Homework 3

ECON 6204 (8204) - 001

Fall 2017

100 Points

Reading Assignment: Lectures 1 - 4

Due: Tuesday, September 26 Student Name: ____________________________

Instruction. On the due date, you should email your completed homework with a single zipped folder named "hw3_YourName. The folder should include a driver file named "main.py" for your Python program and some other items, as specified below.

Problem Experiment on Monte Carlo Integration.

You are asked to program Monte Carlo simulation in Python for computing the values of an European call and put. The call or put option is written on an underlying asset, which follows a geometric Brownian motion,

$$dS_t = (\mu - \delta)S_t dt + \sigma S_t dW_t$$

where S_t is the time t value of the underlying asset, μ is mean return, δ is a continuously compounded dividend yield, σ is volatility, and W_t is a Wiener process with $W_0 = 0$ and $W_t \sim \sqrt{t}\mathcal{N}(0,1)$. The values of the two European options are determined by their discounted risk-neutral expectations:

$$V_C^{eur}(S_0, 0) = e^{-rT} E_Q[V_C^{eur}(S_T, T)]$$

$$V_P^{eur}(t = 0) = e^{-rT} E_Q[V_P^{eur}(S_T, T))$$

where T is the maturity date, and $V_C^{eur}(S_T, T)$ and $V_P^{eur}(S_T, T)$ given by the payoff functions:

$$V_C^{eur} = \max(S_T - K, 0)$$

$$V_P^{eur} = \max(K - S_T, 0)$$

The objective of our Monte Carlo simulation and integration is to approximate risk-neutral expectations by using the law of large numbers. So, the algorithm to price an European call or put is clear: we draw sample paths for S_t and compute

$$\widehat{V}_C^{eur} = e^{-rT}\left(\frac{1}{N}\sum_{k=1}^{N} V_C^{eur}(S_T^{(k)}, T)\right),\ k = 1, 2, \ldots N,$$

$$\widehat{V}_P^{eur} = e^{-rT}\left(\frac{1}{N}\sum_{k=1}^{N} V_P^{eur}(S_T^{(k)}, T)\right),\ k = 1, 2, \ldots N$$

where N is the number of sample paths and k refers to the k^{th} sample path. Note that to generate sample paths for S_t, you must use the **explicit Euler method** and the **Milstein method**, respectively.

The parameter values of the model are: $S_0 = \$100$, $K = \$100$, $\mu = 0.08$, $r = 0.03$, $\delta = 0.025$, $\sigma = 0.75$, and $T = 1$, $\triangle t = 0.01$. Your Python should accomplish the following:

- Simulate $N = 1,000$ sample paths (using either explicit Euler or Milstein) to do the following:
- Compute $\widehat{V}_C^{eur}$ and $\widehat{V}_P^{eur}$,
- Compute $\varepsilon_1 = |\widehat{V}_C^{eur} - V_C^{eur}|$ and $\varepsilon_2 = |\widehat{V}_P^{eur} - V_P^{eur}|$, where the ε terms are absolute errors and V_C^{eur} and V_P^{eur} are the analytical solutions, which you will also need to compute using code for Homework 2.
- Note: you do not need to plot sample paths.
- re-do the above by setting $\triangle t = 0.001$.

Other requirements for this homework:

- You must use a seed number equal to 10.
- The output values of $\hat{V}$ and ε must have six decimals.
- Your program must show your console outputs in table form with headings and captions and and you should briefly comment on your outputs from the perspective of numerical accuracy.
- Your Python code must include remarks at the very beginning briefly specifying the purpose and algorithms of your work and your name. To make your code reader-friendly, you should add remarks when necessary.
- **You must email me with a zipped folder including your Python program, a photocopy of your console outputs and a copy of your analysis (< 2 pages).**