Interim Report SysCom - Naman Kedia

Project Purpose:

This project focused on option pricing and hedging strategies, particularly implementing the Black-Scholes-Merton (BSM) model and delta hedging. The project is divided into two main parts:

1. Randomly simulating stock prices and calculating option prices and hedging errors. The stock prices, call prices and their cumulative hedging errors distributions were observed
2. Using real market data of Google Stock to calculate implied volatilities and perform delta hedging.

Important Functions and Classes:

Code Structure/Files -

- Headers - option.h, stock.h, option_pricing.h, market_data.h
- Code files - main.cpp, option.cpp, stock.cpp, option_pricing.cpp, market_data.cpp
- Unit Test files - pricer_test.cpp, pricer_test.h
- Plotting - plots.py (matplotlib used)
- Outputs
    - stock_simulations_plot.png (*For 100 stock samples)*
    - optionPrices_simulation.png *(For 100 option prices corresponding to the sampled stocks)*
    - Hedging_error_distribution.png *(For HE distribution)*
    - Results.csv *(For tabular output with HE, PnL, PnL with HE (for dates (07/05/2011-07/29/2011)))*

Function Explanations -

1. Option Class (option.h):

- Represents an option contract with properties like strike price, initial stock price, risk-free rate, time to maturity, and volatility.
- Provides getters and setters for these properties.

2. Option_Price Class (option_pricing.h):

- Implements the core option pricing and hedging functionalities.
- Key functions:
    - BSMPricer: Calculates option price and delta using the Black-Scholes-Merton model.
    - deltaHedgingErrors: Computes hedging errors for a given stock price path.
    - calculateImpliedVolatility: Determines the implied volatility from observed market prices using binary search.
    - deltaHedgingErrorsWithImpliedVol: Computes hedging errors for a given stock price, option price path by calculating implied volatility.
    - normalCDF: Calculates the cumulative distribution function for the standard normal distribution.

3. MarketData Class (market_data.h):

- Handles reading and storing market data from CSV files.
- Provides functions to retrieve interest rates, stock prices, and option prices for specific dates.
- Includes utility functions like getDatesBetween and getTimeToMaturity for calculating relevant business days.

4. Stock Class (stock.h):

- Simulates stock price paths.

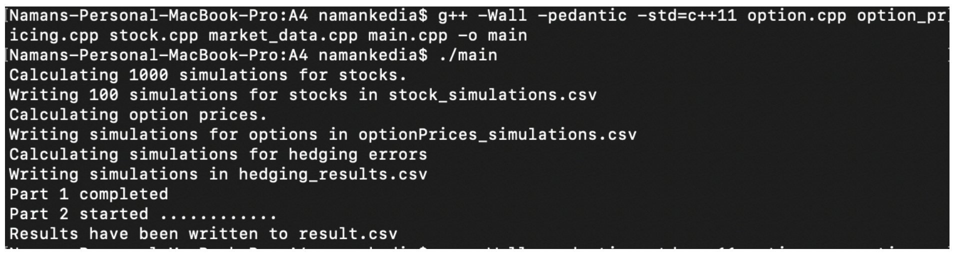
5. Main Function (main.cpp):

- Orchestrates the entire process, divided into two parts:
    - Simulates stock prices, calculates option prices and hedging errors, and writes results to CSV files.
    - Uses real market data to perform delta hedging with implied volatilities.

Interim Report SysCom - Naman Kedia

Output and Analysis

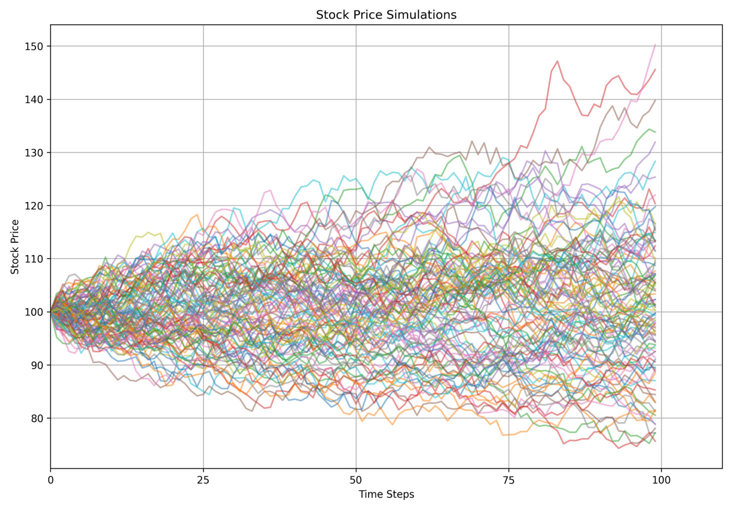
Full output (main.cpp)



```
[Namans-Personal-MacBook-Pro:A4 namankedia$ g++ -Wall -pedantic -std=c++11 option.cpp option_pr
icing.cpp stock.cpp market_data.cpp main.cpp -o main
[Namans-Personal-MacBook-Pro:A4 namankedia$ ./main
Calculating 1000 simulations for stocks.
Writing 100 simulations for stocks in stock_simulations.csv
Calculating option prices.
Writing simulations for options in optionPrices_simulations.csv
Calculating simulations for hedging errors
Writing simulations in hedging_results.csv
Part 1 completed
Part 2 started ...........
Results have been written to result.csv
```

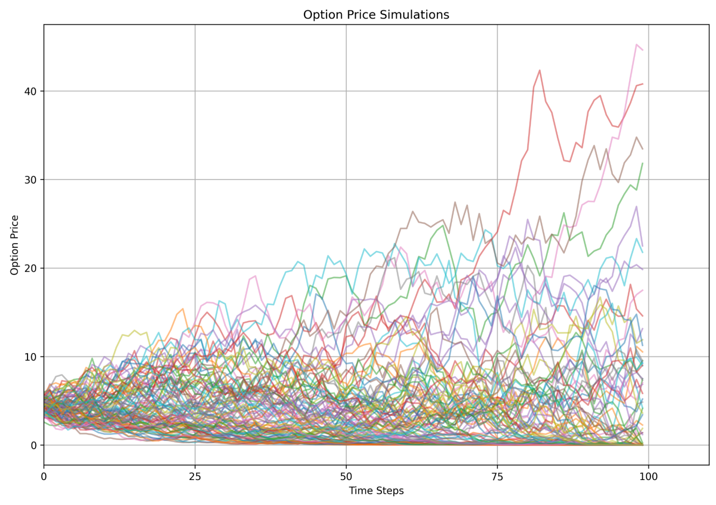
Part 1

Stock simulations were done for 1000 runs. <u>A sample of 100 were plotted as below:</u>



Analysis: *All stocks start from 100. The maximum density of stock paths end up near the start(but slightly higher) due to Z(t) being normally distributed centred around 0 and 1 std dev. Overall marginal upward trend is due to the positive drift (μ = 0.05). The random component (Z(t)) introduces volatility. Some paths will go up more than others, some might even go down despite the positive drift.*

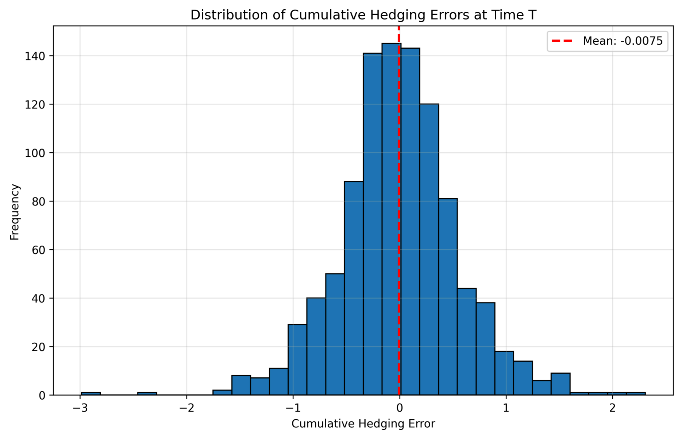
<u>Call prices for the simulations:</u>

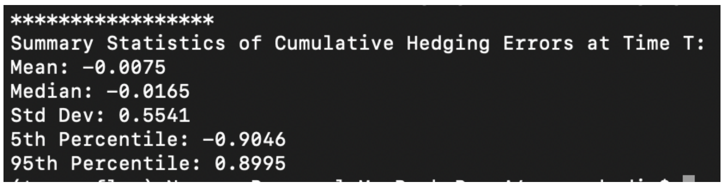


Interim Report SysCom - Naman Kedia

***Analysis:** All options prices start from approx $5 from step 0. The maximum density of stock paths end up near 0 as most stock prices end up back to 100 because For K=105 these options prices will go to 0. The highest call prices (~45) correspond to the outlier stocks that went high up to 150. Call option prices will generally move in the same direction as the stock price. As the stock price increases, the call option becomes more valuable. As we get closer to the expiration date, the time value of the option will decrease. This effect (known as theta) is usually pronounced for at-the-money options.*

<u>Hedging error distribution:</u>



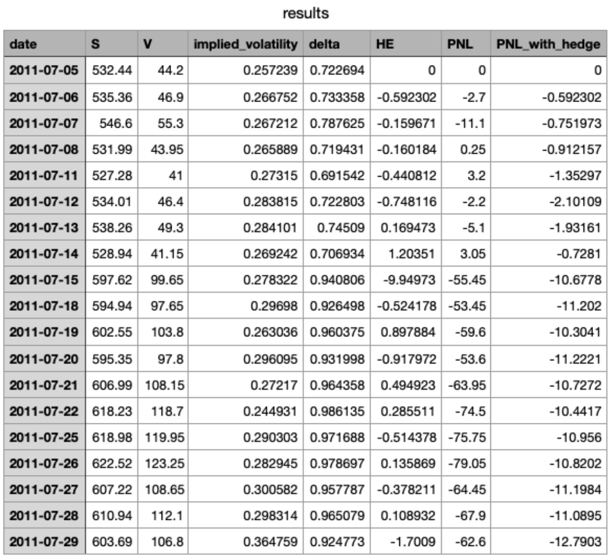


```
*****************
Summary Statistics of Cumulative Hedging Errors at Time T:
Mean: -0.0075
Median: -0.0165
Std Dev: 0.5541
5th Percentile: -0.9046
95th Percentile: 0.8995
```

***Analysis:** Due to the Central Limit Theorem and the assumption of normally distributed stock returns in the Black-Scholes model, the distribution of hedging errors approximate a normal distribution. In an ideal scenario, the mean of the hedging errors should be very close to zero. This is because delta hedging aims to perfectly replicate the option's value, so on average, the errors should cancel out. This is also a result that we can confirm from the plots and stats.*

Part 2

Results.csv has the output for Part 2.

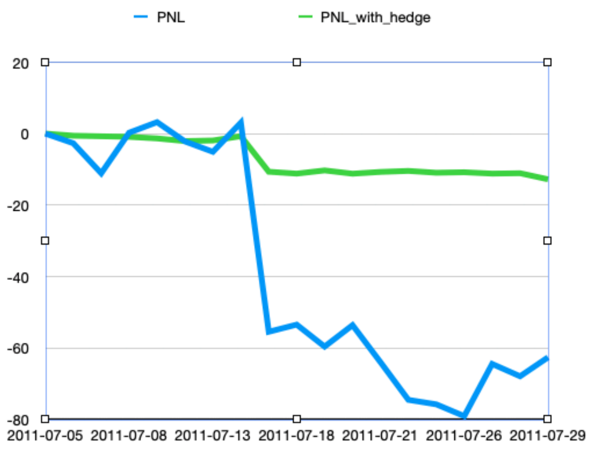


results

date	S	V	implied_volatility	delta	HE	PNL	PNL_with_hedge
2011-07-05	532.44	44.2	0.257239	0.722694	0	0	0
2011-07-06	535.36	46.9	0.266752	0.733358	-0.592302	-2.7	-0.592302
2011-07-07	546.6	55.3	0.267212	0.787625	-0.159671	-11.1	-0.751973
2011-07-08	531.99	43.95	0.265889	0.719431	-0.160184	0.25	-0.912157
2011-07-11	527.28	41	0.27315	0.691542	-0.440812	3.2	-1.35297
2011-07-12	534.01	46.4	0.283815	0.722803	-0.748116	-2.2	-2.10109
2011-07-13	538.26	49.3	0.284101	0.74509	0.169473	-5.1	-1.93161
2011-07-14	528.94	41.15	0.269242	0.706934	1.20351	3.05	-0.7281
2011-07-15	597.62	99.65	0.278322	0.940806	-9.94973	-55.45	-10.6778
2011-07-18	594.94	97.65	0.29698	0.926498	-0.524178	-53.45	-11.202
2011-07-19	602.55	103.8	0.263036	0.960375	0.897884	-59.6	-10.3041
2011-07-20	595.35	97.8	0.296095	0.931998	-0.917972	-53.6	-11.2221
2011-07-21	606.99	108.15	0.27217	0.964358	0.494923	-63.95	-10.7272
2011-07-22	618.23	118.7	0.244931	0.986135	0.285511	-74.5	-10.4417
2011-07-25	618.98	119.95	0.290303	0.971688	-0.514378	-75.75	-10.956
2011-07-26	622.52	123.25	0.282945	0.978697	0.135869	-79.05	-10.8202
2011-07-27	607.22	108.65	0.300582	0.957787	-0.378211	-64.45	-11.1984
2011-07-28	610.94	112.1	0.298314	0.965079	0.108932	-67.9	-11.0895
2011-07-29	603.69	106.8	0.364759	0.924773	-1.7009	-62.6	-12.7903

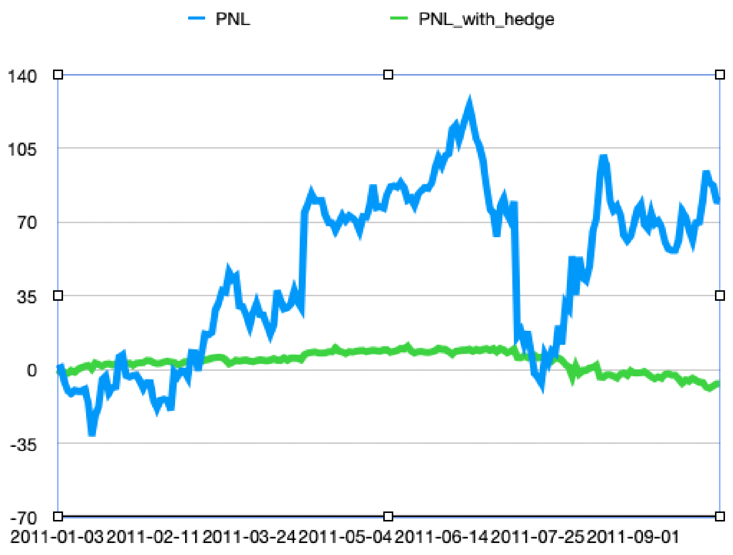
Interim Report SysCom - Naman Kedia

We can notice that the delta hedging strategy is a more stable PnL strategy where both our losses and profits are more stabilised than a simple hold PnL strategy.



- The longer our period of holding is, the more stability we get. Since the given default values of time period were shorter, we can see a spike but overall the green line has a better PnL curve ie. It's more positive than PnL normal at most instances.
- The minor dip is also due to a sudden spike in the stock which is unprecedented usually and not something a hedging strategy would optimise for. We can still notice that the normal PnL got hit much worse and went up to -60 as opposed to the delta hedging only going till -10.

Below is a graph which shows the delta strategy over a longer duration of Google Option with Strike 360.



We can see that PnL(with hedge) is a more stable strategy that is not as affected by major volatility jumps unlike PnL. Even though the profits are higher for PnL this is only because google stock performed better than normal market stocks. In an uncertain market PnL(with hedge) is a better hedging strategy. We can also see that the Hedging PnL is close to mean PnL 0 across time as we had also shown in Part 1.

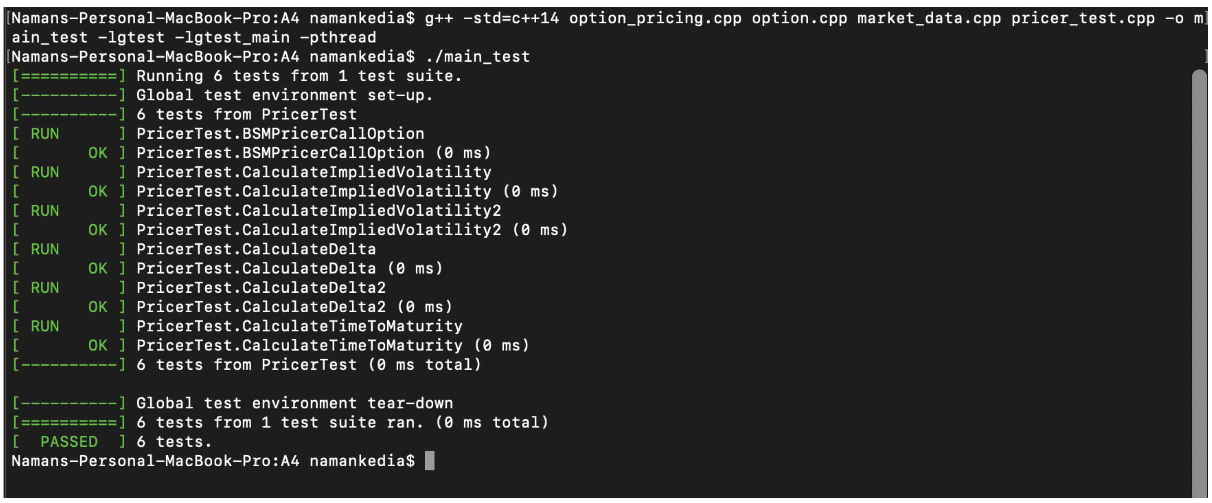
Conclusion

The PnL(with hedge) is highly likely to show significantly **less volatility** compared to the PnL(normal). This is evident from the hedging graphs above. The hedging strategy effectively **mitigates large swings in profit or loss**. While the PnL(normal) strategy would fully benefit from favourable stock price movements (as seen in the graph 2 in Part 2), it would also suffer from unfavourable movements. The hedged strategy, in contrast, limits both potential gains and losses. The hedged strategy is likely to provide **more consistent performance over time.**

Unit Tests

A total of 6 unit tests are covered. 2 for implied volatility, 2 for delta calculation, 1 for BSM pricer and 1 for Maturity Time calculator.

Output of ./main_test



```
[Namans-Personal-MacBook-Pro:A4 namankedia$ g++ -std=c++14 option_pricing.cpp option.cpp market_data.cpp pricer_test.cpp -o m
 ain_test -lgtest -lgtest_main -pthread
[Namans-Personal-MacBook-Pro:A4 namankedia$ ./main_test
[==========] Running 6 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 6 tests from PricerTest
[ RUN      ] PricerTest.BSMPricerCallOption
[       OK ] PricerTest.BSMPricerCallOption (0 ms)
[ RUN      ] PricerTest.CalculateImpliedVolatility
[       OK ] PricerTest.CalculateImpliedVolatility (0 ms)
[ RUN      ] PricerTest.CalculateImpliedVolatility2
[       OK ] PricerTest.CalculateImpliedVolatility2 (0 ms)
[ RUN      ] PricerTest.CalculateDelta
[       OK ] PricerTest.CalculateDelta (0 ms)
[ RUN      ] PricerTest.CalculateDelta2
[       OK ] PricerTest.CalculateDelta2 (0 ms)
[ RUN      ] PricerTest.CalculateTimeToMaturity
[       OK ] PricerTest.CalculateTimeToMaturity (0 ms)
[----------] 6 tests from PricerTest (0 ms total)

[----------] Global test environment tear-down
[==========] 6 tests from 1 test suite ran. (0 ms total)
[  PASSED  ] 6 tests.
Namans-Personal-MacBook-Pro:A4 namankedia$
```

Other Details:

As directed in the project, the function takes the default values and outputs the results.csv for it. (K= 500 startDate: 2011-07-05 enddate: 2011-07-29 expiryDate: 2011-09-17)

User inputs are handled but disabled by default as they were not expected to be done in the assignment according to instructions. (To enable user inputs: in main.cpp switch the while to true for line 81 and while to false for line 120)

File Format

The expected format of the CSV file is as follows:

- All 3 data files must be in a folder called “data”

Error Handling

- The program checks if the date is in the correct format (YYYY-MM) and ensures that the date exists in the dataset.
- If the input format is incorrect or the date does not exist, the program will terminate due to date not being present in the files structure