Project-1

MGMTMFE 405
SPRING 2024

You will need to write codes for all the parts of the project. Make sure the codes work properly and understand the ideas behind each problem below. You may be asked to demonstrate how the codes work, by running them and interpreting the results. Code quality, speed, and accuracy will determine the grades.

LECTURE-1

1. Using the LGM method, generate Uniformly distributed random numbers on [0,1] to do the following:

   (a) Generate 1,000 random numbers with Binomial distribution with $n = 44$ and $p = 0.64$. Compute the probability that the random variable X, that has Binomial (44, 0.64) distribution, is at least 40: $P(X \geq 35)$. Use any statistics textbook or online resources for the exact number for the above probability and compare it with your finding and comment.

   Hint: A random variable with Binomial distribution (n, p) is a sum of n Bernoulli (p) distributed random variables, so you will need to generate 44,000 Uniformly distributed random numbers, to start with).

   (b) Generate 10,000 Exponentially distributed random numbers with parameter $\lambda = 1.5$. Estimate $P(X \geq 1)$; $P(X \geq 4)$; and compute the empirical mean and the standard deviation of the sequence of 10,000 numbers. Draw the histogram by using the 10,000 numbers you have generated.

   Note: Random variable X that is exponentially distributed with parameter λ has the following cdf: $F(t) = P(X \leq t) = 1 - e^{-\frac{t}{\lambda}}$ for $t \geq 0$ (and $E(X) = \lambda.$)

   (c) Generate 5,000 Normally distributed random numbers with mean 0 and variance 1, by using the **Box- Muller** Method.

   (d) Generate 5,000 Normally distributed random numbers with mean 0 and variance 1, by using the **Polar-Marsaglia** method.

   (e) Now compare the efficiencies of the two above-algorithms, by comparing the execution **times** to generate 5,000 normally distributed random numbers by the two methods. Which one is more efficient? If you do not see a clear difference, you need to increase the number of generated realizations of random variables to 10,000, 20,000, etc.

Note: For all parts of problem 1 above, you can ***not*** use the built-in random number generator. You will have to generate random numbers using techniques that were discussed in class.

Note: For all parts of all problems, from here on, you can use the built-in random number generators.

LECTURE-2

2. (a) Estimate the following expected values by simulation:

$$A(t) = E\left(W_t^{\ 2} + sin(W_t)\right) \text{ and } B(t) = E\left(e^{\frac{t}{2}} cos(W_t)\right) \text{ for } t = 1,\ 3,\ 5.$$

Here, W_t is a Standard Wiener Process.

(b) How are the values of $B(t)$ (for the cases $t = 1,\ 3, 5$) related?

(c) Now use a variance reduction technique (whichever you want) to compute the expected value $B(5)$. Do you see any improvements? Comment on your findings.

Inputs: t
Outputs: 1) $A(t)$ and $B(t)$ for all 3 t's. 2) Writeup: comments for parts (b) and (c).

3. Let S_t be a Geometric Brownian Motion process: $S_t = S_0 e^{\left(\sigma W_t + \left(r - \frac{\sigma^2}{2}\right)t\right)}$, where $r = 0.055, \sigma = 0.2, S_0 = \$100$; W_t is a Standard Brownian Motion process (Standard Wiener process).

(a) Estimate the price **c** of a European Call option on the stock with $T = 5,\ X = \$100$ *by using Monte Carlo simulation*.

(b) Compute the exact value of the option **c** using the Black-Scholes formula.

(c) Use variance reduction techniques (whichever one(s) you want) to estimate the price in part (a) again using the same number of simulations. Did the accuracy improve? Compare your findings and comment.

Inputs: r , σ, S_0
Outputs: 1) *Ca and Cb for* parts (a) and (b); 2) Writeup: comments for part (c)

4. (a) For each integer number n from 1 to 10, use 1,000 simulations of S_n to estimate $E(S_n)$, where S_t is a Geometric Brownian Motion process: $S_t = S_0 e^{\left(\sigma W_t + \left(r - \frac{\sigma^2}{2}\right)t\right)}$, where $r = 0.055, \sigma = 0.20, S_0 = \$88$. Plot all of the above $E(S_n)$, for n ranging from 1 to 10, in one graph.

(b) Now simulate 3 paths of S_t for $0 \leq t \leq 10$ (defined in part (a)) by dividing up the interval [0, 10] into 1,000 equal parts.

(c) Plot your data from parts (a) and (b) in one graph.

(d) What would happen to the $E(S_n)$ graph if you increased σ from 20% to 30%? What would happen to the 3 plots of S_t for $0 \leq t \leq 10$, if you increased σ from 20% to 30%?

Inputs: σ
Outputs: 1) Graphs: plots in a .jpg file; 2) writeup: comments in a .pdf file for part (d).

LECTURE-3

5.

(a) Write a code to compute prices of European Call options via Monte Carlo simulation of paths of the stock price process. Use **Euler's** discretization scheme to discretize the SDE for the stock price process.
The code should be generic: for any input of the 5 model parameters - $S_0, T, X, r, \sigma -$ the output is the corresponding price of the European call option and the standard error of the estimate.

(b) Write a code to compute prices of European Call options via Monte Carlo simulation of paths of the stock price process. Use **Milshtein's** discretization scheme to discretize the SDE for the stock price process.
The code should be generic: for any input of the 5 model parameters - $S_0, T, X, r, \sigma -$ the output is the corresponding price of the European call option and the standard error of the estimate.

(c) Write code to compute the prices of European Call options by using the Black-Scholes formula. Use the approximation of $N(\cdot)$ described in Chapter 3.
The code should be generic: for any input values of the 5 parameters - S_0, T, X, r, σ - the output is the corresponding price of the European call option.

(d) Use the results of (a) to (c) to compare the two schemes of parts (a) and (b), by computing the following European call option prices: $X = 100$, $\sigma = 0.25$, $r = 0.055$, $T = 0.5$, and the range [95, 104] for S_0, with a step size of 1.

(e) Estimate the European call option's greeks – delta (Δ), gamma (Γ), theta (θ), and vega (ν) - and graph them as functions of the initial stock price S_0. Use $X = 100$, $\sigma = 0.25$, $r = 0.055$ and $T = 0.5$ in your estimations. Use the range [95, 105] for S_0, with a step size of 1. You will have 4 different graphs for each of the 4 greeks.
In all cases, dt (time-step) should be user-defined. Use $dt = 0.05$ as a default value.

Inputs: $dt,\ S_0, T, X, r, \sigma$
Outputs: Values: *C1 and e1* for part (a); *C2and e2* for part (b); C3 for part (c); writeup: comments in a .pdf file for part (d); $\Delta, \Gamma, \theta, \nu$ for part (e).

6. Consider the following 2-factor model for stock prices with stochastic volatility:

$$\begin{cases} dS_t = rS_t dt + \sqrt{V_t}\, S_t dW_t^1 \\ dV_t = \alpha(\beta - V_t)dt + \sigma\sqrt{V_t}\, dW_t^2 \end{cases}$$

where the Brownian Motion processes above are correlated: $dW_t^1 dW_t^2 = \rho dt$, where the correlation ρ is a constant in $[-1,1]$.
Estimate the price of a European Call option (via Monte Carlo simulation) that has a strike price of X and matures in T years.

Use the following default parameters of the model: $\rho = -0.6,\ r = 0.055,\ S_0 = \$100,\ X = \$100, V_0 = 0.05,\ \sigma = 0.42,\ \alpha = 5.8,\ \beta = 0.0625,\ dt = 0.05, N = 10{,}000$.

Use the Full Truncation, Partial Truncation, and Reflection methods, and provide 3 price estimates by using the tree methods.

Inputs: $\rho,\ r,\ S_0,\ V_0,\ \sigma,\ \alpha,\ \beta$.
Outputs: Values: *C1*, C*2*, C*3*.

7. The objective of this exercise is to compare a sample of Pseudo-Random numbers with a sample of Quasi-Monte Carlo numbers of $Uniform[0,1]x[0,1]$:

Use 2-dimensional Halton sequences to estimate the following integral:

$$I = \int_0^1 \int_0^1 e^{-xy} \left(sin6\pi x + cos^{\frac{1}{3}}2\pi y \right) dxdy$$

Default parameter values: N=10,000; (2,3) for bases.

Inputs: b1; b2 (the bases); N
Outputs: I