Homework 4

ECON 6204 (8204) - 001

Fall 2018

100 Points

Reading Assignment: Lectures 1 - 5

Due: Tuesday, October 9, 7:00pm Student Name: _____

Instruction. On the due date, you should email your completed homework with a single zipped folder named "hw4_YourName. The folder should include a driver file named "main.py" for your Python program and some other items, as specified below.

Problem Monte Carlo simulation for the valuation of American Options.

You are asked to program Monte Carlo simulation in Python for computing the values of an American call and an American 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 these two American options are determined by their discounted risk-neutral expectations:

$$V_C^{am}(S_0, 0) = \sup_{0 \le \tau \le T} E_Q[e^{-r\tau}\Psi_C(S_\tau)|S_0] \tag{1}$$

$$V_P^{am}(S_0, 0) = \sup_{0 \le \tau \le T} E_Q[e^{-r\tau}\Psi_C(S_\tau)|S_0] \tag{2}$$

where Q is a risk-neutral probability measure, T is the maturity date, τ is the stopping time , and Ψ_C and Ψ_P are time-τ payoff functions:

$$\Psi_C(S_\tau) = \max(S_\tau - K, 0); \qquad \Psi_P(S_\tau) = \max(K - S_\tau, 0)$$

To compute the values of the American call and put, you are required to approximate the risk-neutral valuations of (1) and (2) by using the algorithm of regression method I:

Algorithm 3.12 (regression method I)

1. Simulate N paths $S_1(t), \ldots, S_N(t)$ and store the values $S_{ik} \equiv S_k(t_i)$, where $i = 1, \ldots, M$, and $k = 1, \ldots, N$.
2. For $i = M$, set $V_{MK} \equiv \Psi(S_{Mk})$ for all k.
3. For $i = M - 1, \ldots, 1$, compute $\hat{C}$ and V at t_i for each path:
   (a) Approximate $C_i(x) \approx \sum_{l=0}^{L} a_l \phi_l(x) \equiv \hat{C}_i(x)$ by least squares over the N points $(x_k, y_y) \equiv (S_{ik}, e^{-r\triangle t}V_{i+1,k})$, $k = 1, \ldots N$, where $\phi_j(x) = x^j$ (e.g., $\phi_0(x) = x^0 = 1$, $\phi_1(x) = x^1$, and the like).
   (b) and set $V_{ik} = \max\{\Psi(S_{ik}), \hat{C}_i(S_{ik})\}$.
4. Calculate $V_0 = e^{-r\triangle t}\frac{1}{N}(V_{11} + \ldots + V_{1N})$

To implement the algorithm, you must follow the following requirements:

- Apply the **explicit Euler method** to generate $N = 1000$ sample paths with the seed number set at 123.
- Use the parameter values: $S_0 = \$100$, $K = \$100$, $\mu = 0.08$, $r = 0.03$, $\delta = 0.025$, $\sigma = 0.75$, and $T = 1$, $\triangle t = 0.01$.
- Use *scipy.optimize.curve_fit* to approximate the continuation-value function $\hat{C}_i(S_{ik})$ in terms of a polynomial function of order 3; that is, $\hat{C}(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$
- re-do all the above work by setting $\triangle t = 0.001$.
- The computed values of the American call and put must contain 6 decimals.

Other requirements for this homework:

- 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 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 (< or = 1 page).**