Black and Scholes Model for Option Pricing and Greeks

Content:

- Implementation of Option Greeks using BSM (Python)
    - Using Approximation by small changes in inputs
    - Using Analytical formulas derived from BSM
    - Result comparison

    ➔ **The Code can be accessed at this link: Link**

What is an Option?

Defining Options as Financial Instruments

An option is a type of financial derivative that gives the holder the right, but not the obligation, to buy or sell an underlying asset at a predetermined price within a specified time frame. This financial instrument derives its value from the price of the underlying asset, which can be stocks, bonds, commodities, currencies, or market indexes. Options are primarily used for hedging risk or for speculative purposes, offering a way to capitalize on the future price movements of an asset.

Call Options and Put Options

There are two primary types of options: call options and put options.

- **Call Options** give the holder the right to buy the underlying asset. Purchasing a call option is typically a bullish move, where the buyer anticipates an increase in the asset's price. The buyer of a call option expects to profit when the price of the underlying asset rises above the strike price (the predetermined price) before the expiration date.
- **Put Options**, on the other hand, give the holder the right to sell the underlying asset. This is generally a bearish position, where the buyer predicts a decline in the asset's price. The buyer of a put option stands to gain when the price of the underlying asset falls below the strike price before the option expires.

Basic Principles of How Options Work

Options work based on three key principles:

1. **Strike Price**: This is the price at which the holder can buy (for a call option) or sell (for a put option) the underlying asset.
2. **Expiration Date**: Options are time-bound contracts. The expiration date is the final day the option holder can exercise their right to buy or sell the underlying asset.
3. **Premium**: This is the price paid by the buyer to the seller to acquire the rights conferred by the option. The premium is influenced by various factors including the underlying asset's price, time to expiration, volatility, and interest rates.

Different Ways to Value an Option

Valuing options accurately is crucial in financial markets. This section compares three prominent methods for option valuation: the Binomial Tree Model, the Black-Scholes Model (BSM), and the Monte Carlo Simulation, highlighting their advantages and disadvantages.

1. Binomial Tree Model

The Binomial Tree Model is a versatile method for option pricing that involves constructing a binomial tree to represent potential future prices of the underlying asset.

- **Advantages**:
  - **Flexibility**: Efficient in pricing American options and other complex derivatives.
  - **Intuitive Visualization**: Offers a straightforward visual representation of different price paths and outcomes.
- **Disadvantages**:
  - **Computational Intensity**: Requires significant computational power for a large number of time steps.
  - **Approximation Accuracy**: The accuracy depends on the number of time steps and the model's assumptions.

2. Black-Scholes Model (BSM)

The BSM is a mathematical model that provides a closed-form solution for pricing European options.

- **Advantages**:
  - **Efficiency**: Provides quick and easy calculations due to its closed-form solution.
  - **Industry Standard**: Widely accepted and used, especially for its simplicity and reliability under certain conditions.
- **Disadvantages**:
  - **Limiting Assumptions**: Assumes constant volatility and interest rates, which may not always be realistic.
  - **Restricted Applicability**: Primarily suitable for European options.

3. Monte Carlo Simulation

The Monte Carlo Simulation method involves using random sampling and statistical modeling to estimate the potential outcomes of an option's price.

- **Advantages**:

    - **Flexibility and Generality**: Can handle a wide range of options and complex path-dependent structures.
    - **Adaptability**: Capable of incorporating varying market conditions and assumptions.
- **Disadvantages**:
    - **Computational Intensity**: Requires a large number of simulations for accuracy, leading to high computational costs.
    - **Complexity**: More complex to implement and understand than the Binomial Tree Model or BSM.

Comparison

Each of these methods has unique strengths and limitations:

- **Flexibility**: The Binomial Tree Model and Monte Carlo Simulation offer greater flexibility, especially for American and exotic options.
- **Computational Resources**: BSM is less resource-intensive than the other two methods.
- **Accuracy and Complexity**: While BSM is known for its simplicity and efficiency, the Monte Carlo Simulation provides greater accuracy for complex, path-dependent options, albeit with increased computational and implementation complexity.

Introduction to the Black and Scholes Model (BSM)

The Black and Scholes Model (BSM), developed by economists Fischer Black and Myron Scholes in 1973, revolutionized the field of financial economics by providing a systematic method for valuing European-style options. Later, Robert Merton extended this model. This section provides an overview of the BSM and its significance in finance.

General Overview of the BSM

The Black and Scholes Model is a mathematical framework used for determining the fair price of options, specifically European call and put options. It calculates the option price by considering several factors, including the current price of the underlying asset, the option's strike price, the time to expiration, the risk-free interest rate, and the volatility of the underlying asset.

One of the key contributions of the BSM is its formulation of a theoretical estimate for the price of European-style options without the need for any dividend payments. The model assumes that markets are efficient, which means prices of securities reflect all known information. It also presupposes that the price movements of the underlying asset follow a lognormal distribution and that the volatility of the underlying asset is constant over the option's life.

Importance of BSM in Finance

The significance of the Black and Scholes Model in finance cannot be overstated:

1. **Pioneering Role in Financial Engineering**: BSM was one of the first models to apply advanced mathematics to the world of finance, paving the way for the field of financial engineering.

2. **Risk Management Tool**: The model provides a theoretical foundation for risk management strategies, particularly in the pricing and hedging of options.
3. **Market Efficiency**: BSM contributed to the understanding of market efficiency, influencing both academic research and practical trading strategies.
4. **Benchmark for Pricing**: It has become a benchmark in the financial industry for option pricing, against which other models and real market prices are compared.
5. **Innovation and Derivatives Pricing**: The model's simplicity and effectiveness have inspired further innovations in the pricing of more complex financial derivatives.

Factors Involved in the Black and Scholes Model

The Black and Scholes Model (BSM) is underpinned by several key inputs that determine the pricing of options. Understanding these factors is essential for applying the model effectively. This section outlines these inputs and discusses how they influence option pricing.

Key Inputs to BSM

1. **Underlying Asset Price (S)**: This is the current market price of the asset on which the option is based. The potential future value of the underlying asset is a primary determinant of an option's value.
2. **Strike Price (K)**: The strike price is the price at which the option holder can buy (call option) or sell (put option) the underlying asset. It serves as a reference point for determining whether exercising the option is profitable.
3. **Time to Expiration (T)**: This represents the time remaining until the option's expiration date. The more time an option has until expiration, the greater the chance it has to become profitable, affecting its premium.
4. **Volatility (σ)**: Volatility measures the degree of variation in the price of the underlying asset over time. Higher volatility increases the uncertainty of the asset's future price, which typically raises the option's value.
5. **Risk-Free Interest Rate (r)**: This is the theoretical rate of return of an investment with no risk of financial loss. It reflects the time value of money and influences the present value calculations in the BSM.

Factor Influences on Option Pricing

Each of these factors plays a critical role in the BSM and impacts the option's pricing in unique ways:

- **Underlying Asset Price**: A higher asset price for a call option increases its likelihood of profitability and, therefore, its premium. Conversely, for a put option, a lower asset price makes it more valuable.
- **Strike Price**: Options with lower strike prices for calls or higher strike prices for puts are typically more expensive, given their greater likelihood of being exercised profitably.
- **Time to Expiration**: Options with longer durations have higher premiums due to the increased uncertainty and potential for profitability over a more extended period.
- **Volatility**: Higher volatility leads to a wider range of potential prices for the underlying asset in the future, increasing the chance of an option ending in the money and hence its value.

- **Risk-Free Interest Rate**: An increase in the risk-free interest rate raises the cost of holding the underlying asset, increasing the value of call options and decreasing the value of put options.

$$C = N(d_1)S_t - N(d_2)Ke^{-rt}$$

$$\text{where } d_1 = \frac{\ln\frac{S_t}{K} + (r + \frac{\sigma^2}{2})t}{\sigma\sqrt{t}}$$

$$\text{and } d_2 = d_1 - \sigma\sqrt{t}$$

Implementation of BSM (Python)

→ **The Code can be accessed at this link: [Link]**

Taking inputs from the user:

```python
import math
import numpy as np
from scipy.stats import norm

def get_user_input():
    """
    Retrieves user input for parameters and ensures they are valid numbers and non-negative for S and T.
    """
    while True:
        try:
            S = float(input("Enter the current stock price (Spot Price): "))
            if S < 0:
                print("Invalid input: Spot Price cannot be negative. Please re-enter.")
                continue

            X = float(input("Enter the strike price (Strike Price): "))
            if X < 0:
                print("Invalid input: Strike Price cannot be negative. Please re-enter.")
                continue

            T = float(input("Enter the time to maturity (in years, Time to Maturity): "))
            if T < 0:
                print("Invalid input: Time to Maturity cannot be negative. Please re-enter.")
                continue

            sigma = float(input("Enter the volatility (Volatility, as a decimal): "))
            if sigma < 0:
                print("Invalid input: Volatility cannot be negative. Please re-enter.")
                continue

            r = float(input("Enter the risk-free interest rate (Interest Rate, as a decimal): "))
            if r < 0:
                print("Invalid input: Risk-free interest rate cannot be negative. Please re-enter.")
                continue

            return S, X, T, sigma, r
        except ValueError:
            print("Invalid input, please enter numbers.")
```

Result

```
••• Enter the current stock price (Spot Price):
```

Implementation BSM function to calculate price for Calls and Puts

```python
def black_scholes_call_put(S, X, T, r, sigma):
    """
    Calculates the Black-Scholes model prices for European call and put options.
    """
    d1 = (math.log(S / X) + (r + 0.5 * sigma ** 2) * T) / (sigma * math.sqrt(T))
    d2 = d1 - sigma * math.sqrt(T)

    call_price = S * norm.cdf(d1) - X * math.exp(-r * T) * norm.cdf(d2)
    put_price = X * math.exp(-r * T) * norm.cdf(-d2) - S * norm.cdf(-d1)
    return call_price, put_price
```

Result

```
Enter the current stock price (Spot Price): 100
Enter the strike price (Strike Price): 120
Enter the time to maturity (in years, Time to Maturity): 5
Enter the volatility (Volatility, as a decimal): 0.03
Enter the risk-free interest rate (Interest Rate, as a decimal): 0.07
The call option price is: 4.83
The put option price is: 8.12
```

Assumptions in the Black and Scholes Model (BSM)

The Black and Scholes Model (BSM) is predicated on several critical assumptions, which are essential for its application and interpretation. These assumptions are as follows:

1. **Efficient Markets**: The model assumes that markets are efficient, meaning that prices of the underlying assets reflect all available information.
2. **Lognormal Distribution of Stock Prices**: Stock prices are assumed to follow a lognormal distribution, implying that they cannot become negative and their movement is continuous.
3. **No Dividends**: BSM assumes that the underlying asset does not pay dividends during the life of the option.
4. **No Transaction Costs or Taxes**: The model assumes the absence of transaction costs and taxes, allowing for frictionless trading.
5. **Constant Risk-Free Interest Rate**: The risk-free rate is assumed to be constant over the option's life.
6. **Constant Volatility**: Volatility of the underlying asset is assumed to be constant and known over the option's life.
7. **European Options**: The original BSM is designed for European options, which can only be exercised at expiration.

Limitations in the Black and Scholes Model

While the BSM is a groundbreaking tool in financial economics, it has several limitations and has been subject to various criticisms.

Limitations and Criticisms of the Model

1. **Unrealistic Assumptions**: Many of the BSM's assumptions, such as constant volatility and no dividends, are often unrealistic in real-world markets.
2. **Inapplicability to American Options**: The model is not designed to accurately price American options, which can be exercised before the expiration date.
3. **Over-simplification of Market Conditions**: BSM oversimplifies market conditions by ignoring factors like liquidity, bid-ask spreads, and changing interest rates.
4. **Volatility Smile**: Empirical evidence shows a volatility smile pattern in markets, where implied volatility is not constant and varies with strike price and expiration, contrary to BSM's assumptions.

Situations Where BSM Might Not Be Accurate

1. **Dividend Paying Stocks**: For options on dividend-paying stocks, BSM might not provide accurate pricing as it does not account for dividends.
2. **High Volatility Environments**: In markets with high volatility or when the volatility of the underlying asset is not constant, the model's accuracy diminishes.
3. **Short-Dated Options**: BSM may not be as effective for short-dated options where the assumption of constant volatility is less likely to hold.
4. **Exotic Options**: For options with complex features like path-dependency, the model's straightforward approach may not capture the intricacies of such derivatives.

Defining Option Greeks

Option Greeks are financial measures that provide insights into how different factors affect the price of an option. These metrics are essential for traders and investors in managing risks and understanding the sensitivities in option pricing. The primary Greeks are Delta, Gamma, Theta, Vega, and Rho.

Defining Greeks: Delta, Gamma, Theta, Vega, and Rho

1. **Delta (Δ)**: Delta measures the rate of change in the option's price with respect to changes in the underlying asset's price. For call options, Delta ranges from 0 to 1, and for put options, it ranges from -1 to 0. A Delta of 0.5 means the option's price will move $0.50 for every $1 move in the underlying asset.
2. **Gamma (Γ)**: Gamma indicates the rate of change in Delta with respect to changes in the underlying asset's price. It shows the convexity of an option's value relative to the

underlying asset. High Gamma values suggest that Delta is highly sensitive to movements in the underlying asset's price.

3. **Theta (Θ)**: Theta measures the sensitivity of the option's price to the passage of time, often referred to as 'time decay.' It indicates how much an option's price decreases as it approaches its expiration date. Options lose value as time passes, and Theta quantifies this loss for each day.
4. **Vega (V)**: Vega measures the sensitivity of the option's price to changes in the volatility of the underlying asset. It indicates the amount by which the option's price will change with a 1% change in volatility. Vega is particularly important in times of market instability.
5. **Rho (ρ)**: Rho assesses the sensitivity of the option's price to changes in the risk-free interest rate. It shows how much the price of an option would rise or fall with a 1% change in the interest rate. Rho is more significant for options with a longer time to expiration.

Significance of Each Greek in Option Pricing and Risk Management

- **Delta**: Helps in hedging strategies by providing an estimate of the directional risk in an options portfolio.
- **Gamma**: Important for understanding the stability of an option's Delta. A high Gamma can indicate higher risk, as small movements in the underlying asset can result in significant changes in the option's price.
- **Theta**: Critical for option sellers who benefit from time decay, and buyers who face potential value loss over time.
- **Vega**: Essential for gauging the impact of market volatility. A high Vega indicates higher sensitivity to volatility, which can be a significant risk or opportunity.
- **Rho**: Useful in environments where interest rate changes are anticipated. Rho helps in understanding the impact of such changes on an option's price.

Implementation of Option Greeks using BSM (Python)

➔ **The Code can be accessed at this link: Link**

Using Approximation by Small Changes in Inputs

The approximation method for calculating the Greeks involves perturbing each of the input parameters slightly and observing the change in the option price. By comparing these small changes in the option price to the changes in the inputs, we can estimate the sensitivity of the option price to each input parameter, which corresponds to the Greeks.

For instance, to calculate the Delta using this method, we increase the stock price by a small amount and measure how much the option price changes. This gives us an approximation of how sensitive the option price is to the stock price. The same approach is used to calculate Gamma, Theta, Vega, and Rho, by varying the corresponding input parameters and measuring the change in the option price.

```python
def option_greeks_approx(S, X, T, r, sigma):
    base_call_price, base_put_price = black_scholes_call_put(S, X, T, sigma, r) # Calculate the base call and put prices

    dS = S * 0.01  # 1% of stock price
    dT = 1/365  # One day change for Theta
    dr = 0.0001  # 0.01% change for interest rate
    dsigma = 0.01  # 1% change for volatility

    # Delta Approximation
    call_delta = (black_scholes_call_put(S + dS, X, T, sigma, r)[0] - base_call_price) / dS
    put_delta = (black_scholes_call_put(S + dS, X, T, sigma, r)[1] - base_put_price) / dS

    # Gamma Approximation (requires a second order approximation)
    call_gamma = (black_scholes_call_put(S + dS, X, T, sigma, r)[0] - 2 * base_call_price + black_scholes_call_put(S - dS, X, T, sigma, r)[0]) / (dS ** 2)
    put_gamma = (black_scholes_call_put(S + dS, X, T, sigma, r)[1] - 2 * base_put_price + black_scholes_call_put(S - dS, X, T, sigma, r)[1]) / (dS ** 2)

    # Theta Approximation
    call_theta = (black_scholes_call_put(S, X, T - dT, sigma, r)[0] - base_call_price) / dT
    put_theta = (black_scholes_call_put(S, X, T - dT, sigma, r)[1] - base_put_price) / dT

    # Vega Approximation
    call_vega = (black_scholes_call_put(S, X, T, sigma + dsigma, r)[0] - base_call_price) / dsigma
    put_vega = (black_scholes_call_put(S, X, T, sigma + dsigma, r)[1] - base_put_price) / dsigma

    # Rho Approximation
    call_rho = (black_scholes_call_put(S, X, T, sigma, r + dr)[0] - base_call_price) / dr
    put_rho = (black_scholes_call_put(S, X, T, sigma, r + dr)[1] - base_put_price) / dr

    return {
        "call_delta": call_delta, "call_gamma": call_gamma, "call_theta": call_theta,
        "call_vega": call_vega, "call_rho": call_rho,
        "put_delta": put_delta, "put_gamma": put_gamma, "put_theta": put_theta,
        "put_vega": put_vega, "put_rho": put_rho
    }
```

Using Analytical Formulas Derived from BSM

The analytical method for calculating the Greeks uses the closed-form formulas derived from the Black-Scholes model. These formulas take into account the derivatives of the option pricing formula with respect to each input parameter. The analytical Greeks give us a precise measure of the option's sensitivities and are typically more accurate than the approximation method, assuming that all the Black-Scholes assumptions hold true.

The Greeks calculated using the analytical method are exact for infinitesimal changes in the input parameters, whereas the approximation method provides a numerical estimate that is dependent on the size of the input change.

```python
[13] def option_greeks(S, X, T, r, sigma):
         d1 = (math.log(S / X) + (r + 0.5 * sigma ** 2) * T) / (sigma * math.sqrt(T))
         d2 = d1 - sigma * math.sqrt(T)

         # Call option Greeks
         call_delta = norm.cdf(d1)
         call_gamma = norm.pdf(d1) / (S * sigma * math.sqrt(T))
         call_theta = (-S * norm.pdf(d1) * sigma / (2 * math.sqrt(T))) - (r * X * math.exp(-r * T) * norm.cdf(d2))
         call_vega = S * norm.pdf(d1) * math.sqrt(T)
         call_rho = X * T * math.exp(-r * T) * norm.cdf(d2)

         # Put option Greeks
         put_delta = -norm.cdf(-d1)
         put_gamma = call_gamma  # Gamma is the same for calls and puts
         put_theta = (-S * norm.pdf(d1) * sigma / (2 * math.sqrt(T))) + (r * X * math.exp(-r * T) * norm.cdf(-d2))
         put_vega = call_vega  # Vega is the same for calls and puts
         put_rho = -X * T * math.exp(-r * T) * norm.cdf(-d2)

         return {
             "call_delta": call_delta, "call_gamma": call_gamma, "call_theta": call_theta / 365,
             "call_vega": call_vega / 100, "call_rho": call_rho / 100,
             "put_delta": put_delta, "put_gamma": put_gamma, "put_theta": put_theta / 365,
             "put_vega": put_vega / 100, "put_rho": put_rho / 100
         }

     greeks = option_greeks(*inputs)
```

Result Comparison

The comparison of the two methods reveals discrepancies in the calculated values of the Greeks. These discrepancies arise from the fact that the approximation method is just that – an approximation. It is based on finite changes in the input parameters and thus can only provide an estimate of the Greeks. The analytical method, on the other hand, assumes infinitesimal changes and provides a precise calculation.

From the results obtained, it is clear that the analytical method is preferable when precision is required. However, the approximation method has its uses, particularly in scenarios where the analytical formulas are not readily available or when dealing with numerical models that cannot be solved analytically.

The visualization of the comparison (as provided in the uploaded images) illustrates the differences between the two methods. The bar charts highlight the magnitude of the Greeks as computed by each method, while the percentage error chart underscores the relative differences, which can be quite significant in certain cases. These visualizations underscore the importance of using the analytical method for accurate Greek calculations.