DESIGN OF SYSTEMS FOR COMPUTATIONAL FINANCE

ISYE-6767

MIDTERM PROJECT

IMPLEMENTING A DYNAMIC DELTA HEDGING STRATEGY

NAME: ROHAN THANKI

GT ID: 903630513

DATE: 18-Oct-2021

PROBLEM ADDRESSED BY THE PROJECT

- The project introduces the concept of delta-hedging which is a method of forming a hedge in our portfolio after writing an option.
- Writing options is done many a time to earn the premium. Options have negative Theta, i.e., the time value of the options decays to 0 as the option approaches maturity. Thus, if the trader expects the underlying asset to not move too much, writing options is sometimes used for income.
- This is also important for brokers and dealers who must hedge their positions after selling options to their clients.
- Thus delta hedging is a strategy used to hedge a short position in an option.
- The project introduces “dynamic” delta hedging. Basically, as the stock price changes, the price of the underlying option changes and our portfolio is no longer “delta-hedged. Thus, we must “rebalance” our stock holdings to match the new delta. Thus with a short call, we must buy stock to hedge.
- For a call option, delta varies between 0 and 1. As the stock price increases, the delta increases.
- The delta of a put option varies between -1 and 0. As the stock price decrease, the delta becomes more negative. Thus with a short put, we must short stock to hedge.

MODEL(S) USED IN THE PROJECT – PART 1

- We simulate options prices using a Geometric Brownian Motion stochastic process. A Geometric Brownian Motion stochastic process is a continuous-time stochastic process in which the logarithm of the random variable varying quantity follows a Brownian motion (also called a Wiener process) with drift.
- We create 1000 sample paths of the stock price where each path is characterized by the following parameters:
  - Initial stock price = 100
  - Time horizon = 0.4 years
  - Number of time instants = 100
  - Thus, size of timestep = 0.004
  - Risk free interest rate = 2.5%
  - Drift parameter = 0.05
  - Diffusion parameter = 0.24
- For each time instant, we compute the following:
  - The option price using the Black-Scholes Option Pricing Model
  - The Option Delta, for which we get an analytical form from the Black-Scholes model.
- The Hedging Error - this is the profit or loss of our portfolio is we sell a European Call option and buy delta*(number of shares). At each time instant, the delta changes based on how the underlying stock price has moved. We then "rebalance" our portfolio by changing our holdings of the underlying stock – this portfolio is called a Delta-Neutral portfolio.

MODEL(S) USED IN THE PROJECT – PART 2

- The same implementation is done for options on the Google Stock, except we have the following differences from part 1:
  - Instead of generating a series of stock prices, we read the GOOG stock prices from a csv file
  - Instead of computing the option price, we read the price of the GOOG options from a file
  - The risk-free rate changes on every date instead of it being fixed
  - Instead of using a static volatility parameter, we back-track the "implied volatility" from the Black Scholes model using an "Interval Bisection" approach"
  - Hedging error is computed in the same way as part 1
  - Additionally, we also compute the PNL (Profit & Loss) for a naked short call position and for a hedged portfolio.

THE STRUCTURE OF THE MODEL IMPLEMENTATION: FUNCTIONALITY

CLASSES

SR. NO	FILENAME	DESCRIPTION
1	Option.hpp	Contains the definition of the Option class
2	Option.cpp	Contains the implementation of methods in the Option class
3	Hedging_Portfolio.hpp	Contains the definition of the Hedging_Portfolio class
4	Hedging_Portfolio.cpp	Contains the implementation of methods in the Hedging_Portfolio class

Option Class

- This is a base class which contains data of an option

Private Member Variables	flag	To denote whether call (C/c) or put option (P/p)
	K	strike price
	S	spot price
	r	Annualized risk-free rate
	T	time to maturity
	vol	Annualized volatility
Public methods	Default constructor	Constructor and Destructors
	Parameterized Constructor	
	Copy Constructor	
	Destructor	
	getFlag	Getters for each private member variable
	getStrikePrice	
	getSpotPrice	
	getRiskFreeRate	
	getTimeToMaturity	
	getVolatility	
	getFlag	Setters for each private member variable
	getStrikePrice	
	getSpotPrice	
	getRiskFreeRate	
	getTimeToMaturity	
	getVolatility	

Hedging_Portfolio Class

- This class is derived from the Option class
- It contains additional member variables and methods

Private member variables	delta	Delta of the option
	optionPrice	Price of the option - either computed or taken from market data
	B	Used to compute hedging error
	HE	Hedging error of a delta-hedged portfolio
	date	Date of pricing the option in string
	expDate	Option expiry Date in string
	pnlNaked	Profit and loss of a naked short call
	pnlHedged	Profit and loss of a delta hedged portfolio
Public methods	getDelta	Getters
	getOptionPrice	
	getB	
	getHedgingError	
	getDate	
	getExpDate	
	getpnlNaked	
	getpnlHedged	
	setDelta	Setters
	setOptionPrice	
	setB	
	setHedgingError	
	setDate	
	setExpDate	
	setpnlNaked	
	setpnlHedged	
	computeDelta	This method computes the delta of the option
	computeBlackScholesOptionPrice	Computes the option price using the Black-Scholes model
	computeImpliedVol	Computes the implied volatility using an interval bisection approach

FILES

1. **Utilities.cpp**
2. **Project_Parts.cpp**
3. **Main.cpp**

FUNCTIONS IN FILES

Utilities.cpp

Type of Function	Function	Description
General Utility Functions	string_splitter	Splits a string into a vector of string based on a delimiter
	write2DVectorToCSV	Write a 2D vector of string into a csv file
	N	Computes the standard normal CDF - which is used by the Black Scholes model
	printVect	prints a 2D vector to the console - this is used for testing purposes
	stringToDoubleVect	converts a vector of string to a vector of doubles
	stringtoDateVect	converts a vector of string to a vector of boost date objects
	readCSV	reads a csv file where each column is a separate vector
Functions used only in Part 1	simulateStockPrices	returns a 2D
	computeHedgingErrors	Computes a 2D vector of objects of class Hedging_Portoflio - inside each object, we have the delta, option price, hedging error etc.
	getHedgedPortfolioParam	Returns a 2D vector of a specifc parameter of the vector returned by the function computeHedgingErrors
Functions used only in Part 2	countWeekDays	Return the number of business days between 2 string dates in "YYYY-MM-DD" format
	findVal	This function is used to find the interest rate and stock price on a particular day

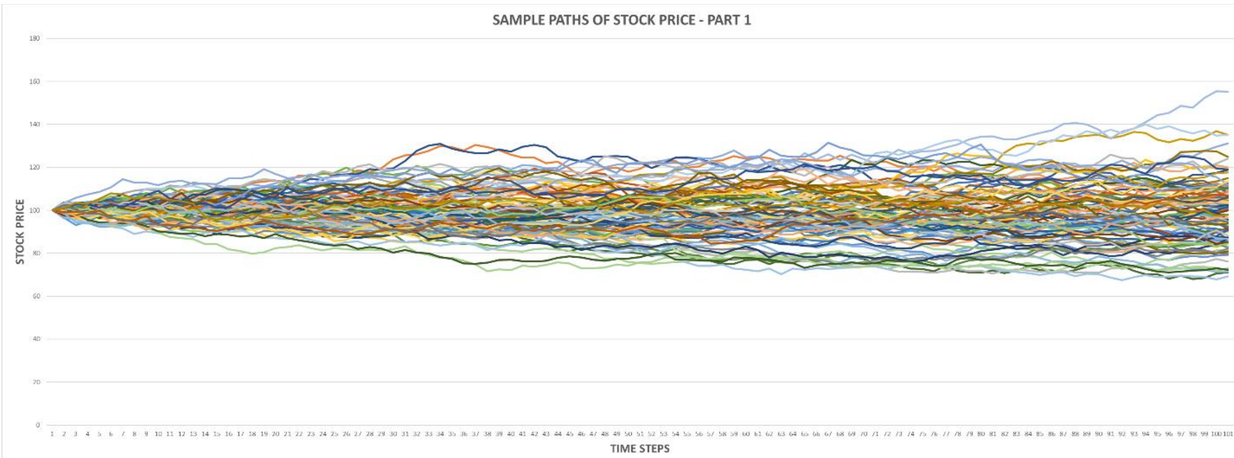
Project_Part.cpp

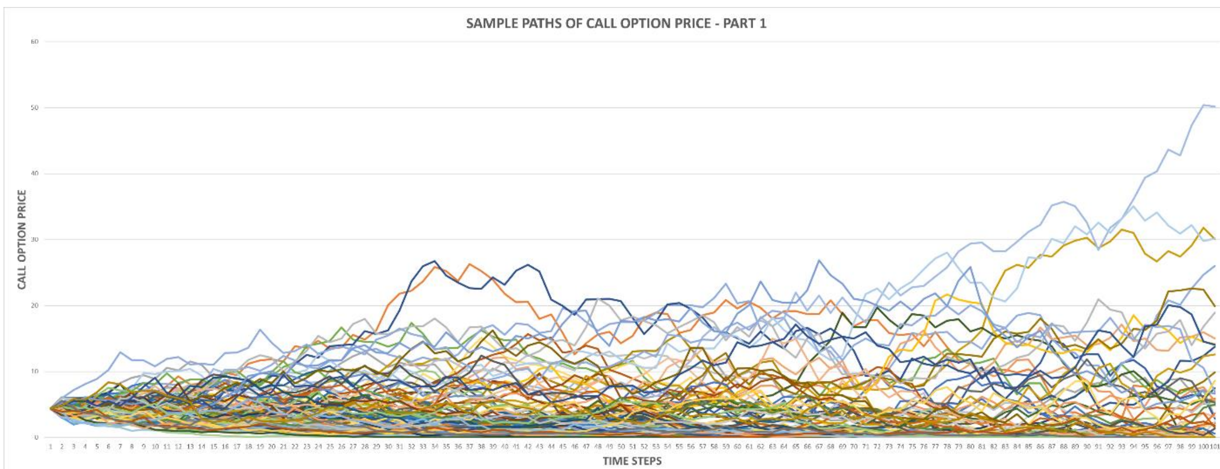
Function	Description
run_part_1	Code for part 1 of the project
run_part_2	Code for part 2 of the project

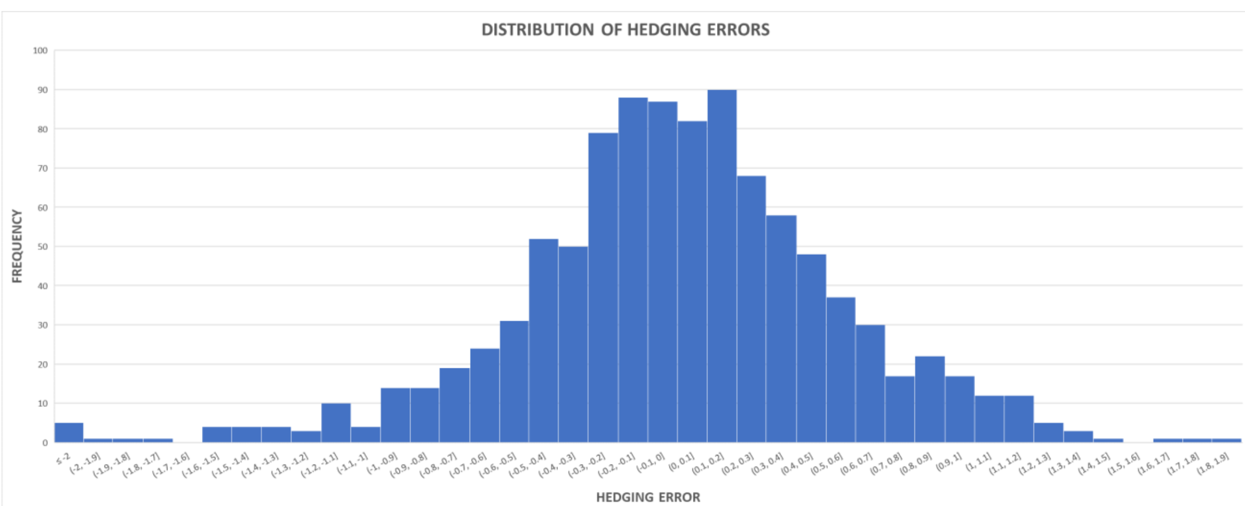
Main.cpp

Function	Description
main	Program execution begins and ends here

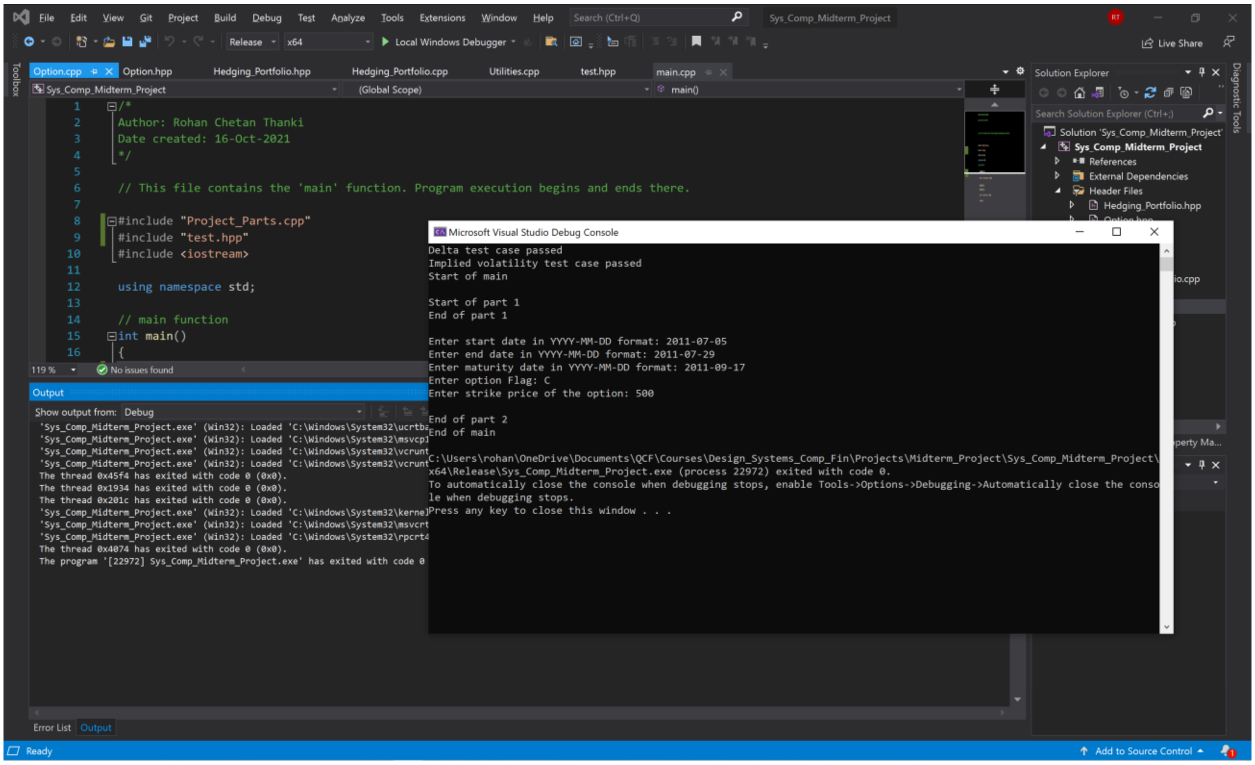
PLOTS AND GRAPHS



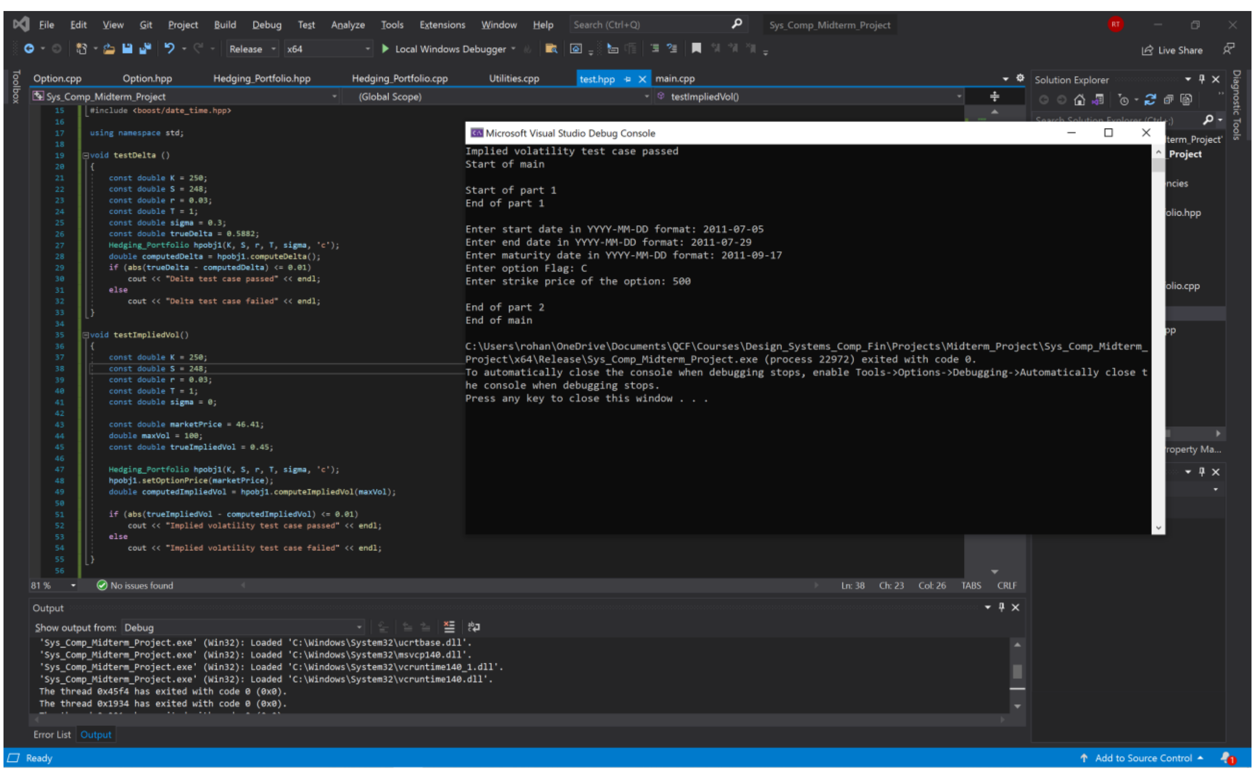




WORKING CODE SCREENSHOT



UNIT TEST CASES SCREENSHOT



OUTCOME OF THE IMPLEMENTATION – OBSERVATIONS AND CONCLUSIONS

- I was able to create a dynamic delta hedging portfolio using the Black Scholes options pricing model.
- In part 1, our hedging errors have somewhat a normal distribution centered around 0. This is because we have assumed a low drift parameter. In the real world, if the drift parameter is higher, many of our options will expire in the money, which will make us incur a loss.
- In part 2, we observe that the PNL for a naked short call position can become much more negative than a hedged portfolio. Thus, selling naked call options can become a very risky trading strategy.
- Even with a hedged portfolio, our PNL is almost always negative. This is because the GOOGLE stock price kept increasing, which increases the delta of the option. So, we had to buy more and more GOOGLE stock at increasing prices to maintain this delta hedge. Thus, we see our hedging errors are negative.
- In fact, the point above had very practical implications when stocks such as GAMESTOP and AMC shot up. As retail investors, bought large number of call options, the brokers had to go out into the market to buy more of these stocks, which increased the stock price and thus the delta of the call option. As more retail investors joined the bandwagon, brokers had to hedge their short call position at higher and higher deltas which pushed the stock price up and created a vicious positive cycle.
- In the real-world delta hedging alone is not enough. Thus, gamma hedging is also done. Gamma is the second derivative of the option price with respect to the stock price, i.e. Gamma is the rate of change of delta with change in stock price. A gamma hedged portfolio provides a better hedge, but it requires more active trading to keep the portfolio balance, which increases transaction cost. Thus portfolio managers have to make a tradeoff between transaction costs and a delta or gamma mismatch.