CS 476/676: Assignment 1
Winter 2018

Instructor: Yuying Li  Office: DC3623  e-mail: yuying@uwaterloo.ca
Lecture Times: MWF 10:30-11:20  AL 208
Yuying Li OH: Thursday 3:00-4:00  DC3623
Ke Nian  knian@uwaterloo.ca  DC3594
Ke Nian OH: Jan 17 Wed 3-4  DC3594
Jan 22 Mon 3-4  Jan 23 Tues 4-5
Jan 30 Tues 11-12  Jan 31 Wed 2-3
Course Web Site: http://www.student.cs.uwaterloo.ca/˜cs476
My Web Site: http://www.cs.uwaterloo.ca/˜yuying

Due February 2, 2018 in class

1. (8 marks) (Imply a Binomial Lattice from Option Prices)

Assume that the ABC stock pays no dividend and is currently priced at \( S_0 = \$10 \). Assume that, at the expiry time \( T > 0 \), the stock price goes up to \( uS_0 \) with probability \( 0 < p < 1 \) and down to \( dS_0 \) with probability \( 1 - p \). We know that \( d < 1 < u \) but do not know \( d \) or \( u \). Assume that there is no arbitrage and the interest rate is zero. Consider the following three options with the same expiry \( T \) on the ABC stock. Assume that a European put option with strike price \( \$9 \) is priced at \( \$14/9 \) while a European put option with strike price \( \$8 \) is priced \( \$8/9 \). What is the fair value of a European call option with a strike price of \( \$7 \)? Explain your answer.

2. (10 marks) (Bound on Lattice Solution)

Consider the no-arbitrage lattice with parameters below

\[
\begin{align*}
  u &= e^{\sigma \sqrt{\Delta t}}, & d &= e^{-\sigma \sqrt{\Delta t}}, \\
  q^* &= e^{r \Delta t} - d, & S_{j+1}^n &= uS_j^n, & S_{j+1}^n &= dS_j^n
\end{align*}
\]  

(1)

(a) Show that for a European call,

\[
V_N^N \to \infty, \quad \text{as} \quad \Delta t \to 0.
\]

(b) Show that, if \( \Delta t \) is sufficiently small, and \( \sigma > 0 \), then the up-move risk neutral probability \( q^* \) satisfies \( 0 \leq q^* \leq 1 \)

(c) Assume that the risk neutral probability \( 0 \leq q^* \leq 1 \). Use induction to show that

\[
V_j^n \geq 0, \quad \forall j, n
\]

(d) Using (2), explicitly verify that

\[
S_j^n = e^{-r \Delta t} (q^* S_{j+1}^{n+1} + (1 - q^*) S_j^{n+1}).
\]  

(3)
3. (8 marks) (Lattice properties)

Consider the binomial lattice with the following properties over the time interval $\Delta t$

$$S(t + \Delta t) = S(t)u,$$ with probability $q^*$
$$S(t + \Delta t) = S(t)d,$$ with probability $(1 - q^*)$

$$u = e^{\sigma \sqrt{\Delta t}},$$
$$d = \frac{e^{r\Delta t} - e^{-\sigma \sqrt{\Delta t}}}{e^{\sigma \sqrt{\Delta t}} - e^{-\sigma \sqrt{\Delta t}}},$$

(a) Show that

$$E^Q[S(t + \Delta t)|S(t)] = S(t)e^{r\Delta t},$$

where $E^Q[\cdot]$ is the expectation using $q^*$.

(b) Show that

$$\text{var}^Q[S(t + \Delta t)|S(t)] = S(t)^2 \left[\sigma^2 \Delta t + O(\Delta t)^2\right],$$

where $\text{var}^Q$ is the variance using $q^*$.

4. (4 marks) (Properties of a Standard Brownian Motion)

Consider $W_t = 10Z_t$ where $Z_t$ is a standard Brownian motion. What are $E(W_t - W_s)$ and $\text{var}(W_t - W_s)$? Provide rigorous mathematical arguments.

**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. By default, you should submit listings of 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.

5. (14 marks) (Binomial Lattice)

In lectures we have considered the lattice

$$u = e^{\sigma \sqrt{\Delta t}}, \quad d = e^{-\sigma \sqrt{\Delta t}}, \quad q^* = \frac{e^{r\Delta t} - d}{u - d},$$

Assume $N$ is an even integer and the underlying stock pays a proportional dividend $d = \rho S(t)$ at time $t = t^n, 0 < t^n < T$, and fixed dividend $d = D$ at time $t = t_2, 0 < t_2 < T$, where $t^n < t_2$ are specified a priori, $\rho > 0$ is a small positive constant, and $D > 0$ is a fixed amount. Note that, when the stock pays dividend at $t$, the stock price at $t^+$ becomes $S(t^-) - d$, where $t^-$ and $t^+$ denote immediately before and after the time of stock dividend payment respectively. Since option is dividend protected, the option value immediately before dividend payment and immediately after dividend payment should remain the same. In §5.5.2 of the course notes, the case when dividend is a fixed amount is discussed. Modify the binomial lattice call and put option pricing to handle the dividend payment case in this assignment question using interpolation similar to the technique described in §5.5.2.

Your code should take only $O(N)$ storage NOT $O(N^2)$, $N = T/\Delta t$. This can be done efficiently if you store the payoff in an array of size $O(N)$ and index it appropriately.
Table 1: Some typical option parameters

<table>
<thead>
<tr>
<th>σ</th>
<th>20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>10%</td>
</tr>
<tr>
<td>Time to expiry</td>
<td>0.5 years</td>
</tr>
<tr>
<td>Initial asset price $S(0)$</td>
<td>$105</td>
</tr>
</tbody>
</table>

(a). (8 marks) Using data in Table 1 and assuming first that $\rho = 0$ and $D = 0$ (i.e., no dividend case), test your code for standard European call and put by comparing your results with the exact solutions from the Matlab function `blsprice`. Show tables with the initial option value $V(S(0), 0)$ (both at-the-money puts and calls, i.e., $K = S(0)$) as a function of $\Delta t$. Start off with a timestep $(\Delta t)^0 = .01$, and show the option value for $(\Delta t)^0/2, (\Delta t)^0/4, \ldots$. You should see your results converging to the `blsprice` value. Your tables should look like Table 2.

Table 2: Convergence Test

<table>
<thead>
<tr>
<th>$\Delta t$</th>
<th>Value</th>
<th>Change</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>.05</td>
<td>$V_1$</td>
<td>$V_2 - V_1$</td>
<td></td>
</tr>
<tr>
<td>.025</td>
<td>$V_2$</td>
<td>$V_3 - V_2$</td>
<td>$\frac{V_2 - V_1}{V_3 - V_2}$</td>
</tr>
<tr>
<td>.0125</td>
<td>$V_3$</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Let $V_0^{exact}$ be the exact initial price $V(S(0), 0)$ (from the solution to the Black-Scholes PDE), and $V_0^{tree}(\Delta t)$ be the price from a lattice pricer with the time step $\Delta t$, then it can be shown that

$$V_0^{tree}(\Delta t) = V_0^{exact} + \alpha \Delta t + O((\Delta t)^{3/2})$$

(8)

where $\alpha$ is a constant independent of $\Delta t$. Computationally determine

$$\lim_{\Delta t \to 0} \frac{V_0^{tree}((\Delta t)/2) - V_0^{tree}(\Delta t)}{V_0^{tree}((\Delta t)/4) - V_0^{tree}((\Delta t)/2)}.$$ (9)

Does your convergence table agree with the theory in terms of rate of convergence?

(b). (6 mark) Assume that $t_d^1 = 0.2$ and $t_d^2 = 0.3$ and let $\Delta t = 0.01$. Compute and tabulate at-the-money call initial option values for the dividend parameters specified in Table 3.

Table 3: Dividend Parameters

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
</tr>
<tr>
<td>0.2</td>
<td>1</td>
</tr>
<tr>
<td>0.3</td>
<td>1.5</td>
</tr>
</tbody>
</table>

How does the initial option value change dividend payment increases? Explain why the observation is or is not reasonable.

6. (12 marks: total) (Monte Carlo)

A European cash-or-nothing put barrier option either pays out a pre-specified cash amount $x$ when the underlying price $S_T$ at $T$ satisfies $S_T < K$ and the underlying price $S_t$ never hits the barrier in $[0, T]$. Otherwise it pays nothing.
(a). (4 marks) Consider a down-and-out cash-or-nothing put option with the expiry of 6 month. The initial asset price $S_0 = 105$, the strike price is $K = 102$, the down barrier is $B = 100$, the cash payout is $x = $15. The analytic formula for pricing this barrier option is

\[ z_1 = \frac{\log(S/K)}{\sigma \sqrt{T}} + \frac{\sigma \sqrt{T}}{2}, \quad z_2 = \frac{\log(S/B)}{\sigma \sqrt{T}} + \frac{\sigma \sqrt{T}}{2}, \]

\[ y_1 = \frac{\log(B^2/(SK))}{\sigma \sqrt{T}} + \frac{\sigma \sqrt{T}}{2}, \quad y_2 = \frac{\log(B/S)}{\sigma \sqrt{T}} + \frac{\sigma \sqrt{T}}{2}. \]

\[ V(S, 0) = xe^{-rT} \left( N(-z_1 + \sigma \sqrt{T}) - N(-z_2 + \sigma \sqrt{T}) + \frac{(S/B)N(y_1 - \sigma \sqrt{T}) - (S/B)N(y_2 - \sigma \sqrt{T})}{\sigma \sqrt{T}} \right). \]

In Matlab, the cumulative distribution function for normal distribution $N(d)$ is `normcdf`. Compute and plot the initial down-and-out cash-or-nothing put option for $S = 100 : 2 : 120$. Other parameters can be found in Table 1.

(b). (8 marks) As we derived in class, for the purposes of pricing options, we assume that the asset price $S$ evolves as follows:

\[ dS = rSdt + \sigma SdZ \]

(10)

where $r$ is the risk free return, $\sigma$ is the volatility, and $dZ$ is the increment of a standard Brownian motion. Let the expiry time of an option be $T$, and let

\[ N = \frac{T}{\Delta t}, \quad S(n\Delta t) = S^n \]

(11)

Then, given an initial price $S(0)$, $M$ realizations of the path of a risky asset are generated using the algorithm

\[ S^{n+1} = S^n e^{(r-\frac{1}{2}\sigma^2)\Delta t + \sigma \phi^n \sqrt{\Delta t}} \]

(12)

where $\{\phi^n\}$ are independent standard normals.

If the $m$th simulation value of $S^N = S(T)$ is denoted by $(S^N)^m$, then an approximate initial value of the option is given by (assuming that $S(0) = S^0$)

\[ \hat{V}(S(0), 0) = e^{-rT} \sum_{m=0}^{M} \text{Payoff}((S^N)^m) \]

(13)

For down-and-out cash-or-nothing put

\[ \text{Payoff}(S^N) = \begin{cases} x, & \text{if } S(T) < K \text{ and } S^n > B, \text{ for } n = 1, \ldots, N \\ 0, & \text{otherwise}. \end{cases} \]

- The price simulation using (12) has no time discretization error. Does the error in the computed value $\hat{V}(S(0), 0)$ depend on the time discretization? Explain.
- Code up this MC algorithm in Matlab to determine the initial value of the down-and-out cash-or-nothing European put.
- For a fixed $\Delta t$, plot the MC computed option values versus the number of simulations for a number of values of $M$. Repeat this computation for different $\Delta t$. Show the exact price computed from (a) on the plot. Explain what you see. You might start out using a timestep of 5 days (assuming 250 trading days in a year) with $M = 1000$.

Submit a listing of your code, plots, and discussion.

7. (Graduate Student Question) (10 marks)

Explain why the Monte Carlo method in Problem 6 is slow in obtaining an accurate barrier option value. Read the recent short paper by Nourier et al, *Digital Barrier option pricing : an improved Monte Carlo algorithm*, Math Sci (2016) 10:65-70. Implement the exiting probability approach and produce results and Figures of Example 1 for the barrier option specification in Problem 6. Note that
the existing probability formula in the paper holds when $D = (B, +\infty)$ as well. The diffusion part for the formula is

$$\beta(x_1) = \sigma x_1$$

Comment on your observations of the computation results. Compare and discuss the improvement of the results compared to your implementation in Problem 6.