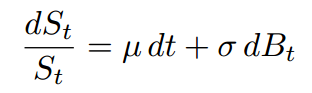
**Report**

Task 1(a) : Simulation Geometric Brownian Motion (GBM)

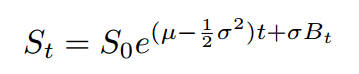
A simple model for stock prices St is obtained by assuming the return over infinitestimal dt is normally distributed with mean and variance both proportional to dt.



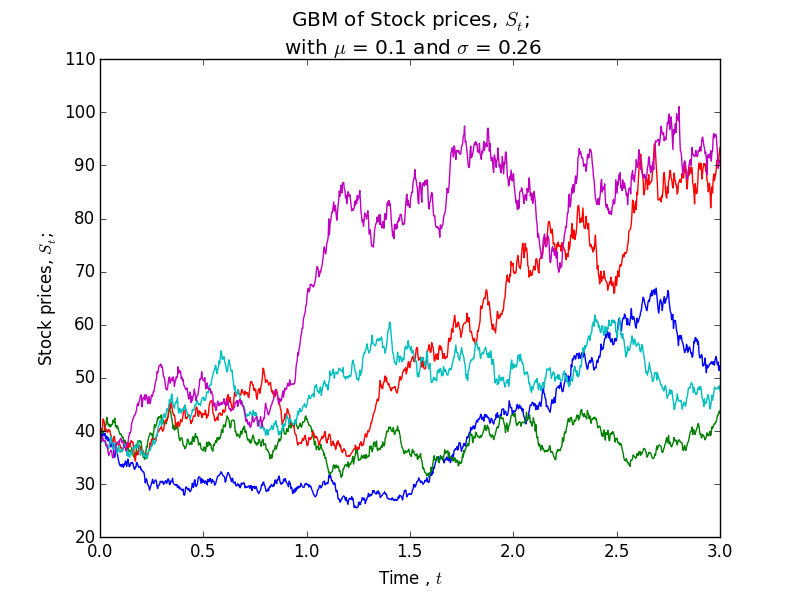
By assuming that the return has a stochastic differential equation (SDE) as following:



The solution can be found by using Ito’s lemma :



This solution is used to simulate 1000 runs of GBM for 0 < t < 3. The time [0,3] is partitioned into 1000 subintervals, each with length 0.003. 5 realizations of the GBM are plotted and the following figure is an example of a simulation:



The expectation value of S(3) can be found by using the property of Brownian motion that a Brownian motion follows a normal distribution:



The variance can also be found by using the formula of variance and also the property stated above:

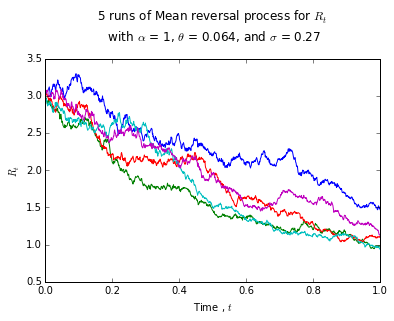


In the code gbm.py, each column of array S represents the stock prices for 1000 runs on each point of time. The last column of array S represents the stock prices for 1000 runs on time 3. The method of finding the expectation value of stock price at time 3, or S(3), based on the simulation is similar to finding the mean; the built-in function *np.mean* is apply on the last column of array S. Variance can also be found by the same way with the built-in function *np.var.*

To find the probability of the S(3) more than 39, namely P(S(3) > 39), a method called mask is used. In the code, the line *mask = S3 > 39* modifies every element in S3, where S3 is an array consist of 1000 different stock prices at time 3, and mask is an array same size with S3. If an element in S3 is more than 39, the corresponding position in mask will be 1, otherwise, 0 is assigned to that position. Every element in mask is then summed up and divided by the number of runs ( 1000 ) to get the probability. Lastly, the expected value of S(3) given S(3) > 39 is also found with the mask method. Stock prices that are more than 39 are taken out, and divided by the number of S(3) > 39 out of the 1000 stock prices.

Task 1(b) : Simulating mean reversal process

Mean reversal process is a random process that the random process tends to go towards the mean price over time. To simulate this process, Euler-Maruyama scheme is using to approximate the solution numerically with Python codes. The plot of 5 realization of the process is as follows:



The expectation value of R at time 1 ( or known as R(1) ) is obtained by taking the mean of all the 1000 values, while the probability of R(1) more than 2 can be calculated by the same way shown in the simulation of Brownian motion, which is the mask method.

Task 2 : Downloading and manipulating stock data

The FTSE Bursa Malaysia KLCI composed of the top 30 companies by market capitalization on the Bursa Malaysia Main market. It is reviewed semi-annually in June and December to ensure the Indices remain representative of the underlying Malaysia market.

A moving average is a type of technical analysis that calculates the average value of a stock’s price over time. To calculate the 5-day moving average of a stock in Python, 1295.KL or Public Bank Berhad in this case, the built-in function *pd.rolling\_mean* can be used. The moving average and the stock price is plotted as the figure below:

Python has also a function to calculate correlation between the stock price of Public Bank Berhad (PBB) and also the price of FTSEKLCI. This function is a function included in the module called *pandas* . The stock prices of PBB and FTSEKLCI is first collected into two columns of a DataFrame variable. In the code, the variable is called *closing*. Next, the correlation is then calculated with the function *closing.corr()* .