Step 1

Team member A

Q1. Pricing ATM European Call and Put options using Black-Scholes model

Question	Type	Exercise Type	GWP 1 Method	GWP 2 Method	GWP 1 Price	GWP 2 Price	%Diff
5	ATM Call	European	Binomial Tree	Black-Scholes	$4.61	$4.61	0%
5	ATM Put	European	Binomial Tree	Black-Scholes	$3.37	$3.37	0%
6	Call Delta	European	Binomial Tree	Black-Scholes	$0.57	$0.57	0%
6	Put Delta	European	Binomial Tree	Black-Scholes	$-0.43	-0.43	0%
7	Vega	European	Binomial Tree	Black-Scholes	19.66	19.64	2%

Q2. Pricing ATM European Call and Put options using Monte Carlo model

With the python code for ATM European, we got the option price, Delta and Vega as below. The data using Binomial Tree method are extracted from GPW.

Question	Type	Exercise Type	GWP 1 Method	GWP 2 Method	GWP 1 Price	GWP 2 Price	%Diff againstGWP1
5	ATM Call	European	Binomial Tree	Monte-Carlo	$4.61	$4.69	-1.73%
6	Call Delta	European	Binomial Tree	Monte-Carlo	$0.57	$0.57	0%
7	Call Vega	European	Binomial Tree	Monte-Carlo	19.66%	19.98%	1.63%

5	ATM Put	European	Binomial Tree	Monte-Carlo	3.78	3.42	-9.52%
6	Put Delta	European	Binomial Tree	Monte-Carlo	-0.43	-0.43	0%
7	Put Vega	European	Binomial Tree	Monte-Carlo	19,6	19.81	1.07%

In general, for the complex simulation, the Monte-Carlo method would give better results (but it costs at computation) (Stringham). However, in the case of call option, as can be seen from the above table, the difference between Binomial Tree and Monte-Carlo on price and Greek delta and Greek Vega are very small. But it is very different at Put option calculation when the price is nearly 10% differently.

Q3. Team Member C

Part A: Put-Call Parity Check for Black-Scholes and Monte Carlo

To check the put-call parity we do several steps which are, doing the black scholes option pricing to calculate the theoretical price of european call and put option, doing the monte carlo simulation for option pricing which we use GBM to simulate multiple stock price path and calculate the call and put option price, then we define a theoretical parity value between the Black Scholes and Monte Carlo model. We define, the theoretical parity value by the following formula (Chen):

We know that put-call Parity is hold when

$$C - P = S0 - K\,x\,e^{(-r\,x\,T)}$$

then

$$theoretical\ parity\ value = S0 - K\,x\,e^{(-r\,x\,T)},$$

where,

S0 = Initial stock price

K = Strike price

T = Time to maturity

r = Risk-free rate

From the question we know that the value of those variables are:

S0 = 100

K = 100

T = 0.25

r = 0.05

Then, we can calculate the theoretical parity value:

$theoretical\,parity\,value\ =\ S0\ -\ K\,x\,e^{(-r\,x\,T)} = 100\ -\ 100\,x\,e^{(-0.05\,x\,0.25)} = 1.24$

Then, we calculate the put-call parity difference as (Chen):

```
bs_parity_diff = round(bs_call_price - bs_put_price - parity_value,10)
mc_parity_diff = round(mc_call_price - mc_put_price - parity_value,10)
```

We want both to be as close as 0. The result of the Put-Call Parity Check are:

Black-Scholes:

Call Price: 4.61

Put Price: 3.37

Monte Carlo:

Call Price: 4.60

Put Price: 3.38

Put-Call Parity:

Theoretical Value: 1.24

Black-Scholes Difference: 0.00

Monte Carlo Difference: -0.02

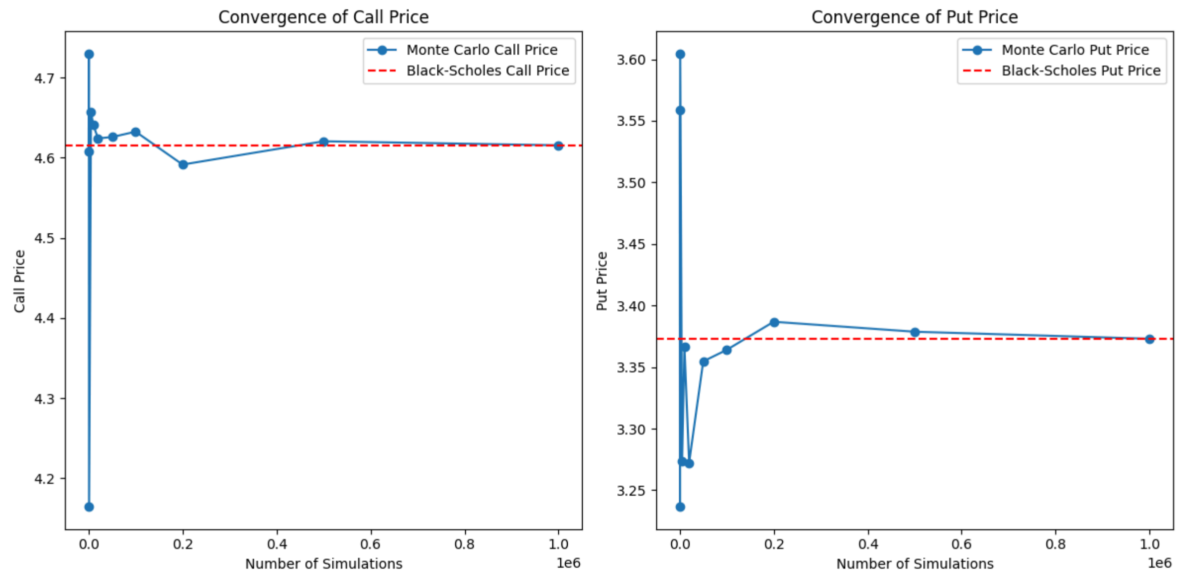
Then using automatic reasoning we got that the Black-Scholes and Monte-Carlo satisfy Put-Call Parity with very small difference (see the python.ipynb for more details).

So, what we can analyze is that bothe Black-Scholes model and Monte Carlo model satisfy Put-Call Parity. We can analyze that the Monte Carlo simulation result is close to Black-Scholes model result indicating that those two models are good. There's small deviation in monte carlo's put-call parity due to the simulation-based nature of it.

Part B: Comparison and discussion the prices obtained

Theoretically, monte carlo option pricing will converge to Black-Scholes model as the number of simulations increase due to law of large numbers. As explained above, the Black Scholes model provides a closed form solution to option pricing. Meanwhile, monte carlo estimated option pricing by simulating many future paths of underlying asset then averaging the results, as the number of paths increase the average price will be approximate the black scholes result (Guo).

To analyze the convergence we try 100 simulations to 1 million simulations. We use 1 million simulations because at that point the Monte Carlo model has reached its stability both in put and call option pricing. Below is the result of the simulation in the form of a plot. In the left one we have the plot of convergence of call option price using monte carlo compared to black-scholes call price, while in the right we have the convergence plot of put option price of monte carlo model to black scholes put option pricing result.



Based on above plot we can analyze that both call and put prices show initial volatility in price but eventually they are converging to their respective Black-Scholes values which is the theory behind Monte-Carlo where it tells us that in the long run the option pricing result will converge to Black-Scholes model (Guo), we also get the same analysis based on above plot where the final values of both Put and Call option of Monte Carlo model are closely matching the Black-Scholes price line (red dashed line). We can analyze that there's initial volatility due to random nature of simulation, we can also analyze that both options have symmetric convergence around their theoretical values. So, in short, both Call options and Put options that are priced with Monte Carlo are converging to Black Scholes model.

Step 2

Team member A

Q4. Use Monte-Carlo methods with regular GBM process and daily simulations on an American Call option.

The code for doing the monte-carlo simulation for the American call option is based on the following paper: Valuing American Options by Simulation: A Simple Least-Squares Approach. Basically it uses regression to develop the optimal early exercise strategy for American options, which probably doesn't matter for call option (since early exercise is not optimal in this case).
Additionally, I choose a small increase in stock price and volatility (5% in line with what was in GWP 1) and we use 252 days (daily simulations) for MC, which is different to the number of steps used in binomial tree.

Question	Type	Exercise Type	GWP 1 Method	GWP 2 Method	GWP 1 Price	GWP 2 Price	%Diff
5	ATM Call	American	Binomial Tree	Monte-Carlo	$4.61	$4.56	-1.09%
6	Call Delta	American	Binomial Tree	Monte-Carlo	$0.57	$1.19	109%
7	Call Vega	American	Binomial Tree	Monte-Carlo	19.67	19.89	1.19%

On the aspect of Price and Vega, the difference is very small (~+-1%). However, the disparity between Delta outputs might be due to the use of monte-carlo (MC) instead of an exact solution in the binomial tree.

Team member B

Q5. Monte-Carlo methods with regular GBM process and daily simulations on an American Put option.

The table below includes data of Put option using Binomial Tree (extracted from our GPW1 report) and Monte-Carlo methods with regular GBM process.

Question	Type	Exercise Type	GWP 1 Method	GWP 2 Method	GWP 1 Price	GWP 2 Price	%Diff
5	Put Price	American	Binomial Tree	Monte-Carlo	3.48	3.53	1.44%
6	Put Delta	American	Binomial Tree	Monte-Carlo	-0.45	-0.48	-6.67%
7	Put Vega	American	Binomial Tree	Monte-Carlo	19.6	21.62	10.31%

It is interesting to see that the difference between Monte-Carlo and Binomial Tree method on the price are not significant, but the differences on Greek Delta and Greek Vega are noticeable - 6.67% and 10.31% respectively.

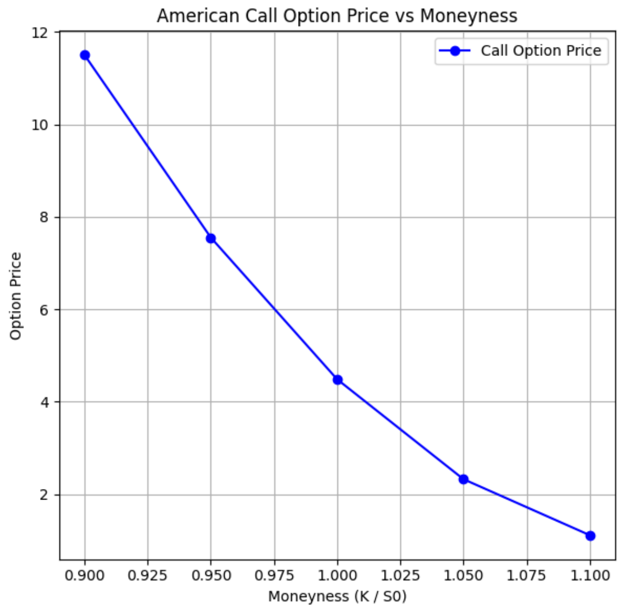
From both outputs of Call and Put options using Binomial Tree and Monte Carlo, the prices would be very similar, but the sensitive data (Delta and Vega) would be very different.

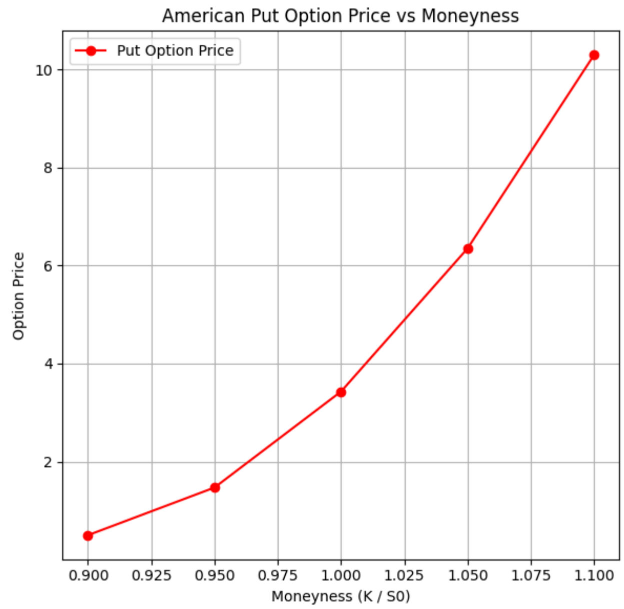
Team member C

Q6. Graph the relationship between option price and moneyness

Based on numbers 15 and 16, we know that the money we are going to use is 90%, 95%, 100%, 105%, 110%. We also have the code for option pricing from previous parts Q4 and Q5 for American Put and American Call using Monte Carlo SImulations. So, we calculate prices of American Call and Put Options for various moneyness levels. This will give us the idea how option prices change relative to the current price of the asset as the strike price shifts. The result that we got:

Moneyness	Call Price	Put Price
90%	11.5	0.5
95%	7.55	1.47
100%	4.49	3.43
105%	2.32	6.35
110%	1.11	10.29





From the result we can analyze that the call price decreases when the moneyness increases, this makes sense because as the strike prices higher the calls are less valuable. While the put price increases as moneyness increases, this also makes sense because as the strike price higher the Put becomes valuable. We can also analyze that the pricing curves show that there's expected convexity due to option time value. For trading we can analyze that the Deep ITM options are more expensive but have higher delta, while the OTM options are cheaper but have lower probability to expire in the money.

Step 3

Team member A

Q7.

a. **Pricing European options with same characteristics as GWP#1 under different levels of moneyness:**

Moneyness	Exercise Type	GWP 1 Method	GWP 2 Method	GWP 1 Price	GWP 2 Price	%Diff
110%	European call	Trinomial Tree	Black-Scholes	$1.19	$1.19	0%
95%	European put	Trinomial Tree	Black-Scholes	$1.53	$1.53	0%

b. **Build a portfolio that buys the previous Call and Put options**

For part b) since Delta is linear, the Delta of a portfolio is simply the sum of the deltas of the call and put options in part b) and the difference in part c)

As the delta of the portfolio in b) is negative, this portfolio benefits slightly from a decline in the underlying stock price. Hence, to delta-hedge this portfolio, we would need to buy approximately 0.0275 shares of the underlying stock per portfolio unit.

c. **Build a second portfolio that buys the previous Call option and sells the Put.**

As the delta of the portfolio in c) is positive, this portfolio benefits from an increase in the underlying stock price. Hence, to delta-hedge this portfolio, we would need to short approximately 0.464 shares of the underlying stock per portfolio unit.

Team member B

Q8. Work with Monte-Carlo methods with daily time steps to price an Up-and-Out (UAO) barrier option.

An Up-and-out option is a type of options contract that will exist if the underlying moves above a certain price point (a barrier) (Mitchell and Kvilhaug). In this question, the Monte-Carlo method is combined with an Up-and-out barrier option which allows the Monte-Carlo loop to exist when the stock price is over the barrier value.
From the python code, the Up-and-Out Barrier Option Price is 0.68. The next question would give more interpretation of Up-and-Out Barrier Option.

Team member C

Q9. Repeat Q8 and Consider Up and In Barrier (UAI) Option

Based on Scott, Up and In Barrier Option is a type of exotic option. Its structures are more complex than the standard vanilla options. This exotic option is often traded in OTC Markets, which makes the variation of this kind of option high. This option in some condition of liquidity of the underlying (forex or stocks), may be offered in bespoke manner (Scott). In addition, UAI is a special kind of option that depends not only on the final asset price but also on whether the price crosses a certain level ("barrier") during the option's life. We price the UAI option by simulating asset price, then we calculate the payoff, then we calculate the option price. The differences are in the payoff section. We calculate the payoff by giving condition that payoff is based solely on the difference between final asset price and strike price (Zhou et.al): The payoff of barrier option happens when:

- Up-and-Out (UAO): If price hits the barrier, option is canceled (no payoff).
- Up-and-In (UAI): option only pays if the barrier was hit at some point.

Then the result of calculating of the option price of UAO and UAI is:

- Up-and-In Barrier Option (UAI) Price: 13.13
- Up-and-Out Barrier Option (UAO) Price: 0.68

We also calculate the vanilla option price as the question given and we got:

- Vanilla Option Price: 13.81

Then, the relationships between prices are:

A. Price of the Vanilla option is usually higher than both UAI and UAO options because the Vanilla option doesn't have barrier restrictions.
B. UAO (Up and Out) option price is usually lower than Vanilla option since it "knocks out" if the barrier level is breached, reducing chance of payoff.
C. The UAI (Up and In) option price usually between Vanilla and UAO prices because it only has payoff when the barrier is breached, when the barrier is not breached the option doesn't have value.
D. Usually, UAI Price + UAO Price ≈ Vanilla Price.

From what we got from our calculation the price of UAO is indeed the lowest, follows up by UAI in the middle and the Vanilla is the most expensive one, and the price of UAI, UAO and Vanilla indeed fulfill the equation of UAI Price + UAO Price ≈ Vanilla Price. Which implies that what we have done is correct.

References

Mitchell, Cory, and Suzanne Kvilhaug. “Up-and-Out Option: What it is, How it Works, Example.” *Investopedia*, 2024, https://www.investopedia.com/terms/u/up-and-outoption.asp. Accessed 12 November 2024.

Stringham, Talon C. “Option Pricing: Black-Scholes v Binomial v Monte Carlo Simulation.” *LinkedIn*, 13 February 2015, https://www.linkedin.com/pulse/option-pricing-black-scholes-v-binomial-monte-carlo-stringham/. Accessed 12 November 2024.

Francis A. Longstaff; Eduardo S. Schwartz. The Review of Financial Studies, Vol. 14, No. 1. (Spring, 2001), pp. 113-147

Guo, Taoer. "Black-Scholes Process and Monte Carlo Simulation-Based Options Pricing." *Shanghai Weiyu International High School*, 10664820@qq.com, 2021.

Chen, James. “Put-Call Parity: Definition, Formula, How It Works, and Examples.” Investopedia, Investopedia, www.investopedia.com/terms/p/putcallparity.asp. Accessed 13 Nov. 2024.

Scott, Gordon. “Up-and-In Option: What it Means, How it Works.” Investopedia, 11 May 2022, www.investopedia.com/terms/u/up-and-inoption.asp.

Zhou, Shaoling, Huixin Chai, and Xiaosheng Wang. "Barrier Option Pricing with Floating Interest Rate Based on Uncertain Exponential Ornstein–Uhlenbeck Model." AIMS Mathematics, vol. 9, 2024, pp. 25809-25833, doi:10.3934/math.20241261.