CS522 Assignment 2

1

(a) From the definition of the model we get that:

\[ \frac{S_{t+1}}{S_t} = 1 + \mu \delta + \sigma \sqrt{\delta} \cdot \epsilon_t \]

So the up ratio \( u \) and the down ratio \( d \) are given by:

\[ u = 1 + \mu \delta + \sigma \sqrt{\delta} \]

\[ d = 1 + \mu \delta - \sigma \sqrt{\delta}. \]

(b) We know that the volatility is the standard deviation of the (continuously compounding) annual return. In the last homework we calculated that for one time period we have:

\[ \text{var}(R_t) = \sigma^2 \delta, \text{ where } R_t \text{ is the return for one time period } [t, t+1]. \]

We also know that for \( n \) independent random variables \( X_1, X_2, ..., X_n \) the variance is equal to the sum of variances:

\[ \text{var}(X_1 + X_2 + ... + X_n) = \text{var}(X_1) + \text{var}(X_2) + ... + \text{var}(X_n) \]

So in our case we apply this theoretical result for \( 1/\delta \) so we get that that the total variation on one year is \( \frac{1}{\delta} \times \sigma^2 \delta = \sigma^2 \).

This implies that the volatility of the stock is \( \sigma \).

(c) (i) In order to construct the tree for the stock price we need to know the up and down ratios:

\[ u = 1.0908 \]

\[ d = 0.9176 \]

In this way we get the following tree just by considering \( S_{t+1}^u = u S_t \) and \( S_{t+1}^d = d S_t \):

```
          132.38
         /    |
     121.36 /  |
    /   /  |   |
111.26 111.36
 /  /   |
93.59 93.67
 /  /   |
85.87 78.79
 /  /   |
78.79 78.79
```
(ii) In order to evaluate the European option we need the risk neutral probability \( p = \frac{e^{r\delta} - d}{u - d} \). This formula gives us that \( p = .5 \).

We can construct now the corresponding binomial tree for the European option taking into account that the discount factor in each period is equal to \( e^{-r\delta} = .996 \).

\[
\begin{align*}
\xi S_{t+1}^u + \eta e^{r\delta} &= V_{t+1}^u \\
\xi S_{t+1}^d + \eta e^{r\delta} &= V_{t+1}^d
\end{align*}
\]

where \( \xi \) is the quantity of stock in our hedge and \( \eta \) is the quantity of money market account.

The value of the option at the corresponding vertex from time \( t \) will be given by:

\[
\xi S_t + \eta
\]

where the value \( S_t \) can be taken from the binomial tree of the stock price. We will use this value calculated here to get the hedge at time \( t - 1 \).

We can observe that at time 2 we have only one nonzero value for the value of the European option, so at each timestep we have to calculate only one hedge, all the other ones are \((0,0)\). So we only have to find three hedges, that will confirm the same binomial tree of the option.
At time 2 the only nonzero hedge will be $\xi_2 \approx .589$ and $\eta_2 \approx -65.32$. This gives the value of the option in the corresponding node of the tree to be 6.15.

At time 1 the only nonzero hedge will be $\xi_1 \approx .32$ and $\eta_1 \approx -32.45$. This gives the value of the option in the corresponding node of the tree to be 3.06.

At time 0 the only we have hedge will be $\xi_0 \approx .173$ and $\eta_0 \approx -16.14$. This gives the value of the option in the corresponding node of the tree to be 1.52.

(iii) In order to hedge our operation at each step we should borrow an amount of money equal to 100 times the corresponding absolute value of $\beta$ and we should sell 100 times the corresponding $\alpha$ shares of the stock.

(iv) In this situation the market is not arbitrage free. For example we could sell 100 of such options with 5 dollars each, we buy 17 shares of stock and borrow 1614 dollars. Buying 17 shares of stock and borrowing 1614 dollars will hedge 100 options. This operation will give us at time 0: 500-152=348 dollars. Depending on which branch of the tree we will be at the next time steps, we have to rebalance our position in the portfolio, so at maturity we will have a perfect hedge for the 100 options, which implies that we won’t have to take money out of our pocket. We observe that this hedging strategy is self-financing and gives an arbitrage opportunity.

(d) function price=MyBinModel(S0,E,T,mu,r,s,M,flag)
    dt=T/M;
    u=1+mu*dt+s*sqrt(dt);
    d=1+mu*dt-s*sqrt(dt);
    pu=(exp(r*dt)-d)/(u-d);
    pd=1-pu;
    discount=exp(-r*dt);
    for i=1:(M+1)
        S(i) = S0 * (u^i * d^M) * (d^M + 1 - i);
    end
    if flag==1
        V(i)=max(S(i)-E,0);
    elseif flag==0
        V(i)=max(E-S(i),0);
    end
    for i=M:-1:1
        for j=1:(i+1)
            payoff(j)=V(j);
        end
        for j=1:i
            V(j)=(pu*payoff(j+1)+pd*payoff(j))*discount;
        end
    end
    price=V(1);
end

We observe that this program gives us the same result as in the calculations from before.
(e) For \( r = 0.05 \) we get the following values of the call options:

<table>
<thead>
<tr>
<th>( M )</th>
<th>( \mu = 0.05 )</th>
<th>( \mu = 0.3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.4072</td>
<td>1.2669</td>
</tr>
<tr>
<td>100</td>
<td>1.3750</td>
<td>1.3736</td>
</tr>
<tr>
<td>200</td>
<td>1.3736</td>
<td>1.3734</td>
</tr>
<tr>
<td>400</td>
<td>1.3714</td>
<td>1.3727</td>
</tr>
</tbody>
</table>

The value of the call option given by \( \text{blsprice} \) is 1.3712.

Since the value of the option does not depend on \( \mu \) both sequences should converge to the true value of the option. When \( M = 10 \), the difference between the binomial and Black-Scholes values is unacceptably large for most purposes. But as \( M \) increases, the differences will decrease showing that higher values of \( m \) tend to bring us closer to the Black-Scholes value than lower values of \( m \). You can try to find the results also for \( M \) an odd number. Perhaps what is most striking about the rate of convergence is that the difference oscillates around 0, always being positive for odd values of \( n \) and negative for even values of \( n \). It is easy to see why this happens. Think of the structure of a binomial tree of the underlying asset price. Along the "spine" (or horizontal center) of the tree, after an even number of moves (2, 4, 6, ...), the price is always the same and equal to the current price \( S \). A tree with an even number of moves will have a central ending node exactly equal to \( S \). A currently at-the-money option, as in our example, will end up finishing exactly at-the-money at this node and be worthless. This node does not contribute to its current value. It is as if any probability concentrated there is thrown away. However, for a tree with an odd number of moves, half the ending nodes lie above and half below the at-the-money point so that no probability is "wasted". Thus, for currently at-the-money options, odd-move binomial trees will tend to overshoot Black-Scholes values; and even-move trees will tend to undershoot Black-Scholes values.

This is not just a curiosity, since we can use this observation to speed up the calculation of option values. Instead of using a tree with a large number of moves, say \( n = 400 \), we could instead calculate two option values, one from a tree with \( n = 100 \) and one from a tree with \( n = 99 \). We know that the \( n = 99 \) option value will be too high and the \( n = 100 \) value too low. But if we average the two values, the errors will tend to cancel. This technique is known in numerical analysis as "Richardson’s extrapolation".

(f) function price=MyBinPrice(S0,E,T,r,s,M,flag)
dt=T/M;
u=1+r*dt+s*sqrt(dt);
d=1+r*dt-s*sqrt(dt);
pu=(exp(r*dt)-d)/(u-d);
pd=1-pu;
discount=exp(-r*dt);
for i=1:(M+1)
    S(i) = S0 * (u(i-1)) * (d(M + 1 - i));
if flag==1
    V(i)=max(S(i)-E,0);
elseif flag==0
V(i) = max(E - S(i), 0);
end
for i = M-1:1
for j = 1:(i+1)
    payoff(j) = V(j);
end
for j = 1:i
    V(j) = (pu*payoff(j + 1) + pd*payoff(j))*discount;
end
end
price = V(1);
end

The value of the call option given by blsprice is 1.3712. Next is the program that plots the errors for different numbers of time steps.

M = 10:100:1000;
n = length(M);
S0 = 102;
r = .05;
sig = .3;
T = 1/4;
E = 120;
V = zeros(n, 1);
err = zeros(n, 1);
v = blsprice(S0, E, r, T, sig);
for i = 1:n
    V(i) = MyBinPrice(S0, E, T, r, sig, M(i), 1);
    err(i) = abs(V(i) - v);
end
plot(M, err);
From the first graph we observe that the value of a call option is always greater than the payoff. From the second graph we see that the value of the put option is sometimes less than the payoff.
(a) Suppose we start with a process $G$ that satisfies
\[ dG_t = A(G_t, t)dX + B(G_t, t)dt. \]

We consider a function $f$ depending on both $G$ and $t$. Ito’s lemma gives us that $f$ will satisfy the following stochastic differential equation:
\[ df = A \frac{\partial f}{\partial G} dX + \left( B \frac{\partial f}{\partial G} + \frac{1}{2} A^2 \frac{\partial^2 f}{\partial G^2} + \frac{\partial f}{\partial t} \right) dt. \]

In our case the process $G$ is just the Wiener process $X_t$, so $A(G, t) = 1$ and $B(G, t) = 0$. The function for which we want to calculate the differential is:
\[ f(X_t, t) = S_0 e^{(\nu - \frac{1}{2} \sigma^2) t + \sigma X_t}. \]

Replacing these in the formulae from above we get:
\[ df = \frac{\partial f}{\partial X} dX + \left( \frac{1}{2} \frac{\partial^2 f}{\partial X^2} + \frac{\partial f}{\partial t} \right) dt. \]

Calculating the partial derivatives of $f$ we get:
\[ \frac{\partial f}{\partial X} = \sigma S_t. \]
\[ \frac{\partial^2 f}{\partial X^2} = \sigma^2 S_t \]

\[ \frac{\partial f}{\partial t} = (\mu - \frac{1}{2} \sigma^2) S_t. \]

All these formulas give us:

\[ dS_t = \sigma S_t dX_t + \left( \frac{1}{2} \sigma^2 + \mu - \frac{1}{2} \sigma^2 \right) S_t dt \]

So:

\[ dS_t = \mu S_t dt + \sigma S_t dX_t. \]

(b) From the definition of a standard brownian motion we have that \( E(dX) = 0 \) and also that the next asset price \((S + dS)\) depends only on the today’s price, so:

\[ E(dS) = E(\mu S dt + \sigma S dX_t) = \mu S dt. \]

So the expected rate of change is \( \mu \).

(c) If the exponential term in the definition of \( S_t \) is:

\[ S_t = S_0 e^{\mu t + \sigma X_t}. \]

Doing the same calculations as above we would get:

\[ dS_t = \sigma S_t dX_t + \left( \frac{1}{2} \sigma^2 + \mu \right) S_t dt. \]

Now the expected rate of change in \( S \) will be: \( \frac{1}{2} \sigma^2 + \mu \).