Homework 2

ECON 6204 (8204) - 090

Fall 2018

10 Points

Reading Assignment: Lectures 1 - 3

Due: Tuesday, September 18 Student Name: ___________________________

Instruction. On the due date, you should email your completed homework with a single zipped folder named "hw2_YourName. The folder should include at least the main.py and a photocopy of your console output. In addition, homework submission must follow the requirements specified below.

Problem. Assuming a geometric Brownian motion

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

where S_t is the time t value of the underlying asset, $S_0 = 100$ $\mu = 0.1$, $r = 0.05$, $\delta = 0.025$ $\sigma = 0.2$, W_t is a Wiener process with $W_0 = 0$. Write a Python program to simulate **five sample paths** of S_t, $t \in [0, 1]$, respectively, based on Algorithm 1.11 (Euler discretization of a SDE).

To apply this algorithm, you must discretize the time domain by using $\triangle t = 0.001$. Given this time step, you will need to generate 1000 simulated values for S. Executing your driver file (main.py) must be able to show five sample path on the console.

Requirements for this homework:

- You must use a seed number equal to 123. That is, $numpy.random.seed(seed = 123)$.
- Your program must include two Python files:
  - The driver: main(),
  - The implementation function: sample_path(parameter 1, ...), where you can design the function, sample_path, to draw either a sample path or multiple sample paths in one call, subject to your preference.
- Your program must show a 2-dimensional diagram of five sample paths on the console, where the horizontal axis is labeled "time (t)" and the vertical axis is labeled "S(t)".
- Your Python program must include remarks briefly specifying the code's purpose and algorithm as well as your name. To make your code reader-friendly, you should add remarks elsewhere when necessary.

Suggested Python functions useful for the homework:

- $numpy.random.standardnormal(...)$, or $scipy.stats.norm.rvs(...)$
- $numpy.random.seed(...)$
- $numpy.cumsum(...)$
- $matplotlib.pyplot.plot(...)$