CS 476/676: Assignment 2

Winter 2015

Instructor: Wei Xu	Office: DC 1312	e-mail: *wei.xu@uwaterloo.ca*
Lecture Times :	MWF 11:30-12:20	MC 2054
Office Hours Until March 2, 2015:	Wei Xu, Mondays 1:30-3:00pm, DC1312 Kai Ma, DC3594: Wednesday 2:00-4:00pm, Feb. 26 (Thursday) 10:00-12:00,	

Due March 2, in class

IMPORTANT: In this and in future assignments, most of the marks for programming questions are allocated for explanations of algorithms and discussions 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 (nicely commented). The TA will take off marks for poor documentation. Finally, **vectorize your code**.

Assignment Questions

1. (6 marks) (Hedging under a Bionomial Lattice). Modify your code of pricing under a binomial lattice so that it now returns the hedging positions delta.

(a). Write a Matlab function $[\text{delta}, S, t] = \text{deltaBinomial}(S_0, r, \sigma, T, N, \text{fpayoff})$ which returns the delta, the binomial nodal price S, and discrete time t (a vector) with N time period in $[0, T]$, assuming a constant interest rate r and volatility σ. The argument fpayoff specifies a function which returns the payoff of a specified option. At each binomial node, the option value is computed by risk neutral pricing

$$V_{t_i} = e^{-r\Delta t}(q^* V^u_{t_i+1} + (1-q^*)V^d_{t_i+1}).$$

The hedging position is given by

$$\alpha(t_i) = \frac{V^u_{t_{i+1}} - V^d_{t_{i+1}}}{(u-d)S_{t_i}}.$$

(b). Using linear interpolation and hedging positions on a binomial lattice, write a Matlab function delta_i=interpBinomial(delta, S, t, t_i, simuS) to determine hedging positions at t_i when the underlying price is given in a vector simuS. You can assume that time t_i always coincides with one of the time discretization points t of the binomial lattice. When a simulated price in simuS does not equal to any price at binomial nodes, use linear interpolation from the adjacent nodes to approximate the hedging position. If an underlying price exceeds the range of those on the binomial nodes, set the hedging position equal to that of the nearest binomial node.

2. (14 marks) (Delta Hedging for a European Option). The basic idea of these tasks is to evaluate discrete delta hedging effectiveness using simulations. Assume that

$$dS = \mu S dt + \sigma S dZ \tag{1}$$

where μ is the drift rate, S is the asset price, σ is the volatility, and dZ is the increment of a standard Brownian motion.

In practice, continuous hedging is impossible and discrete hedging leads to a hedging error. Using the data given in Table 1, determine the hedging error using discrete delta hedging. You will need to produce *efficient* Matlab code to carry out this simulation.

Suppose that a writer sold an option at the price $V(S(0), 0)$ at $t = 0$. To hedge the risk in this position, the writer will take position $\alpha(t_{i-1})$ in $[t_{i-1}, t_i)$ in the underlying asset $S(t)$ where hedging times are

$$0 = t_0 < t_1 < \cdots < t_N = T, \quad \Delta t = t_i - t_{i-1}$$

In other words, $\alpha(t) = \alpha(t_{i-1})$ for all $t \in [t_{i-1}, t_i)$.

The writer will balance positions using a risk-free bank account $B(t)$. In $[t_{i-1}, t_i)$, the writer holds a portfolio $\Pi(t) = \{-V(S(t), t), \alpha(t_{i-1})S(t), B(t)\}$, i.e., $\Pi(t)$ consists of:

- A short option position $-V(S(t), t)$.
- A total of $\alpha(t_{i-1})$ shares at price $S(t)$.
- An amount $B(t)$ in a risk-free bank account.

Initially, we have

$$\begin{aligned} \Pi(0) &= -V(S(0), 0) + \alpha(0)S(0) + B(0) = 0 \\ & B(0) = V(S(0), 0) - \alpha(0)S(0) \end{aligned}$$

The hedge is rebalanced at time t_i: the position in the underlying is changed from $\alpha(t_{i-1})$ to $\alpha(t_i)$ and the hedging is self-financed using the bank account, i.e.,

$$\alpha(t_{i-1})S(t_i) + e^{r\Delta t}B(t_{i-1}) = \alpha(t_i)S(t_i) + B(t_i)$$

The left side is the value of positions in the underlying and bank account immediately before rebalancing; the right hand side is the value immediately after rebalancing. In other words, the bank account is updated by

$$B(t_i) = e^{r\Delta t}B(t_{i-1}) - (\alpha(t_i) - \alpha(t_{i-1}))\,S(t_i)$$

Immediately after rebalancing time t_i, the hedging error is the portfolio value

$$\Pi(t_i) = -V(S(t_i), t_i) + \alpha(t_i)S(t_i) + B(t_i)$$

At time T, liquidate all positions and the portfolio has the value

$$\Pi(T) = -V(S(T), T) + \alpha(t_{N-1})S(T) + B(t_{N-1})e^{r\Delta t}.$$

Let the delta hedging positions and option values be computed as using matlab functions deltaBinomial and interpBinomial in Question 1. Now, simulate a path of an asset price forward in time using a Monte Carlo method. At each discrete hedging interval, update your portfolio, using the current asset price and the corresponding value of delta. For each simulated path, determine the discounted relative P&L (hedging error)

$$P\&L = \frac{e^{-rT}\Pi(T)}{V(S_0, 0)} \tag{2}$$

where $\Pi(T)$ is the liquidated value of the portfolio at $t = T$ and $V(S_0, 0)$ is the initial Black-Scholes price.

In this Monte Carlo implementation, you can avoid timestepping error in the simulated price by using the exact solution to the SDE (1). Note that when advancing the solution of the SDE from t_i to t_{i+1}, the exact solution for $S(t_{i+1})$, given $S(t_i)$, is

$$S(t_{i+1}) \quad = \quad S(t_i) \exp\left[(\mu - \sigma^2/2)(t_{i+1} - t_i) + \sigma \phi_i \sqrt{t_{i+1} - t_i}\right] \tag{3}$$

where $\phi_i \sim \mathcal{N}(0, 1)$.

Consider a fixed time horizon $\bar{t}$ and a given portfolio of risky assets. Let L be a random variable denoting the portfolio loss, for example, $L = -(P\&L)$ in the hedge example. Let $P(L(\bar{x}, S) \leq \alpha)$ denote the probability of the portfolio loss not exceeding α. For a given confidence level β, e.g., $\beta = 95\%$,

$$\text{VaR}_\beta(L) \stackrel{\text{def}}{=} \min\{\alpha : P(L \leq \alpha) \geq \beta\}. \tag{4}$$

In other words, with a β-confidence level, the portfolio loss does not exceed α. When the loss distribution is continuous, VaR is the value α which solves

$$P(L \leq \alpha) = \beta.$$

Alternative to VaR, Conditional Value-at-Risk (CVaR) of the portfolio is:

$$\text{CVaR}_\beta(L) = \mathbf{E}\left(L | L \geq \text{VaR}_\beta\right) \tag{5}$$

assuming that the loss distribution is continuous.

For a MC sample distribution, VaR and CVaR can be computed based on definition (4) and a slightly modified definition for CVaR for a distribution with jumps (for example a sample distribution) as follows, see e.g., [Rockfellar and Uryasev(2002)]:

(1) Compute M independent samples for the loss.

(2) Sort the loss: $L_1 < L_2 < \cdots < L_M$

(3) Let $i_\beta \leq M$ be such that

$$\frac{i_\beta}{M} \geq \beta > \frac{i_\beta - 1}{M}$$

Then

$$\text{VaR}_\beta = L_{i_\beta}$$

$$\text{CVaR}_\beta = \frac{1}{1 - \beta}\left[(\frac{i_\beta}{M} - \beta) L_{i_\beta} + \frac{1}{M} \sum_{i = i_\beta + 1}^{M} L_i\right]$$

Using MC simulations, conduct hedging performance anaylysis for $N = 250,\ 52,\ 12$ respectively, when hedging positions are computed from the binomial lattice as in Question 1 with the same number of periods respectively. This corresponds approximately daily, weekly and monthly rebalancing.

(a) Plot the histogram of the **probability density** of the relative hedging error, for no hedging, rebalancing monthly, weekly, and daily. The Matlab functions *histc*, *bar* might prove useful. Use at least 50 bins in your histogram.

Table 1: Data for Hedging Simulations Using Analytic Formula

σ	.40
r	.02
μ	0.1
Time to expiry	1.0 years
Strike Price	\$120
Initial asset price S^0	\$100
μ	.10
Payoff	put

(b) Write a Matlab function [var,cvar]=dVaRCVaR(L, β) which returns VaR and CVaR for a discrete loss distribution L with M independent samples using the procedure described. In Matlab, sorting can be done using **sort**.

(c) Compute the standard deviation, VaR (95%) and CVaR (95%). Report in a table, mean, standard deviation, VAR (95%), CVAR (95%), of the relative P&L for no hedging, rebalancing monthly, weekly, and daily. Note that the loss L here corresponds to $-P\&L$.

Submit your Matlab code, a short pseudo-code description and discussion on how hedging performance changes with the rebalancing frequency.

Note: you should be using *at least* 10,000 simulations to get a reasonable histogram for the probability density. Look at the previous handout for hints on how to vectorize your code.

3. (5 marks) (Generate Random Numbers for a Distribution). A Cauchy-distributed random variable has the density function

$$f(x) = \frac{c}{\pi}\frac{1}{c^2 + x^2}$$

Show that its distribution function F and its inverse F^{-1} are

$$F(x) = \frac{1}{\pi}\arctan\left(\frac{x}{c}\right) + \frac{1}{2}, \quad F_c^{-1}(y) = c\,\tan\left(\pi(y - \frac{1}{2})\right).$$

How can this be used to generate Cauchy-distributed random numbers out of random numbers for a uniform distribution?

4. (5 marks) (Numerical Scheme for SDE) The variance in the stochastic volatility in the Heston model

$$dv = -\lambda(v - \bar{v})dt + \eta\sqrt{v}dZ^{(2)}$$

is a mean reverting Cox-Ingersoll-Ross (CIR) process.

Due to possible negative variance in the Euler-Maruyama MC simulation method for the variance, the accuracy of the option value can be poor. This can cause problems in calibrating a Heston model (more discussion on model calibration in the next assignment). Show that the Milstein method for the SDE for the variance gives the following scheme

$$v_{i+1} = \left(\sqrt{v_i} + \frac{\eta}{2}\sqrt{\Delta t}\phi_i^{(2)}\right)^2 - \lambda(v_i - \bar{v})\Delta t - \frac{\eta^2}{4}\Delta t$$

Prove that $v_{i+1} > 0$ if $v_i = 0$ and $\frac{4\lambda\bar{v}}{\eta^2} > 1$.

Table 2: Data for the Heston Model

r	.02
Initial variance v	0.0174
$\bar{v}$	0.0354
η	0.3877
ρ	- 0.7165
λ	1.3253
Initial asset price S^0	\$100
Strike K	\$100
Expiry T	1

5. **(10 marks) (Stochastic Volatility Model)** The Heston stochastic volatility model [Heston(1993)] for pricing options is given by

$$\begin{aligned} \frac{dS}{S} &= r dt + \sqrt{v} dZ^{(1)} \\ dv &= -\lambda(v - \bar{v})dt + \eta\sqrt{v}dZ^{(2)} \\ \mathbf{E}(dZ^{(1)}dZ^{(2)}) &= \rho dt \end{aligned} \tag{6}$$

where λ is the speed of reversion of the variance v to its long-term mean $\bar{v}$.

(a) **(5 marks)** Write a Matlab code to compute the value of a European call with strike K and expiry T using the following Milstein method

$$\begin{aligned} x_{i+1} &= x_i + r\Delta t - \frac{v_i}{2}\Delta t + \sqrt{v_i \Delta t}\phi_i^{(1)} \\ v_{i+1} &= \left(\sqrt{v_i} + \frac{\eta}{2}\sqrt{\Delta t}\phi_i^{(2)}\right)^2 - \lambda(v_i - \bar{v})\delta t - \frac{\eta^2}{4}\Delta t \\ E(\phi_i^{(1)}\phi_i^{(2)}) &= \rho \end{aligned}$$

where $x = \log(S)$ and $\phi_i^{(1)}$ and $\phi_i^{(2)}$ are standard normals with correlation ρ. If at any time $v_i < 0$ then simply set $v_i := |v_i|$ and continue simulation.
Using data in Table 2, $M = 10000$, and $\Delta t = \frac{1}{100}$, compute the initial value of an at-the-money call (i.e., $K = S_0$) with the expiry $T = 1$.

(b) **(5 marks)** Use Matlab function **blsimpv** to plot the implied volatility of option prices computed from (a)

- against the strike, e.g., K=linspace($0.7S_0, 1.2S_0, 30$) and $T = 1$;
- against the expiry, e.g., T=linspace($0.25, 1, 20$) and $K = S_0$.

Discuss characteristics of the implied volatility from the Heston model.

Submit your matlab code, a short pseudo-code description, and discussion.

6. **(Graduate Student Question) (10 marks)** Read Section 1 in Heston's paper [Heston(1993)]. Using the formulae (10), (17), (18) and the expressions therein, write a Matlab code to compute integration in (18) numerically to compute the call option values. However, replace the

formula for $C(\tau;\phi)$ below (17) by the following

$$C(\tau;\phi) = r\phi i\tau + \frac{a}{\sigma^2}\left\{(b_j - \rho\sigma\phi i - d)\tau - 2\ln\left[\frac{e^{-d\tau} - g}{1-g}\right]\right\}$$

The above formula has shown to lead to better stability in computation. You can assume risk neutral pricing, i.e., letting price of volatility risk $\lambda = 0$. You can use Matlab function **quadl** for integration. Compute the call option value for the data given in Table 2. Note that you may see some Matlab warnings in this computation, which can be turned off.

Submit your matlab code, a short pseudo-code description, and discussion.

References

[Heston(1993)] S. Heston. A closed-form solution for options with stochaastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6:327–343, 1993.

[Rockfellar and Uryasev(2002)] R.T. Rockfellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26(7):1443–1471, 2002.