AY24/25 Winter Semester

FINE3300R Python for Finance Professionals

Title: FINE 3300 (Winter 2025) Group Project

Tutor: Prof Faisal Habib

Team: Group 8

Date of Submission: 31 March 2025

Group Members	**Student ID**
ALEX KHOO SHIEN HOW	221943691
ALICIA NG YAN LENG	221943493
ABDIKARIM ABUKAR	219660679

*With this submission, all group members acknowledge their equal meaningful contributions to the project. There are no disputes regarding individual efforts, and no further arguments on this matter shall be entertained.

Repository link: https://github.com/alexksh2/FINE3300_Final_Project

ALEX KHOO SHIEN HOW

ALICIA NG YAN LENG

ABDIKARIM ABUKAR

Introduction

Asian options are a class of derivatives that are characterized by their path-dependent nature, where payoffs happen to be determined by the average price of the underlying asset over a specified time period. This averaging aspect provides protection against market volatility in the short-term and price manipulation near expiration dates.

In this project, we employ the Monte Carlo simulation techniques to price different European-style, non-dividend paying Asian options with fixed strike and floating strike options. Each has either arithmetic or geometric averaging methods. Using Geometric Brownian Motion (GBM) to predict and plot stock price movements under a risk-neutral environment, we analyze how option values are affected by changes in market parameters and simulation iterations.

Discussion

For easy reference, we have appended Table 2 from the Group Project document which showcases the 5 options we will be looking at for our analysis. We have organised the report by question for discussion of the findings. Further elaboration on the code behind our results can be found in the file **FINE3300_Group_8_Group_Project.ipynb** attached together with this report.

Table 2.

	Strike (K)	μ	σ	Expiration
#1	120.00	8%	20%	30 days
#2	80.00	5%	20%	30 days
#3	100.00	8%	25%	90 days
#4	80.00	4%	15%	60 days
#5	150.00	8%	40%	90 days

Question 0: Design A Class - Pricing the 5 Options

With reference to the code file, we have created a class called **AsianOptionMonteCarlo**. We have included the methods generate_paths and price_option. The method **generate_paths** simulates the stock price using GBM. The method **price_option** prices the asian option with the relevant parameters using Monte Carlo simulation.

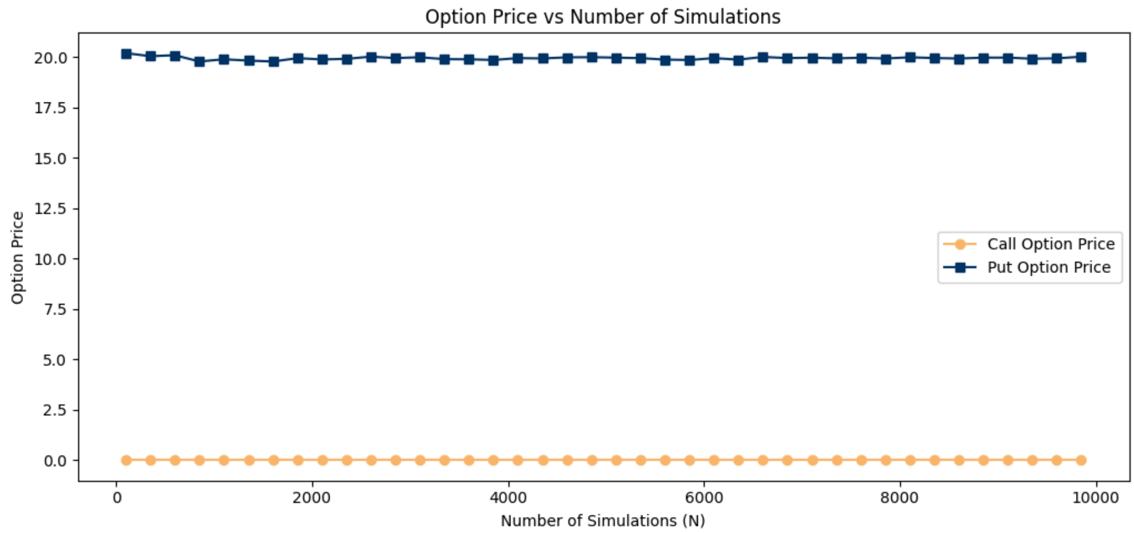
Using the class and its methods, we have listed the prices and standard errors in parentheses for each of the 5 options in Table 2. We have taken into account the following parameters: (1) **option type**: call/put, (2) **averaging type**: arithmetic/geometric, (3) **strike type**: fixed/floating. Below we show a sample output for Option #1. The rest of the sample output can be found in Appendix A/the code file.

```
==== Option #1 ====
Call | arithmetic avg | fixed strike => (Price, Standard Error): (0.0000, 0.0000)
Put | arithmetic avg | fixed strike => (Price, Standard Error): (19.9102, 0.0331)
Call | arithmetic avg | floating strike => (Price, Standard Error): (1.3527, 0.0202)
Put | arithmetic avg | floating strike => (Price, Standard Error): (1.2789, 0.0183)
Call | geometric avg | fixed strike => (Price, Standard Error): (0.0000, 0.0000)
Put | geometric avg | fixed strike => (Price, Standard Error): (19.9629, 0.0332)
Call | geometric avg | floating strike => (Price, Standard Error): (1.3423, 0.0203)
Put | geometric avg | floating strike => (Price, Standard Error): (1.2984, 0.0184)
```

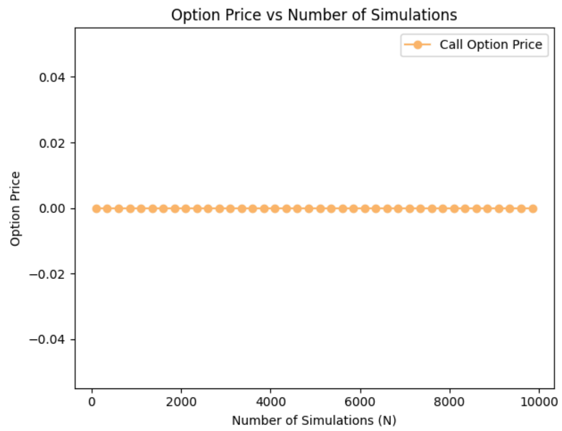
Question 1: Option #1 - Simulation Performance

With reference to Table 2, the following graphs were plotted for Option #1. The option prices were determined based on the assumptions - strike type was **fixed** and averaging type was **arithmetic**.

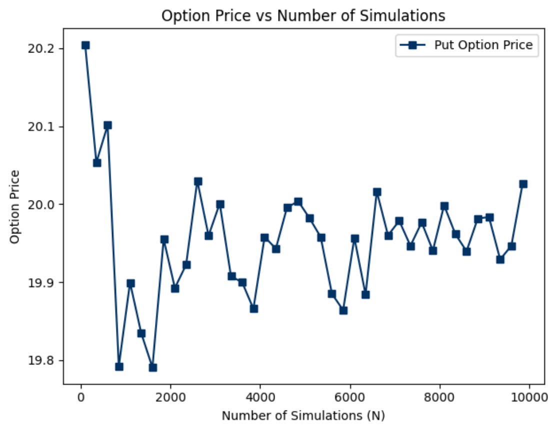
What value of N is suitable to use?



Graph 1.1: Call & Put Option Price vs No. of Simulations (N)



Graph 1.2: Call Option Price vs N

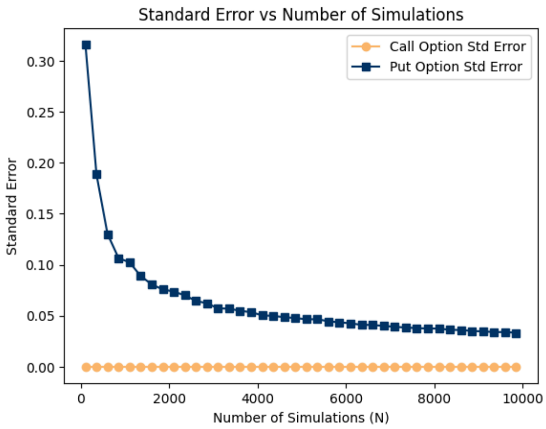


Graph 1.3: Put Option Price vs N

From the combined graph Graph 1.1, it can be seen that the price seems relatively constant across changes in N. However, taking a closer look at the individual graphs, in particular Graph 1.3, it can be seen that the put option price does show fluctuations across changes in N. Analyzing the trend, it can be deduced that the option price stabilizes as N increases, particularly from around N = 7000 to N = 1000. To ensure computational efficiency while ensuring higher precision in option prices, we believe that an arbitrary number of around **N = 8500** is a suitable number for the number of simulations.

 # NOTE: Graph 1.2 was ignored from analysis as in this case the call position for option #1 is out-of-the-money where the underlying stock price does not go above the strike price, resulting in a call option price of 0 regardless of the number of simulations.

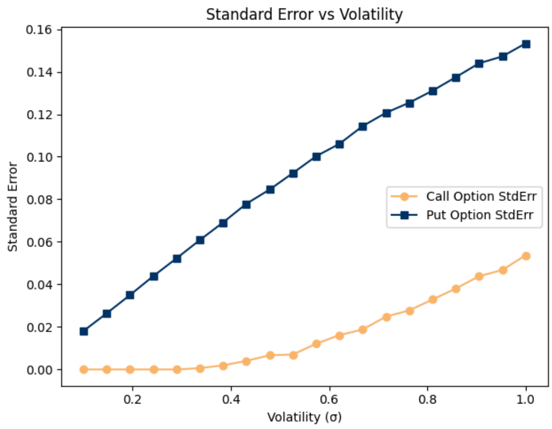
What happens to the standard error as the number of simulations increases?



Referring to the graph on the left, it can be seen that the **standard error decreases** as the number of simulations increases. An increasing N provides a more accurate estimate of the option price by averaging out random fluctuations.

 # NOTE: Call Option Standard Errorwas ignored from analysis since call option price is 0 regardless of the number of simulations.

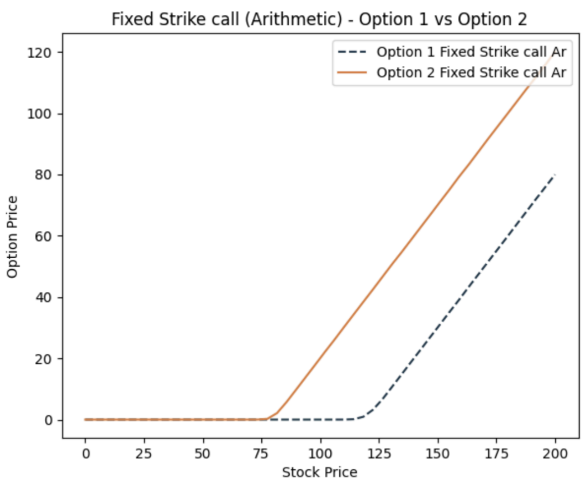
If volatility of the returns increase, how well does your Monte Carlo simulation perform?



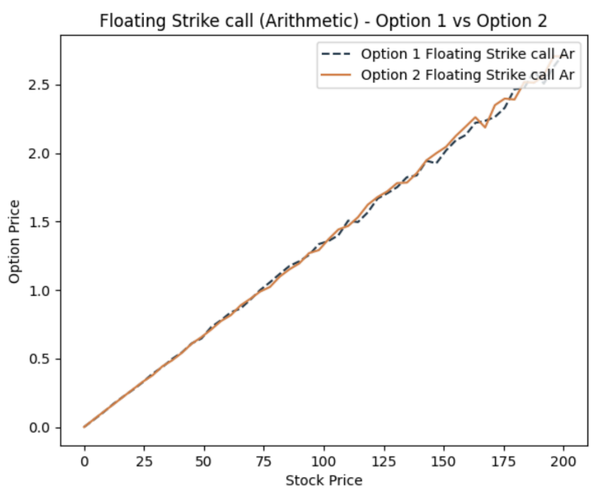
Higher volatility leads to greater variability in stock price paths. Referring to the graph on the left, it can be seen that the standard error increases as volatility increases. This **increase in standard error suggests that an increased number of simulations** would be required for accuracy and precision in valuing the option. Hence, this suggests that it would be **more computationally expensive** to execute the Monte Carlo simulation for volatile assets.

Question 2: Option #1 & #2 - Varying Stock Price

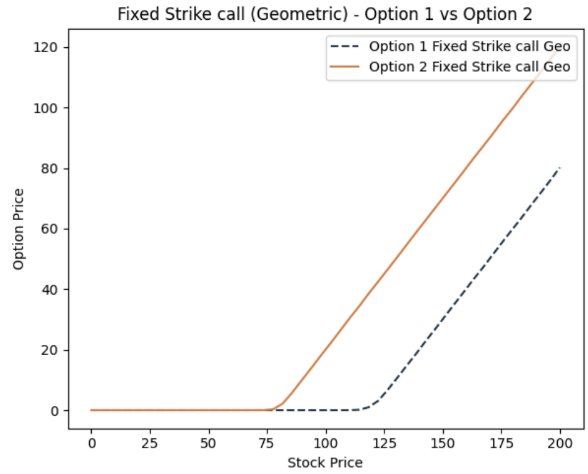
With reference to Table 2, the following graphs were plotted for Option #1 and Option #2. The number of simulations was set at what we determined to be a suitable N in Question 1 at N = 8500.



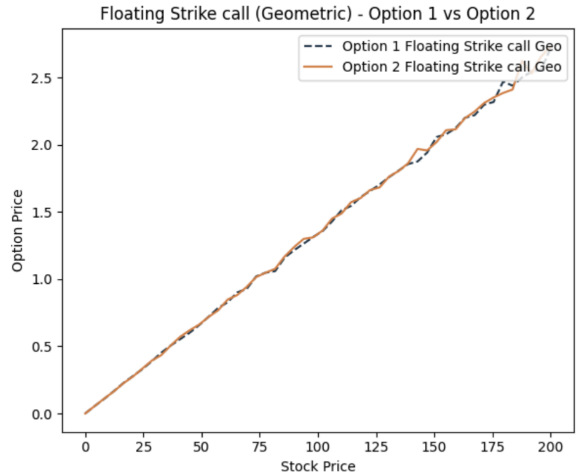
Graph 2.1



Graph 2.2



Graph 2.3



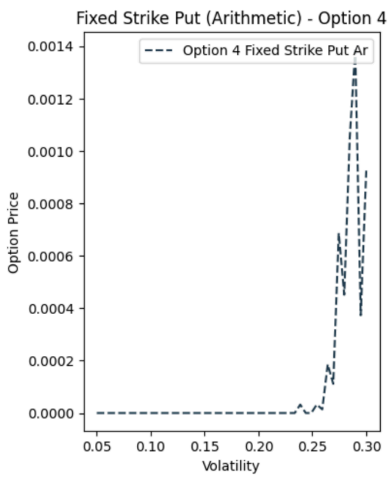
Graph 2.4

Based on the above graphs, it can be seen that regardless of the type of Asian option with a call position, the **option price for both Option #1 and Option #2 increases as the underlying stock price increases**. Looking particularly at Graphs 2.1 and 2.3 (i.e., fixed strike type), it can be seen that when the stock price is below the Option #1 and Option #2 strike prices of $120 and $80 respectively, the **option price remains close to 0, as the option is out-of-the-money**. As the stock price moves above the respective strike prices, the option becomes in-the-money, where the call option price increases almost linearly.

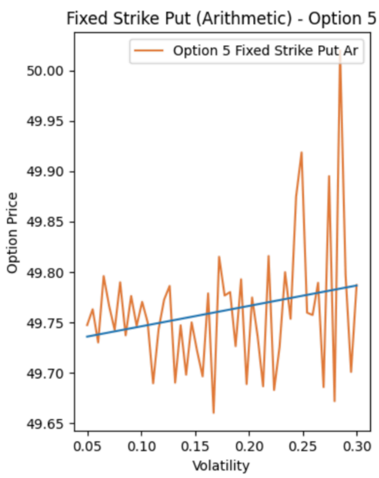
Additionally, just by comparing between the fixed strike call (Graph 2.1 and 2.3) and the floating strike call (Graph 2.1 and 2.4) graphs, it can be observed that the **option price increases more gradually for the floating strike call compared to the fixed strike call**. For fixed strike, a higher underlying stock price directly increases the intrinsic value of the call option. For floating strike price, the strike price is based on the **average stock price**, reducing the impact of short-term fluctuations. This results in **lower volatility** in option prices and a **smoother increase** as the stock price rises.

Question 3: Option #4 & #5 - Varying Volatility

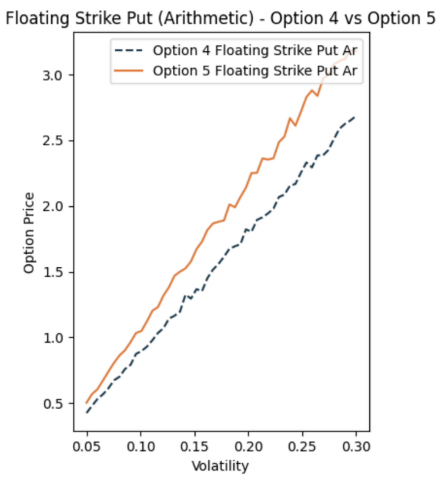
With reference to Table 2, the following graphs were plotted for Option #4 and Option #5. The number of simulations was set at what we determined to be a suitable N in Question 1 at N = 8500.



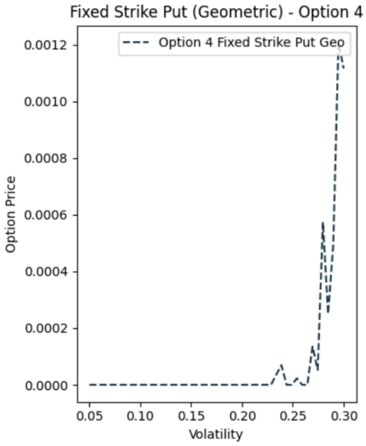
Graph 3.1



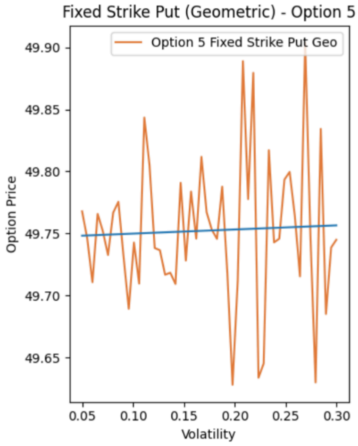
Graph 3.2



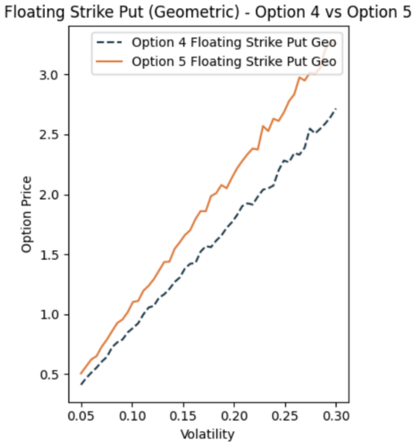
Graph 3.3



Graph 3.4



Graph 3.5



Graph 3.6

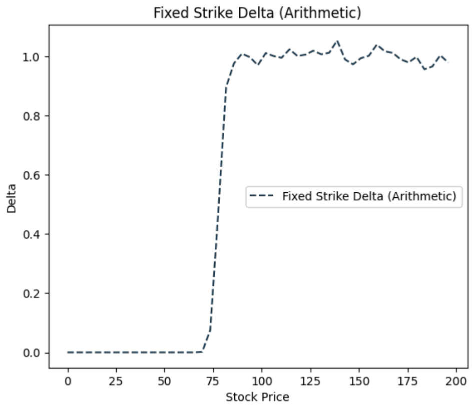
Looking at the fixed strike graphs for Option #4 (Graphs 3.1 and 3.4), we can see that both the arithmetic and geometric methods show very close behavior. This means the price yield of the put **option stays around 0 at low volatility (5-25%) but spikes sharply with volatility thereafter**. This sudden spike indicates that at higher volatility levels, there's an increased probability that the average price will fall below the strike price of $80, making the put option more valuable.

Looking at the fixed strike graphs for Option #5 (Graphs 3.2 and 3.5), we can see from the trendline that **option price increases with volatility**. Additionally, the put option price is quite high at approximately 40 even for low volatility. This suggests that **Option #5 is deeply in-the-money** with such a high strike price of $150 compared to the stock price of $100. That being said, Graphs 3.2 and 3.5 can be observed to have large fluctuations across different volatility levels, but it lacks the dramatic spike seen in Option #4. This could potentially suggest limitations in the Monte Carlo simulation with regards to the number of simulations or the fact that small changes in simulated paths could lead to large fluctuations in the option price. Regardless, it can still be seen that option price increases with volatilities. The geometric averaging (Graph 3.5) shows similar behavior to arithmetic averaging (Graph 3.2), despite more pronounced fluctuations.

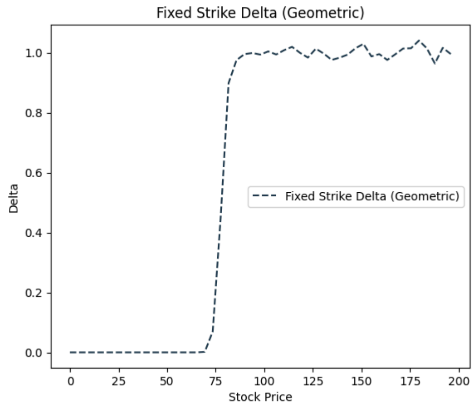
For floating strike put options (Graphs 3.3 and 3.6), Options #4 and #5 shows a **linear upward price increase with increasing volatility**. This is consistent with the trend shown in the fixed strike graphs. The reason for the linear increase could be explained by how floating strike put options are less affected by extreme values allowing the trend to remain smooth.

Question 4: Option #4 - Delta

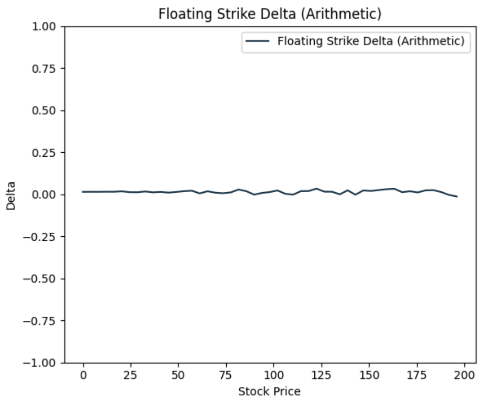
With reference to Table 2, the following graphs were plotted for Option #4. The number of simulations was set at what we determined to be a suitable N in Question 1 at N = 8500. *(Graphs are displayed on the next page).*



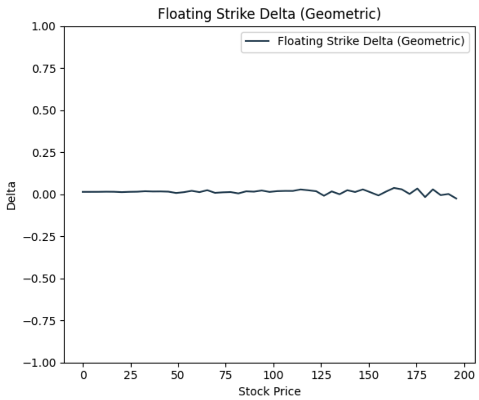
Graph 4.1



Graph 4.2



Graph 4.3



Graph 4.4

Looking at the fixed strike options graphs (Graphs 4.1 and 4.2), the trend observed can be discussed by dividing the graph into 3 different sections.

(1) **Low Stock Price Region ($0-75):** When the stock price is way below the strike price $80, the **delta will be near 0**. This indicates that small changes in the underlying stock price have a very small impact on the option price, as the option is deep out-of-the-money.

(2) **Transition Region ($75-85)**: As the stock price heads towards and supasses the strike price, there's a **rapid increase in delta from nearly 0 to approximately 1**. This very steep slope happens around the strike price of $80, indicating that the option's sensitivity to price fluctuations is at a peak when the option is nearing in-the-money.

(3) **High Stock Price Region ($85 and above)**: When the stock price happens to exceed the strike price by a large margin, the **delta stabilizes around 1**, this shows that the option price moves almost at one-to-one with the underlying price as the option becomes in-the-money.

Looking at the floating strike options graphs (Graphs 4.3 and 4.4), it can be seen that both graphs are consistent in showing that the **delta is close to 0 throughout with minor fluctuations**. Since floating strike options adjust the strike price based on the average of past stock prices, at high stock prices, the average price also rises. Hence, the floating strike price also increases. As a result, this causes the option price to be less sensitive to price changes, allowing delta to be stable at around 0.

Conclusion

In conclusion, this report has analyzed the 4 types of Asian options using Monte Carlo simulations. Monte Carlo simulations tend to perform better with lower volatility and an increased number of simulations. The performance of the simulation should be balanced between accuracy and computational efficiency. Additionally, it has been determined that fixed strike options are more sensitive to stock price and volatility while floating strike options tend to have smooth and linear changes in option price. Delta analysis also suggests that fixed strike options react more towards changes in stock price while floating strike options are less reactive to stock prices.

Appendix A: Output for Pricing the 5 Options in Table 2

```
==== Option #1 ====
Call | arithmetic avg | fixed strike => (Price, Standard Error): (0.0000, 0.0000)
Put | arithmetic avg | fixed strike => (Price, Standard Error): (19.9102, 0.0331)
Call | arithmetic avg | floating strike => (Price, Standard Error): (1.3527, 0.0202)
Put | arithmetic avg | floating strike => (Price, Standard Error): (1.2789, 0.0183)
Call | geometric avg | fixed strike => (Price, Standard Error): (0.0000, 0.0000)
Put | geometric avg | fixed strike => (Price, Standard Error): (19.9629, 0.0332)
Call | geometric avg | floating strike => (Price, Standard Error): (1.3423, 0.0203)
Put | geometric avg | floating strike => (Price, Standard Error): (1.2984, 0.0184)

==== Option #2 ====
Call | arithmetic avg | fixed strike => (Price, Standard Error): (20.0203, 0.0332)
Put | arithmetic avg | fixed strike => (Price, Standard Error): (0.0000, 0.0000)
Call | arithmetic avg | floating strike => (Price, Standard Error): (1.3118, 0.0200)
Put | arithmetic avg | floating strike => (Price, Standard Error): (1.2789, 0.0183)
Call | geometric avg | fixed strike => (Price, Standard Error): (20.0264, 0.0336)
Put | geometric avg | fixed strike => (Price, Standard Error): (0.0000, 0.0000)
Call | geometric avg | floating strike => (Price, Standard Error): (1.3958, 0.0208)
Put | geometric avg | floating strike => (Price, Standard Error): (1.2799, 0.0183)
```

```
==== Option #3 ====
Call | arithmetic avg | fixed strike => (Price, Standard Error): (2.9331, 0.0444)
Put | arithmetic avg | fixed strike => (Price, Standard Error): (2.8087, 0.0395)
Call | arithmetic avg | floating strike => (Price, Standard Error): (3.0016, 0.0459)
Put | arithmetic avg | floating strike => (Price, Standard Error): (2.8295, 0.0387)
Call | geometric avg | fixed strike => (Price, Standard Error): (2.8380, 0.0436)
Put | geometric avg | fixed strike => (Price, Standard Error): (2.8690, 0.0400)
Call | geometric avg | floating strike => (Price, Standard Error): (2.9596, 0.0467)
Put | geometric avg | floating strike => (Price, Standard Error): (2.6558, 0.0375)

==== Option #4 ====
Call | arithmetic avg | fixed strike => (Price, Standard Error): (19.9313, 0.0354)
Put | arithmetic avg | fixed strike => (Price, Standard Error): (0.0000, 0.0000)
Call | arithmetic avg | floating strike => (Price, Standard Error): (1.4122, 0.0214)
Put | arithmetic avg | floating strike => (Price, Standard Error): (1.3333, 0.0191)
Call | geometric avg | fixed strike => (Price, Standard Error): (20.0728, 0.0352)
Put | geometric avg | fixed strike => (Price, Standard Error): (0.0000, 0.0000)
Call | geometric avg | floating strike => (Price, Standard Error): (1.4736, 0.0219)
Put | geometric avg | floating strike => (Price, Standard Error): (1.3227, 0.0190)
```

```
==== Option #5 ====
Call | arithmetic avg | fixed strike => (Price, Standard Error): (0.0014, 0.0009)
Put | arithmetic avg | fixed strike => (Price, Standard Error): (49.6650, 0.1154)
Call | arithmetic avg | floating strike => (Price, Standard Error): (4.5966, 0.0761)
Put | arithmetic avg | floating strike => (Price, Standard Error): (4.4146, 0.0592)
Call | geometric avg | fixed strike => (Price, Standard Error): (0.0017, 0.0011)
Put | geometric avg | fixed strike => (Price, Standard Error): (50.2904, 0.1144)
Call | geometric avg | floating strike => (Price, Standard Error): (4.8146, 0.0793)
Put | geometric avg | floating strike => (Price, Standard Error): (4.3569, 0.0577)
```