility. Common methods include:

- **Newton-Raphson method**: A root-finding algorithm that iteratively adjusts σ by considering the difference between the observed option price and the Black-Scholes theoretical price.
- **Bisection method**: A bracketing method where the volatility is incrementally adjusted within an interval until the observed option price is closely matched.
- **Secant method**: A faster iterative approach that approximates the derivative of the option price function using secant lines.

These methods are necessary because an explicit closed-form solution for σ does not exist within the Black-Scholes framework, making implied volatility estimation a non-trivial problem.

4.3.2 Market Significance of Implied Volatility

Implied volatility is widely used to infer the market's expectations of future volatility. When investors observe the prices of options, they can deduce the level of volatility that the market deducts over the life of the option. High implied volatility typically indicates that the market expects significant price movements in the underlying stock, whereas low implied volatility suggests that the stock is expected to remain relatively stable.

Traders and investors rely on implied volatility as a measure of risk and potential price fluctuations in the stock.

4.4 Dividends in European and American Options

Dividends play an important role in the pricing and exercise strategy of options. When a stock pays dividends, the treatment of those dividends affects both the price of European options (which cannot be exercised early) and the exercise strategy of American options (which can be exercised before maturity).

4.4.1 Dividends and European Options

For European options, the Black-Scholes (B&S) formula can still be used to price options on dividend-paying stocks, provided that we adjust the stock price to account for the present value of future dividends.

Let:

- S_0 be the current stock price,
- D be the present value of all dividends expected to be paid before the option's expiration.

The adjusted stock price used in the Black-Scholes formula becomes:

$$S_0 - D$$

Thus, the pricing formulas for European call and put options on dividend-paying stocks become:

$$c = (S_0 - D)N(d_1) - Ke^{-rT}N(d_2)$$
$$p = Ke^{-rT}N(-d_2) - (S_0 - D)N(-d_1)$$

where d_1 and d_2 are defined in the usual manner, with S_0 replaced by $S_0 - D$ in the logarithmic term.

4.4.2 Dividends and American Call Options

For American call options, the presence of dividends introduces a situation where early exercise may become optimal. Specifically:

- If a stock does not pay dividends, it is never optimal to exercise an American call option early. This is because the holder of the call option benefits from delaying exercise until maturity in order to maximize the time value of the option.
- If a stock pays dividends, it may be optimal to exercise the American call option just before the ex-dividend date, especially if the dividend is large and the dividend payment date is close to the option's maturity. In such cases, the stock price is expected to drop by the dividend amount after the ex-dividend date, so early exercise allows the option holder to capture the dividend and avoid the price drop.

The optimal exercise strategy for American call options on dividend-paying stocks depends on both the magnitude of the dividend and the timing of its payment relative to the option's maturity. Typically, early exercise is optimal if:

- The dividend payment is large, and
- The ex-dividend date is close to the expiration of the option.

4.4.3 American Put Options and Dividends

Unlike American call options, the presence of dividends does not significantly affect the exercise strategy for American put options, since put options benefit from stock price declines, which are unaffected by dividend payments.

4.5 Black Approximation

The Black approximation is a widely used method to estimate the price of American call options on stocks that pay dividends. Since American call options allow for early exercise, particularly when dividends are involved, this method offers a practical approach to estimate the option price when dividends complicate the direct application of the Black-Scholes model.

4.5.1 Method of the Black Approximation

The Black approximation simplifies the problem by calculating the prices of two European call options:

- The first price is calculated for the option maturing at the actual expiration date T taking into account the dividend payment
- The second price is calculated for the option just before the last dividend payment, denoted t_n, where t_n is the ex-dividend date closest to the option's expiration date.

The logic behind this method is that if the dividend is significant and the ex-dividend date is close to the option's maturity, it may be advantageous for the holder to exercise the option early, just before the stock price drops due to the dividend.

Thus, the price of the American call option is the maximum of these two European option prices:

$$C_{\text{American}} = \max\left(C(T), C(t_n)\right)$$

where:

- $C(T)$ is the price of the European call option with expiration at T,
- $C(t_n)$ is the price of the European call option just before the last dividend payment at time t_n.

This approach approximates the price of the American call option by considering both the potential early exercise due to dividends and the time value of waiting until maturity.

4.5.2 Dividends and American Put Options

In contrast to American call options, dividends typically do not encourage the early exercise of American put options. Since put options benefit from a declining stock price, the dividend payment, which reduces the stock price, is advantageous for put holders.

4.5.3 Implementation

A Python implementation of the Black Approximation can be found here, in the script I computed the two prices of a European call at maturity T after the dividend has been paid and at t_n before the payment of the dividend, using B&S formula. The final price using Black approximation was then set equal to the maximum of the two.
The parameters used for this example are the following:

- $S_0 = 100$ Current stock price
- $K = 100$ Strike price
- $r = 0.05$ Risk-free interest rate
- $T = 1.0$ Time to maturity (years)
- $\sigma = 0.2$ Volatility
- $D_1 = 2.0$ Dividend payment
- $t_1 = 0.5$ Time of dividend payment (years)

The final results are shown in the table below.

Description	Value
Call before dividend	6.88
Call after dividend	9.24
Black Approximation Price	9.24

4.6 Roll-Geske-Whaley

The Roll-Geske-Whaley formula provides an exact solution for the price of an American call option on a stock that pays one discrete dividend. If several dividends are expected before maturity, the formula can still be used by adjusting S_0 as the stock price at the initial time minus the present value of all dividends except the last one. Thus, D_1 is the value of the final dividend and t_1 is the time of the final dividend payment. We will follow the formulas presented in Geske and Roll [1984].

4.6.1 The Roll-Geske-Whaley Formula

The price of the American call option is given by:

$$C = (S_0 - D_1 e^{-rt_1}) M\left(a_1, -b_1; -\sqrt{\frac{t_1}{T}}\right) + (S_0 - D_1 e^{-rt_1}) N(b_1) +$$
$$-K e^{-rT} M\left(a_2, -b_2; -\sqrt{\frac{t_1}{T}}\right) - (K - D_1) e^{-rT} N(b_2)$$

where:

$$a_1 = \frac{\ln\left(\frac{S_0 - D_1 e^{-rt_1}}{K}\right) + \left(r + \frac{\sigma^2}{2}\right) T}{\sigma\sqrt{T}}$$

$$a_2 = a_1 - \sigma\sqrt{T}$$

$$b_1 = \frac{\ln\left(\frac{S_0 - D_1 e^{-rt_1}}{S^*}\right) + \left(r + \frac{\sigma^2}{2}\right) t_1}{\sigma\sqrt{t_1}}$$

$$b_2 = b_1 - \sigma\sqrt{t_1}$$

Here, σ is the volatility of the stock (adjusted for dividends), and S^* is the critical stock price that satisfies the early exercise condition, which is obtained by solving:

$$c(S^*) = S^* + D_1 - K$$

where $c(S^*)$ is the Black-Scholes price for a European call option when the stock price is S^*, and the time to maturity is $T - t_1$.

$M(a, b; \rho)$ is the cumulative distribution function for a standardized bivariate normal distribution, which represents the probability that the first variable is less than a and the second is less than b, with correlation ρ. $N(x)$ is the cumulative distribution function for a standard normal distribution.

4.7 Drezner Approximation for $M(a, b; \rho)$

For the case where $a \leq 0$, $b \leq 0$, and $\rho \leq 0$, Drezner's approximation for $M(a, b; \rho)$ is given by:

$$M(a, b; \rho) = \frac{\sqrt{1-\rho^2}}{\pi} \sum_{i,j=1}^{4} A_i A_j f(B_i, B_j)$$

where:

$$f(x, y) = \exp\left[a'(2x - a') + b'(2y - b') + 2\rho(x - a')(y - b')\right]$$

with:

$$a' = \frac{a}{\sqrt{2(1-\rho^2)}}, \quad b' = \frac{b}{\sqrt{2(1-\rho^2)}}$$

The constants A_i and B_i are as follows:

$$A_1 = 0.3253030, \quad A_2 = 0.4211071, \quad A_3 = 0.1334425, \quad A_4 = 0.006374323$$

$$B_1 = 0.1337764, \quad B_2 = 0.6243247, \quad B_3 = 1.3425378, \quad B_4 = 2.2626645$$

If the product of a, b, and ρ is negative or zero, we have the following identities:

$$M(a, b; \rho) = N(a) - M(-a, -b; -\rho)$$

$$M(a, b; \rho) = N(b) - M(-a, -b; -\rho)$$

$$M(a, b; \rho) = N(a) + N(b) - 1 + M(-a, -b; \rho)$$

When the product of a, b, and ρ is positive, we use the identity:

$$M(a, b; \rho) = M(a, 0; \rho_1) + M(b, 0; \rho_2) - \delta$$

where:

$$\rho_1 = \frac{\rho a - b}{\sqrt{a^2 - 2\rho ab + b^2}} \operatorname{sgn}(a), \quad \rho_2 = \frac{\rho b - a}{\sqrt{a^2 - 2\rho ab + b^2}} \operatorname{sgn}(b)$$

$$\delta = \frac{1 - \operatorname{sgn}(a)\operatorname{sgn}(b)}{4}$$

The sign function $\mathrm{sgn}(x)$ is defined as:

$$\mathrm{sgn}(x) = \begin{cases} +1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

4.8 Implementation of Roll-Geske-Whaley

The implementation of the Roll-Geske-Whaley formula in Python code can be found here.

4.8.1 1. Libraries and Dependencies

The code begins by importing the necessary libraries:

- `numpy`: Used for numerical operations, such as logarithms and square roots.
- `scipy.stats.norm`: Provides functions for working with the standard normal cumulative distribution function (CDF).
- `scipy.stats.mvn`: Contains functions to compute the cumulative distribution for multivariate normal distributions, which is essential for calculating the bivariate normal CDF.
- `scipy.optimize.fsolve`: Used for solving nonlinear equations, specifically for solving for the critical stock price S^*.

4.8.2 2. Bivariate Normal CDF Calculation

The Roll-Geske-Whaley formula involves the bivariate normal cumulative distribution function (CDF), denoted by $M(a, b; \rho)$, where a and b are the limits for two correlated normal random variables, and ρ is the correlation between them.

The function `M_bivariate_normal(a, b, rho)` computes the CDF for two variables with correlation ρ using the `mvn.mvnun` function from the `scipy.stats` module. The input parameters are:

- `lower = [-inf, -inf]`: Specifies the lower bounds for the integration, which are negative infinity for both variables.
- `upper = [a, b]`: Specifies the upper bounds of integration, which are a and b.
- `mean = [0, 0]`: The mean of the two normal variables is assumed to be 0 (standard normal).
- `cov_matrix = [[1, rho], [rho, 1]]`: This is the covariance matrix where the diagonal elements represent the variance (1 for a standard normal), and the off-diagonal elements represent the correlation ρ between the two variables.

The result, `p`, is the value of the bivariate normal CDF.

4. Solving for S^*

The function `bs_condition(S_star)` represents the equation that needs to be solved for the critical stock price S^*:

$$c(S^*) = S^* + D_1 - K$$

where $c(S^*)$ is the Black-Scholes price for a European call option when the stock price is S^*, and the maturity is $T - t_1$. The function `fsolve` from `scipy.optimize` is used to numerically solve for S^*.

5. Roll-Geske-Whaley Formula

The main function `roll_geske_whaley(S0, K, r, T, sigma, D1, t1)` implements the Roll-Geske-Whaley formula to compute the American call option price. The key steps are:

- **Calculate a_1 and a_2:**

$$a_1 = \frac{\ln\left(\frac{S_0 - D_1 e^{-rt_1}}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$

$$a_2 = a_1 - \sigma\sqrt{T}$$

- **Solve for S^*** using the condition:

$$c(S^*) = S^* + D_1 - K$$

- **Calculate b_1 and b_2:**

$$b_1 = \frac{\ln\left(\frac{S_0 - D_1 e^{-rt_1}}{S^*}\right) + \left(r + \frac{\sigma^2}{2}\right)t_1}{\sigma\sqrt{t_1}}$$

$$b_2 = b_1 - \sigma\sqrt{t_1}$$

- **Compute the correlation coefficient ρ:**

$$\rho = -\sqrt{\frac{t_1}{T}}$$

- **Calculate the bivariate normal CDF values M_1 and M_2** using the function `M_bivariate_normal`:

$$M_1 = M(a_1, -b_1; \rho)$$

$$M_2 = M(a_2, -b_2; \rho)$$

- **Compute the final American call option price** using the Roll-Geske-Whaley formula:

$$C = (S_0 - D_1 e^{-rt_1})M_1 + (S_0 - D_1 e^{-rt_1})N(b_1) - Ke^{-rT}M_2 - (K - D_1)e^{-rT}N(b_2)$$

6. Parameters and Output

The parameters are set as follows, they are the same as before to compare the result obtained with the Black approximation:

- $S_0 = 100$ (current stock price),
- $K = 100$ (strike price),
- $r = 0.05$ (risk-free interest rate),
- $T = 1.0$ (time to maturity in years),
- $\sigma = 0.2$ (volatility),
- $D_1 = 2.0$ (dividend payment),
- $t_1 = 0.5$ (time of dividend payment in years).

The final result is similar to the one obtained with the Black approximation, this indicates that with a certain tolerance the approaches are equivalent.

Output:

Roll-Geske-Whaley American Call Option Price: 9.25

Part II

Exotic Options and Portfolio Management

Chapter 5

Introduction of Exotic Options

5.1 Exotic Options

Exotic options are a category of financial derivatives that have more complex structures compared to standard options (like European and American options). They are usually traded over-the-counter (OTC), meaning they are customized contracts between two parties rather than standardized and traded on exchanges. Many exotic options are linked to an underlying asset that provides a continuous dividend yield, denoted as q, which affects their pricing. For this Part about Exotic Options and Portfolio management we will closely follow the book of Hull [2021] and Capiński and Zastawniak [2011].

5.1.1 Bermuda Options

Bermuda options give the holder the right to exercise the option, but only on specific dates during the life of the option, rather than at any time (as in American options) or only at maturity (as in European options). These pre-defined exercise dates provide more flexibility than European options but less than American ones. Early exercise is only permitted on certain dates, typically spaced out at regular intervals. Investors use Bermuda options when they want the flexibility of potential early exercise, but with limited and specific opportunities to do so.

5.1.2 Warrants

Warrants are similar to long-term options that give the holder the right, but not the obligation, to buy or sell an underlying asset (usually company shares) at a specific price, known as the strike price, within a certain period. Unlike typical options, the strike price of a warrant can change during its lifetime. Additionally, early exercise is allowed only during a specific period, rather than throughout the entire life of the option. Warrants are often issued by the company itself, meaning exercising them can result in the issuance of new shares, potentially diluting the company's existing share capital. Warrants are often used as sweeteners in debt or equity deals, giving investors an incentive for long-term investment.

5.2 Forward Start Options

A forward start option is a type of option that begins at a future date. The option itself is agreed upon in advance, but it only comes into existence at the specified future start date. While the option is structured and priced at inception, the actual strike price is typically set when the option starts, often at the underlying asset's price at the start date (i.e., the at-the-money price at that future time). These options are commonly used for hedging purposes, particularly when an investor anticipates needing protection or exposure starting at a future date. For example, they are popular with corporations managing long-term compensation plans, where the need for an option begins only in future periods.

5.2.1 Pricing European Forward-Start Call Option

Consider a European forward-start call option on a non-dividend paying stock (i.e., the dividend yield $q = 0$). Let the following notation be defined:

- Initial time: $t = 0$
- Start time: T_1 (this is when the option becomes active, and the strike price is determined)
- Maturity time: T_2 (this is when the option expires)
- Stock price at time $t = 0$: S_0
- Stock price at time T_1: S_1
- Risk-free interest rate: r

Using the risk-neutral measure, the value of the forward-start European call option at time $t = 0$ is given by the discounted expectation of the option payoff under the risk-neutral measure $\hat{E}$. Since the option starts at time T_1, the present value of the option depends on the stock price S_1 at that time.

The value of the option at $t = 0$ is:

$$V(0) = e^{-rT_1} \hat{E} \left[C(S_1, T_1, T_2) \right]$$

where $C(S_1, T_1, T_2)$ is the value of the European call option at time T_1 with the underlying price S_1, strike price K, and maturity at T_2.

For a non-dividend paying stock (i.e., $q = 0$), under the risk-neutral measure, the expected value of the future stock price S_1 is known. Specifically, it is given by:

$$\hat{E}[S_1] = S_0 e^{rT_1}$$

Substituting this into the valuation equation simplifies the expression. Since the expected value of S_1 is known, the option value at time $t = 0$ simplifies to:

$$V(0) = Ce^{-rT_1}$$

where C is a constant representing the price of a European call option under these conditions using B&S formula. The discount factor e^{-rT_1} accounts for the time T_1 until the option becomes active.

5.3 Compound Options

Compound options are a type of exotic option where the underlying asset is itself an option. These options provide the holder with the right, but not the obligation, to purchase or sell another option at a specified future date. Compound options are highly leveraged and useful when the price of an option is uncertain, or the investor wants flexibility in deciding whether to acquire an option.

There are four main types of compound options:

- **Call on Call**: A call option on a call option
- **Call on Put**: A call option on a put option
- **Put on Call**: A put option on a call option
- **Put on Put**: A put option on a put option

Compound options are characterized by:

- **Two strike prices**:
  - K_1 (the strike price of the first option)
  - K_2 (the strike price of the second option)
- **Two exercise dates**:
  - T_1 (the maturity of the first option)
  - T_2 (the maturity of the second option), where $T_2 > T_1$

At the first maturity T_1, the holder can exercise the first option by paying the strike price K_1. If exercised, the holder obtains a second option with strike price K_2 and maturity T_2. The first option is exercised if the value of the second option at T_1 exceeds K_1, i.e.,

$$C(T_1) > K_1$$

where $C(T_1)$ is the value of the second option at time T_1.

5.3.1 Pricing a Call on Call Compound Option

Consider the case of a call on call compound option. The key parameters are:

- The strike price of the first option is K_1
- The maturity of the first option is T_1
- The strike price of the second option is K_2
- The maturity of the second option is T_2, where $T_2 > T_1$

At time T_1, if the value of the second call option exceeds K_1, the first option will be exercised. The value of the second option, denoted $C(T_1)$, depends on the expected value of the underlying asset at time T_2:

$$C(T_1) = e^{-r(T_2-T_1)}\hat{E}\left[\max(S_{T_2} - K_2, 0)\right]$$

If $C(T_1) > K_1$, the holder pays K_1 to acquire the second option, which can be exercised at T_2.

5.4 General models to price Compound Options

For all cases, the compound option has two strike prices and two maturities:

- K_1 is the strike price for the first option, expiring at T_1
- K_2 is the strike price of the second option, which expires at T_2

The first option is exercised at time T_1 if and only if the value of the second option at T_1 exceeds K_1. The pricing formulas below rely on the cumulative bivariate normal distribution M, since the valuation requires considering two sources of uncertainty: the price at T_1 and the price at T_2.

The value of a European call option on a call option is given by:

$$V_{\text{call on call}} = S_0 e^{-qT_2} M(a_1, b_1; \sqrt{\frac{T_1}{T_2}}) - K_2 e^{-rT_2} M(a_2, b_2; \sqrt{\frac{T_1}{T_2}}) - e^{-rT_1} K_1 N(a_2)$$

where the parameters are defined as:

$$a_1 = \frac{\ln\left(\frac{S_0}{S^*}\right) + \left(r - q + \frac{\sigma^2}{2}\right)T_1}{\sigma\sqrt{T_1}}, \quad a_2 = a_1 - \sigma\sqrt{T_1}$$

$$b_1 = \frac{\ln\left(\frac{S_0}{K_2}\right) + \left(r - q + \frac{\sigma^2}{2}\right)T_2}{\sigma\sqrt{T_2}}, \quad b_2 = b_1 - \sigma\sqrt{T_2}$$

here:

- $M(a, b; \rho)$ is the cumulative bivariate normal distribution with correlation ρ
- $N(x)$ is the cumulative univariate normal distribution

The value of a European put option on a call option is given by:

$$V_{\text{put on call}} = K_2 e^{-rT_2} M(-a_2, b_2; -\sqrt{\frac{T_1}{T_2}}) - S_0 e^{-qT_2} M(-a_1, b_1; -\sqrt{\frac{T_1}{T_2}}) + e^{-rT_1} K_1 N(-a_2)$$

The value of a European call option on a put option is:

$$V_{\text{call on put}} = K_2 e^{-rT_2} M(-a_2, -b_2; \sqrt{\frac{T_1}{T_2}}) - S_0 e^{-qT_2} M(-a_1, -b_1; \sqrt{\frac{T_1}{T_2}}) - e^{-rT_1} K_1 N(-a_2)$$

The value of a European put option on a put option is:

$$V_{\text{put on put}} = S_0 e^{-qT_2} M(a_1, -b_1; -\sqrt{\frac{T_1}{T_2}}) - K_2 e^{-rT_2} M(a_2, -b_2; -\sqrt{\frac{T_1}{T_2}}) + e^{-rT_1} K_1 N(a_2)$$

The parameters used in the formulas are defined as follows:

- S_0 is the initial price of the underlying asset
- K_1 is the strike price of the first option (compound option)
- K_2 is the strike price of the second option (underlying option)
- T_1 is the time to maturity of the first option
- T_2 is the time to maturity of the second option, with $T_2 > T_1$
- r is the risk-free interest rate
- q is the continuous dividend yield of the underlying asset
- σ is the volatility of the underlying asset
- S^* is the critical asset price at time T_1, where the value of the second option is exactly K_1
- $M(a, b; \rho)$ is the cumulative bivariate normal distribution with correlation ρ
- $N(x)$ is the cumulative univariate normal distribution

The first option will be exercised at time T_1 if the value of the second option at time T_1 exceeds K_1. This means that if the price of the asset at T_1 is greater than the critical level S^*, the first option will be exercised:

$$S_{T_1} > S^*$$

where S^* is the price at which the value of the second option equals K_1 at time T_1. Otherwise, the first option will not be exercised.

5.5 Chooser Options

A chooser option is a type of exotic option that grants the holder the right to decide, at a specific future time T_1, whether the option will become a call or a put. This flexibility allows the holder to defer the decision until T_1, depending on how the underlying asset's price evolves over time.

The parameters used for this analysis are the following:

- S_0 is the initial price of the underlying asset
- S_1 is the price of the underlying asset at time T_1
- K is the strike price of the option
- T_1 is the time at which the holder decides if the option will be a call or a put
- T_2 is the maturity of the option (after the choice is made)
- r is the risk-free interest rate
- q is the continuous dividend yield
- $C(T_1)$ is the value of the European call at time T_1
- $P(T_1)$ is the value of the European put at time T_1

If the option is bought at $t = 0$, the holder makes the decision at time T_1. The value at T_1 of the chooser option is:

$$V_{\text{chooser}}(T_1) = \max(C(T_1), P(T_1))$$

where:

- $C(T_1)$ is the value of the option if it were a European call at time T_1
- $P(T_1)$ is the value of the option if it were a European put at time T_1

To value the chooser option, we can exploit put-call parity. The put-call parity relationship for European options with the same strike price K and maturity T_2 is:

$$C(T_1) - P(T_1) = S_1 e^{-q(T_2 - T_1)} - K e^{-r(T_2 - T_1)}$$

where:

- S_1 is the price of the underlying asset at time T_1
- K is the strike price
- T_2 is the maturity of both the call and the put
- r is the risk-free interest rate

- q is the continuous dividend yield of the underlying asset

Using put-call parity, we can express the value of the chooser option as follows:

$$V_{\text{chooser}}(T_1) = \max\left(C(T_1), C(T_1) + Ke^{-r(T_2-T_1)} - S_1 e^{-q(T_2-T_1)}\right)$$

Simplifying this expression:

$$V_{\text{chooser}}(T_1) = C(T_1) + \max\left(0, Ke^{-(r-q)(T_2-T_1)} - S_1\right) e^{-q(T_2-T_1)}$$

From the final equation, we can interpret the chooser option as a portfolio consisting of:

- A European call option with strike K and maturity T_2
- A quantity of $e^{-q(T_2-T_1)}$ European put options with strike price $Ke^{-(r-q)(T_2-T_1)}$ and maturity at T_1

At time T_1 the chooser option's holder selects whether to exercise the option as a call or a put, at T_2 if the call is chosen, the payoff will be $\max(S_{T_2} - K, 0)$; if the put is selected, the payoff will be $\max(K - S_{T_2}, 0)$. By exploiting put-call parity, we rewrite the chooser option's value in terms of a call option and an embedded put option.

Thus, the chooser option can be viewed as a strategy combining a long position in a European call and a certain quantity of European puts, where the strike and maturity of the put are adjusted based on interest rate and dividend yield differences.

5.6 Implementation to price Chooser Options

In this section, we analyze a Python code that can price Chooser Options. The Black-Scholes formula for a European call option is given by:

$$C = Se^{-qT}\Phi(d_1) - Ke^{-rT}\Phi(d_2)$$

where:

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r - q + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

Here:

- S is the current asset price
- K is the strike price
- T is the time to maturity
- r is the risk-free interest rate

- q is the dividend yield
- σ is the volatility
- Φ is the cumulative distribution function of the standard normal distribution

The Black-Scholes formula for a European put option is:

$$P = Ke^{-rT}\Phi(-d_2) - Se^{-qT}\Phi(-d_1)$$

5.6.1 Pricing of Chooser Options in Python

Now, a chooser option can be priced as follows:

$$\text{Chooser Price} = C + e^{-q(T2-T1)} \cdot P$$

where:

- C is the price of the European Call
- $T1$ is the time to maturity of the put option
- $T2$ is the time to maturity of the call option
- P is the price of the put option with strike $K \cdot e^{-(r-q)(T2-T1)}$ and maturity $T1$

The put option price P is calculated using:

$$P = K_{\text{put}}e^{-rT1}\Phi(-d_2) - Se^{-qT1}\Phi(-d_1)$$

where $K_{\text{put}} = K \cdot e^{-(r-q)(T2-T1)}$.

5.6.2 Code Implementation

The Python implementation involves the following steps:

1. Define functions to calculate Black-Scholes prices for call and put options.
2. Use these functions to calculate the price of the chooser option.

Here are the parameters used in the pricing of the chooser option:

- $S_0 = 100$: Initial asset price
- $K = 105$: Strike price of the chooser option
- $T_1 = 0.5$ years : Time to maturity of the put option
- $T_2 = 1.0$ years : Time to maturity of the call option
- $r = 0.03$: Risk-free interest rate (3%)

- $q = 0.02$: Dividend yield (2%)
- $\sigma = 0.25$: Volatility (25%)
- $c = 1.0$: Constant term in the chooser option price

The price of the chooser option, given the parameters above, is:

$$\text{Price} = 10.15$$

Chapter 6

Pricing Exotic Options

6.1 Barrier Options

Barrier options are a class of exotic options whose payoff at maturity depends not only on the terminal value of the underlying asset but also on whether the underlying asset's price has reached or crossed a certain threshold level during the life of the option. This threshold is called the *barrier level*, and the behavior of the option changes depending on whether the barrier is reached.

There are two main types of barrier options:

- Knock-Out Options: These options become worthless if the price of the underlying asset reaches or exceeds (or falls below, depending on the option type) a specified barrier level

- Knock-In Options: These options only come into existence and acquire value if the price of the underlying asset touches or crosses the barrier level

6.1.1 Down-and-out Call

A **down-and-out call** is a type of knock-out option. It behaves like a standard European call option but will become worthless if the price of the underlying asset falls to or below a specified barrier level H at any point before the option's maturity. The barrier H is set below the initial asset price S_0.

The key features of a down-and-out call are:

- The option holder has the right to buy the underlying asset at the strike price K at maturity if the asset price remains above the barrier level H

- If the underlying asset price reaches or falls below the barrier level H before maturity, the option is immediately knocked out and becomes worthless

6.1.2 Down-and-in Call

A **down-and-in call** is a type of knock-in option. It only becomes active and acquires value if the price of the underlying asset touches or falls below a specified barrier level H during the life of the option. If the barrier is reached, the option becomes a standard call option with the following features:

- If the asset price reaches the barrier H, the option holder has the right to buy the underlying asset at the strike price K at maturity
- If the underlying asset price never reaches or falls below the barrier H, the option expires worthless

6.1.3 Down-and-In Call Option

The formula for the down-and-in call option, denoted c_{di}, is given by:

$$c_{di} = S_0 e^{-qT} \left(\frac{H}{S_0}\right)^{2\lambda} N(y) - Ke^{-rT} \left(\frac{H}{S_0}\right)^{2\lambda-2} N(y - \sigma\sqrt{T})$$

where:

$$\lambda = \frac{r - q + \frac{1}{2}\sigma^2}{\sigma^2}$$

and

$$y = \frac{\ln\left(\frac{H^2}{S_0 K}\right)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}$$

Here, $N(x)$ is the cumulative distribution function (CDF) of the standard normal distribution.

6.1.4 Down-and-Out Call Option

The price of the down-and-out call option, denoted as c_{do}, is given by:

$$c_{do} = c - c_{di}$$

where c is the price of a standard European call option, calculated using the Black-Scholes formula.

For cases where the barrier level $H \geq K$, the down-and-out call price is calculated using the following formula:

$$c_{do} = S_0 N(x_1) e^{-qT} - Ke^{-rT} N(x_1 - \sigma\sqrt{T}) - S_0 e^{-qT} \left(\frac{H}{S_0}\right)^{2\lambda} N(y_1) +$$

$$+ Ke^{-rT} \left(\frac{H}{S_0}\right)^{2\lambda-2} N(y_1 - \sigma\sqrt{T})$$

where:

$$x_1 = \frac{\ln(S_0/H)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}$$
$$y_1 = \frac{\ln(H/S_0)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}$$

6.1.5 Up-and-Out Call Option

An **up-and-out call** option becomes worthless if the asset price reaches a barrier level H, which is higher than the current price of the underlying asset. If the barrier is not reached, the option behaves like a standard call option.

6.1.6 Case 1: $H \leq K$

When the barrier level H is less than or equal to the strike price K:

$$c_{uo} = 0 \quad \text{(up-and-out call is worthless)}$$
$$c_{ui} = c \quad \text{(up-and-in call is equivalent to a regular European call)}$$

6.1.7 Case 2: $H > K$

When the barrier level H is greater than the strike price K, the up-and-in and up-and-out call prices are calculated as follows:

Up-and-In Call Price

The formula for the up-and-in call price c_{ui} is:

$$c_{ui} = S_0 N(x_1)e^{-qT} - Ke^{-rT}N(x_1 - \sigma\sqrt{T}) - S_0 e^{-qT}\left(\frac{H}{S_0}\right)^{2\lambda}\left[N(-y_1) - N(-y_1 + \sigma\sqrt{T})\right] +$$
$$+Ke^{-rT}\left(\frac{H}{S_0}\right)^{2\lambda-2}\left[N(-y_1 + \sigma\sqrt{T}) - N(-y_1)\right]$$

Up-and-Out Call Price

The up-and-out call price c_{uo} is computed by subtracting the up-and-in call price from the price of a regular European call c:

$$c_{uo} = c - c_{ui}$$

6.1.8 Barrier Put Options

For put options, the two main categories are up-and-out and up-and-in put options:

- An **up-and-out put** ceases to exist if the asset price moves above the barrier H
- An **up-and-in put** comes into existence only if the asset price reaches the barrier H

6.1.9 Case 1: $H \geq K$

When the barrier H is greater than or equal to the strike price K, the up-and-in put price p_{ui} is determined by subtracting the up-and-out put price p_{uo} from the price of a regular put option p:

$$p_{uo} = p - p_{ui}$$

The up-and-in put price is calculated by:

$$p_{ui} = -S_0 e^{-qT} \left(\frac{H}{S_0}\right)^{2\lambda} N(-y) + Ke^{-rT} \left(\frac{H}{S_0}\right)^{2\lambda-2} N(-y + \sigma\sqrt{T})$$

6.1.10 Case 2: $H < K$

When the barrier H is less than the strike price K, the up-and-out put price is given by:

$$p_{uo} = -S_0 N(-x_1) e^{-qT} + Ke^{-rT} N(-x_1 + \sigma\sqrt{T}) + S_0 e^{-qT} \left(\frac{H}{S_0}\right)^{2\lambda} N(-y_1) +$$

$$-Ke^{-rT} \left(\frac{H}{S_0}\right)^{2\lambda-2} N(-y_1 + \sigma\sqrt{T})$$

The up-and-in put price is then:

$$p_{ui} = p - p_{uo}$$

6.1.11 Down-and-Out and Down-and-In Put Options

The pricing of down-and-out and down-and-in put options follows a similar logic, with the key difference being that the barrier H is below the current asset price.

The price of a down-and-out put is given by:

$$p_{do} = -S_0 N(-x_1) e^{-qT} + Ke^{-rT} N(-x_1 + \sigma\sqrt{T}) + S_0 e^{-qT} \left(\frac{H}{S_0}\right)^{2\lambda} \left[N(y_1) - N(y_1 - \sigma\sqrt{T})\right] +$$

$$-Ke^{-rT} \left(\frac{H}{S_0}\right)^{2\lambda-2} \left[N(y_1 - \sigma\sqrt{T}) - N(y_1)\right]$$

The down-and-in put price p_{di} is computed by subtracting the down-and-out put price from the regular put price p:

$$p_{di} = p - p_{do}$$

6.1.12 Barrier Types: Summary

There are several variations of barrier options depending on whether the barrier is set above or below the current price of the asset.
The primary types are:

- Up-and-Out Option: A knock-out option that becomes worthless if the asset price rises above the barrier level

- Up-and-In Option: A knock-in option that becomes active if the asset price rises above the barrier level

- Down-and-Out Option: A knock-out option that becomes worthless if the asset price falls below the barrier level

- Down-and-In Option: A knock-in option that becomes active if the asset price falls below the barrier level

Barrier options can be advantageous for investors seeking to reduce premiums, as they are typically cheaper than standard options. However, the complexity of their payoff structure, which depends on the asset's path, adds an additional layer of risk.

Barrier options offer flexible risk management tools but also come with the risk of becoming worthless if the underlying asset touches or breaches the barrier level. They are often used by traders who anticipate significant price movements but want to limit the cost of option premiums.

6.2 Implementation to Price Barrier Options

In this section we will analyze a Python code to price Barrier Options, we will focus on down-and-in-call and down-and-out-call.
The code implements the following functions for pricing barrier options:

- `down_and_in_call`: This function calculates the price of a down-and-in call option using the derived formula. It computes:

  - λ: A parameter that depends on the risk-free rate r, the dividend yield q, and the volatility σ.
  - y: A term involving the logarithm of the barrier and strike prices.
  - The cumulative normal distribution values $N(y)$ and $N(y - \sigma\sqrt{T})$.

- `european_call`: This function calculates the price of a standard European call option using the Black-Scholes formula. It computes the terms d_1 and d_2, which are used to evaluate the call price based on the risk-neutral probability framework.

- `down_and_out_call`: This function calculates the price of a down-and-out call option. It does so by subtracting the price of the down-and-in call from the price of a standard European call option:

$$c_{do} = c - c_{di}$$

  where c is the price of the European call and c_{di} is the price of the down-and-in call.

6.2.1 Example in Python

The following parameters are used to calculate the option prices:

- $S_0 = 100$: The current asset price
- $H = 90$: The barrier level
- $K = 95$: The strike price
- $T = 1$: Time to maturity (in years)
- $r = 0.05$: The risk-free interest rate (5%)
- $q = 0.02$: The dividend yield (2%)
- $\sigma = 0.2$: The volatility (20%)

Using these parameters, the code computes the down-and-in call price and the down-and-out call price.
The code produces the following results for barrier option pricing:

- **Down-and-In Call Price**: 2.50
- **Down-and-Out Call Price**: 9.44

6.3 Binary Options

Binary options are financial instruments that allow traders to speculate on the price movement of an underlying asset. Unlike traditional options, where the payoff depends on the magnitude of the price movement, binary options offer only two possible outcomes: the trader either receives a fixed payoff or nothing at all.

There are two main types of binary options:

1. **Cash-or-nothing options**
2. **Asset-or-nothing options**

Each type can further be classified into call options and put options, depending on whether the trader expects the underlying asset price to go up or down.

6.3.1 Cash-or-Nothing Options

A **cash-or-nothing call option** pays a fixed amount, denoted as Q, if the price of the underlying asset at maturity, S_T, is greater than the strike price, K. Otherwise, the payoff is zero.

In a risk-neutral framework, the probability that the asset price ends up above the strike price at maturity is given by $N(d_2)$, where $N(d_2)$ represents the cumulative distribution function (CDF) of the standard normal distribution.

The value of a cash-or-nothing call option at time t is:

$$\text{Value of cash-or-nothing call} = Qe^{-r(T-t)}N(d_2)$$

where:

- Q is the fixed payoff if the option finishes in-the-money (i.e., $S_T > K$)
- r is the risk-free interest rate
- $T - t$ is the time to maturity
- $N(d_2)$ is the probability that the asset price will exceed the strike price at time T

A **cash-or-nothing put option** pays the fixed amount Q if the asset price at maturity is less than the strike price. The value of a cash-or-nothing put option is given by:

$$\text{Value of cash-or-nothing put} = Qe^{-r(T-t)}N(-d_2)$$

where $N(-d_2)$ represents the probability that the asset price will be below the strike price at maturity.

6.3.2 Asset-or-Nothing Options

An **asset-or-nothing call option** pays the price of the underlying asset, S_T, if the asset price is greater than the strike price at maturity. If the asset price is below the strike price, the payoff is zero. The value of an asset-or-nothing call option is:

$$\text{Value of asset-or-nothing call} = S_0e^{-q(T-t)}N(d_1)$$

where:

- S_0 is the current asset price
- q is the dividend yield of the underlying asset
- $N(d_1)$ is the risk-neutral probability that the option finishes in-the-money (i.e., $S_T > K$)

Similarly, an **asset-or-nothing put option** pays the asset price if the price at maturity is less than the strike price. The value of an asset-or-nothing put option is:

$$\text{Value of asset-or-nothing put} = S_0e^{-q(T-t)}N(-d_1)$$

6.3.3 The d_1 and d_2 Parameters

The parameters d_1 and d_2 used above are the usual parameters of the B&S formula. These parameters are defined as:

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - q + \frac{\sigma^2}{2}\right)(T - t)}{\sigma\sqrt{T - t}}$$

$$d_2 = d_1 - \sigma\sqrt{T - t}$$

where:

- S_0 is the current asset price
- K is the strike price
- r is the risk-free rate
- q is the dividend yield
- σ is the volatility of the underlying asset
- $T - t$ is the time to maturity

6.3.4 Relationship with European Options

Interestingly, binary options can be linked to traditional European options. A European call option can be considered as a combination of a long position in an asset-or-nothing call and a short position in a cash-or-nothing call. The payoff structure of a European call can be reconstructed as:

$$\text{European call} = (\text{Asset-or-nothing call}) - (\text{Cash-or-nothing call with payoff } K)$$

Similarly, a European put option can be thought of as a combination of a cash-or-nothing put and an asset-or-nothing put:

$$\text{European put} = (\text{Cash-or-nothing put}) - (\text{Asset-or-nothing put with payoff } K)$$

6.3.5 Summary of Pricing Formulas

For reference, here are the key pricing formulas for binary options:

- **Cash-or-nothing call option:**

$$\text{Call} = Qe^{-r(T-t)}N(d_2)$$

- **Cash-or-nothing put option:**

$$\text{Put} = Qe^{-r(T-t)}N(-d_2)$$

- **Asset-or-nothing call option:**

$$\text{Call} = S_0 e^{-q(T-t)} N(d_1)$$

- **Asset-or-nothing put option:**

$$\text{Put} = S_0 e^{-q(T-t)} N(-d_1)$$

Binary options are financial instruments that provide traders with an all-or-nothing payout structure. Their pricing can be derived using risk-neutral probabilities, with both cash-or-nothing and asset-or-nothing variations available. Although binary options are relatively simple in their payout structure, they are closely linked to traditional European options, and their valuation shares similar components, such as the parameters d_1 and d_2.

6.4 Lookback Options

Lookback options are path-dependent options whose payoff depends on the maximum or minimum asset price reached during the life of the option. These options allow the holder to "look back" over the entire life of the option to determine the payoff based on the most favorable asset price. For more details see M. Barry Goldman and Gatto [1979].

6.4.1 Payoff Structure

For a European-style **lookback call option**, the payoff is the difference between the final asset price and the minimum asset price achieved during the life of the option. That is, the option allows the holder to buy at the lowest price the asset reached during the option's life and sell at the final price.

For a European-style **lookback put option**, the payoff is the difference between the maximum asset price achieved during the life of the option and the final price. This allows the holder to sell at the highest price the asset reached and buy at the final price.

6.4.2 Valuation of European Lookback Call Option

The valuation of a European lookback call option at time zero is given by the formula:

$$S_0 e^{-qT} N(a_1) - S_{\min} e^{-rT} N(-a_2) - S_{\min} e^{-rT} \left(\frac{\sigma^2}{2(r-q)} \right) N(-a_3) + S_{\min} e^{-rT} e^{Y_1} N(-a_3)$$

where:

- S_0 is the current asset price
- $S_{\min}$ is the minimum asset price achieved to date (i.e., the lowest price the asset has reached so far)
- r is the risk-free interest rate

- q is the dividend yield
- σ is the asset price volatility
- T is the time to maturity
- $N(x)$ represents the cumulative distribution function (CDF) of the standard normal distribution

6.4.3 Definitions of a_1, a_2, a_3, and Y_1

The terms a_1, a_2, a_3, and Y_1 are defined as follows:

$$a_1 = \frac{\ln\left(\frac{S_0}{S_{\min}}\right) + \left(r - q + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$

$$a_2 = a_1 - \sigma\sqrt{T}$$

$$a_3 = \frac{\ln\left(\frac{S_0}{S_{\min}}\right) + \left(r - q + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$

$$Y_1 = \frac{-2(r - q - \frac{\sigma^2}{2})\ln\left(\frac{S_0}{S_{\min}}\right)}{\sigma^2}$$

6.4.4 Explanation of Parameters

- **Asset Price** (S_0): The current price of the underlying asset
- **Minimum Asset Price** ($S_{\min}$): The lowest price reached by the asset over the life of the option up until the current time
- **Risk-Free Rate** (r): The continuously compounded risk-free interest rate
- **Dividend Yield** (q): The continuously compounded rate at which dividends are paid by the underlying asset
- **Volatility** (σ): The annualized volatility of the underlying asset's returns
- **Time to Maturity** (T): The time remaining until the option expires
- **Cumulative Normal Distribution** ($N(x)$): Represents the probability that a standard normally distributed variable is less than x

The above formula applies to European-style lookback call options. For lookback put options, the structure is similar, but with the roles of the maximum and minimum asset prices reversed.

6.4.5 European lookback put options

European lookback put options provide the holder with the payoff depending on the maximum asset price, $S_{\max}$, achieved during the option's life. The payoff at maturity is the difference between the maximum asset price and the asset price at expiry. The payoff for the European lookback put option is:

$$\max(S_{\max} - S_T, 0)$$

6.4.6 Valuation of European Lookback Put Option

The valuation formula for a European lookback put option at time zero is given by:

$$S_{\max} e^{-rT} N(b_1) - S_0 e^{-qT} \left(N(-b_2) + \frac{\sigma^2}{2(r-q)} e^{Y_2} N(-b_3) \right)$$

where:

$$b_1 = \frac{\ln\left(\frac{S_{\max}}{S_0}\right) + (r - q + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$b_2 = b_1 - \sigma\sqrt{T}$$

$$b_3 = \frac{\ln\left(\frac{S_{\max}}{S_0}\right) + (r - q - \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$Y_2 = \frac{-2(r - q - \sigma^2/2)\ln\left(\frac{S_{\max}}{S_0}\right)}{\sigma^2}$$

In the above equations:

- $S_{\max}$ is the maximum asset price achieved during the life of the option (if the lookback has just been originated, $S_{\max} = S_0$)
- S_0 is the current price of the underlying asset
- r is the risk-free interest rate
- q is the dividend yield
- σ is the volatility of the asset
- T is the time to maturity
- $N(\cdot)$ is the cumulative distribution function of the standard normal distribution

Lookback options offer significant flexibility to the holder, allowing them to base their payoff on the maximum or minimum asset price observed during the life of the option. This feature makes lookback options more expensive than standard European or American options. The continuous nature of lookback options means the asset price is assumed to be monitored at all times; however, in practice, discrete monitoring may be used to approximate continuous observation.

Chapter 7

Asian Options

An **Asian option** is a type of financial derivative where the payoff depends on the average price of the underlying asset over a specific period, rather than its price at a single point in time, such as at maturity.

7.0.1 Key Features

- **Average Price**: The payoff is determined by the average price of the asset over a defined period.
- **Two Main Types of Asian Options:**
  - **Average Price Options**: The payoff depends on the difference between the average price of the underlying asset and the strike price
  - **Average Strike Options**: The payoff depends on the difference between the terminal price of the underlying asset and the average price over the life of the option

7.0.2 Formulas for Pricing Asian Options

For **European-style Asian options** (where the option can only be exercised at maturity), the payoff depends on whether it is an average price or average strike option.

7.0.3 Asian Option Payoff

$$\text{Payoff} = \max\left(\frac{1}{N}\sum_{i=1}^{N} S_i - K, 0\right)$$

where:

- S_i is the price of the underlying asset at time t_i.
- K is the strike price.
- N is the number of time intervals over which the average is calculated.

7.0.4 Stock Price Simulation under Geometric Brownian Motion (GBM)

A stock price under Geometric Brownian Motion (GBM) follows the stochastic differential equation:

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

Where:

- μ is the drift rate (expected return)
- σ is the volatility of the underlying asset
- dW_t is the Wiener process (random noise)

7.0.5 Monte Carlo Simulation for Pricing an Asian Option

We can price the Asian option using Monte Carlo simulation by following these steps:

1. Simulate multiple stock paths using the Geometric Brownian Motion (GBM) model
2. Compute the average price of the underlying asset for each simulated path
3. Calculate the payoff of the Asian option based on the average price
4. Discount the expected payoff to the present value using the risk-free rate

The Python code provided simulates this process and calculates the price of the Asian option using real parameters such as:

- Initial stock price $S_0 = 100$
- Strike price $K = 105$
- Time to maturity $T = 1$ year
- Risk-free rate $r = 0.05$ (5%)
- Volatility $\sigma = 0.2$ (20%)
- Number of time steps $N = 252$ (daily time steps for 1 year)
- Number of simulations $M = 10{,}000$

7.1 Implementation to Price Asian Options

In this Python code we analyze a way to price Asian Options using Monte Carlo simulation for the paths of a stock. In this method, multiple paths of the stock price are simulated using the Geometric Brownian Motion (GBM) model. The payoff of the option is based on the average price of the stock over the life of the option, which is then discounted to the present value to determine the option price.

7.1.1 Stock Price Simulation

The stock price follows a Geometric Brownian Motion, which models the price evolution using a stochastic differential equation. The stock prices are simulated for a large number of paths, where each path represents the possible price evolution of the underlying asset over time.

7.1.2 Average Price Calculation

Once the stock price paths are simulated, the code computes the average stock price for each path. This average price is critical because, for an Asian option, the payoff depends on the average price of the asset, rather than just the final price at maturity.

7.1.3 Payoff Calculation

The option payoff is computed based on the difference between the average stock price and the strike price for a call option. For each simulated path, if the average price exceeds the strike price, the payoff is positive; otherwise, it is zero.

7.1.4 Discounting to Present Value

After calculating the payoff for all simulated paths, the expected payoff is discounted to the present value using the risk-free interest rate. This discounting accounts for the time value of money, and the average discounted payoff across all simulations gives an estimate of the option price.

In summary, the code simulates many possible stock price paths, calculates the average price for each path, computes the option payoff, and then discounts the payoff to estimate the price of the Asian option. The Monte Carlo simulation approach allows for a flexible and numerical way to price options when closed-form solutions are not available.

7.1.5 Results

The parameters used for this pricing are as follows:

- $S_0 = 100$: Initial stock price
- $K = 105$: Strike price
- $T = 1.0$: Time to maturity (in years)

- $r = 0.05$: Risk-free interest rate
- $\sigma = 0.2$: Volatility of the underlying asset
- $N = 252$: Number of time steps (daily observations over one year)
- $M = 10{,}000$: Number of Monte Carlo simulations

After running the Monte Carlo simulation to estimate the value of the Asian option, the resulting option price is approximately **3.41**.

The price reflects the discounted expected payoff of the Asian option, taking into account the averaging feature and the randomness introduced by the daily fluctuations in asset prices.

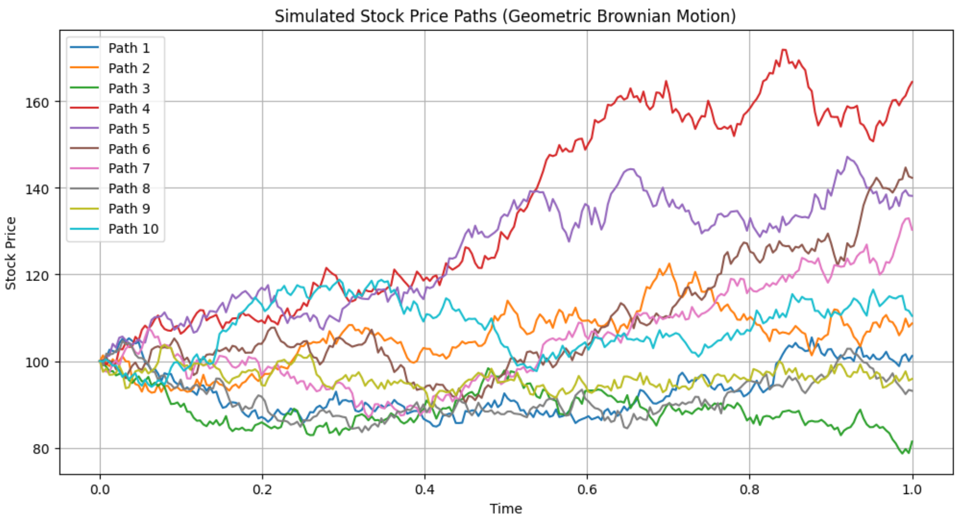

Figure 7.1. 10 of the 10000 Monte Carlo simulations for paths of the stock used to price the Asian Option

Chapter 8

Portfolio Management

8.1 Measuring Risk

In financial analysis, risk is often quantified using statistical measures. Let K be a random variable representing the return on an investment over a given period. The expected return, $E(K)$, and the dispersion of potential outcomes around this expected return, measured by the variance $\text{Var}(K)$, are key components in assessing the risk associated with an investment. In order to introduce some concepts of Portfolio management and risk measuring, we will follow the book of Capiński and Zastawniak [2011]. For this Chapter we will follow the book of Capiński and Zastawniak [2011] also to derive some notable formulas.

8.1.1 Variance

The **variance**, $\text{Var}(K)$, measures the spread or dispersion of the returns around the expected value. It quantifies the degree to which the actual returns deviate from the expected return, and is calculated as:

$$\text{Var}(K) = E\left[(K - E(K))^2\right]$$

This gives a squared measure of risk, where larger values of $\text{Var}(K)$ indicate higher risk, implying that the returns can vary significantly from their expected value.

8.1.2 Standard Deviation

While the variance provides a useful measure of risk, it is expressed in squared units of return. To obtain a measure of risk that is in the same units as the return itself, we take the square root of the variance, known as the **standard deviation**, denoted σ_K. The standard deviation is a widely used risk measure and is computed as:

$$\sigma_K = \sqrt{\text{Var}(K)} = \sqrt{E\left[(K - E(K))^2\right]}$$

The standard deviation is particularly useful because, unlike the variance, it is expressed in the same unit of measure as the return K, allowing for a more intuitive interpretation of risk. A higher standard deviation indicates a greater level of volatility or uncertainty in the potential returns of the investment.
In summary :

- The **expected return** $E(K)$ provides a measure of the average or typical outcome
- The **variance** $\text{Var}(K)$ measures the dispersion of returns around the expected value, giving an indication of the risk involved in the investment
- The **standard deviation** σ_K, being the square root of the variance, offers a more intuitive and easily interpretable measure of risk, expressed in the same units as the return K

Understanding these concepts helps investors evaluate both the potential profitability and the uncertainty associated with their investments.

8.2 Portfolio with two assets

8.2.1 Portfolio Return and Risk

In financial analysis, when constructing a portfolio consisting of two securities, we are often interested in both the portfolio's expected return and the associated risk (variance). Let the portfolio consist of x_1 shares of security 1 and x_2 shares of security 2. The return and risk of the portfolio can be derived as follows.

8.2.2 Initial and Final Values of the Portfolio

The **initial value** of the portfolio at time $t = 0$ is given by the sum of the values of each security:

$$V(0) = x_1 S_1(0) + x_2 S_2(0)$$

where $S_1(0)$ and $S_2(0)$ are the initial prices of securities 1 and 2, respectively.

At time $t = 1$, the **final value** of the portfolio is:

$$V(1) = x_1 S_1(0)(1 + K_1) + x_2 S_2(0)(1 + K_2)$$

where K_1 and K_2 represent the returns on securities 1 and 2, respectively, over the period.

We can factor the initial portfolio value as:

$$V(1) = V(0)\left(w_1(1 + K_1) + w_2(1 + K_2)\right)$$

where w_1 and w_2 are the proportions of the total initial value invested in securities 1 and 2, defined as:

$$w_1 = \frac{x_1 S_1(0)}{V(0)}, \quad w_2 = \frac{x_2 S_2(0)}{V(0)}$$

8.2.3 Portfolio Return

The **return on the portfolio**, K_V, is defined as the relative change in the portfolio value from time $t = 0$ to $t = 1$:

$$K_V = \frac{V(1) - V(0)}{V(0)} = w_1 K_1 + w_2 K_2$$

Thus, the return on the portfolio is a weighted average of the returns on the individual securities, where the weights are determined by the proportion of the portfolio invested in each security.

8.2.4 Expected Return of the Portfolio

The **expected return** of the portfolio, $E(K_V)$, is the weighted average of the expected returns of the individual securities:

$$E(K_V) = w_1 E(K_1) + w_2 E(K_2)$$

where $E(K_1)$ and $E(K_2)$ are the expected returns of securities 1 and 2, respectively.

8.2.5 Variance of the Portfolio

The **variance of the portfolio**, $\text{Var}(K_V)$, is a measure of the risk associated with the portfolio. To derive the variance, we substitute the expression for K_V and expand:

$$\text{Var}(K_V) = E(K_V^2) - E(K_V)^2$$

Since $K_V = w_1 K_1 + w_2 K_2$, we can expand K_V^2:

$$K_V^2 = w_1^2 K_1^2 + w_2^2 K_2^2 + 2 w_1 w_2 K_1 K_2$$

Thus, the variance becomes:

$$\text{Var}(K_V) = w_1^2 \left(E(K_1^2) - E(K_1)^2 \right) + w_2^2 \left(E(K_2^2) - E(K_2)^2 \right)$$

$$+ 2 w_1 w_2 \left(E(K_1 K_2) - E(K_1) E(K_2) \right)$$

Finally, recognizing that the terms $E(K_1^2) - E(K_1)^2$ and $E(K_2^2) - E(K_2)^2$ are simply the variances of K_1 and K_2, and that $E(K_1 K_2) - E(K_1)E(K_2)$ is the covariance of K_1 and K_2, we can express the portfolio variance as:

$$\text{Var}(K_V) = w_1^2 \text{Var}(K_1) + w_2^2 \text{Var}(K_2) + 2 w_1 w_2 \text{Cov}(K_1, K_2)$$

This shows that the variance of the portfolio depends on both the variances of the individual securities and the covariance between them.

8.3 Expectation, Variance, and Correlation Coefficient

To simplify notation, we introduce symbols to represent the **expectation** and **variance** of the portfolio's return, as well as the return of its individual components. This will allow us to express the portfolio's expected return and variance more concisely.

Let K_V, K_1, and K_2 represent the returns on the portfolio and the two individual securities, respectively.

8.3.1 Expectations and Standard Deviations

Define the following:

$$\mu_V = \mathbb{E}(K_V), \quad \sigma_V = \sqrt{\text{Var}(K_V)}$$

where μ_V is the **expected return of the portfolio**, and σ_V is the **standard deviation** (risk) of the portfolio's return.

Similarly, for the individual securities:

$$\mu_1 = \mathbb{E}(K_1), \quad \sigma_1 = \sqrt{\text{Var}(K_1)},$$

$$\mu_2 = \mathbb{E}(K_2), \quad \sigma_2 = \sqrt{\text{Var}(K_2)},$$

where μ_1 and μ_2 represent the **expected returns** of securities 1 and 2, respectively, and σ_1 and σ_2 are their **standard deviations**.

8.3.2 Correlation Coefficient

The **correlation coefficient** ρ_{12} measures the linear relationship between the returns on securities 1 and 2. It is defined as:

$$\rho_{12} = \frac{\text{Cov}(K_1, K_2)}{\sigma_1 \sigma_2}$$

The correlation coefficient ranges between -1 and 1, where:

- $\rho_{12} = 1$ indicates a perfect positive correlation
- $\rho_{12} = -1$ indicates a perfect negative correlation
- $\rho_{12} = 0$ indicates no linear relationship between the returns

8.3.3 Expected Return of the Portfolio

Using the notation introduced above, we can express the **expected return** of the portfolio as the weighted average of the expected returns of the two securities:

$$\mu_V = w_1 \mu_1 + w_2 \mu_2$$

where w_1 and w_2 are the proportions of the total value invested in securities 1 and 2, respectively.

8.3.4 Variance of the Portfolio

The **variance of the portfolio** can also be expressed in a more compact form using the correlation coefficient. The formula for the variance becomes:

$$\sigma_V^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2\rho_{12}\sigma_1\sigma_2$$

where σ_1^2 and σ_2^2 are the variances of the returns on securities 1 and 2, and the final term accounts for the correlation between the two securities.

This formula shows that the variance of the portfolio depends not only on the variances of the individual securities but also on the degree of correlation between them. When the securities are highly correlated (ρ_{12} close to 1 or -1), the portfolio's variance is more heavily influenced by this relationship.

8.4 Upper Bound on Portfolio Variance

We aim to prove that the portfolio variance σ_V^2 is bounded above by the largest of the two individual variances, σ_1^2 and σ_2^2.

Let us assume without loss of generality that $\sigma_1^2 \leq \sigma_2^2$. We also assume that short sales are not allowed, meaning that the portfolio weights satisfy $w_1, w_2 \geq 0$ and thus $w_1 + w_2 = 1$.

8.4.1 Step 1: Bound on Weighted Standard Deviations

Given that $\sigma_1^2 \leq \sigma_2^2$, the weighted sum of the standard deviations can be bounded as follows:

$$w_1\sigma_1 + w_2\sigma_2 \leq (w_1 + w_2)\sigma_2 = \sigma_2$$

This implies that the weighted average of the two standard deviations cannot exceed the larger of the two, which is σ_2.

8.4.2 Step 2: Upper Bound on Portfolio Variance

Recall the formula for the variance of a two-asset portfolio:

$$\sigma_V^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2\rho_{12}\sigma_1\sigma_2$$

where ρ_{12} is the correlation coefficient between the two securities. Since the correlation coefficient satisfies $-1 \leq \rho_{12} \leq 1$, we can establish the following inequality:

$$\sigma_V^2 \leq w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2\sigma_1\sigma_2$$

This inequality holds because the term $2w_1w_2\rho_{12}\sigma_1\sigma_2$ is maximized when $\rho_{12} = 1$.

8.4.3 Step 3: Simplification of the Variance Expression

The right-hand side of the inequality simplifies as follows:

$$\sigma_V^2 \leq w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2\sigma_1\sigma_2 = (w_1\sigma_1 + w_2\sigma_2)^2$$

From Step 1, we know that $w_1\sigma_1 + w_2\sigma_2 \leq \sigma_2$, so it follows that:

$$(w_1\sigma_1 + w_2\sigma_2)^2 \leq \sigma_2^2$$

Thus, we conclude that:

$$\sigma_V^2 \leq \sigma_2^2$$

8.4.4 Step 4: The Case $\sigma_1^2 \geq \sigma_2^2$

If $\sigma_1^2 \geq \sigma_2^2$, the proof proceeds analogously. In this case, we would have:

$$w_1\sigma_1 + w_2\sigma_2 \leq \sigma_1$$

and similarly:

$$\sigma_V^2 \leq \sigma_1^2$$

Thus, we have shown that the portfolio variance σ_V^2 is bounded above by the largest of the two individual variances, σ_1^2 and σ_2^2. Specifically, we have proven that:

$$\sigma_V^2 \leq \max(\sigma_1^2, \sigma_2^2)$$

8.5 Implementation of a Portfolio with two assets

In this Python code we analyze a simulation of a portfolio with two assets.

In this section, we analyze how to calculate the expected return, variance, and standard deviation for a portfolio of two assets based on scenario-based probabilities using Python.

The initial parameters for this calculation are:

- Probabilities of scenarios: $[0.2, 0.5, 0.3]$
- Returns for Asset 1: $[0.10, 0.12, 0.08]$
- Returns for Asset 2: $[0.15, 0.10, 0.05]$
- Portfolio weights: $w_1 = 0.6$ (60% invested in Asset 1), and $w_2 = 0.4$ (40% invested in Asset 2)

8.5.1 Scenario Probabilities and Returns

We are given three market scenarios, each with an associated probability. For each scenario, we know the expected returns for two different assets. Let:

- p_i: the probability of scenario i
- r_{1i}: the return of Asset 1 in scenario i
- r_{2i}: the return of Asset 2 in scenario i

8.5.2 Expected Return

The expected return for each asset is the weighted average of returns across the scenarios:

$$\mu_1 = \sum_{i=1}^{n} p_i r_{1i}, \quad \mu_2 = \sum_{i=1}^{n} p_i r_{2i}$$

Given the probabilities and returns, we compute:

$$\mu_1 = 0.104, \quad \mu_2 = 0.095$$

8.5.3 Variance and Standard Deviation

The variance for each asset measures the spread of returns around the expected return. It is calculated as:

$$\text{Var}(r_1) = \sum_{i=1}^{n} p_i (r_{1i} - \mu_1)^2, \quad \text{Var}(r_2) = \sum_{i=1}^{n} p_i (r_{2i} - \mu_2)^2$$

The standard deviation is the square root of the variance:

$$\sigma_1 = \sqrt{\text{Var}(r_1)}, \quad \sigma_2 = \sqrt{\text{Var}(r_2)}$$

For Asset 1 and Asset 2, the calculated variances and standard deviations are:

$$\text{Var}(r_1) = 0.000304, \quad \sigma_1 = 0.017436,$$

$$\text{Var}(r_2) = 0.001225, \quad \sigma_2 = 0.035000$$

8.5.4 Covariance

Covariance measures how two assets move together:

$$\text{Cov}(r_1, r_2) = \sum_{i=1}^{n} p_i (r_{1i} - \mu_1)(r_{2i} - \mu_2)$$

The covariance between Asset 1 and Asset 2 is:

$$\text{Cov}(r_1, r_2) = 0.000321$$

8.5.5 Correlation Coefficient

The correlation coefficient normalizes the covariance, making it easier to interpret:

$$\rho_{12} = \frac{\text{Cov}(r_1, r_2)}{\sigma_1 \sigma_2}$$

For Asset 1 and Asset 2, the correlation coefficient is:

$$\rho_{12} = 0.524$$

8.5.6 Portfolio Construction

The portfolio is constructed by allocating 60% to Asset 1 and 40% to Asset 2. The weights are denoted by $w_1 = 0.6$ and $w_2 = 0.4$.

8.5.7 Portfolio Expected Return

The expected return of the portfolio is the weighted average of the expected returns of the two assets:

$$\mu_V = w_1 \mu_1 + w_2 \mu_2$$

Substituting the values, we get:

$$\mu_V = 0.1004$$

8.5.8 Portfolio Variance and Standard Deviation

The variance of the portfolio is calculated as:

$$\sigma_V^2 = w_1^2 \sigma_1^2 + w_2^2 \sigma_2^2 + 2 w_1 w_2 \text{Cov}(r_1, r_2)$$

For our portfolio, the variance is:

$$\sigma_V^2 = 0.000459$$

The standard deviation of the portfolio is the square root of the variance:

$$\sigma_V = \sqrt{\sigma_V^2} = 0.021425$$

8.5.9 Summary Table

The following table summarizes the calculated values for the expected return, variance, standard deviation, and correlation coefficient for both assets and the portfolio.

8.5.10 Conclusion

This analysis explains the calculation of the expected return, variance, and standard deviation for a portfolio composed by two assets. Using these metrics, we can evaluate the risk and return characteristics of a portfolio and understand the benefits of diversification.

Asset	(μ)	(σ^2)	(σ)	(ρ_{12})
Asset 1	0.1040	0.000304	0.017436	-
Asset 2	0.0950	0.001225	0.035000	-
Portfolio	0.1004	0.000459	0.021425	0.524

Table 8.1. Portfolio Comparison Table

8.6 Zero Variance Portfolio for $\rho_{12} = 1$ and $\rho_{12} = -1$

We aim to show under what conditions the portfolio variance σ_V^2 is zero when the correlation coefficient ρ_{12} is either 1 or -1. These cases correspond to perfect positive and perfect negative correlation, respectively.

8.6.1 Case 1: Perfect Positive Correlation ($\rho_{12} = 1$)

Let $\rho_{12} = 1$. In this case, the formula for the portfolio variance simplifies to:

$$\sigma_V^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2\sigma_1\sigma_2 = (w_1\sigma_1 + w_2\sigma_2)^2$$

In order for the portfolio variance to be zero, we need:

$$\sigma_V^2 = 0 \quad \text{if and only if} \quad w_1\sigma_1 + w_2\sigma_2 = 0$$

This condition holds when $\sigma_1 \neq \sigma_2$, and the weights w_1 and w_2 are given by:

$$w_1 = -\frac{\sigma_2}{\sigma_1 - \sigma_2}, \quad w_2 = \frac{\sigma_1}{\sigma_1 - \sigma_2}$$

Since the weights w_1 and w_2 sum to 1, we can see that this configuration requires short sales because either w_1 or w_2 must be negative. Specifically, the sign of w_1 is negative when $\sigma_1 > \sigma_2$, which implies the need for short selling in security 1.

8.6.2 Case 2: Perfect Negative Correlation ($\rho_{12} = -1$)

Now let $\rho_{12} = -1$. In this case the equation takes the following form:

$$\sigma_V^2 = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 - 2w_1w_2\sigma_1\sigma_2 = (w_1\sigma_1 - w_2\sigma_2)^2$$

For the portfolio variance to be zero, we need:

$$\sigma_V^2 = 0 \quad \text{if and only if} \quad w_1\sigma_1 - w_2\sigma_2 = 0.$$

This condition holds when the weights w_1 and w_2 are given by:

$$w_1 = \frac{\sigma_2}{\sigma_1 + \sigma_2}, \quad w_2 = \frac{\sigma_1}{\sigma_1 + \sigma_2}$$

Since both w_1 and w_2 are positive in this case, short sales are not required. The weights represent a balanced portfolio where the variance is eliminated due to the perfect negative correlation between the two securities.

In summary, when $\rho_{12} = 1$ (perfect positive correlation), the portfolio variance σ_V^2 can be zero if the portfolio involves short sales, with the weights given by the equations of w_1 and w_2 in case 1. In contrast, when $\rho_{12} = -1$ (perfect negative correlation), the portfolio variance σ_V^2 can also be zero, but no short sales are required, with the weights given by the equations of w_1 and w_2 in case 2.

8.7 Implementation of Portfolio Relationships

In this section we analyze an implementation in Python that outlines the process of simulating the variance and expected return of a two-asset portfolio, considering different correlation scenarios between the assets. The objective is to visualize how varying the allocation between the two assets affects the overall portfolio characteristics.

We begin by defining the key parameters for the two assets involved in the portfolio:

- **Volatility of Asset 1** (σ_1): Set to 1.
- **Expected Return of Asset 1** (μ_1): Set to 1.5.
- **Volatility of Asset 2** (σ_2): Set to 2.
- **Expected Return of Asset 2** (μ_2): Set to 2.5.

A variable s is introduced to represent the proportion of asset 2 in the portfolio. This variable ranges from -1 to 2, which allows for exploration of short selling (where $s < 0$) as well as over-investing in asset 2 (where $s > 1$).

The analysis includes two primary correlation scenarios, denoted by ρ_{12}:

8.7.1 Case 1: Perfect Negative Correlation ($\rho_{12} = -1$)

In this scenario, the portfolio variance and expected return are calculated. The portfolio variance is represented by the absolute value of the linear combination of the asset volatilities, while the expected return is computed as a weighted sum of the individual asset returns.

8.7.2 Case 2: Perfect Positive Correlation ($\rho_{12} = 1$)

Similarly, the calculations for the portfolio variance and expected return are performed under the assumption of perfect positive correlation between the assets.

8.7.3 Visualization

The results are plotted to illustrate the relationship between portfolio variance and expected return across the defined range of s:

- The plot includes dashed lines for both correlation scenarios.
- A highlighted segment is shown for $0 < s < 1$, which represents the case without short selling.
- Individual assets are marked on the plot for visual clarity.

This simulation provides insights into how different correlation scenarios influence the risk-return profile of a two-asset portfolio. By adjusting the allocation parameter s, investors can better understand the implications of their investment choices.

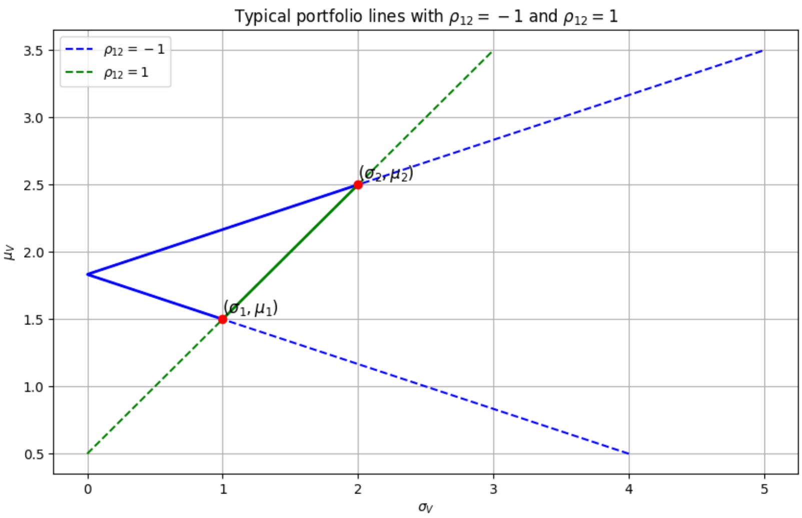

Figure 8.1. Relationship between Variance and Exp. Return

8.8 Finding minimum variance portfolio for any ρ_{12}

8.8.1 Portfolio Characteristics

The expected return μ_V and variance σ_V^2 of the portfolio are defined as:

$$\mu_V = (1 - s)\mu_1 + s\mu_2$$
$$\sigma_V^2 = (1 - s)^2\sigma_1^2 + s^2\sigma_2^2 + 2s(1 - s)\rho_{12}\sigma_1\sigma_2$$

where:

- μ_1, μ_2: Expected returns of assets 1 and 2
- σ_1, σ_2: Volatilities (standard deviations) of the two assets
- ρ_{12}: Correlation between the returns of the two assets

The expected return μ_V is a linear function of s, while the variance σ_V^2 is a quadratic function of s with a positive leading coefficient, subject to the condition:

$$\sigma_1^2 + \sigma_2^2 - 2\rho_{12}\sigma_1\sigma_2 \geq 0$$

To minimize the variance σ_V^2 (or the standard deviation σ_V), we consider the following:

Theorem 5.5 from Capiński and Zastawniak [2011]:
For $-1 < \rho_{12} < 1$, the portfolio with minimum variance is attained at:

$$s_0 = \frac{\sigma_2^2 - \rho_{12}\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2 - 2\rho_{12}\sigma_1\sigma_2}$$

If short sales are not allowed, the minimum variance is attained at:

$$s_{\text{min}} = \begin{cases} 0, & \text{if } s_0 < 0, \\ s_0, & \text{if } 0 \leq s_0 \leq 1, \\ 1, & \text{if } s_0 > 1 \end{cases}$$

8.8.2 Derivation of Minimum Variance Portfolio

To find the minimum variance, we compute the derivative of σ_V^2 with respect to s and set it to zero:

$$\frac{d}{ds}\sigma_V^2 = -2(1-s)\sigma_1^2 + 2s\sigma_2^2 + 2(1-s)\rho_{12}\sigma_1\sigma_2 - 2s\rho_{12}\sigma_1\sigma_2 = 0$$

Solving for s yields the expression for s_0. The second derivative is:

$$\frac{d^2}{ds^2}\sigma_V^2 = 2\sigma_1^2 + 2\sigma_2^2 - 4\rho_{12}\sigma_1\sigma_2,$$

which simplifies to:

$$2(\sigma_1 - \sigma_2)^2 \geq 0$$

This indicates that the second derivative is positive, confirming a global minimum at s_0.

If short sales are not allowed, we find the minimum within the bounds $0 \leq s \leq 1$:

- If $0 \leq s_0 \leq 1$, the minimum occurs at s_0
- If $s_0 < 0$, the minimum is at $s = 0$
- If $s_0 > 1$, the minimum is at $s = 1$

8.9 Implementation of Min-Var Portfolio

In this implementation using Python, we analyze plots that represent the minimum variance portfolios. We base our plots on the following equations. From Theorem 5.5, the variance of a portfolio, σ_V^2, as a function of s is given by:

$$\sigma_V^2(s) = (1-s)^2\sigma_1^2 + s^2\sigma_2^2 + 2s(1-s)\rho_{12}\sigma_1\sigma_2$$

The expected return μ_V of the portfolio as a function of s is:

$$\mu_V(s) = (1-s)\mu_1 + s\mu_2$$

The optimal value s_0 minimizing variance is given by:

$$s_0 = \frac{\sigma_2^2 - \rho_{12}\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2 - 2\rho_{12}\sigma_1\sigma_2}$$

We also consider the constraints for short selling, where $s_{\min}$ is defined as:

$$s_{\min} = \begin{cases} 0 & \text{if } s_0 < 0 \\ s_0 & \text{if } 0 \leq s_0 \leq 1 \\ 1 & \text{if } s_0 > 1 \end{cases}$$

8.9.1 Plotting Approach

We generated two types of plots:

1. Plotting $\sigma_V^2(s)$ for different values of ρ_{12}, highlighting the region where short selling is not allowed ($0 \leq s \leq 1$).
2. Plotting the efficient frontier in the (μ_V, σ_V) space for ρ_{12} values between 0 and 1.

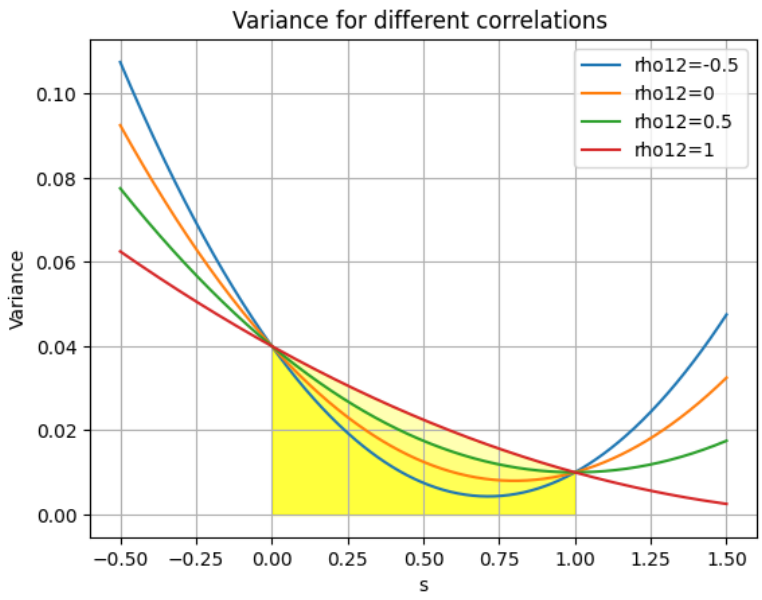

Figure 8.2. Variance for different coefficients

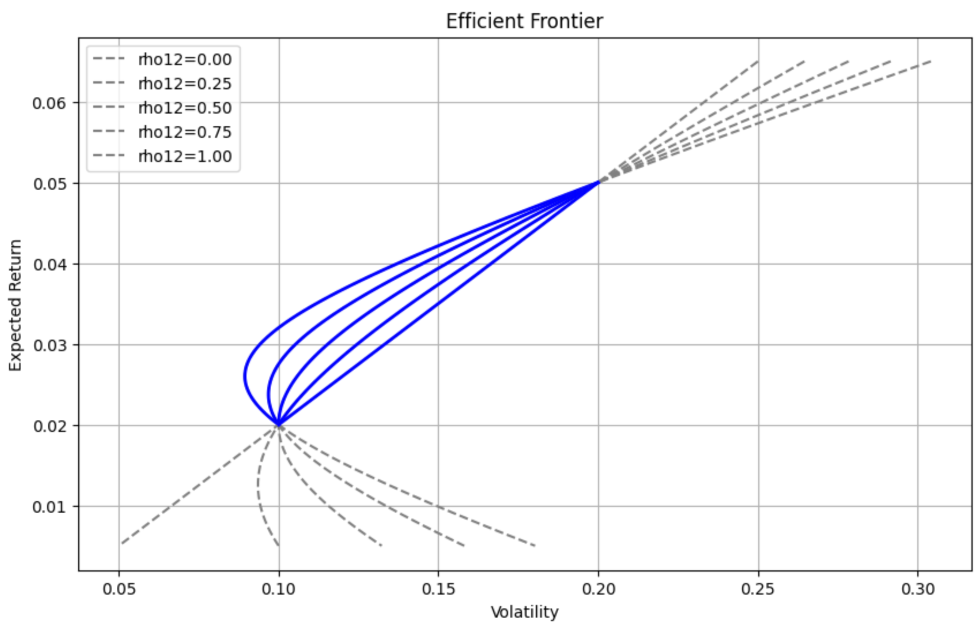

Figure 8.3. Expected return vs Volatility for different coefficients

8.10 Portfolios with a Risk-Free Security

We conclude this section with a brief discussion of portfolios that include a risk-free security. The variance of the risky security (a stock) is positive, while the risk-free component (a bond) has zero variance.

The standard deviation σ_V of a portfolio consisting of a risky security with expected return μ_1 and standard deviation $\sigma_1 > 0$, and a risk-free security with return r_F and standard deviation zero, depends on the weight w_1 of the risky security as follows:

$$\sigma_V = |w_1|\sigma_1$$

Let $\sigma_1 > 0$ be the standard deviation of the risky asset and $\sigma_2 = 0$ for the risk-free asset. Then the equation reduces to:

$$\sigma_V^2 = w_1^2\sigma_1^2$$

and the formula for σ_V follows by taking the square root:

$$\sigma_V = |w_1|\sigma_1$$

8.11 Risk and Expected Return on a Portfolio with many risky assets

A portfolio constructed from n different securities can be described in terms of their weights:

$$w_i = \frac{x_i S_i(0)}{V(0)}, \quad i = 1, \ldots, n,$$

where x_i is the number of shares of security i in the portfolio, $S_i(0)$ is the initial price of security i, and $V(0)$ is the amount initially invested in the portfolio. We can arrange the weights into a one-row matrix:

$$w = [w_1\, w_2\, \cdots\, w_n]$$

The weights must add up to 1, which is written in matrix form as:

$$1 = uw^T$$

where

$$u = [1\, 1\, \cdots\, 1]$$

is a one-row matrix with all n entries equal to 1, and w^T is the transpose of w.

The set of attainable portfolios consists of all portfolios with weights w satisfying equation (5.14).

Suppose the returns on the securities are $K_1, \ldots, K_n$, with expected returns arranged into a one-row matrix:

$$m = [\mu_1\, \mu_2\, \cdots\, \mu_n], \quad \mu_i = \mathbb{E}(K_i).$$

The covariances between returns, denoted by $c_{ij} = \mathrm{Cov}(K_i, K_j)$, form the $n \times n$ covariance matrix:

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

The matrix C is symmetric and positive definite, with diagonal elements $c_{ii} = \mathrm{Var}(K_i)$. We assume that C has an inverse C^{-1}.

The expected return $\mu_V = \mathbb{E}(K_V)$ and variance $\sigma_V^2 = \mathrm{Var}(K_V)$ of a portfolio with weights w are given by:

$$\mu_V = mw^T$$
$$\sigma_V^2 = wCw^T$$

8.12 Portfolio with the smallest variance

The portfolio with the smallest variance in the attainable set has weights

$$w = \frac{uC^{-1}}{uC^{-1}u^T}$$

provided that the denominator is non-zero.

8.13 Portfolio with the smallest variance whose expected return is equal to a given number

The portfolio with the smallest variance among attainable portfolios with expected return μ_V has weights

$$w = \frac{\begin{vmatrix} 1 & uC^{-1}m^T \\ \mu_V & mC^{-1}m^T \end{vmatrix} uC^{-1} + \begin{vmatrix} uC^{-1}u^T & 1 \\ mC^{-1}u^T & \mu_V \end{vmatrix} mC^{-1}}{\begin{vmatrix} uC^{-1}u^T & uC^{-1}m^T \\ mC^{-1}u^T & mC^{-1}m^T \end{vmatrix}}$$

provided that the determinant in the denominator is non-zero. The weights depend linearly on μ_V.

Part III

Machine Learning Models to Price American Put Options

Chapter 9

Introduction and Related works

9.1 Motivation and goals

In this work, we showed how to implement standard mathematical models to price Vanilla and Exotic Options. Usually, pricing can be performed using simulations of the underlying price paths, but this method has some limitations specifically in the case of American Put Options. We are now trying to analyze modern approaches to price American Put Options in more detail, starting from a paper published in September 2024.

9.2 Related works

One of the most recent paper involving Machine Learning to price American Put Options is Djagba and Ndizihiwe [2024]. As reported in the introduction of Djagba and Ndizihiwe [2024], traditional pricing models, like Black-Scholes formulas, work well for European options but struggle with the complexity of American options. Machine learning pricing models can improve option pricing accuracy by adapting to changing market conditions and capturing complex patterns. In addition, Machine learning pricing models overcome many of the mathematical difficulties arising from the pricing of American Put Options. The Least-Squares Monte Carlo (LSM) method by Longstaff-Schwartz (Longstaff and Schwartz [2001]) is a popular technique for pricing American options. It estimates the continuation value of the option at each step, using least-squares regression. Machine learning models are used in the paper of Djagba and Ndizihiwe [2024] instead of using a regression model to estimate the continuation values. This should improve significantly the accuracy and speed of estimation of the continuation values.

9.3 Formulation of the problem

Following the paper of Djagba and Ndizihiwe [2024], we now make a summary of the reasoning behind using machine learning in the pricing of derivatives. We may say that American option pricing is a challenging task due to the feature that allows for early exercise, which complicates the valuation process. The research carried out in Djagba and Ndizihiwe [2024] investigates how machine learning models can enhance the Least Squares Monte Carlo (LSM) method, a widely used approach for pricing American options.

Traditional pricing models, such as the Black-Scholes model, have been useful for European options but are not well-suited for American options because they do not account for the early exercise feature. This limitation creates the need for more advanced approaches that can better capture the complexities of American options. With the growing availability of data and advancements in computational methods, machine learning, especially deep learning, has emerged as a tool to improve option pricing by recognizing complex patterns and relationships in historical market data.

Machine learning models can adapt to market fluctuations and incorporate a broader set of input variables, leading to more accurate pricing predictions. By analyzing historical data, these models can learn from past market dynamics and better predict option prices under varying conditions, while the role of history was not taken into account in the paper of Longstaff and Schwartz [2001]. This ability to adapt and evolve makes machine learning a powerful tool for handling the complexities of option pricing, especially when dealing with the non-linear payoff structures of American options.

The study of machine learning role in enhancing the LSM approach has the potential to provide more reliable pricing methods, which can contribute to better decision-making and risk management for financial professionals. By combining Monte Carlo simulations with machine learning models, the research aims to overcome the limitations of traditional approaches, offering greater flexibility and accuracy in pricing. This improved understanding of option pricing could benefit the broader financial system by enhancing the tools available for assessing and managing financial derivatives.

9.4 Base Model

We want to determine the price P of an American option, where the holder has the right to exercise the option at any time before the maturity T.

Let:

- $t = 0, 1, \ldots, M$ be the discrete time steps.
- S_t be the stock price at time t.
- K be the strike price.
- r be the risk-free rate.
- σ be the volatility of the stock.
- P_t be the price of the option at time t.

- $\mathbb{E}[\cdot]$ be the expectation operator under the risk-neutral measure.

The objective is to maximize the payoff of the option, either by exercising at time t or by continuing to hold it, using backward induction.

9.4.1 Variables

Let:

- $C(S_t)$ be the continuation value of the option at time t, approximated by a polynomial regression model.
- $E(S_t, K) = \max(K - S_t, 0)$ be the exercise value of a put option at time t.
- τ be the optimal stopping time (i.e., the time to exercise the option).

9.4.2 Objective

The objective is to maximize the expected payoff by selecting the optimal stopping time τ, balancing between exercising and continuing:

$$\max_{\tau \in \{t,\dots,T\}} \mathbb{E}\left[e^{-r\tau} E(S_\tau, K)\right]$$

9.4.3 Constraints

1. **Stock price dynamics**: The stock price follows a Geometric Brownian Motion (GBM):

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

where W_t is a Wiener process under the risk-neutral measure.

2. **Polynomial regression for continuation value**: The continuation value $C(S_t)$ at time t is modeled by fitting a polynomial regression on the in-the-money paths. Let X be the set of stock prices S_t where the option is in-the-money. The continuation value is approximated by a polynomial $\hat{C}(S_t)$:

$$\hat{C}(S_t) = \beta_0 + \beta_1 S_t + \beta_2 S_t^2 + \cdots + \beta_d S_t^d$$

where $\beta_0, \beta_1, \dots, \beta_d$ are the polynomial coefficients and d is the degree of the polynomial.

3. **Exercise condition**: At each time step t, the option holder decides whether to exercise the option if the exercise value exceeds the continuation value:

$$\text{If } E(S_t, K) > \hat{C}(S_t), \text{ then exercise at time } t.$$

4. **No exercise at maturity**: At maturity T, the continuation value is zero, and the exercise decision is based purely on the intrinsic value:

$$P_T = E(S_T, K).$$

5. **Backward recursion**: The price at time t is recursively calculated by discounting the continuation value and comparing it with the exercise value:

$$P_t = \max\left(E(S_t, K), \mathbb{E}\left[e^{-r\Delta t} P_{t+1} \mid S_t\right]\right).$$

6. **Discounting future cash flows**: The continuation value is computed using the discounted future cash flows:

$$Y_t = P_{t+1} e^{-r\Delta t}$$

where P_{t+1} represents the option price at the next time step, discounted by the risk-free rate r.

9.4.4 Baseline Algorithm

The LSM method proceeds as follows:

1. Simulate N asset price paths S_t for $t = 0, 1, \ldots, M$ using a geometric Brownian motion model.
2. At maturity T, set the option payoff to the exercise value: $P_T = E(S_T, K)$.
3. For each time step $t = M-1, M-2, \ldots, 1$:
   - Identify the in-the-money paths where $E(S_t, K) > 0$.
   - Perform a polynomial regression on the in-the-money paths to approximate the continuation value $\hat{C}(S_t)$.
   - Compare the continuation value $\hat{C}(S_t)$ with the exercise value $E(S_t, K)$ to determine whether to exercise.
   - If exercise occurs, set the cash flow to $E(S_t, K)$. Otherwise, set it to the discounted continuation value.
4. Discount the cash flows to the present time $t = 0$ to compute the option price P_0.

The final objective can be expressed as:

$$\max_{\tau \in \{t, \ldots, T\}} \mathbb{E}\left[\max\left(E(S_\tau, K), \hat{C}(S_\tau)\right) e^{-r\tau}\right]$$

subject to:

$$P_T = E(S_T, K),$$
$$C(S_t) = \hat{C}(S_t) = \beta_0 + \beta_1 S_t + \beta_2 S_t^2 + \cdots + \beta_d S_t^d,$$
$$S_t = S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right) t + \sigma W_t\right).$$

9.4.5 Pseudo-Code

The algorithm presented in Longstaff and Schwartz [2001] is summarized in the following pseudo-code.

Algorithm 1 Longstaff-Schwartz Algorithm for Pricing American Option

1: **Start**
2: Create random paths $\tilde{S}_j(t_i), i = t_0, \ldots, n_{\text{step}}, j = 1, \ldots, n_{\text{path}}, t_i = t_0 + i \cdot \Delta t$
3: For each path, set payoff at maturity to the payoff $V_j(T) = h(\tilde{S}_j(T))$
4: Start at maturity $i = n_{\text{timestep}}$
5: **while** $i > 1$ **do**
6:     $i \leftarrow i - 1$
7:     **for** each path j **do**
8:         Discount the price $V_j(t_i) = e^{-r\Delta s} V_j(t_{i+1})$
9:     **end for**
10:     Solve regression problem to find β coefficients
11:     **for** each path j **do**
12:         Compute the continuation value c_j
13:         **if** $h_j > c_j$ **then**
14:             Exercise, i.e., set $V_j(t_i) = h(\tilde{S}_j(t_i))$
15:         **end if**
16:     **end for**
17: **end while**
18: **for** each path j **do**
19:     Discount the price $V_j(t_0) = e^{-r\Delta s} V_j(t_1)$
20: **end for**
21: The price today is the average over all paths $V(\tilde{S}(t_0)) = \frac{1}{n_{\text{path}}} \sum_{j=1}^{n_{\text{path}}} V_j(t_0)$
22: **Stop**

Step 12 is where the paper of Djagba and Ndizihiwe [2024] applies machine learning instead of solving the regression problem. In the paper Djagba and Ndizihiwe [2024] the training of the models was carried out on simulations of paths described by Geometric Brownian Motion. In our analysis, we will carry out the training of the models on historical data in order to improve the role of history and we will use the trained model in Step 12 to produce the estimated continuation value.

9.5 Formulation with Machine Learning

Let:

- $t = 0, 1, \ldots, M$ be the discrete time steps.
- S_t be the stock price at time t.
- K be the strike price.
- r be the risk-free rate.
- σ be the volatility of the stock.
- P_t be the price of the option at time t.
- $\mathbb{E}[\cdot]$ be the expectation operator under the risk-neutral measure.

The objective is to maximize the payoff of the option, either by exercising at time t or by continuing to hold it, using backward induction.

9.5.1 Variables

Let:

- $C(S_t, K, r, \sigma)$ be the continuation value of the option at time t, approximated by a neural network.
- $E(S_t, K) = \max(K - S_t, 0)$ be the exercise value of a put option at time t.
- τ be the optimal stopping time (i.e., the time to exercise the option).

9.6 Objective

The objective is to maximize the expected payoff by selecting the optimal stopping time τ, balancing between exercising and continuing:

$$\max_{\tau \in \{t, \ldots, T\}} \mathbb{E}[e^{-r\tau} E(S_\tau, K)]$$

9.7 Constraints

1. **Stock price dynamics**: The stock price follows a Geometric Brownian Motion (GBM):

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

where W_t is a Wiener process under the risk-neutral measure.

2. **Neural network approximation**: The continuation value $C(S_t, K, r, \sigma)$ at time t is modeled by a neural network $\mathcal{NN}(S_t, K, r, \sigma; \theta)$ with parameters θ, trained on historical data.

3. **Exercise condition**: At each time step t, the option holder decides whether to exercise the option if the exercise value exceeds the continuation value:

$$\text{If } E(S_t, K) > C(S_t, K, r, \sigma), \text{ then exercise at time } t.$$

4. **No exercise at maturity**: At maturity T, the continuation value is zero, and the exercise decision is based purely on the intrinsic value:

$$P_T = E(S_T, K).$$

5. **Backward recursion**: The price at time t is recursively calculated by discounting the continuation value and comparing it with the exercise value:

$$P_t = \max\left(E(S_t, K), \mathbb{E}\left[e^{-r\Delta t} P_{t+1} \mid S_t\right]\right).$$

9.8 Neural Network Training

The neural network is trained on historical data to minimize the mean squared error between predicted and actual continuation values:

$$\min_\theta \sum_{t=1}^{M} \left(\mathcal{NN}(S_t, K, r, \sigma; \theta) - \text{True Continuation Value}\right)^2$$

where the true continuation value is estimated by: $\mathcal{NN}(S_t, K, r, \sigma; \theta)$
The true continuation value is defined as:

$$\text{True Continuation Value at time } t = P_{t+1} e^{-r\Delta t}.$$

9.8.1 Final Formulation

The final objective can be rewritten as:

$$\max_{\tau \in \{t, \dots, T\}} \mathbb{E}\left[\max\left(E(S_\tau, K), \mathcal{NN}(S_\tau, K, r, \sigma; \theta)\right) e^{-r\tau}\right]$$

subject to:

$$P_T = E(S_T, K),$$
$$C(S_t, K, r, \sigma) = \mathcal{NN}(S_t, K, r, \sigma; \theta),$$
$$S_t = S_0 \exp\left(\left(r - \frac{1}{2}\sigma^2\right) t + \sigma W_t\right).$$

9.9 Neural Network Implementation

The neural network used in the code is a feedforward deep learning model created using the TensorFlow/Keras library. The network architecture consists of the following components:

- **Input Layer**: The input to the neural network consists of four variables: stock price (S), strike price (K), risk-free rate (r), and volatility (σ). These variables are combined into a 2D array that forms the training data.
- **Hidden Layers**:
  - The model has two dense (fully connected) hidden layers, each with 64 neurons. These layers use ReLU (Rectified Linear Unit) activation functions, which introduce non-linearity into the model, allowing it to learn complex relationships in the input data.
- **Output Layer**: The output layer consists of a single neuron with a linear activation function, which represents the predicted continuation value of the option. The continuation value helps determine whether the option should be exercised early or held for future exercise.
- **Compilation**: The model is compiled using the Adam optimizer, which adjusts learning rates based on past gradient information, and the mean squared error (MSE) is used as the loss function. The goal is to minimize the error between predicted and actual continuation values during training.
- **Training**: The network is trained on a dataset containing historical stock prices, option prices, strike prices, risk-free rates, and volatilities. The training process aims to minimize the error between predicted continuation values and discounted option prices using backpropagation and optimization techniques.
- **Validation**: The dataset is split into training and testing sets to evaluate the model's performance. The loss values for both training and testing sets are monitored over multiple epochs (in this case, 100), and accuracy metrics such as Mean Squared Error (MSE) and R-squared (R^2) are computed to assess the model's predictive capabilities.

9.10 Differences Between Neural Network LSM and Classical LSM

The classical Least Squares Monte Carlo (LSM) method, proposed by Longstaff and Schwartz, relies on **polynomial regression** to estimate the continuation value based on in-the-money paths. Typically, a polynomial of degree 2 or 3 is used to fit stock prices at each time step, and the decision to exercise or hold the option is based on this estimate.

In contrast, this implementation replaces the polynomial regression with a **neural network**. Instead of using only the stock price to estimate continuation value, the neural network takes as input four variables: the stock price (S), strike price (K), risk-free rate (r), and volatility (σ). This allows for a more flexible and complex relationship between inputs and continuation value.

9.10.1 Key Differences:

1. **Model Flexibility:** Classical LSM uses a fixed polynomial regression, which limits the complexity of relationships that can be captured. The neural network, however, is capable of modeling much more intricate, non-linear relationships between inputs, leading to potentially more accurate estimates of continuation value.

2. **Input Variables:** In classical LSM, only the stock price is used to estimate the continuation value. In this neural network approach, additional variables (strike price, risk-free rate, volatility) are considered, enabling the model to incorporate a wider range of factors in its estimates.

3. **Learning from Data:** The polynomial regression in classical LSM is a one-time fit for each time step, based solely on available paths. The neural network, however, learns from historical data and can generalize better to new scenarios, especially in dynamic markets.

4. **Adaptability:** Classical LSM is static and unable to adapt to changing market conditions, such as fluctuations in volatility or interest rates. The neural network, on the other hand, can adapt to different market conditions through training on historical data.

9.11 Efficiency of the Neural Network Implementation

This neural network implementation is more efficient than classical LSM for several reasons:

- **Higher Accuracy:** The neural network can model non-linear relationships more effectively than polynomial regression. By incorporating additional market variables, the model provides more accurate estimates of continuation value, leading to improved option pricing.

- **Generalization to Market Changes:** By learning from historical data, the neural network can generalize better when market conditions change. The classical LSM model does not learn from historical data of stock prices.

- **Handling Complex Relationships:** Classical LSM assumes a simple quadratic relationship between stock prices and continuation values. The neural network, with its multi-layered structure, can handle much more complex relationships, leading to better decision-making regarding early exercise.

- **Dynamic Input Consideration:** The neural network takes multiple variables into account, such as stock price, strike price, risk-free rate, and volatility. This leads to better predictions of the optimal exercise strategy, which in turn improves pricing accuracy.

9.12 Advantages

In conclusion, the neural network-based LSM implementation offers a more advanced and flexible approach to American option pricing compared to the classical LSM method. By using machine learning and historical data, the model can capture complex market patterns, adapt to changing conditions, and provide better estimates of continuation values, thereby enhancing option pricing accuracy and decision-making.

Chapter 10

Implementation

10.1 Dataset

The dataset used in this analysis contains data relative to American Put Options for the stock of Apple quoted on the NASDAQ index.
The full dataset was cleaned and only the relevant features of the Options were taken into account. The dataset contains about 400,000 rows from 1st January 2021 to 31st December 2023, this number of observations allows for good training of a neural network. Let's see more in detail the features:

- **UNDERLYING_PRICE**: The stock price of Apple at the time of observation, representing the most recent price quote.
- **STRIKE**: The strike price of the American put option. This is the price at which the option holder has the right to sell the underlying stock.
- **OPTION_TRADE_PRICE**: The price of the American put option for the specific strike price and maturity. In my analysis, it was computed as the average between the ask and the bid price.
- **DTE (Days to Expiration)**: The time until the option's maturity, given in days. This variable helps determine how much time the option has before it expires.

10.2 Training and Testing

In order to evaluate the performance of the model in training and testing I split the dataset into two partitions: one made of 90% of data to train the model, while the remaining 10% of data was used to test the trained model. The performance evaluation was carried out using the R^2 metric and the loss.

10.2.1 R^2 metric

The R^2 (R-squared) metric, also known as the coefficient of determination, measures the proportion of the variance in the dependent variable that is predictable from the independent variables in a regression model. The formula for R^2 is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

where:

- y_i is the actual value of the dependent variable
- $\hat{y}_i$ is the predicted value from the regression model
- $\bar{y}$ is the mean of the actual values y_i
- n is the number of data points

In this equation:

- The numerator $\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ represents the sum of squared residuals (or the variance of the error term)
- The denominator $\sum_{i=1}^{n} (y_i - \bar{y})^2$ is the total variance of the data (the total sum of squares)

An R^2 value of 1 indicates that the model perfectly explains the variance in the data, while a value of 0 means the model explains none of the variance.

10.2.2 Loss

The **loss** in a neural network represents the difference between the predicted output and the actual target values. It quantifies how well or poorly the model is performing on a given dataset during training. The lower the loss, the closer the model's predictions are to the true values.

1. **Loss Function**: The loss function (also called the cost or objective function) measures this error and guides the training process. Common loss functions include:

- **Mean Squared Error (MSE)** for regression tasks:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where y_i is the actual target value, $\hat{y}_i$ is the predicted value, and n is the number of samples.

2. **Optimization Process**: During training, the neural network adjusts its internal parameters (weights and biases) to minimize the loss function. This is done through algorithms like **gradient descent**, which updates the parameters in the direction that reduces the loss the most.

- A **low loss** indicates that the model is making predictions that are close to the true target values, meaning it is learning effectively from the training data.
- A **high loss** suggests that the model's predictions are far from the target, meaning the model is either underfitting (not learning enough from the data) or overfitting (focusing too much on specific details and noise in the training data).

The loss function is a crucial measure that reflects how well the neural network is performing during training. A low loss value signifies a well-trained model that makes accurate predictions, while a high loss indicates poor performance, either due to a need for more training or other issues like an improper model architecture or noisy data.

10.3 More details on the Algorithm

This code combines data preprocessing, neural network training, and the Least Squares Monte Carlo (LSM) method to price an American put option. Below is a step-by-step breakdown of the procedure. The code can be found on Github.

10.3.1 Importing Libraries

The code begins by importing essential libraries:

- `numpy` for numerical computations.
- `matplotlib.pyplot` for plotting graphs.
- `tensorflow.keras` to build and train the neural network model.
- `pandas` to load and manipulate the dataset.
- `sklearn.model_selection` for splitting the dataset into training and testing sets.
- `sklearn.preprocessing.StandardScaler` for feature normalization.
- `sklearn.metrics` for evaluating the model performance using metrics like Mean Squared Error (MSE) and R-squared (R^2).

10.3.2 Loading the Dataset and Extracting Relevant Columns

The dataset is loaded from a CSV file using `pandas`. The dataset contains historical information about various features required for option pricing.
The relevant columns represent the key variables required for American option pricing:

- **Stock Prices** (S_hist): The underlying stock prices over time.
- **Option Prices** (P_hist): The observed option prices.
- **Strike Prices** (K_hist): The strike price of the option.
- **Risk-Free Rate** (r_hist): The interest rate used for discounting.
- **Volatility** (σ_hist): The implied volatility used to predict stock price movements.

10.3.3 Neural Network Model Definition

A neural network is created using TensorFlow's Keras API. The architecture consists of:

- Two hidden layers, each with 64 neurons and ReLU activation.
- An output layer with a linear activation function to produce a continuous value (the continuation value).

The model is compiled using the Adam optimizer and MSE loss function, which is typical for regression tasks.

10.3.4 Calculating the Target Values (`Y_hist`)

The target variable Y_hist, which represents the discounted future option prices, is calculated by iterating backward over the option prices:

- For each time step t, the option price at $t + 1$ is discounted using the risk-free rate.
- At maturity, the continuation value is set to zero since there is no continuation value at the end of the option's life.

10.3.5 Preparing the Input Data (`X_hist`)

The input features for the neural network consist of:

- Stock price (S_train)
- Strike price (K_train)
- Risk-free rate (r_train)
- Volatility (σ_train)

These features are combined into a single array using `numpy.column_stack`.

10.3.6 Feature Scaling

The features are standardized using `StandardScaler` to improve the neural network's performance. This transforms the features to have zero mean and unit variance.

10.3.7 Train-Test Split

The normalized dataset is split into training and testing sets using `train_test_split`. 90% of the data is used for training and 10% for testing, allowing the model to generalize well.

10.3.8 Training the Neural Network

The neural network is trained on the training data for 100 epochs with a batch size of 32. The validation data is used to track performance and detect overfitting. Loss values for both training and validation sets are plotted over time.

10.3.9 Model Evaluation

After training, the model makes predictions on the test set. The predictions are evaluated using two metrics:

- **Mean Squared Error (MSE)**: Measures the average squared difference between predictions and actual values.
- **R-squared** (R^2): Represents the proportion of variance in the target that is predictable from the features.

10.3.10 Simulating Stock Price Paths

Stock price paths are simulated using Geometric Brownian Motion (GBM). This step generates N paths over M time steps, using the initial stock price S_0, risk-free rate r, and volatility σ.

10.3.11 Payoff Function for American Put Option

The payoff function for the American put option is defined. At each time step, the payoff is calculated as the maximum of $K - S$ or 0, representing early exercise if the option is in the money.

10.3.12 LSM Algorithm for Option Pricing

The LSM algorithm is used to price the American put option:

- At each time step, the exercise payoff is compared to the continuation value, which is predicted using the neural network.

- The neural network inputs stock price, strike price, risk-free rate, and volatility to estimate the continuation value.
- If the exercise value is greater than the continuation value, the option is exercised early, and the cash flows are updated accordingly.

10.3.13 Final Option Price Calculation

After iterating through all time steps, the final option price is calculated by averaging the discounted cash flows at the present time. This provides the estimated fair price of the option.

10.3.14 Output of the Option Price

Finally, the computed American put option price is printed as the result of the LSM algorithm combined with the neural network predictions.

10.4 Results

1. **Model Loss Over Epochs:**
   The graph shows the training and testing loss of the neural network over 100 epochs.
   - **Train Loss (Blue Line):** The model's loss on the training set decreases consistently over time, with some fluctuations. This indicates that the neural network is learning and improving its performance on the training data
   - **Test Loss (Orange Line):** The test loss is more erratic, with significant peaks and valleys, but overall shows a downward trend. This suggests that the model is generalizing to unseen data, albeit with some variance

2. R-Squared Value of 74%:
An R^2 value of 0.74 means that the model explains 74% of the variance in the target variable (option prices). While not perfect, this indicates that the neural network captures most of the underlying relationship between the inputs (stock price, strike, risk-free rate, volatility) and the continuation value (future option prices). A higher R^2 value would suggest better predictive power, but 74% suggests a reasonably good model in practice.
3. Mean Squared Error (MSE) of 350:
The Mean Squared Error (MSE) quantifies the average squared difference between the predicted continuation values and the actual option prices. An MSE of 350 means that, on average, the model's predictions deviate from the actual values by a squared error of 350. This is acceptable depending on the scale of the target variable (option price). Lower MSE indicates better model performance.
4. Model Behavior:
The fluctuations in test loss could indicate slight overfitting at certain points, as the model may perform well on the training data but struggles to generalize perfectly on the test data, as seen in the spikes. However, the general downward trend in both training

and testing loss suggests that the model is learning and improving, and the final model is usable for predicting option prices.

5. Coherent Price Generation:

The combination of an acceptable R^2 score and MSE, alongside the converging loss curves, indicates that the model is capable of generating a coherent price for the American put option. This suggests that the neural network integrated within the Longstaff-Schwartz (LSM) algorithm provides reliable continuation value approximations, which result in a realistic option pricing estimation.

Conclusion:

Overall, the neural network model, while not perfect, shows a solid performance. With an R^2 of 0.74, it explains a significant portion of the variance in the target variable. The fluctuations in test loss and the MSE of 350 indicate some room for improvement, but the algorithm generates a coherent and reasonable price for the American put option based on the given data.

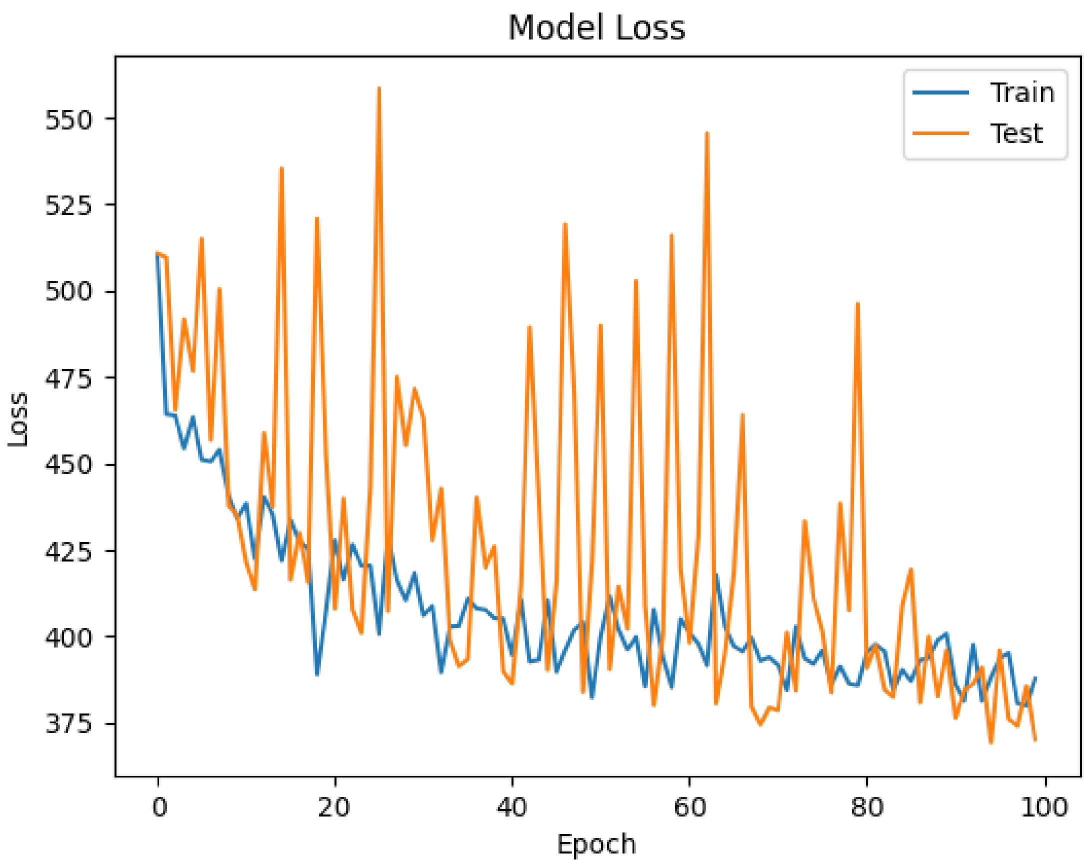

Figure 10.1. Train and Test Loss for the training process

10.5 Test

The following results summarize the prices of American put options under various parameter settings, calculated using the algorithm with the neural network explained previously:

Case 1: Initial Stock Price $S_0 = 100$

- Initial stock price: $S_0 = 100$
- Strike price: $K = 100$
- Risk-free rate: $r = 0.05$
- Volatility: $\sigma = 0.30$
- Time to maturity: $T = 1.0$ year
- Number of time steps: $M = 50$
- Number of simulations: $N = 10{,}000$
- **American Put Option Price: 10.13**

Case 2: Initial Stock Price $S_0 = 105$

- Initial stock price: $S_0 = 105$
- Strike price: $K = 100$
- Risk-free rate: $r = 0.05$
- Volatility: $\sigma = 0.30$
- Time to maturity: $T = 1.0$ year
- Number of time steps: $M = 50$
- Number of simulations: $N = 10{,}000$
- **American Put Option Price: 8.33**

Case 3: Strike Price $K = 105$

- Initial stock price: $S_0 = 100$
- Strike price: $K = 100$
- Risk-free rate: $r = 0.05$
- Volatility: $\sigma = 0.30$
- Time to maturity: $T = 1.0$ year

- Number of time steps: $M = 50$
- Number of simulations: $N = 10{,}000$
- **American Put Option Price: 12.87**

Case 4: Volatility $\sigma = 0.2$

- Initial stock price: $S_0 = 100$
- Strike price: $K = 100$
- Risk-free rate: $r = 0.05$
- Volatility: $\sigma = 0.20$
- Time to maturity: $T = 1.0$ year
- Number of time steps: $M = 50$
- Number of simulations: $N = 10{,}000$
- **American Put Option Price: 6.21**

Case 5: Stock Price $S = 120$

- Initial stock price: $S_0 = 120$
- Strike price: $K = 100$
- Risk-free rate: $r = 0.05$
- Volatility: $\sigma = 0.30$
- Time to maturity: $T = 1.0$ year
- Number of time steps: $M = 50$
- Number of simulations: $N = 10{,}000$
- **American Put Option Price: 4.44**

Chapter 11

Conclusions for the model

The models presented in the previous chapter represent a significant shift from traditional financial modeling methods towards a more advanced, data-driven approach using machine learning, to price American put options. In the previous section, the focus is on overcoming the limitations of classical pricing techniques, such as the Longstaff-Schwartz least-squares Monte Carlo (LSM) method, by incorporating the flexibility and learning capabilities of neural networks.

11.1 Traditional methods

The LSM method has been widely used in the financial industry to price American options, particularly due to its ability to handle the early exercise feature. However, the method's reliance on polynomial regression to estimate the continuation value often struggles with capturing complex, nonlinear relationships between the variables influencing the option's price, such as stock price, volatility, and time to maturity. While polynomial regression can provide a decent approximation, its inherent simplicity limits its ability to model intricate financial data patterns, especially in real-world markets where relationships between variables are seldom linear.

Another limitation of traditional approaches like LSM is their reliance on pre-specified basis functions, which restricts the model's ability to adapt to new data.

11.2 Machine Learning

The implementation of neural networks in this thesis offers a powerful alternative to classical methods. Neural networks are particularly well-suited for option pricing because of their ability to model complex, nonlinear relationships without requiring explicit feature engineering. Unlike traditional regression models, neural networks automatically learn the most relevant features and relationships from historical data during training, which allows them to better approximate the continuation value for American options.

Neural networks can capture subtle patterns in the data that classical methods may miss, leading to more accurate pricing models. For example, the continuation value, which represents the expected payoff of holding an option rather than exercising it, can be highly sensitive to small changes in the underlying variables. Neural networks, with their ability to process large amounts of data and recognize intricate patterns, can provide a more refined approximation of this value compared to polynomial-based methods used in LSM.

Additionally, the flexibility of neural networks means they can be easily extended to handle more complex derivative products or datasets with different characteristics. This adaptability is one of the key strengths of the machine learning approach: as financial markets evolve, neural networks can learn and adapt to new patterns in the data, offering a more robust and scalable solution for pricing American options.

11.3 Results

The thesis provides a comprehensive walkthrough of the implementation process, from data preparation to neural network architecture and training.

The design of the neural network itself, with its layers of interconnected neurons, allows the model to learn from the historical data of stock prices, strike prices, and other variables. The network is trained to minimize error in the predicted continuation values, which in turn improves the overall accuracy of the option pricing model.

Throughout the thesis, Python code is provided to demonstrate the step-by-step implementation of the neural network. The inclusion of Monte Carlo simulations for stock price paths and the use of backpropagation for optimizing the neural network's weights are key technical contributions.

The results of the neural network implementation show a marked improvement in accuracy over the classical LSM approach. The ability of the neural network to capture more complex relationships between variables results in more precise predictions of American option prices. Additionally, the neural network approach demonstrates greater computational efficiency in some cases, as it reduces the need for repeated regression fitting during the LSM simulation process.

11.4 Future Improvements

The shift towards machine learning in American option pricing offers several clear advantages. First, the accuracy of the pricing model improves due to the neural network's ability to handle nonlinearities and complex relationships between the input variables. Second, the scalability of the neural network model means it can be easily extended to more complex derivatives or larger datasets without requiring major modifications to the core architecture. Finally, the adaptability of the model allows it to learn from new data and adjust to changing market conditions, something that traditional models struggle with.

One of the most promising aspects of this approach is its potential for future research and application. Neural networks could be further optimized by experimenting with different architectures (e.g., deeper networks, recurrent neural networks), loss functions, or optimization techniques. Additionally, other machine learning techniques, such as reinforcement learning or ensemble methods, could be explored to further enhance the performance of option pricing models. There is also potential to apply this methodology to other types of derivatives, such as exotic options, where traditional pricing methods are even more limited.

In conclusion, the thesis successfully demonstrates that neural networks provide a viable and superior alternative to traditional LSM methods for pricing American put options. The combination of theoretical insights, practical Python implementations, and empirical results highlights the significant advantages of integrating machine learning into financial modeling. This work not only opens the door for further advancements in option pricing but also underscores the broader potential for machine learning in the field of quantitative finance.

Part IV

Conclusions

Chapter 12

Final Conclusion

In this work, we focused on implementing pricing models for financial derivatives with the ultimate goal of obtaining a modern approach to price American Put Options using machine learning. The focus has been on comparing traditional approaches like the Least-Squares Monte Carlo (LSM) method with newer machine learning models, specifically neural networks, to improve pricing accuracy and computational efficiency.

12.1 Overview of Classical Methods

In the initial chapters, we discussed classical option pricing models for vanilla options and exotic options, such as the Black-Scholes-Merton model and the Roll-Geske-Whaley model. The Black-Scholes model, though highly influential, is limited to European options, as it fails to account for the early exercise feature present in American options. To address this limitation, the LSM method, introduced by Longstaff and Schwartz [2001], was utilized to price American options by estimating the continuation value at each step using polynomial regression.

The LSM method has proven to be a robust approach, particularly for pricing American options where early exercise is a significant factor. However, it suffers from limitations in accurately modeling complex financial relationships and struggles when dealing with high-dimensional data. As such, we sought to improve upon this with modern machine learning techniques. A new approach using machine learning techniques can be seen in Djagba and Ndizihiwe [2024].

12.2 Neural Network Implementation for American Option Pricing

The latter part of this thesis presents a significant shift from classical financial models to a machine learning-driven approach, focusing on the pricing of American put options. We proposed a neural network-based model integrated into the LSM framework. The neural network was designed to approximate the continuation value, a key element in the pricing of American options, where decisions about early exercise are crucial.

The neural network was trained on a large dataset of American put options on Apple stock, with features including the stock price, strike price, volatility, and days to expiration. A multi-layered architecture with ReLU activation functions was employed to learn the complex, non-linear relationships between these variables and the continuation value. This implementation provided several advantages over classical methods:

- **Improved Accuracy**: The neural network's ability to learn from historical data allowed it to capture complex patterns in market dynamics that classical polynomial regression models in LSM could not. By considering additional variables, such as the risk-free rate and volatility, the neural network produced more accurate estimates of continuation values, resulting in more precise option pricing.
- **Flexibility and Scalability**: Unlike the classical LSM, which relies on pre-defined basis functions, the neural network can adapt to new data patterns without requiring manual adjustments. This makes it scalable to handle different types of financial instruments beyond American options, such as exotic options.
- **Computational Efficiency**: By integrating the neural network into the LSM algorithm, we reduced the need for repeated regression fitting during simulations. The network's ability to generalize from training data made it computationally more efficient, especially in scenarios involving high-dimensional datasets.

12.3 Empirical Results

The neural network demonstrated a higher accuracy in predicting option prices, with an R-squared value of 0.74, indicating that 74% of the variance in option prices was explained by the model. Additionally, the Mean Squared Error (MSE) of 350 further highlighted the model's effectiveness in generating realistic option prices. Even if some peaks in high values in the MSE while training the model can be seen from the orange line in Figure 10.1, these peaks represent a normal problem that arises while training neural networks with real-world data because the model tends to slightly overfit to the train data. Examples of this behavior can be seen frequently in the field of Reinforcement Learning.

Moreover, the neural network exhibited greater flexibility in adapting to various market conditions. Unlike classical regression models that fit each time step independently, the neural network leveraged historical data and market variables, which allowed it to generalize better to unseen market scenarios. This adaptability is crucial in real-world financial markets, where conditions can change rapidly.

12.4 Future Improvements

While the neural network-based model significantly improved the accuracy and efficiency of option pricing, there remains room for future research. Several potential enhancements could be explored:

- **Advanced Architectures**: Experimenting with deeper networks or recurrent neural networks (RNNs) may improve the model's ability to handle sequential data and time-dependent features, such as volatility clustering.
- **Reinforcement Learning**: Integrating reinforcement learning techniques could further enhance the model's ability to make optimal early exercise decisions by learning through exploration and exploitation of the option payoff landscape.
- **Generalization to Other Derivatives**: The methodology outlined in this thesis could be extended to other types of financial derivatives, such as barrier or exotic options, where traditional pricing methods face even greater limitations.
- **Handling Extreme Market Events**: Incorporating stress testing and training the model on datasets that include extreme market events, such as financial crises, could improve its robustness in unpredictable market conditions.

12.5 Conclusion

This thesis has demonstrated that machine learning, particularly neural networks, offers a powerful alternative to traditional option pricing models. By integrating neural networks into the LSM framework, we were able to address the limitations of classical methods, such as their inability to handle complex, non-linear relationships in financial data. The neural network model not only provided more accurate and efficient pricing for American put options but also showcased its potential to adapt to evolving market conditions.

The results of this study suggest that machine learning will play a crucial role in the future of financial modeling. As financial markets become more complex and data-driven, the ability of machine learning models to learn from vast datasets and adapt to new patterns will be indispensable for financial professionals. This work serves as a foundation for further exploration and research into the use of machine learning for derivative pricing and opens up various ways for research into more sophisticated and robust pricing models.

Bibliography

Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 1973.

Marek Capiński and Tomasz Zastawniak. *Mathematics for Finance: An Introduction to Financial Engineering*. Springer, 2011.

Prudence Djagba and Callixte Ndizihiwe. Pricing american options using machine learning algorithms, 2024.

Robert Geske and Richard Roll. On valuing american call options with the Black-Scholes european formula. *The Journal of Finance 39*, 1984. doi: https://doi.org/10.2307/2327870.

John Hull. *Options, Futures, and Other Derivatives (11th Edition)*. Pearson College Div, 2021.

Francis Longstaff and Eduardo Schwartz. Valuing american options by simulation: A simple least-squares approach. *Review of Financial Studies*, 14:113–47, 02 2001. doi: 10.1093/rfs/14.1.113.

Howard B. Sosin M. Barry Goldman and Mary Ann Gatto. Path dependent options: "buy at the low, sell at the high". *The Journal of Finance*, 1979.

Robert C. Merton. *Theory of Rational Option Pricing*. Legare Street Press, 2022.