CS 476/676 Assignment 3

Winter 2022

Instructor: Yuying Li Office: DC3623 *yuying@uwaterloo.ca*
Lecture Times: MW 4-5:20pm
Yuying Li: OH: Tues 3-4pm
TA: Chendi Ni OH: Mar 4 11:30 am -12:30 pm chendi.ni@uwaterloo.ca
Chendi Ni OH: Mar 11 12 - 1pm
Chendi Ni OH: Mar 15 1 - 2pm
My Web Site: http://cs.www.uwaterloo.ca/~yuying

Due: 4pm Mar 17, 2022

Programming Questions

IMPORTANT: In this and in future assignments, most of the marks for programming questions are allocated for explanations of algorithms (e.g. pseudo-code) and discussion of results. If all you hand in is the listing of the "Raw Code" or "Raw Output" by itself, you will get poor marks. All coding should be done in Matlab. All the plots should be appropriately labeled. You should submit all Matlab code used in your assignment. Be sure to document (i.e. add liberal comments) your code. The TA will take off marks for poor documentation.

1 [(10 marks)] (Numerical Scheme for SDE)

Assume that the interest rate r_t is modeled by the mean-reverting process below

$$dr_t = a(b - r_t)dt + \sigma\sqrt{r_t}dZ_t$$

where Z_t is a stabdard Brownian motion, and a, b, σ are positive constants.

Let the time step Δt be given. Assume that the interest rate r_n at time $t_n = n\Delta t$ is given.

(a) Given r_n at t_n, write down the Euler-Maruyama formula for computing the interest rate r_{n+1} at t_{n+1}. Assuming $r_n = b$, write down the condition on the standard normal sample which leads to $r_{n+1} < 0$. Explain.

(b) Given r_n at t_n, write down the Milstein method for computing r_{n+1}. Assumming $r_n = 0$, provide an inequality using parameters a, b, σ, which guarantees that r_{n+1} is always positive.

.

2 [(8 marks)] (Numerical Scheme for SDE)

For a long time horizon, the following risk neutral CIR interest rate model can be considered

$$dr_t \quad = \quad a(b - r_t)\ dt + \sigma\sqrt{r_t}\ dZ_t$$

where $a, b, \bar{r}, \sigma$ are positive constants, representing speed of mean reverting, average rate and volatility respectively. A zero coupon bond, paying off one dollar at $t = T$, has the following no-arbitrage value

$$\text{Bond Value} \quad = \quad E_Q\left[\exp\left(-\int_0^T r(t)\ dt\right)\right] \ . \tag{1}$$

Provide a pseudo code for using a Monte Carlo method with M simulations, N time steps, and Milstein time-stepping to price the above zero coupon bond. Determine the value at $t = t_0$, assuming $r = r_0$ at $t = t_0$. Assume the existence of a function which returns a random variable $\phi \sim \mathcal{N}(0, 1)$. .

Table 1: Data for Hedging Simulations Using a Binomial Lattice

σ	.20
r	.03
μ	0.15
Time to expiry T	1 year
Initial asset price $S(t_0)$	\$10
Strike Price K	\$$0.95S(t_0)$
N	250

3 [(20 marks)](Compute Delta from a Binomial Lattice)

Assume that the (no dividend) underlying price follows

$$\frac{dS_t}{S_t} = \mu dt + \sigma dZ_t \tag{2}$$

where Z_t is a standard Brownian, μ and σ are constants.

Consider the European straddle, whose payoff equals the sum of the call payoff and put payoff with the same strike K, i.e., $\max(S-K,0)+\max(K-S,0)$.

You are to conduct hedging effectiveness analysis for this power option using MC simulations of (2) for the underlying price so that there is no time discretization error in the simulated price.

In addition, you will compute hedging positions by modifying your code for option pricing under a binomial lattice in Assignment 2.

Assume that the initial option price is computed from your binomial model.

Specifically,

(a) Write a Matlab function $[V_0, S, \delta] =$ **binomialDeltaStraddle**(S_0, r, σ, T, N) which returns, the initial option value V_0, S and δ, representing the underlying price and delta hedging positions on the binomial lattice nodes over equally spaced N time periods in $[0, T]$. The interest rate r and volatility σ are constants.

(b) Now consider a vector of (simulated) underlying price S (not necessarily equal to any binomial lattice price), using linear interpolation to determine the corresponding hedging positions $\delta(S)$ from the binomial lattice price vector S^n and delta δ^n as follows: When a simulated price in S does not equal to any lattice price S^n, interpolate using the hedging values at lattice nodes at each rebalancing time to approximate the hedging position (you can use matlab function **interp1**). To avoid potentially large extrapolation error, reset the hedging position equal to that of the nearest binomial price S^n, if an underlying price exceeds the range of the price S^n. Write a Matlab function δ=**interpDelta**(δ^n, S^n, S) to implement this interpolation.

(c) Using the data in Table 1, **binomialDeltaStraddle**, and **interpDelta**, write a matlab script to compute hedging positions at time t_n, $n = 0.8N$, for a European straddle $S = \text{linspace}(0.8S_0, 1.4S_0, 100)$ and plot the computed hedge position δ against S.

(d) Using the parameter values given in Table 1, determine the hedging error using discrete delta hedging for the straddle. Plot the histogram of the relative hedging error P&L, for no hedging, $n = 1 : 1 : N$ (daily), $n = 5 : 5 : N$ (weekly), $n = 20 : 20 : N$ (monthly) respectively. assuming hedging positions are computed using **binomialDeltaStraddle** and **interpDelta**,. The Matlab functions *histc*, *bar* might prove useful. Use at least 50 bins in your histogram. Comment on your observations.

(e) Write a Matlab function [var,cvar]=dVaRCVaR(P&L, β) which returns VaR and CVaR for a discrete P&L distribution with M independent samples using the procedure described. In Matlab, ordering can be done using **sort**. For the straddle, compute and report in a table, mean, standard deviation, VAR (95%), CVAR (95%), of $P\&L$ for no hedging, rebalancing monthly, weekly, and daily. Discuss how hedging performance changes with the rebalancing frequency.

(f) In general, a hedging rebalancing time t_n^{rb} may not coincide with the discrete time t_n of the lattice. How would you change your code to implement the hedging analysis?

.

4 [(10 marks)] (Trading simulation)

You think you have discovered an arbitrage dynamic trading strategy. Does it work?

Simulation can be used to analyze your strategy.

Assume that it is possible to trade continuously in time, without transaction cost, and you can short sell stock, i.e., borrow stock to sell and return later without cost.

For simplicity, assume that the interest $r = 0$. Let S_t and P_t denote time t values of the stock and a European put with strike K and expiry T.

Suppose that today you sell the European put option with strike K and expiry T. Assume $S_0 < K$.

Let $\delta_t S_t + B_t$ denote the value of the portfolio of the underlying and cash account at t. The cash account B_t is used to balance the transactions.

At $t = 0$, you sell a share in the stock, i.e., $\delta_0 = -1$. Subsequently, for $t > 0$, you implement the following simple hypothetic trading strategy: As soon as $S_t > K$, buy a stock share. As soon as $S_t < K$, you sell the stock.

At $t = 0$, the cash account is set such that $-P_0 + \delta_0 S_0 + B_0 = 0$.

(a) Write down a mathematical expression for your trading strategy δ_t, $0 < t < T$, where δ_t denotes the number of shares in the stock immediately after trading at t.

(b) Under specified assumptions, time t value of the hedging portfolio $\delta_t S_t + B_t$ has two differential expressions in terms of S_t, K. Provide the precise expressions and conditions under which each expression is valid. Explain.

(c) Show that this is an arbitrage by analyze the distribution for the relative P&L, $(-P_T + \delta_T S_T + B_T)/(P_0 + S_0)$.

(d) Now perform the following computation to investigate whether this is true. Assume that the underlying follows a Black-Scholes model, with the model parameters given in Table 1. Simulate your strategy above, assuming $S_0 = 100$, $K = 105$. Compute the relative P&L $(-P_T + \delta_T S_T + B_T)/(P_0 + S_0)$ by MC simulations. Assume P_0 equals the initial call value computed by Black-Scholes formula. Use about $80,000$ simulations. Use $100, 200, 400, 800$ rebalancing times (which are different from the value of N in Table 1). Show a table of results (mean, standard deviation, 95% VAR and CVAR) versus number of rebalancing times. Generate plot of the probability density of the relative P&L for the case of 800 rebalancing times.

What do you observe about the mean and variance of the hedging error?

(e) **Bonus**. Explain the discrepancy in the variance hedging error between the theoretical value under the continuous hedging assumption and the computed values using MC simulations.

.

5 [(12 marks)] (MC Pricing Under Jump Model)

Assume that Z_t is a standard Brownian motion. Let q_t denotes a Poisson counting process where the change dq_t is defined as

$$dq_t = \begin{cases} 0 & \text{with probability } 1 - \lambda dt \\ 1 & \text{with probability } \lambda dt, \end{cases}$$

Table 2: Model Parameter Data

σ	.15
r	.05
Time to expiry	1 years
Strike Price	\$95
Initial asset price $S(0)$	\$95
μ_u	0.304
μ_d	0.308
p_u	0.34
Intensity of Poisson Process λ	.1
Δt	1/1000

where $\lambda > 0$ denotes the jump intensity. Consider the following risk neutral jump-diffusion process

$$\frac{dS(t)}{S(t^-)} = (r - \lambda\kappa)dt + \sigma dZ + (J-1)dq_t \tag{3}$$

where the value $S(t^-)$ is the price just before jump and $\kappa = E[J-1]$. Note that, if $dq_t = 1$, then after the jump

$$S(t) = S(t^-)J$$

where J corresponds to the random jump amplitude. In Merton's jump model, $\log J$ has a normal distribution, see §9, 9.1, 9.2, 9.4, 9.5 in the course notes for discussions on the Merton's jump diffusion model and its implementation.

In contrast to the original Merton's jump process, here we assume, instead, a double exponential jump process. Specifically, assume that, when jump occurs, the up-jump occurs with a probability p_u and down-jump occurs with a probability $1 - p_u$ and the density function of the jump amplitude J is a double exponential, i.e., the density function $f(y)$ of $y = \log(J)$ is

$$f(y) = p_u\mu_u \exp(-\mu_u y)\mathbf{1}_{y>=0} + (1-p_u)\mu_d \exp(\mu_d y)\mathbf{1}_{y<0} \tag{4}$$

where $\mathbf{1}_{y>=0}$ is an indicator which equals 1 if $y >= 0$ and 0 otherwise.

(a) Verify that

$$\kappa = E[J-1] = \frac{p_u\mu_u}{\mu_u - 1} + \frac{(1-p_u)\mu_d}{\mu_d + 1} - 1$$

(b) Assuming the underlying follows a jump diffusion process (3) with the density of log jump amplitude $\log(J)$ given in (4), using Monte Carlo method with Euler time stepping., write a Matlab function to compute the fair value of a European capped-call

$$\text{val} = \text{CappedCall}(S_0, \sigma, p_u, \mu_u, \mu_d, \lambda, K, T, \mathcal{C})$$

with strike K and expiry T. The payoff of a European capped-call is

$$\min\left\{\max(S_T - K, 0), \mathcal{C}\right\}$$

Using parameter values in Table 2,, plot the computed option values with the number of (timesteps, simulations) = $(800, 25000)$ for $\mathcal{C} = 20:10:100$. How does the computed option value depend on the cap $\mathcal{C}$? Explain why your observation is reasonable. You can use the matlab function **exprnd** which generates a random sample from an exponential distribution in $[0, +\infty)$ with a probability density function

$$f_X(x) = \begin{cases} \frac{1}{\mu} e^{-\frac{x}{\mu}} & \text{if } x \in [0, +\infty) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\mu > 0$ is a parameter.

(c) Use Matlab function **blsimpv** to plot the implied volatility of the standard option price using your CappedCall$(S_0, \sigma, p_u, \mu_u, \mu_d, \lambda, K, T, \mathcal{C})$, by setting $\mathcal{C} = +\infty$, with (timesteps, simulations) $= (800, 25000)$ against the strike, e.g., K=linspace$(70, 120, 20)$ and $T = 1$.

(d) Discuss observed characteristics of the implied volatility from the assumed jump model.

6 [(Graduate Student Question)] (10 marks)

Practitioner's BS delta hedging consists of a portfolio $\Pi(t)$ with the following components at rebalancing time $t = t_n$: an option position $-V_n$, δ_n shares at price S_n, and the dollar amount B_n in a risk-free cash account, where

$$\delta_n = \frac{\partial V_{BS}}{\partial S}(S_n, t_n; \breve{\sigma}_n) \tag{6}$$

and $\frac{\partial V_{BS}}{\partial S}(S_n, t_n; \breve{\sigma}_n)$ is the BS delta at t_n using the implied volatility $\breve{\sigma}_n$ computed from the market option price V_n at the rebalancing time t_n. In [Hull and White(2017)], hedging position function for the underlying is learned from the market price data [1]. More specifically, the mean variance delta, δ_{MV} is modelled in a quadratic form

$$\delta_{MV} = \delta_{BS} + \frac{\text{VEGA}_{BS}}{S\sqrt{T}} \left(a + b \cdot \delta_{BS} + c \cdot \delta_{BS}^2 \right), \tag{7}$$

see (5) in [Hull and White(2017)]. Your task is to read this paper to understand their proposed method and then conduct a comparative computational study on the data below.

You are given the S&P 500 index option data, in the file RawData.mat, from 01-Jan-2009 to 31-Dec-2010 as the training data set and the data from 01-Jan - 2011 to 31-Dec-2011 as the testing data set. In RawData.mat, values of following additional variables are also provided for you to use for training and testing:

- CSTrain: Daily change of the market index price for the training set.
- CVTrain: Daily change of the market index option price for the training set.
- CSTest: Daily change of the market index price for the testing set.
- CVTest: Daily change of the market option price for the testing set.
- DeltaTrain: BS delta δ_{BS} computed using implied volatility σ_{imp} for the training set.
- DeltaTest: BS delta δ_{BS} computed using implied volatility σ_{imp} for the testing set.
- VegaTrain: BS vega VEGA_{BS} computed using implied volatility σ_{imp} for the training set.
- VegaTest: BS vega VEGA_{BS} computed using implied volatility σ_{imp} for the testing set.
- STrain: market index price, S, for the training set.
- STest: market index price, S, for the testing set.
- TauTrain: time to expiry, τ, for the training set.
- TauTest: time to expiry, τ, for the testing set.

Consider daily hedging error $\Delta V - \delta \Delta S$, where $\Delta V = V(S(t_{n+1}), t_{n+1}) - V(S(t_n), t_n)$, $\Delta S = S(t_{n+1}) - S(t_n)$, δ is the position of the underlying at t_n. Using the provided data, you are to

(a) For practitioner's BS delta hedging defined in (6), plot the histogram of the hedging error $\Delta V - \delta \Delta S$, tabulate its mean, standard deviation, 95% VaR and CVaR when the hedging position $\delta = \delta_{BS}$ using given training data and testing data respectively.

[1] *Optimal delta hedging for options*, Journal of Banking and Finance, 82, 180-190, 2017.

(b) Learn the MV hedging model (7), see (5) in [Hull and White(2017)] , using the training data.

(1) Report the ratio GAIN defined in (3) in [Hull and White(2017)] for the learned MV hedging model (7) using the training data and testing data respectively.

(2) In addition, plot the histogram of the hedging error $\Delta V - \delta \Delta S$, tabulate its mean, standard deviation, 95% VaR and CVaR with respect to the training data and testing data respectively, when the hedging position δ comes from the MV model (7) learned from the training data. How does the testing performance compare to training performance?

(c) Assume now that the hedging position function δ at time t has a different parametric form given below

$$\delta = c_0 + c_1 \cdot S + c_2 \cdot \delta_{BS} + c_3 \cdot \delta_{BS}^2 + c_4 \cdot \text{VEGA}_{BS} \cdot \delta_{BS} + c_5 \cdot \text{VEGA}_{BS}^2 + c_6 \cdot \delta_{BS}^3 \tag{8}$$

where δ_{BS} denotes the BS delta and VEGA_{BS} denotes the BS vega. Learn this parametric function for the δ position by minimizing the MV error as defined in [Hull and White(2017)].

Repeat the same performance assessment as in (b) using the position in the underlying δ learned from training data using the parametric form (8). Compare results from (c) with the parameterizations in (b). What do you observe? Is the observation reasonable? Why?

You can use Matlab function **blsprice**, **blsimpv**, **blsdelta**.