Progress Report

Option Pricing using QAE

DESIGN CREDIT PROJECT

SHIVANGINI RATHORE
B22PH012

INTRODUCTION

In this project, I explored how quantum computing can be applied to **Option pricing** specifically, for a *European Call Option*. The idea was to compare the classical Monte Carlo approach, which uses thousands of random simulations, with a **Quantum amplitude estimation method**, which encodes uncertainty into qubits and estimates the expected payoff more efficiently. This allowed me to study how quantum algorithms can potentially speed up complex financial computations while maintaining accuracy.

LITERATURE REVIEW

Quantum computing has shown significant potential in accelerating financial computations, particularly those involving stochastic modeling and uncertainty. Several studies have explored the integration of quantum algorithms, such as Amplitude Estimation, with classical finance models to improve efficiency and precision.

Woerner and Egger (2019) introduced one of the earliest frameworks for quantum risk analysis, where they modeled the distribution of asset prices using a *log-normal distribution* and applied quantum algorithms to estimate expected payoffs. Their work established the foundation for encoding financial uncertainty into quantum states.

Stamatopoulos et al. (2019) extended this by developing a comprehensive quantum framework for option pricing. They have demonstrated how amplitude estimation could be applied to both European and exotic options, achieving quadratic speedup compared to classical Monte Carlo simulations in their research.

These studies provide a strong theoretical and practical foundation for quantum-enhanced financial modeling. My project builds upon these contributions by simulating a European call option using six qubits to represent uncertainty and comparing the quantum estimation results with a classical Monte Carlo simulation of 5,000 paths.

Classical Approach

In the classical part of my project, I have used the **Monte Carlo simulation method** to calculate the price of a European call option. This method works by generating many random possible paths for how a stock price could move in the future and then averaging the results to estimate the option's value.

I assumed that the stock price follows a *Geometric Brownian Motion (GBM)*, which basically means the *price changes randomly over time* but also depends on the average market return and volatility.

For my simulation:

- I used 5000 random paths,
- Each path had 100 time steps
- The parameters I have used are -
    a. **Initial stock price** S0 = 50
    b. **Strike price** K = 55 (the fixed price at which an option holder can buy or sell the underlying asset.)
    c. **Risk-free rate** r = 0.05 (the return on an investment with zero risk, usually based on government bonds.)
    d. **Volatility sigma** = 0.4 (how much the price of an asset fluctuates over time)
    e. **Time to maturity** T = 1 year (the remaining time until a financial contract or option expires.)

At the end of each path, I calculated the payoff using this formula -

Payoff = max(S_T − K , 0)

where S_T = final stock price at the end of the simulation.

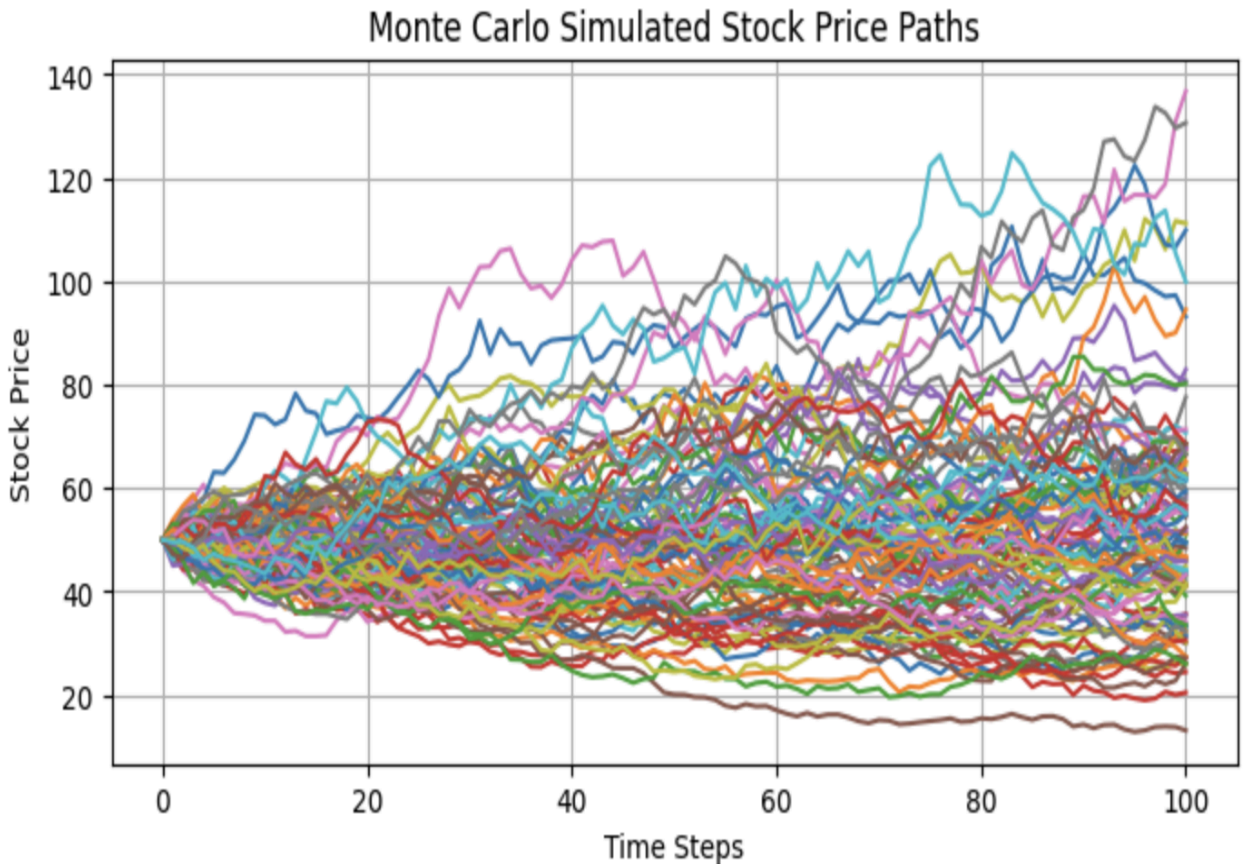
Then, I took the average of all payoffs and discounted it back to the present using the risk-free rate to get the final option price.

This classical result acts as a baseline that I have compared later with my quantum estimation results.

Results from Monte Carlo Simulation -

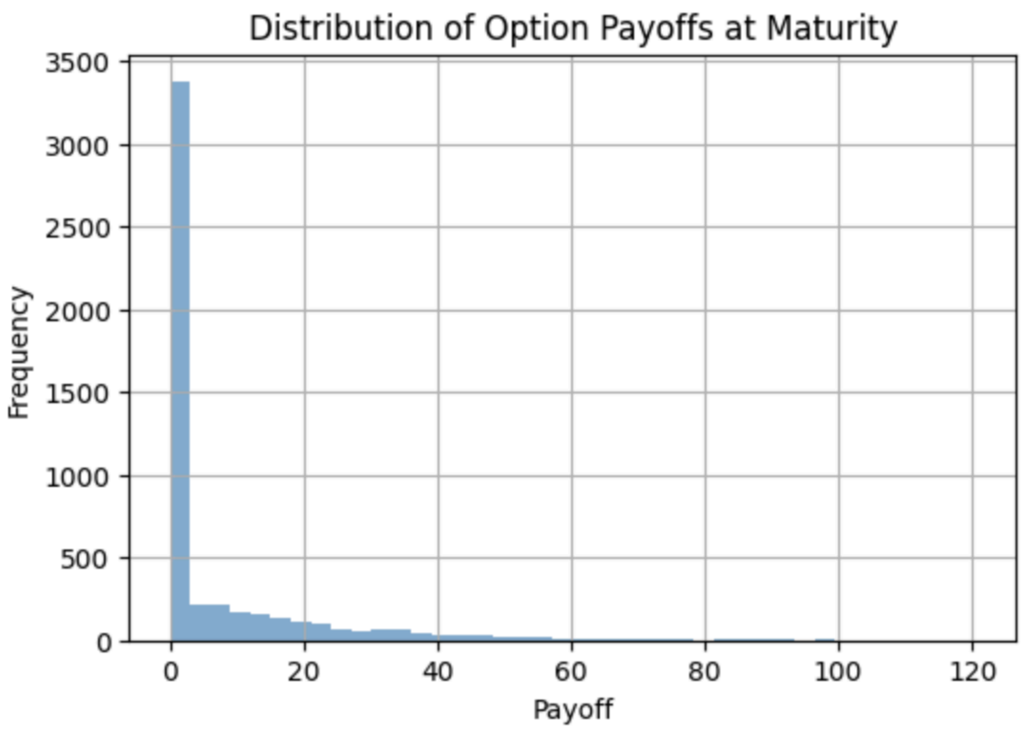
```
Estimated European Call Option Price: $6.7909
95% Confidence Interval: ±0.4016
```

Visualization of Monte Carlo Simulated Stock Price Paths - This helps visualize the uncertainty and randomness in price movement over time.



Distributions of Option Payoffs -

Each bar represents how frequently a certain payoff occurred. It helps visualize how most outcomes cluster around lower payoffs, while a few high payoffs occur when the stock price rises well above the strike price.



Quantum Approach - QAE

I have explored how the same problem can be approached using quantum computing. The goal is to understand how quantum methods, specifically Quantum Amplitude Estimation (QAE), can provide faster and more efficient estimation of option prices compared to classical simulations.

I have used **Qiskit** to represent the stock price distribution as a *quantum state*.

For this, I have used **6 qubits** to model the uncertainty in the stock price. Each qubit represents a slice of the possible stock price range, meaning **2^6=64** possible price outcomes were encoded simultaneously, something that is computationally heavy for classical methods but natural for quantum systems. The stock price distribution is assumed to follow a **log-normal distribution**, which is the same assumption used in the classical Black-Scholes model. This is loaded into a quantum circuit using $\mathrm{LogNormalDistribution()}$ from Qiskit Finance.

How 6 qubit framework is working?

We used 6 qubits to represent the possible stock price values —
Each combination of 6 qubits (|000000>........|111111>) represents one possible stock price level.

That gives us 2^6=64 possible stock prices.

The LogNormalDistribution circuit loads the probabilities (converted into amplitudes) of each price level into those 6 qubits.

So before measurement:

- The amplitude for each qubit combination encodes *how likely* that stock price is.
- Measuring the qubits gives a price according to that probability.

Then, I have constructed the option payoff function max(0,S_T − K) and used Iterative Amplitude Estimation (a quantum algorithm) to estimate the expected payoff.
This quantum method achieves a theoretical quadratic speed-up — it can reach the same accuracy with far fewer samples compared to classical Monte Carlo.

I have also compared the **quantum-estimated option value** with the **exact expected value**.
The results were close, showing that even with limited qubits and simulated environments, quantum approaches can produce accurate pricing results with significantly improved efficiency potential.

Comparison & Results

In the classical Monte Carlo simulation, I obtained an estimated European Call Option price of $6.79, with a 95% confidence interval of ±0.40 based on 5,000 simulated paths.

In the quantum amplitude estimation (QAE) method, the algorithm estimated an amplitude value of around **0.58**, compared to the classical reference value of **0.46**. This value corresponds to the expected probability of the option being in the money, which can later be interpreted to obtain the option price in financial terms.

While the numerical scales differ (since QAE operates on normalized quantum amplitudes), both methods show consistent behavior - the quantum estimation approximates the classical expectation with a small estimation error (nearly 0.12).

This confirms that the quantum method can reproduce classical outcomes with fewer queries, theoretically offering a quadratic speedup.

```
Exact Classical Monte Carlo Value: 0.4605
Estimated Quantum Value: 0.5870
Estimation Error: 0.1265
Confidence Interval: (np.float64(0.524012919098793), np.float64(0.6499433961065993))
```

Conclusion

The Quantum Amplitude Estimation (QAE) algorithm estimated the normalized expected payoff as 0.5870, compared to the classical exact value of 0.4605, with an estimation error of 0.1265 within the confidence interval (0.5240, 0.6499). The values are normalized, meaning 0.46 represents 46% of the maximum possible payoff. While QAE achieves comparable accuracy with exponentially fewer samples than classical Monte Carlo methods, practical challenges such as noise, limited qubit precision, and circuit depth still affect accuracy on real quantum hardware.