UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

# MIE1622H
Computational Finance and Risk Management

# Assignment 4 – Asset Pricing

Professor: Dr. Oleksandr Romanko

Student name: Jianhui Li

Student number:1002116907

Date: April 4, 2020

## Introduction

The purpose of this assignment is to determine the value of call and put price of European option and Barrier knock-in option of underlying stock. The value of the option will be computed by using Black-Scholes pricing formula, Monte Carlo pricing procedure respectively. There are three pricing functions are implemented in this assignment, including BS_european_price, MC_european_price and MC_barrier_knockin_price. For the Monte Carlo simulation pricing procedure, the time steps and a number of scenarios are set manually for computations. We will try one-step and multi-step for MC simulation. After that, three pricing strategies for European options are compared to analyze the differences and discuss their performance relative to each other, which include Black-Scholes price of European option, One-step MC price of European option and Multi-step MC price of European option. Then the European and Barrier options need to compare to analyze the difference. At last, we compute prices of Barrier options with volatility increased and decreased by 10% from the original inputs to see the difference.

# Implementation

*Black-Scholes pricing formula for European option*

European option mean the timeframe when holders of an options contract may exercise their contract rights. The rights for the option holder include buying the underlying asset or selling the underlying asset at the specified contract price—the strike price. With European options, the holder may only exercise their rights on the day of expiration. In this function, Black–Scholes formula is used, which gives a theoretical estimate of the price of European-style options and shows that the option has a unique price regardless of the risk of the security and its expected return. I assume that the price of the underlying asset follows Geometric Brownian Motion model. The value of a European call or put option for a non-dividend-paying underlying stock is given by the solution of the Black-Scholes equation:

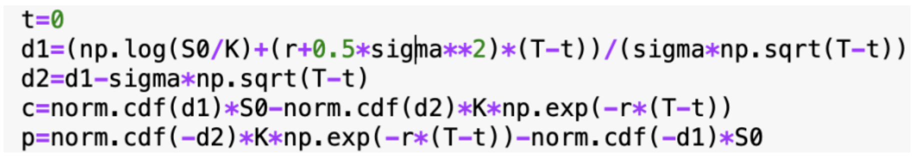
$$\begin{aligned} P(S,t) &= Ke^{-r(T-t)} - S + C(S,t) \\ &= \mathcal{N}(-d_2)Ke^{-r(T-t)} - \mathcal{N}(-d_1)S \end{aligned}$$

$$C(S,t) = \mathcal{N}(d_1)S - \mathcal{N}(d_2)Ke^{-r(T-t)}$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right]$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

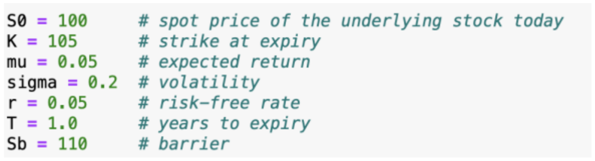
The code was written as:



```
t=0
d1=(np.log(S0/K)+(r+0.5*sigma**2)*(T-t))/(sigma*np.sqrt(T-t))
d2=d1-sigma*np.sqrt(T-t)
c=norm.cdf(d1)*S0-norm.cdf(d2)*K*np.exp(-r*(T-t))
p=norm.cdf(-d2)*K*np.exp(-r*(T-t))-norm.cdf(-d1)*S0
```

*Monte Carlo pricing procedure for European option*

A price of an underlying stock can be modeled by Monte Carlo simulations using Geometric Brownian Motion (GBM) with constant drift μ and volatility σ. The discretized version of GBM is known as Geometric Random Walk equation: $S_{t+1} = S_t \cdot e^{(\mu - \frac{1}{2} \cdot \sigma^2) + \sigma \cdot \epsilon_t}$. The prices of the European option is calculated from Monte Carlo simulations by computing call and put payoffs and discounting those back to current time with the riskless short rate r. Price of underlying stock today (t = 0), i.e., spot price today, is S0 = 100, strike at expiry is K = 105, years to expiry T = 1, risk-free rate r = 0.05, drift μ = 0.05, volatility σ = 0.2. The barrier is $110 for this assignment and all other variables. This is written in code:

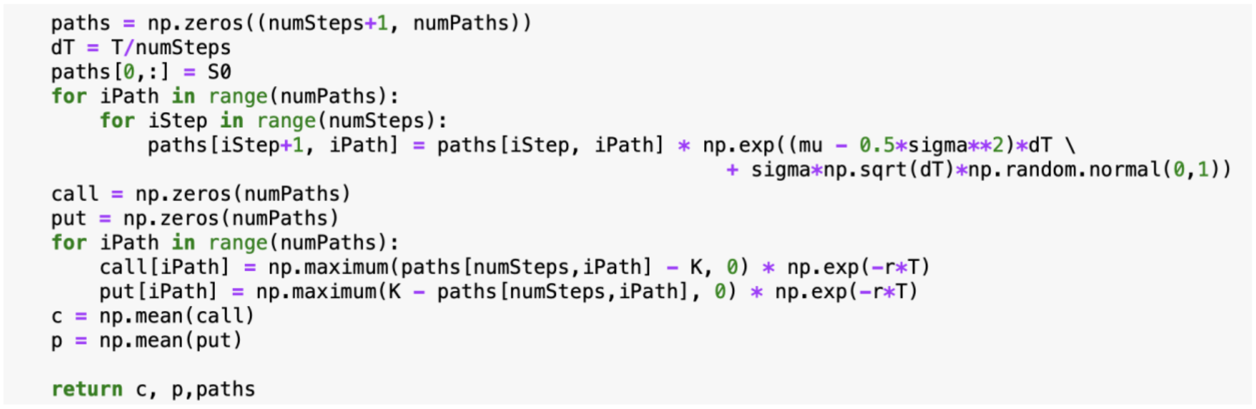


```
S0 = 100      # spot price of the underlying stock today
K = 105       # strike at expiry
mu = 0.05     # expected return
sigma = 0.2   # volatility
r = 0.05      # risk-free rate
T = 1.0       # years to expiry
Sb = 110      # barrier
```

To implement Monta Carlo simulation, the exact expression for the asset price can be obtained by using by using Ito's Lemma: $S_t = S_0 \cdot e^{(\mu - \frac{1}{2} \cdot \sigma^2) \cdot t + \sigma \cdot \sqrt{t} \cdot \epsilon}$ . After that, the vaule of European call and put option is calculated for each scenario.

$$P_c(T) = \max\{S(T) - K, 0\}, P_p(T) = \max\{K - S(T), 0\}$$

Then calculating the mean of all price of call and put options to get the simulated price. The code for Monta Carlo simulation pricing for put and call of European option is



```
    paths = np.zeros((numSteps+1, numPaths))
    dT = T/numSteps
    paths[0,:] = S0
    for iPath in range(numPaths):
        for iStep in range(numSteps):
            paths[iStep+1, iPath] = paths[iStep, iPath] * np.exp((mu - 0.5*sigma**2)*dT \
                                                  + sigma*np.sqrt(dT)*np.random.normal(0,1))
    call = np.zeros(numPaths)
    put = np.zeros(numPaths)
    for iPath in range(numPaths):
        call[iPath] = np.maximum(paths[numSteps,iPath] - K, 0) * np.exp(-r*T)
        put[iPath] = np.maximum(K - paths[numSteps,iPath], 0) * np.exp(-r*T)
    c = np.mean(call)
    p = np.mean(put)

    return c, p,paths
```

In this function, two methods are implemented, which are one-step and multi-step. One-step means if maturity of an option on a stock is 1 year from today, I can compute a price of an underlying stock once at the end of the year. Multi-step means computing the price of an underlying every week, every month, every day of the year, in other words I chop the time interval into as many units as we need. Usually, the multi-step can choose 2, 12, 24, 52, or 252 that represent half-year, every month, every half month, every week, and every day.

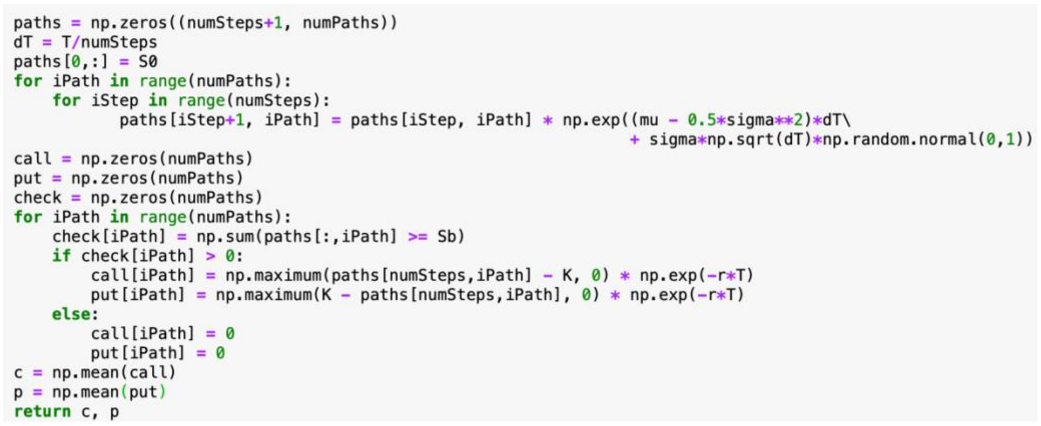
## *Monte Carlo pricing procedure for Barrier knock-in option*

A knock-in option is a type of contract that is not an option until a certain price is met. So if the price is never reached, it is as if the contract never existed. However, if the underlying asset reaches a specified barrier, the knock-in option comes into existence. The difference between a knock-in and knock-out option is that a knock-in option comes into existence only when the underlying security reaches a barrier, while a knock-out option ceases to exist when the underlying security reaches a barrier. Therefore, this function is going to use the same method to simulate the price of the call and put option by using Monte Carlo simulation. The equation is same as previous function: $S_t = S_0 \cdot e^{(\mu - \frac{1}{2}\cdot\sigma^2)\cdot t + \sigma\cdot\sqrt{t}\cdot\epsilon}$ . Once multiple asset paths for GBM have been simulated, the next step is to determine the payoff for each asset path. This is done by evaluating each path to see whether it has hit the barrier. Barrier knock-in option is used to find the price of call and put option. If the maximum of each path exceeds the strike price means that we need to pay out if the barrier is crossed during the life of the option. If not, the price of call and put option is zero, which means no trade.

$$P_c^B(T) = \max\{S(T) - K, 0\}\mathbb{1}_{\{\max_t\{S(t)\}>B\}}$$

$$P_p^B(T) = \max\{K - S(T), 0\}\mathbb{1}_{\{\max_t\{S(t)\}>B\}}$$

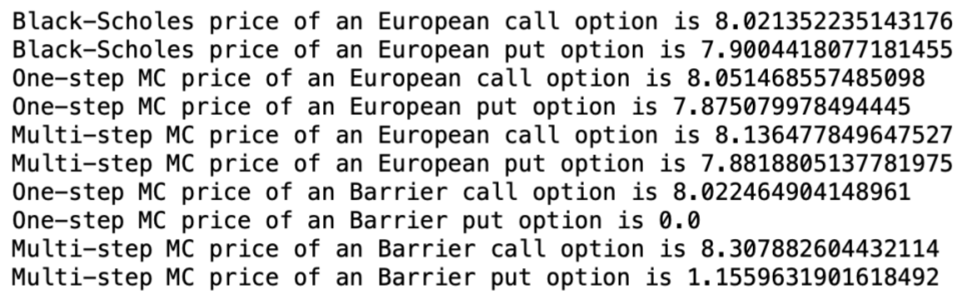
The code is shown below:



```
paths = np.zeros((numSteps+1, numPaths))
dT = T/numSteps
paths[0,:] = S0
for iPath in range(numPaths):
    for iStep in range(numSteps):
           paths[iStep+1, iPath] = paths[iStep, iPath] * np.exp((mu - 0.5*sigma**2)*dT\
                                                     + sigma*np.sqrt(dT)*np.random.normal(0,1))
call = np.zeros(numPaths)
put = np.zeros(numPaths)
check = np.zeros(numPaths)
for iPath in range(numPaths):
    check[iPath] = np.sum(paths[:,iPath] >= Sb)
    if check[iPath] > 0:
        call[iPath] = np.maximum(paths[numSteps,iPath] - K, 0) * np.exp(-r*T)
        put[iPath] = np.maximum(K - paths[numSteps,iPath], 0) * np.exp(-r*T)
    else:
        call[iPath] = 0
        put[iPath] = 0
c = np.mean(call)
p = np.mean(put)
return c, p
```

## Result and Discussion

*Results of European Option and Barrier-in Option*



```
Black–Scholes price of an European call option is 8.021352235143176
Black–Scholes price of an European put option is 7.9004418077181455
One–step MC price of an European call option is 8.051468557485098
One–step MC price of an European put option is 7.875079978494445
Multi–step MC price of an European call option is 8.136477849647527
Multi–step MC price of an European put option is 7.8818805137781975
One–step MC price of an Barrier call option is 8.022464904148961
One–step MC price of an Barrier put option is 0.0
Multi–step MC price of an Barrier call option is 8.307882604432114
Multi–step MC price of an Barrier put option is 1.1559631901618492
```

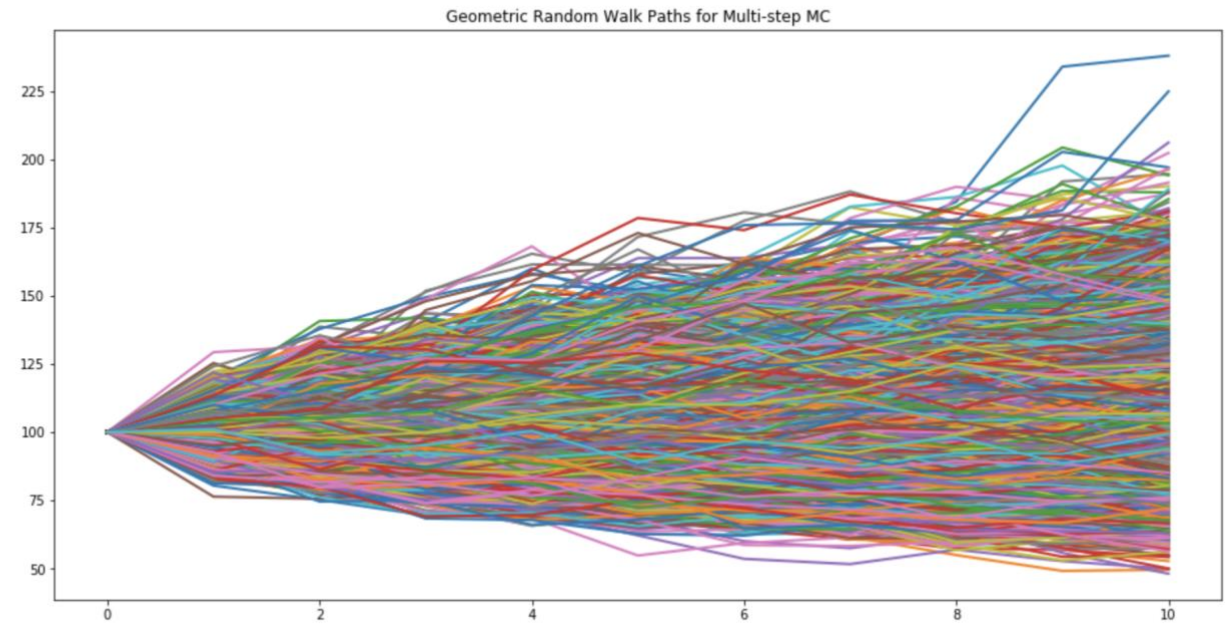
In one-step Monte Carlo pricing procedure for European option, number of Paths is 10000, number of step is 1

In multi-step Monte Carlo pricing procedure for European option, number Paths is 10000, number of steps is 10, which means 10 times to  compute the price of an underlying stock every year.

In my one-step Monte Carlo pricing procedure for Barrier option, number of Paths is 10000, number of step is 1.

In my multi-step Monte Carlo pricing procedure for Barrier option, number Paths is 10000, number of steps is 10.

*Plot one chart in Python that illustrates your Monte Carlo pricing procedure is the best way*



This plot demonstrates the Geometric Random Walk Paths for Multi-step Monte Carlo. There are 10000 scenarios for the underlying stocks, and 10 times per year to calculate the price of underlying stocks. These 10000 paths spread from 225 to 50 (prices). The x-axis represents the times of calculating underlying stocks. Overall, this plot simulates the 10000 scenarios from Monte Carlo simulation, with spot price of the underlying stock today 100, expected return 5%, volatility 0.2 , years to expiry 1. It shows how MC pricing procedure to generate all scenarios. Next step is to get the mean of all scenarios as the price of call and put option.

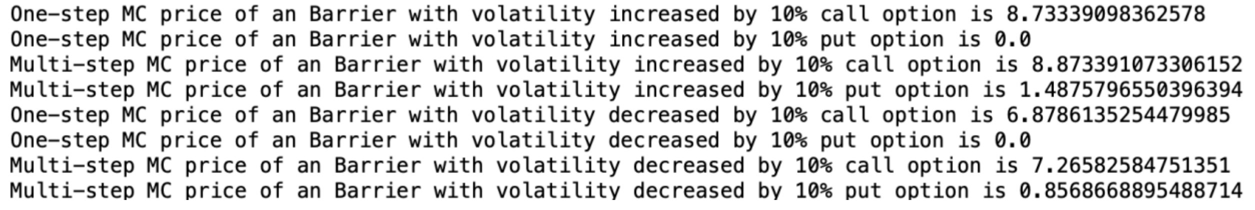
*Compare three pricing strategies for European option*

The Monte Carlo simulation results are very closed to the closed form of Black Scholes equation results. This is due to the price of the underlying asset follows Geometric Brownian Motion model. Therefore, we can use closed form of Black Scholes equation to get a very close value compared to MC simulation procedure.  Between multi-step MC and one-step MC for European option, the difference is very small. I can't tell the exactly differences between these two methods, only the MC price of European option is higher than the other one. The algorithm of BS and MC are quite different. For the Black-Scholes equation, all of the variables are fixed, which indicates that the variables will not change during its calculation process. However, the MC simulation is based on the random variable generation in a normal probability distribution. The variables will be fluctuating between 0 to 1 during the calculation process and therefore the determination process is not fixed. Although the methods algorithms are different, their results are very similar to each other within an acceptable range of estimation error.

*Explain the difference between call and put prices obtained for European and Barrier options*

From my results, the price of barrier call option and European call option do not have much difference. Only the barrier put option is very smaller than the European put option. It make senses due to the barrier is 110 dollars, but the strike price is 105 dollar. When the barrier is breached, no one like to sell their option at 105, because the market price is around 110. Therefore, when the barrier knocks in, there is almost very few chances the price will go down to the 105. That's why the Barrier put option is very cheap comparing to the European put option. No one likes to buy an option looks like useless.

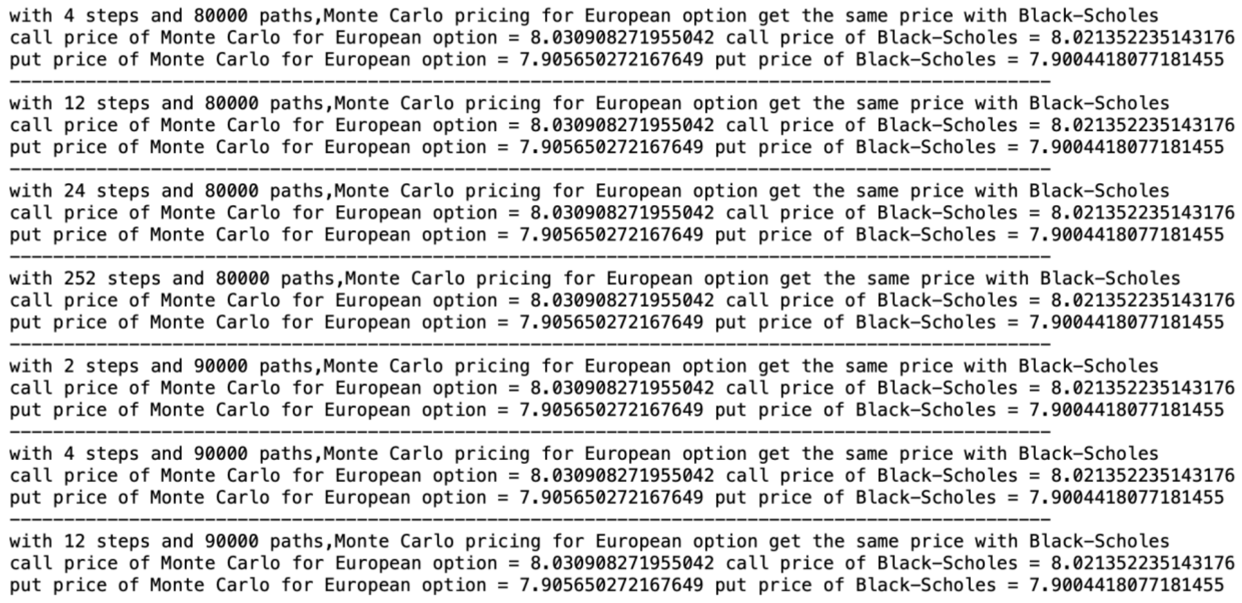
*prices of Barrier options with volatility increased and decreased by 10% from the original inputs*



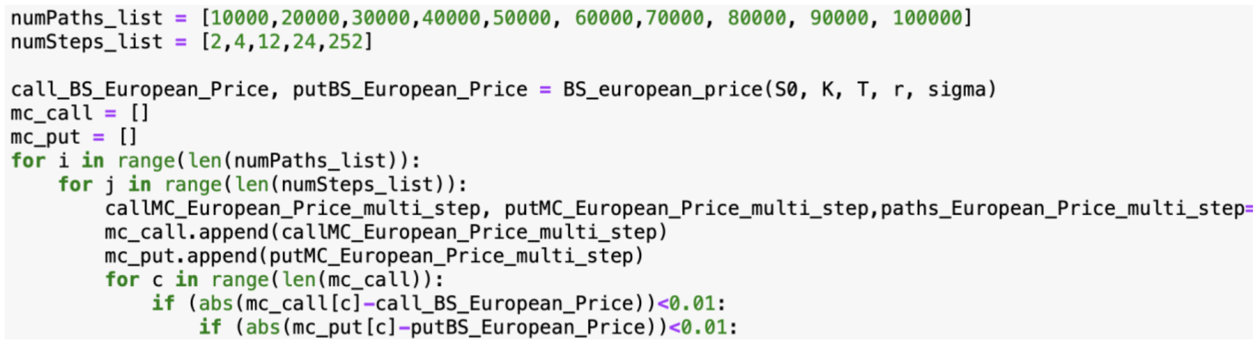
```
One-step MC price of an Barrier with volatility increased by 10% call option is 8.73339098362578
One-step MC price of an Barrier with volatility increased by 10% put option is 0.0
Multi-step MC price of an Barrier with volatility increased by 10% call option is 8.873391073306152
Multi-step MC price of an Barrier with volatility increased by 10% put option is 1.4875796550396394
One-step MC price of an Barrier with volatility decreased by 10% call option is 6.8786135254479985
One-step MC price of an Barrier with volatility decreased by 10% put option is 0.0
Multi-step MC price of an Barrier with volatility decreased by 10% call option is 7.26582584751351
Multi-step MC price of an Barrier with volatility decreased by 10% put option is 0.8568668895488714
```

If the volatility goes up by 10%, the price of Barrier-in call options increase, and price of Barrier-in put option decrease. If the volatility goes down by 10%, the price of Barrier-in call options decrease, and price of Barrier-in put option also decrease. My interpretation is when volatility increase, the price of stock will have more chance to reach the barrier, and reach a higher price. Therefore, the call option increase because people believe they will make more money. In contrast, when volatility decrease, the price of stock will have less chances to reach barrier, even though, it reach the barrier, the price of stock won't go so high. As to the put option, if people think the price of stock will go much higher than the barrier, also means it can't go down anymore. The price of put option will because zero. If the volatility goes down, once the barrier is breached, the price of stock may become very stable, which means it is hard to go down. As the same result, the price of put option will because zero because people don't like them.

## Possible Strategies



```
with 4 steps and 80000 paths,Monte Carlo pricing for European option get the same price with Black-Scholes
call price of Monte Carlo for European option = 8.030908271955042 call price of Black-Scholes = 8.021352235143176
put price of Monte Carlo for European option = 7.905650272167649 put price of Black-Scholes = 7.9004418077181455
------------------------------------------------------------------------------------------------
with 12 steps and 80000 paths,Monte Carlo pricing for European option get the same price with Black-Scholes
call price of Monte Carlo for European option = 8.030908271955042 call price of Black-Scholes = 8.021352235143176
put price of Monte Carlo for European option = 7.905650272167649 put price of Black-Scholes = 7.9004418077181455
------------------------------------------------------------------------------------------------
with 24 steps and 80000 paths,Monte Carlo pricing for European option get the same price with Black-Scholes
call price of Monte Carlo for European option = 8.030908271955042 call price of Black-Scholes = 8.021352235143176
put price of Monte Carlo for European option = 7.905650272167649 put price of Black-Scholes = 7.9004418077181455
------------------------------------------------------------------------------------------------
with 252 steps and 80000 paths,Monte Carlo pricing for European option get the same price with Black-Scholes
call price of Monte Carlo for European option = 8.030908271955042 call price of Black-Scholes = 8.021352235143176
put price of Monte Carlo for European option = 7.905650272167649 put price of Black-Scholes = 7.9004418077181455
------------------------------------------------------------------------------------------------
with 2 steps and 90000 paths,Monte Carlo pricing for European option get the same price with Black-Scholes
call price of Monte Carlo for European option = 8.030908271955042 call price of Black-Scholes = 8.021352235143176
put price of Monte Carlo for European option = 7.905650272167649 put price of Black-Scholes = 7.9004418077181455
------------------------------------------------------------------------------------------------
with 4 steps and 90000 paths,Monte Carlo pricing for European option get the same price with Black-Scholes
call price of Monte Carlo for European option = 8.030908271955042 call price of Black-Scholes = 8.021352235143176
put price of Monte Carlo for European option = 7.905650272167649 put price of Black-Scholes = 7.9004418077181455
------------------------------------------------------------------------------------------------
with 12 steps and 90000 paths,Monte Carlo pricing for European option get the same price with Black-Scholes
call price of Monte Carlo for European option = 8.030908271955042 call price of Black-Scholes = 8.021352235143176
put price of Monte Carlo for European option = 7.905650272167649 put price of Black-Scholes = 7.9004418077181455
```



```python
numPaths_list = [10000,20000,30000,40000,50000, 60000,70000, 80000, 90000, 100000]
numSteps_list = [2,4,12,24,252]

call_BS_European_Price, putBS_European_Price = BS_european_price(S0, K, T, r, sigma)
mc_call = []
mc_put = []
for i in range(len(numPaths_list)):
    for j in range(len(numSteps_list)):
        callMC_European_Price_multi_step, putMC_European_Price_multi_step,paths_European_Price_multi_step=
        mc_call.append(callMC_European_Price_multi_step)
        mc_put.append(putMC_European_Price_multi_step)
        for c in range(len(mc_call)):
            if (abs(mc_call[c]-call_BS_European_Price))<0.01:
                if (abs(mc_put[c]-putBS_European_Price))<0.01:
```

The purpose of this section is to design a procedure to make the prices between the BS and the MC model become the same price (up to cent). The BS model is a fixed variable algorithm, therefore, I design a process for search the best number of scenarios and number of steps to allow Monte Carlo simulation to achieve the same price as the Black Scholes Formula. The algorithm looks like grid search, which is a simple loop to search both numPath and numStep. In this part, I get the result that when numPath is 80000 and 90000, the numStep can be 4, 12 , 24, 252. This part can tell us MC simulation can give us a very close value to BS Formula in some specific scenarios.