Pullak Parag Khetan
RU Id: 189007407

Project Objective: To estimate the implied volatility in the stock market for a particular stock by inverting the Black-Scholes model and plotting the volatility surface

Description: Estimating a call option price using Black-Scholes model makes use of historical volatility. This doesn't reflect the prevailing sentiment of the market. By using a current market call option price for a particular stock, we can gauge the implied volatility by inverting the Black-Scholes model. This estimate of implied volatility can aid in determining the future trajectory of the stock price and also overall market sentiment

Files:
Main program: ImpliedVolatility.cpp
Header file: BlackScholes.h
Graph plotter: ImpliedVolatility.py

Method used: Newton-Raphson approximation

In general, approximation using Newton-Raphson method is as follows

$$x_{n+1} = ((y - g(x_n))/g'(x_n)) + x_n$$

For Black Scholes method, this equation will transform to

$$\sigma_{n+1} = ((C_t - B(\sigma_n))/vega) + \sigma_n$$

where C_t=current European Call option price in the market
$B(\sigma_n)$ = Black-Scholes model price given the volatility
Vega=partial derivative of the Black-Scholes model w.r.t. to the volatility

Inputs:

- Spot price
- Risk-free interest rate
- Maturity
- Strike Price
- Historical volatility
- Current Market price

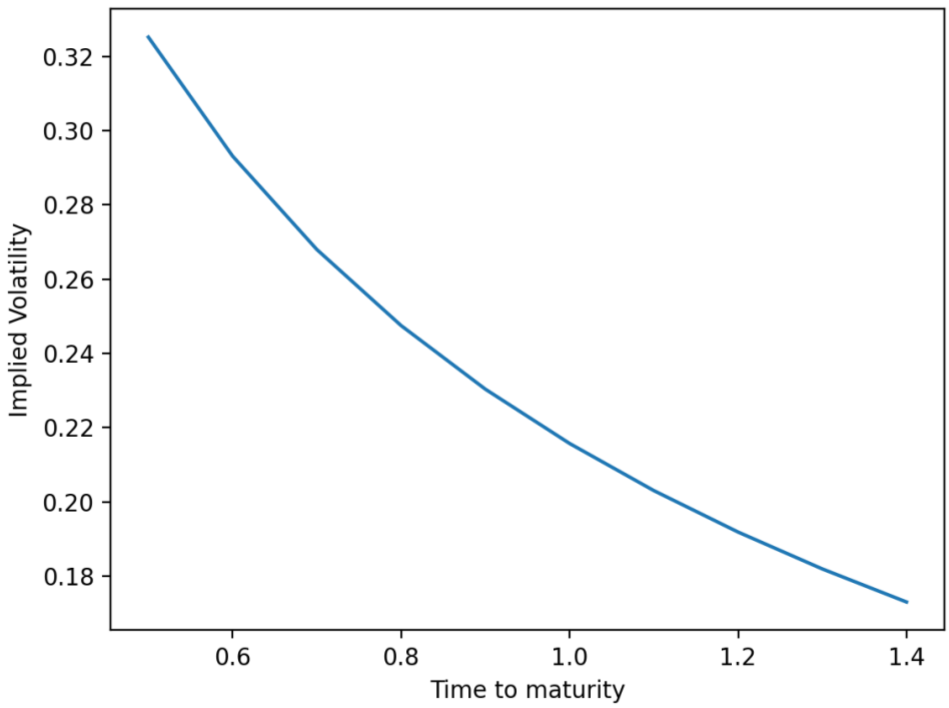
Methods:

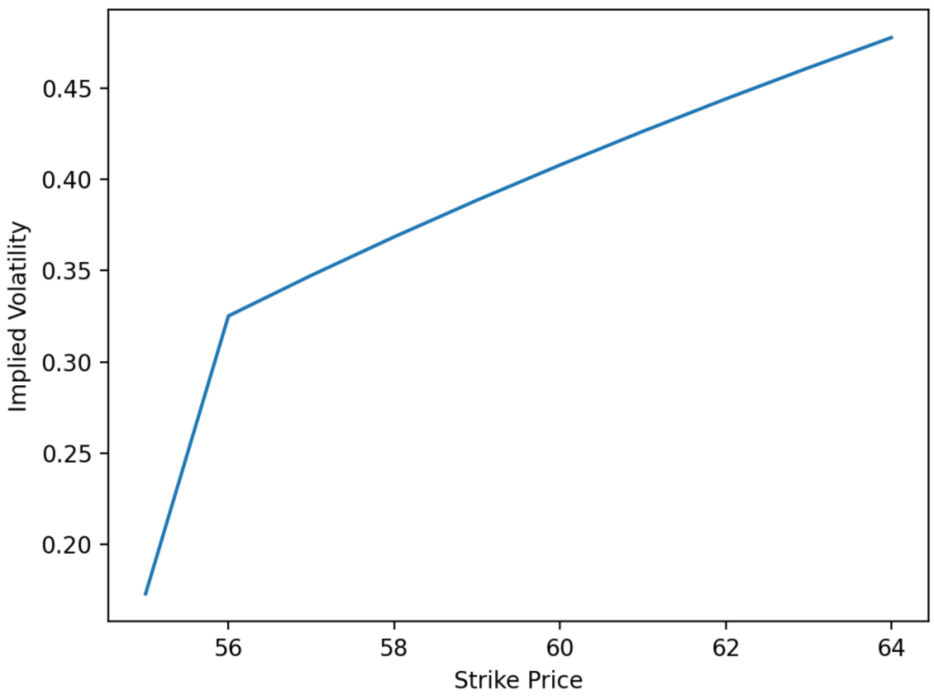
1. Calc_price(): Generates Black-Scholes model price
2. Vega(): Approximates partial derivative of Black-Scholes model w.r.t the volatility
3. Inverse_volatility_maturity: Generates implied volatility values for different maturities
4. Inverse_volatility_strike(): Generates implied volatility values for different strike price

Libraries used:

1. <iostream>
2. <array>
3. <cmath>
4. <string>
5. <iomanip>
6. matplotlib (in python)
7. sys (in python)

Output:




Implied Volatility
0.45
0.40
0.35
0.30
0.25
0.20
56
58
60
62
64
Strike Price