Math942 Numerical Methods in Finance

Assignment IV: Monte Carlo Method for Option Pricing

Preface: The Monte Carlo Method itself is an extremely broad class of computational algorithms that are performing random sampling to obtain numerical results from a huge number of repetitions to generate distributions of said random sampling. This allows us to price Options under the Martingale approach by generating tens of thousands of random paths according to Geometric Brownian Motion, which is assumed that underlying stock prices follow for many other models, including the Black Scholes Merton Model. The Stochastic process (SDE) we have used is defined below.

$$dS_t = \mu S_t + \sigma S_t dW_t$$

$$Where\ W_t\ is\ the\ Wiener\ process$$

With solution to said SDE,

$$ln\frac{S_t}{S_0} = \left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t$$

It is also important to note the computational inefficiency of MC methods in comparison to using standard solutions from the BSM PDE. In general, you will see that as we **increase the number of simulations, the computational time will have a linear increase** when keeping time steps constant.

1. Monte Carlo Computational Algorithm

```
def MCarlo(sim, T, N, r, vol, K, S0):
    dt = T/N
    nudt = (r - 0.5*vol**2)*dt
    volsdt = vol*np.sqrt(dt)
    lnS0 = np.log(S0)

    sum_CT = 0


    start = time.time() #Record start time

    for i in range(sim): #How many Simulations

        lnSt = lnS0

        for j in range(N):#Time steps
            lnSt = lnSt + nudt + volsdt*np.random.normal() #GMB

        ST = np.exp(lnSt)
        CT = max(0, ST - K)
        sum_CT = sum_CT + CT


    C0 = np.exp(-r*T)*sum_CT/sim #Compute Expected Payoff



    stop = time.time() #End computation time
    duration = stop-start #Time to complete computation with number of sim

    return C0, duration
```

2. Price Comparison

All calculations have used the below inputs.

S0	K	T	r	volatility	Time Step	BS Price
100	100	1	0.01	0.2	100	8.43

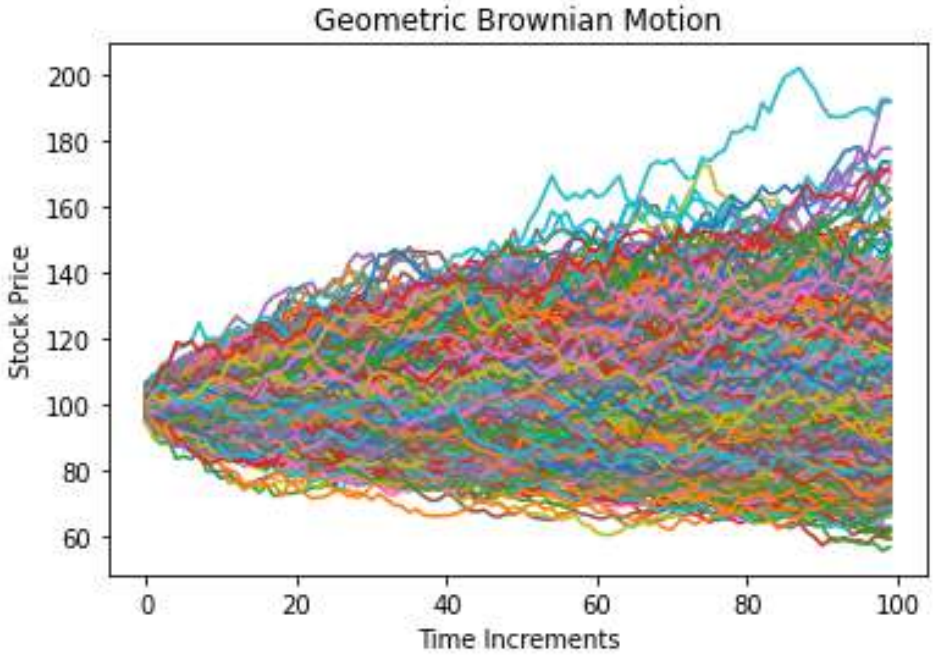

Figure 1: Thousands of samples from the SDE describing Geometric Brownian Motion

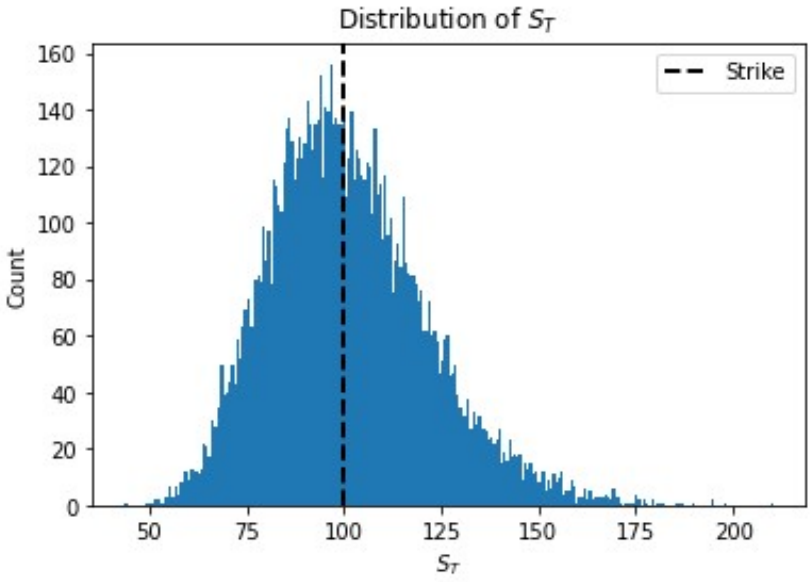

Figure 2: Distribution of the underlying Stock price

2.1 Not Enough Simulations

It is clear that without enough number of Simulations, the Price given by the MC Method is very unstable. This is because to generate an accurate distribution of price paths, we need a huge number of samples from the above SDE. However, this comes at a cost of computational speed.

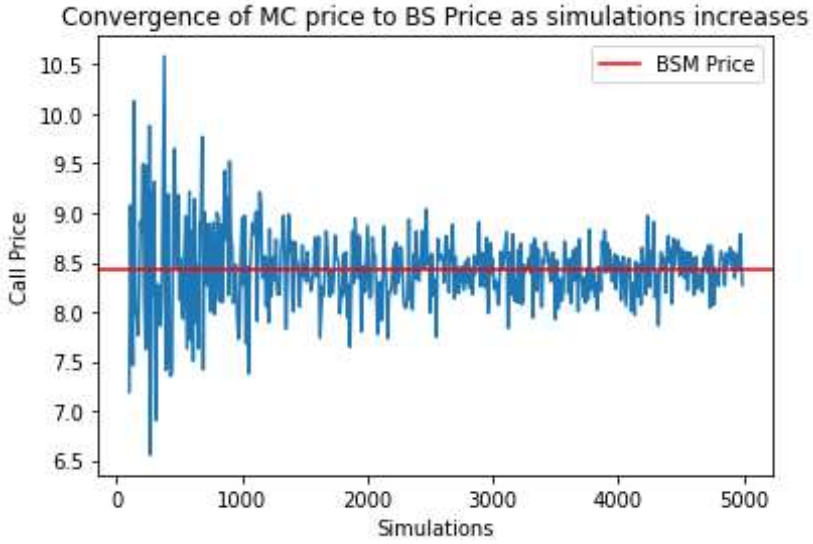

Figure 3: MC Carlo prices as function of simulations. It is clear that <5000 simulations is insufficient to give a stable price. The approximate prices below <2000 simulations range from between 10.5 and 6.5.

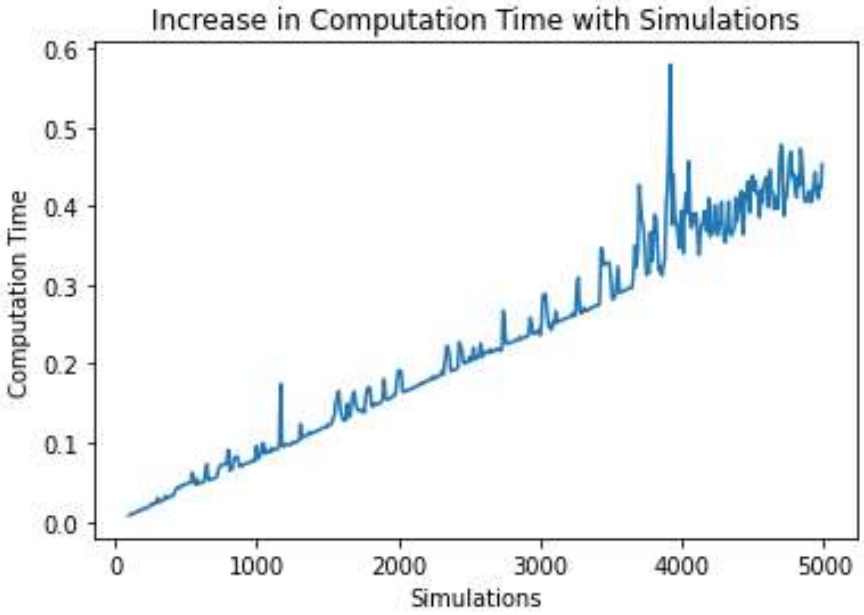

Figure 4: Computational times as function of Simulations when keeping time steps constant.

2.2 Appropriate number of Simulations

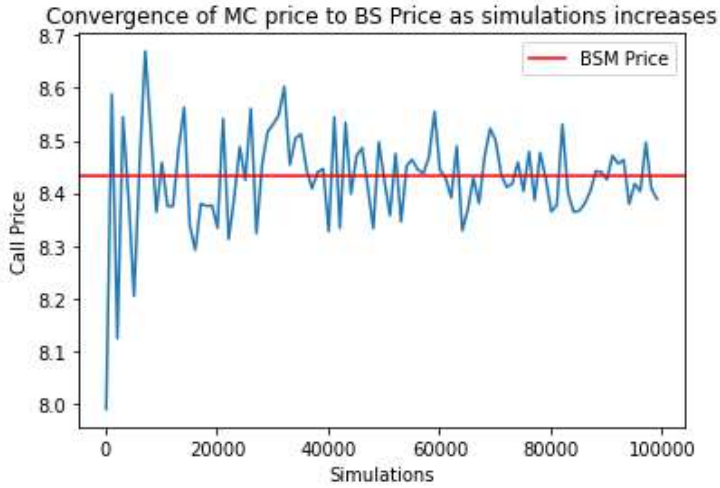

Figure 5: Price Convergence when using up to 100,000 simulations. Notice that the range for prices is much more compact compared to Figure 1

Column1	
Mean	8.419272
Standard Error	0.011082
Median	8.426467
Mode	#N/A
Standard Deviation	0.110819
Sample Variance	0.012281
Kurtosis	9.949271
Skewness	-2.64494
Range	0.740863
Minimum	7.844098
Maximum	8.584961

Figure 6: Descriptive Statistics on the Price Distribution given by the Monte Carlo Method when using up to 100,000 simulations which took a grand total of ~ 428.5299 seconds or ~7.13 minutes .

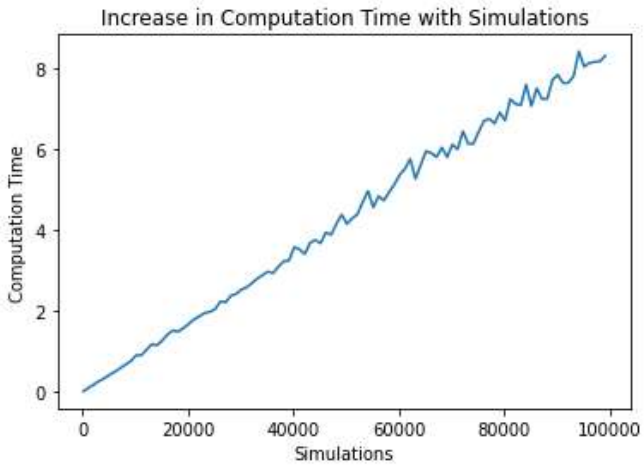

Figure 7: Linear Increase in Computation Time still present

In fact, we can fit easily fit a straight line through this as it is clearly following a linear trend, which we can obtain as,

$$y = (9E - 05)x - 0.016$$

Thus, with time steps set to 100, we can expect our rate of increase in computational time to be 9E-05 time units, which in short means every 1 unit increase in simulations, gives a 9E-05 unit computational time increase. (To do this we very crudely exported the data into an excel file and fitted a straight line through). Also it is immediately obvious if we increased time steps, computational time would also increase

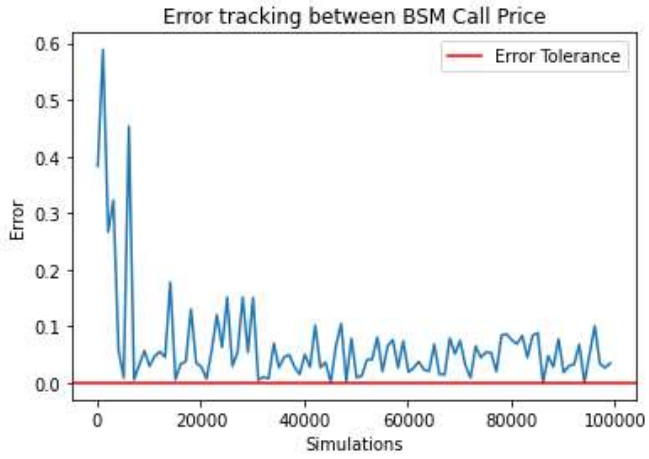

Figure 8: Tracking the absolute relative error between MC price and BSM model price. Even with the increased number of simulations, we still require >100,000 simulations to obtain a price range below the 10^-3 tolerance in relative error.

3. Different time to expiry, interest rates and volatilities

Figure 9: As expected, Call Option Price increases with respect to increasing Maturity.

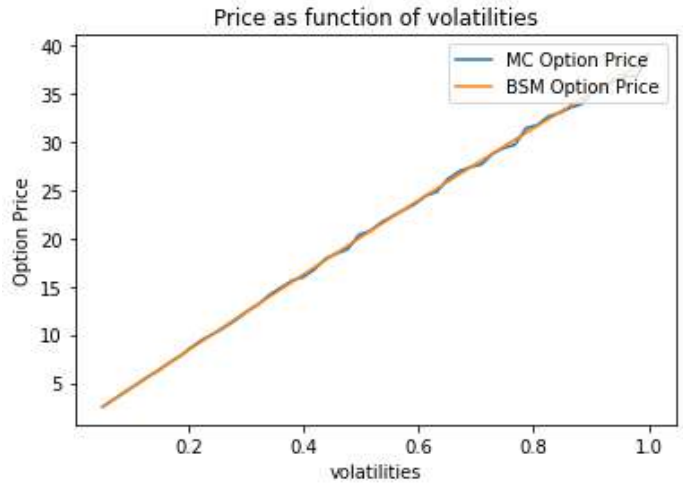

Figure 10: Call Option Price increases with respect to increasing volatilities. This is the same for Put Options as Vega is the same for both types of contracts by Put Call Parity

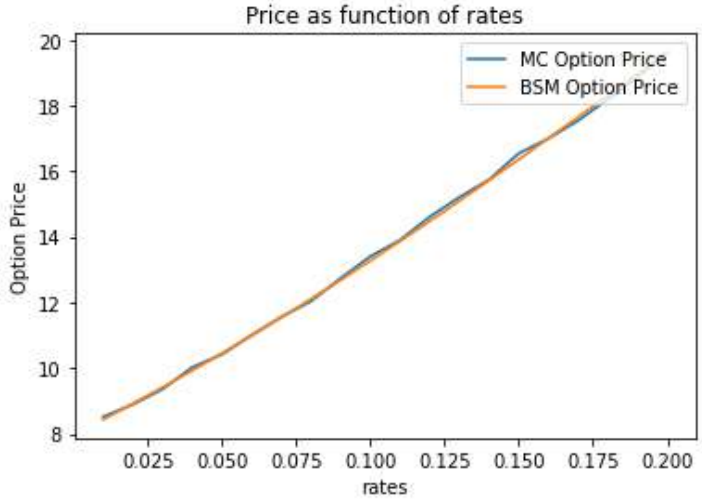

Figure 11: Call Option Price increases with respect to increasing interest rates. This is expected as Call Options have positive Rho.