Option Pricing and Greeks using Black-Scholes Merton model

Introduction

The Black-Scholes-Merton (BSM) option pricing model is a mathematical model for valuing European call and put options. The model assumes that the underlying asset (such as a stock) follows a geometric Brownian motion process, which means that its logarithmic returns are normally distributed with a constant drift and volatility. The BSM model can be used to calculate the theoretical price of a European call or put option based on its input parameters, such as the spot price of the underlying asset, the strike price of the option, the risk-free interest rate, the time to maturity, and the volatility of the underlying asset.

The "greeks" of an option are sensitivities of the option's value to various factors. The most used greeks are:

- Delta: The sensitivity of the option's value to a small change in the spot price of the underlying asset.
- Gamma: The sensitivity of the option's delta to a small change in the spot price of the underlying asset.
- Vega: The sensitivity of the option's value to a small change in the volatility of the underlying asset.
- Theta: The sensitivity of the option's value to a small change in the time to maturity.
- Rho: The sensitivity of the option's value to a small change in the risk-free interest rate.

The greeks are useful for hedging purposes, as they can help traders determine how to adjust their positions in the underlying asset and the option to minimize their exposure to changes in the underlying market conditions.

The Black-Scholes-Merton (BSM) option pricing model uses the following formulas to calculate the theoretical price of a European call or put option:

- Call option price
  C = S * N(d1) - K * exp(-r * t) * N(d2)

- Put option price
  P = K * exp(-r * t) * N(-d2) - S * N(-d1)

Formulas for Greeks:

- Delta (call) = N(d1)
- Gamma = N'(d1) / (S * sigma * sqrt(t))
- Vega = S * sqrt(t) * N'(d1)
- Theta = -(S * N'(d1) * sigma / (2 * sqrt(t))) - (r * K * exp(-r * t) * N(d2))
- Rho (call) = K * t * exp(-r * t) * N(d2)

Rho (put) = -K * t * exp(-r * t) * N(-d2)

Where,

- C is the call option price
- S is the spot price of the underlying asset
- K is the strike price of the option
- r is the risk-free interest rate
- t is the time to maturity of the option
- N(x) is the standard normal cumulative distribution function

d1 and d2 are defined as follows:

d1 = (ln(S/K) + (r + sigma^2 / 2) * t) / (sigma * sqrt(t))
d2 = d1 - sigma * sqrt(t)

1. **BSM_model.cpp**
   The Option class is a representation of an option in a financial context. The class has several public methods for accessing various properties of the option such as the price, time to maturity, and risk-free rate. It also has methods for calculating and returning the option's "greeks" such as delta, gamma, vega, rho, and theta. These are all measures of the sensitivity of the option's price to changes in various underlying parameters such as the price of the underlying asset and the option's volatility.
   The Option class also has several private member variables, including spot and strike which represent the current price of the underlying asset and the option's strike price respectively, as well as rate, time, and price which represent the option's risk-free rate, time to maturity, and price. The class also has a private vol variable which represents the option's volatility.
   The Option class' constructor initializes these member variables using the parameters passed to it. It also uses the Integration class to calculate the option's price using the Black-Scholes model, depending on whether the option is a call or a put option and whether the user has input the option's price or its volatility. The call_price and put_price methods use the Black-Scholes formula to calculate the price of a call or put option, respectively. The volatility method uses an iterative method to calculate the option's implied volatility given its price, underlying asset price, strike price, risk-free rate, and time to maturity.
   The vega_for_vol method calculates the vega of the option given the same inputs as the volatility method. The calculate_delta, calculate_gamma, calculate_vega, calculate_rho, and calculate_theta methods are used to calculate the option's delta, gamma, vega, rho, and theta, respectively. These methods use the Black-Scholes formula to calculate the values of the

respective "greeks" measures. The option_type method returns a string representation of whether the option is a call or a put option.

2. **BSM_model.h**
   This is the class definition of the Option class. It defines the member variables and member functions of the class. The class has several public methods for accessing various properties of the option such as the price, time to maturity, and risk-free rate. It also has methods for calculating and returning the option's "greeks" such as delta, gamma, vega, rho, and theta. These are all measures of the sensitivity of the option's price to changes in various underlying parameters such as the price of the underlying asset and the option's volatility. The Option class also has several private member variables, including spot and strike which represent the current price of the underlying asset and the option's strike price respectively, as well as rate, time, and price which represent the option's risk-free rate, time to maturity, and price. The class also has a private vol variable which represents the option's volatility.

3. **Integration.h**
   The Integration class is a simple class that calculates the definite integral of a function between two bounds (x_min and x_max) using the rectangle rule. The class has a public integral method which takes in the number of steps (rectangles) to use in the calculation and returns the result. It also has a constructor that takes in the bounds x_min and x_max and initializes the corresponding member variables. The func method is a private method that defines the function that will be integrated. The default number of steps used in the calculation is 5000.

4. **Integration.cpp**
   The integral function takes an integer argument width, which represents the number of rectangles used to approximate the area under the curve of the func function. The func function itself appears to be a mathematical formula for the standard normal distribution.

5. **Main.cpp**
   The code prompts the user to input the values of various parameters of the option, such as the spot price, strike price, risk-free interest rate, and time to maturity, and uses these values to create an instance of the Option class. The Option class is then used to calculate either the option's price and to output a summary of the option's input parameters and the option's "greeks" (sensitivities of the option's value to various factors). The code uses a while loop to repeatedly prompt the user for input and output the requested information until the user decides to exit the program.

Future work

Implementation of code to find out implied volatility when the spot, strike, risk free rate, time to maturity and price of the option are known