COM501: Scientific programming

Term Paper

Question 1. Suppose $X = (S_t : 0 \le t \le T)$ is the price process of a stock, and X_t follows below process.

$$\triangle X_t = \mu X_t \triangle t + \sigma X_t \sqrt{\triangle t}$$

where $X_0 \in \mathbb{R}$, $0 \le t \le T < \infty$, $\sum \sigma^2 < \infty$, i.e., all mathematical requirements for ordinary stock price process are satisfied, while $\frac{\triangle X_t}{X_t} \sim N(\mu, \sigma^2)$. In other words, today's stock price X_t is an aggregation of all historical Gaussian random process upto today, plus initial stock price. (Although risk-neutrality assumptions are needed for this process to be real stock price, let's skip the change of measure and only focus on Monte-Carlo simulation for a random process.) This is a discretized version of Brownian motion, an infinite process of Gaussian random.

Since the problem is fairly complicated, we use an ordered transformation by taking log.

$$x_t = \ln X_t \sim N(\mu, \sigma^2)$$

Then,

$$x_t = x_0 \times \exp(\sum_{s=1}^{t} \mu \times s + \sum_{s=1}^{t} \sigma \sqrt{s})$$

In mathematics, it is called that X_t follows log-Normal distribution, because x_t follows normal distribution. (The above equation is called a discrete version of Geometric Brownian motion. For arithmetic Brownian verison, it is $x_t = x_0 + \sum_{s=1}^{t} \mu \times s + \sum_{s=1}^{t} \sigma\sqrt{s}$.)
You only need superficial understanding of the above stochastic process for below question. What is needed for our purpose is to see if one computer approximation can be computationally more efficient than the others in a certain condition.
This is where we start. Let's set a put option price (or ReLU type non-linear transformation of X_t) by

$$P(k) = \mathbb{E}[(K - X_T)_+] = \mathbb{E}[e^{x_T} - e^k] = \sum_{0}^{k} (e^k - e^{x_T}) q_T(x)$$

where $k = \ln K$, $q_t(x)$ the probability density of the log-price x_t, which is Gaussian. Then, we can call this function f as,

$$f(t, x) = \mathbb{E}_t[(K - X_T)_+] = \mathbb{E}_t[e^k - e^{x_T}] = \sum_{0}^{k} (e^k - e^{x_T}) q_t(x)$$

the subscript t denotes an intermittent time between $[0, T]$. In other words, we have a ReLU type non-linear transformation for a Gaussian process.

1) Compute the initial price of the put option with time period $T \in \{\frac{1}{4}, \frac{1}{2}, 1, 5\}$ and anchor value $K \in \{70, 90, 100, 110, 130\}$ given $X_0 = 100$, $\mu = 0$, and $\sigma = 0.8$. First, use a rectangular quadrature based on $N \in \{10, 50, 100, 1000\}$ subintervals. Choose the truncation level accordingly to guarantee convergence of the approximation.

2) Plot the mean squared error of the quadrature approximations across all put option prices against N. What is the rate of convergence of your quadrature approximation? You may assume that the correct price is the price you obtain from a rectangular quadrature approximation with $N = 10^5$ subintervals.

3) Do the absolute error and the computational efficiency of your quadature approximations depend on the given time period and the current price to anchor value ratio? What do your results suggest?

Now move on to Fast Fourier transform and see if we can achieve any better computational efficiency.

4) Use the Fast Fourier transform method with $N \in \{10, 50, 100, 1000\}$ quadrature subintervals. Use $\{k_n = \beta + (n-1)\delta : 1 \le n \le N\}$ in order to obtain approximation of the option prices at anchor $K \in \{70, 90, 100, 110, 130\}$

5) Plot the mean squared error of the FFT approximations across all option prices against N. What is the rate of convergence of your quadrature approximation? Again, use the correct price above.

6) Compare your results above to quadrature case. Is FFT more computationally efficient than rectangular quadrature across all possibilities?

Now let's fix $T = 1$. Given the logged return $x_T = \log \frac{X_T}{X_0}$, we are given to approximate the function f, or option price c in our case, by Monte-Carlo simulation. For the simulation process, we invite one other variation of the above model, which can be translated to (G)ARCH, in a sense that variance is no longer constant and depends on time. The process now becomes,

$$\triangle X_t = \mu X_t \triangle t + \sigma_t X_t \sqrt{\triangle t}$$
$$\triangle \sigma_t = \kappa \sigma_t \sqrt{\triangle t}$$

where σ follows mean 0 Gaussian distribution with increasing variance, and κ is a constant. Assume $\kappa = 2$ for remaining questions. Note that the logged return $s_T = \log \frac{S_T}{S_0}$ follows below process

$$\tilde{x} = x_0 - \frac{1}{2}v + Z$$

where $v_t = \sum_{s=1}^{t} \sigma_s$ for $t \in [0, T]$ and $Z \sim N(0, v)$.

7) Suppose you are given a Monte-Carlo sample $\tilde{x}^1$ of $\tilde{x}$. Write a pseudo-code that constructs a Monte-Carlo sample of X_T using the samples $\tilde{x}^1$.

8) In words, how can you use the Monte-Carlo samples $\tilde{x}^1$ to construct a Monte-Carlo estimator of the price?

9) The file "v simul.txt" contains 10,000 i.i.d. samples of v. Use these samples to construct a Monte-Carlo estimator of f based on Monte-Carlo sample of $Q \in \{100, 500, 1000, 5000, 10000\}$. In other words, use the i.i.d. samples of v to construct i.i.d. samples of $\tilde{x}$. Then, construct Monte-Carlo samples of X_T, and use it to find the function f.

10) Measure the time it takes to compute, price, and the MSE for all Ks. Plot MSE as a function of the time necessary to compute the approximations.

11) Is Monte-Carlo a superior approximation to other methods? If so, in which case?

12) What happens if $\kappa = 0.5$?

In your answer, provide necessary arguments and graphs. For questions that mathematical derivation can sharpen your argument, precise intuitive reasoning can be sufficient. Good luck with the term paper. Merry Christmas and happy new year!