CQF Exam 2

Viranca Balsingh

In partial fulfilment of the Certificate in Quantitative Finance
As offered by Fitch Learning under the supervision of Dr. Paul Wilmott

October 4, 2024

1 Outline of the finance problem and numerical procedure

1.1 Euler-Marayuma scheme for Simulation of the Underlying Stock Price

Analytically, the underlying stock price follows a Geometric Brownian Motion (GBM), which is modelled by

$$dS_t = rS_t\,dt + \sigma S_t\,dW_t \tag{1.1}$$

Where S_t is the stock price at time t, r is the risk free rate (constant), σ is the volatility of the underlying stock and W_t is the Wiener process.

The Euler-Maruyama method gives the following solution to the GBM for a discretized time dimension

$$S_{t+\Delta t} - S_t = r_t S_t \Delta t + \sigma_t \phi \sqrt{\Delta t} \tag{1.2}$$

Where Δt is the time step, and ϕ is a standard normal random variable.
The simulation is initialized by setting the parameters to the following starting conditions:

$$\begin{aligned}
\text{Today's stock price } S_0 &= 100\\
\text{Strike } E &= 100\\
\text{Time to expiry } (T-t) &= 1 \text{ year}\\
\text{Volatility } \sigma &= 20\%\\
\text{Constant risk-free interest rate } r &= 5\%
\end{aligned}$$

Moreover, regarding the payoff scheme, the expected value of the discounted payoff under the risk neutral density $\mathbb{Q}$ is given by the following

$$V(S,t) = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}[\text{Payoff}(S_T)] \tag{1.3}$$

The payoff for the Asian options will consider the average price as compared to the strike price over the simulation period. The payoff for the Lookback options will be derived from the maximum and minimum prices as compared to the strike price over the simulation period.

$$\begin{aligned}
\text{Payoff}_{\text{Asian Call}} &= \max(\bar{S} - E, 0) = H(\bar{S} - E)(\bar{S} - E)\\
\text{Payoff}_{\text{Asian Put}} &= \max(E - \bar{S}, 0) = H(E - \bar{S})(E - \bar{S})
\end{aligned}$$

$$\begin{aligned}
\text{Payoff}_{\text{Lookback Call}} &= \max(S_{\max} - E, 0) = H(S_{\max} - E)(S_{\max} - E)\\
\text{Payoff}_{\text{Lookback Put}} &= \max(E - S_{\min}, 0) = H(E - S_{\min})(E - S_{\min})
\end{aligned}$$

1.2 Numerical Procedure for Simulation of the Underlying Stock Price

With the full code attached, this section will go over the key components of the numerical simulation. Initializing the variables and simulating the stock price along M paths as follows:

```python
import numpy as np

# Parameters
S0 = 100        # Initial stock price
E = 100         # Strike price
T = 1           # Time to expiry (1 year)
r = 0.05        # Risk-free rate
sigma = 0.2     # Volatility
N = 252         # Number of time steps (daily)
M = 10000       # Number of simulations
dt = T / N      # Time increment

# Euler-Maruyama simulation of stock prices
np.random.seed(42)
S_paths = np.zeros((M, N + 1))
S_paths[:, 0] = S0

for t in range(1, N + 1):
    Z = np.random.normal(0, 1, M) # Random values from standard normal distribution
    S_paths[:, t] = S_paths[:, t - 1] * np.exp((r - 0.5 * sigma**2) * dt + sigma * np.sqrt(dt) * Z)
```

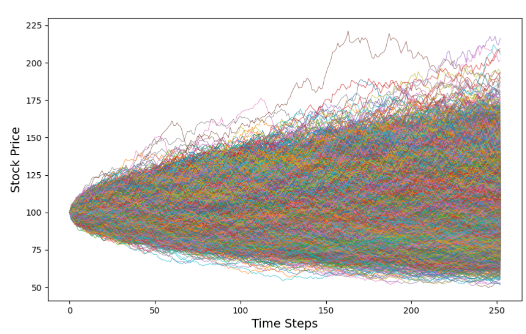

Figure 1.1: Monte Carlo Simulation of a Stock Price (GBM)

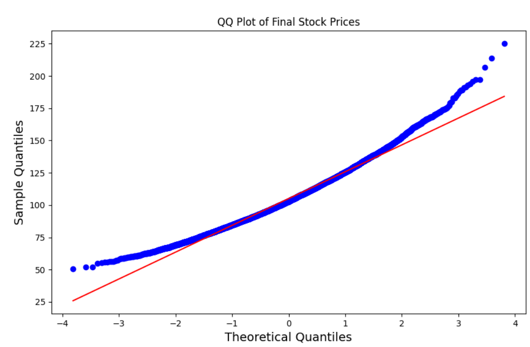

Figure 1.2: QQ plot of simulated stock prices versus normal distribution

To understand these paths better, the QQ plot is made to compare against a normal distribution in Figure 1.2. The curve in the stock prices is explained by the fact that the final stock prices are log-normally distributed due to the exponential nature of the model.
For the initial parameters, the payouts can be easily computed by considering the average price across the simulation paths and then applying the Heaviside functions for each option type. Then the payouts are discounted by the risk free rate as follows:

```
# Asian Option: Calculate the average stock price for each path
average_price = np.mean(S_paths[:, 1:], axis=1)
asian_call_payouts = np.maximum(average_price - E, 0)
asian_put_payouts = np.maximum(E - average_price, 0)

# Lookback Option: Calculate the maximum and minimum stock price for each path
max_price = np.max(S_paths[:, 1:], axis=1)
min_price = np.min(S_paths[:, 1:], axis=1)
lookback_call_payouts = np.maximum(max_price - E, 0)
lookback_put_payouts = np.maximum(E - min_price, 0)

# Discounted payout for both call and put options
asian_call_option_value = np.exp(-r * T) * np.mean(asian_call_payouts)
asian_put_option_value = np.exp(-r * T) * np.mean(asian_put_payouts)
lookback_call_option_value = np.exp(-r * T) * np.mean(lookback_call_payouts)
lookback_put_option_value = np.exp(-r * T) * np.mean(lookback_put_payouts)
```

Running the Monte Carlo Simulation using various total simulation counts, while keeping all other variables constant to the initial settings, gives the following results:

# Simulations	**Asian**		**Lookback**	
	Call	**Put**	**Call**	**Put**
1,000	6.04765	3.02884	19.06955	11.12771
10,000	5.76594	3.30847	18.29730	11.64586
100,000	5.80175	3.34589	18.38701	11.72133

1.3 Model Extensions

Variable Volatility - Heston Model

There are several methods to make this model more realistic. First of all, the Black-Scholes assumes a constant Volatility. Rather, the Heston model proposes to use the below equation to model the volatility more realistically

$$dV_t = k(\theta - V_t)dt + \sigma\sqrt{V_t}dW_{2t} \tag{1.4}$$

Where:

- V_t is the variance of the asset price at time t,
- k is the rate of mean reversion,
- θ is the long-term variance mean,
- σ is the volatility of the variance,

In python this is implemented into our initial model as follows:

```
# Parameters
S0 = 100        # Initial stock price
E = 100         # Strike price
T = 1           # Time to expiry (1 year)
r = 0.05        # Risk-free rate
v0 = 0.04       # Initial variance
kappa = 2.0     # Rate of reversion
theta = 0.04    # Long-term variance
xi = 0.1        # Volatility of volatility
rho = -0.7      # Correlation between the two Brownian motions
N = 252         # Number of time steps (daily)
M = 10000       # Number of simulations
dt = T / N      # Time increment

# Euler-Maruyama simulation of stock prices and variance using the Heston model
np.random.seed(1)
S_paths = np.zeros((M, N + 1))
v_paths = np.zeros((M, N + 1))
S_paths[:, 0] = S0
v_paths[:, 0] = v0

for t in range(1, N + 1):
    Z1 = np.random.normal(0, 1, M) # Random values for stock price
    Z2 = np.random.normal(0, 1, M) # Random values for variance
    Z2 = rho * Z1 + np.sqrt(1 - rho**2) * Z2 # Correlated Brownian motion
```

```
v_paths[:, t] = np.maximum(v_paths[:, t - 1] + kappa * (theta - v_paths[:, t
    - 1]) * dt + xi * np.sqrt(v_paths[:, t - 1] * dt) * Z2, 0)
S_paths[:, t] = S_paths[:, t - 1] * np.exp((r - 0.5 * v_paths[:, t]) * dt +
    np.sqrt(v_paths[:, t] * dt) * Z1)
```

To following graph shows how the Heston volatility looks over time, as well as the old constant volatility:

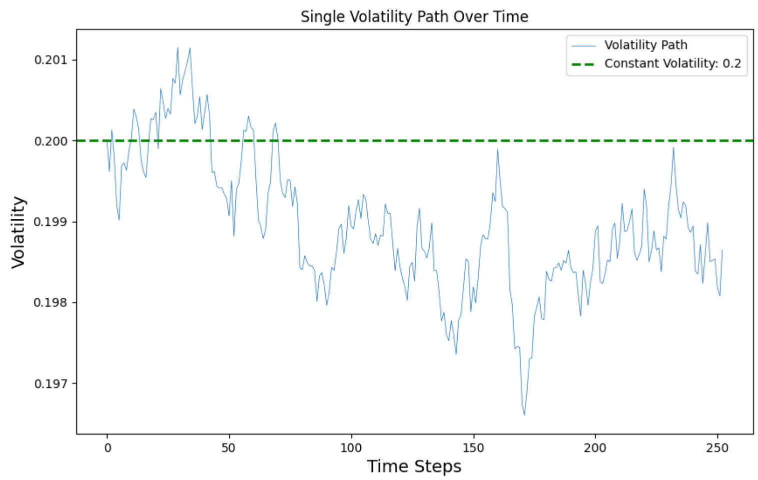

Figure 1.3: Variable Heston volatility vs constant volatility

This gives the following results, again for varying counts of simulation paths for good measure:

# Simulations	**Asian**		**Lookback**	
	Call	**Put**	**Call**	**Put**
1,000	6.15338	3.17819	18.65926	11.38911
10,000	5.54680	3.33989	17.90667	11.85505
100,000	5.67045	3.41627	18.04865	11.90456

Variable Interest Rate - Vasicek Model

The interest rate model is described by a mean-reverting process, specifically a Vasicek model. This model is characterized by the following stochastic differential equation:

$$dr_t = \kappa_r(\theta_r - r_t)\,dt + \sigma_r\,dW_t^r \tag{1.5}$$

where:

- r_t is the interest rate at time t.
- κ_r is the rate of reversion.
- θ_r is the long-term mean level of the interest rate.
- σ_r is the volatility of the interest rate.
- dW_t^r is a Wiener process (Brownian motion).

In python this is implemented into our current model as follows:

```
# Parameters
S0 = 100        # Initial stock price
E = 100         # Strike price
T = 1           # Time to expiry (1 year)
r0 = 0.05       # Initial risk-free rate
v0 = 0.04       # Initial variance
kappa = 3.0     # Rate of reversion for variance
theta = 0.04    # Long-term variance
xi = 0.01       # Volatility of volatility
rho = -0.7      # Correlation between the two Brownian motions
kappa_r = 0.1   # Rate of reversion for interest rate
theta_r = 0.05  # Long-term interest rate
sigma_r = 0.01  # Volatility of interest rate
N = 252         # Number of time steps (daily)
M = 10000       # Number of simulations
dt = T / N      # Time increment

# Euler-Maruyama simulation of stock prices, variance, and interest rate using
    the Heston model
np.random.seed(1)
S_paths = np.zeros((M, N + 1))
v_paths = np.zeros((M, N + 1))
r_paths = np.zeros((M, N + 1))
S_paths[:, 0] = S0
v_paths[:, 0] = v0
r_paths[:, 0] = r0

for t in range(1, N + 1):
```

```
Z1 = np.random.normal(0, 1, M) # Random values for stock price
Z2 = np.random.normal(0, 1, M) # Random values for variance
Z3 = np.random.normal(0, 1, M) # Random values for interest rate
Z2 = rho * Z1 + np.sqrt(1 - rho**2) * Z2 # Correlated Brownian motion for
    variance

v_paths[:, t] = np.maximum(v_paths[:, t - 1] + kappa * (theta - v_paths[:, t
    - 1]) * dt + xi * np.sqrt(v_paths[:, t - 1] * dt) * Z2, 0)
r_paths[:, t] = r_paths[:, t - 1] + kappa_r * (theta_r - r_paths[:, t - 1]) *
    dt + sigma_r * np.sqrt(dt) * Z3
S_paths[:, t] = S_paths[:, t - 1] * np.exp((r_paths[:, t - 1] - 0.5 *
    v_paths[:, t]) * dt + np.sqrt(v_paths[:, t] * dt) * Z1)
```

To following graph shows how the Vasicek Interest Rate looks over time, as well as the old constant Interest Rate:

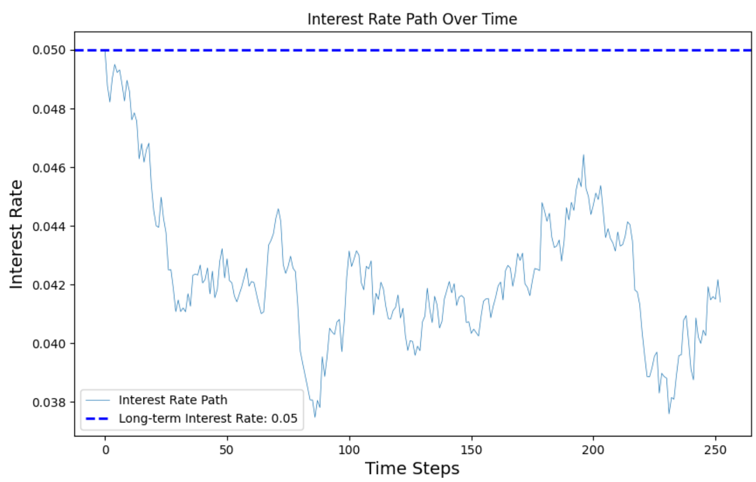

Figure 1.4: Variable Vasicek IR vs constant initial interest rate

This gives the following results, again for varying counts of simulation paths for better comparability:

# Simulations	Asian		Lookback	
	Call	Put	Call	Put
1,000	6.10265	3.33950	18.71642	11.44191
10,000	5.65735	3.46227	18.02039	11.95172
100,000	5.67476	3.41152	18.10253	11.90648

1.4 Observations and Challenges Encountered

1.4.1 Observations

1. **Monte Carlo Convergence**: Increasing the number of simulation paths improved accuracy, with significant convergence by 10,000 paths.
2. **Volatility Impact**: Stochastic volatility in the Heston model led to more realistic pricing, particularly for long-term and Lookback options, compared to the constant volatility assumption.
3. **Variable Interest Rates**: The Vasicek model showed that variable rates affect long-term options more, highlighting the limitations of using constant rates.
4. **Computational Trade-offs**: While the Euler-Maruyama method is easy to implement, it becomes computationally expensive with high-resolution simulations.

1.4.2 Challenges

1. **Negative Variance**: The Heston model sometimes produced negative variances, requiring an approximation to avoid instability.
2. **Numerical Stability**: Stability issues arose with small time steps, necessitating careful tuning of simulation parameters.
3. **Correlation Handling**: Managing the correlation between Brownian motions in the Heston model added complexity and potential for errors.
4. **Long Computation Times**: Higher simulation counts significantly increased computation time, making efficiency a key concern.

1.5 Conclusion

This report provides a detailed analysis of the numerical methods and financial models used to simulate the underlying stock price and compute option payoffs, focusing on Asian and Lookback options. The stock price follows a Geometric Brownian Motion (GBM), and the Euler-Maruyama scheme is applied for the numerical simulation.

The analysis begins with an explanation of the stock price dynamics under the GBM assumption, followed by the discretized form for the simulation. The model initialization sets standard parameters such as stock price, strike price, time to expiry, volatility, and risk-free rate.

After simulating the stock price paths using the Euler-Maruyama method, the payoff schemes for Asian and Lookback options are introduced. The payoff depends on the average, maximum, or minimum prices over the simulation. The Monte Carlo simulation allows for the estimation of option values using a large number of simulation paths, demonstrating the accuracy and convergence as the number of simulations increases.

Next, the basic model is extended by incorporating the Heston model for stochastic volatility, which captures more realistic volatility behavior. The Heston model replaces the constant volatility assumption in the Black-Scholes model with a mean-reverting stochastic variance process, adding complexity and realism to the simulation.

Additionally, the impact of variable interest rates is explored using the Vasicek process. The Vasicek model is a mean-reverting process that captures the dynamics of interest rates more accurately than a constant risk-free rate assumption.

Through simulations with varying total paths and using both the GBM and Heston models, the effects of different levels of complexity in the volatility and interest rate models on the pricing of Asian and Lookback options are observed. The model extensions, including stochastic volatility and variable interest rates, demonstrate more nuanced behaviors and potential improvements over constant parameter models.

Bibliography

[1] QuantInsti Blog, "Heston Model: Stochastic Volatility Model for Pricing Options", QuantInsti, https://blog.quantinsti.com/heston-model/, [Accessed: October 3, 2024].

[2] Paul Wilmott, *Paul Wilmott on Quantitative Finance*, 2nd edition, John Wiley & Sons, Chapter 49.

[3] Fischer Black and Myron Scholes, "The Pricing of Options and Corporate Liabilities", *Journal of Political Economy*, Vol. 81, No. 3, pp. 637-654, 1973.

[4] Steven L. Heston, "A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options", *The Review of Financial Studies*, Vol. 6, No. 2, 1993, pp. 327-343. https://doi.org/10.1093/rfs/6.2.327.

[5] Paul Glasserman, *Monte Carlo Methods in Financial Engineering*, Springer, 2004.

[6] University of Virginia Library, "Understanding Q-Q Plots", University of Virginia Library, https://library.virginia.edu/data/articles/understanding-q-q-plots#:~:text=A%20QQ%20plot%20is%20a,truly%20come%20from%20normal%20distributions., [Accessed: October 3, 2024].

[7] Valparaiso University, "Estimating Option Prices with Heston's Stochastic Volatility Model", Valparaiso University, https://www.valpo.edu/mathematics-statistics/files/2015/07/Estimating-Option-Prices-with-Heston%E2%80%99s-Stochastic-Volatility-Model.pdf, [Accessed: October 3, 2024].

[8] Oldrich Vasicek, "An Equilibrium Characterization of the Term Structure", *Journal of Financial Economics*, Vol. 5, No. 2, 1977, pp. 177-188.