MA374-Assignment 3

Sourav Bikash
11012338

January 24, 2014

1 Question 1

Here we need to determine the price of an American Call and an American Put option in the Binomial model with the following data:
$S(0) = 100; K = 100; T = 1; M = 100; r = 0.08\%; \sigma = 0.2$
We also need to determine the above for two sets of (u,d) given by:
(a)Set1: $u = e^{\sigma\sqrt{\Delta t}}; d = e^{-\sigma\sqrt{\Delta t}};$
(b)Set2: $u = e^{\sigma\sqrt{\Delta t}+(r-\frac{1}{2}\sigma^2)}; d = e^{-\sigma\sqrt{\Delta t}+(r-\frac{1}{2}\sigma^2)};$
where $\Delta t = \frac{T}{M}$ M being the number of subintervals in the time interval $[0, T]$. We will also we use the compounding convention in our calculations.

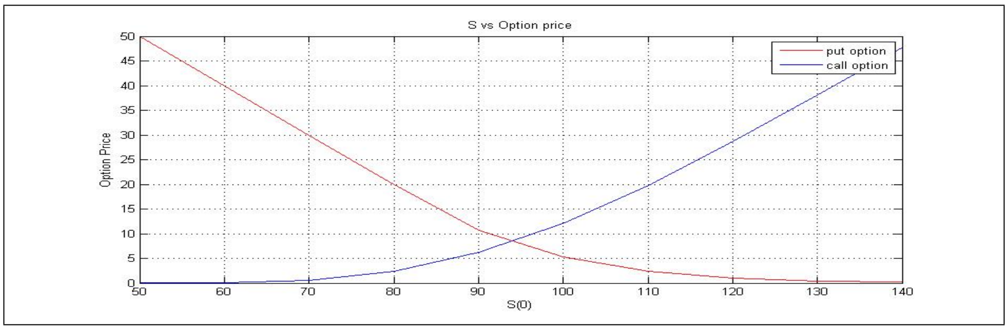
1.1 The Call/Put Option Prices

The following table gives the American Call/Put Option Values at t=0.

Option	Set1	Set2
Call	12.08538001	12.12301638
Put	5.26674759	5.279837145

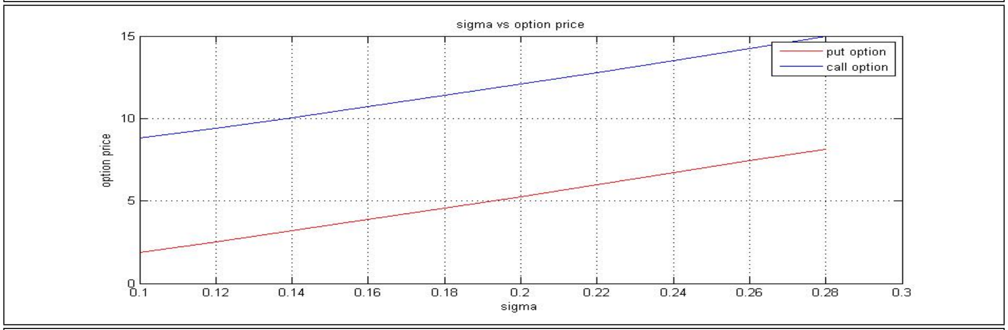
1.2 Varying the Parameters

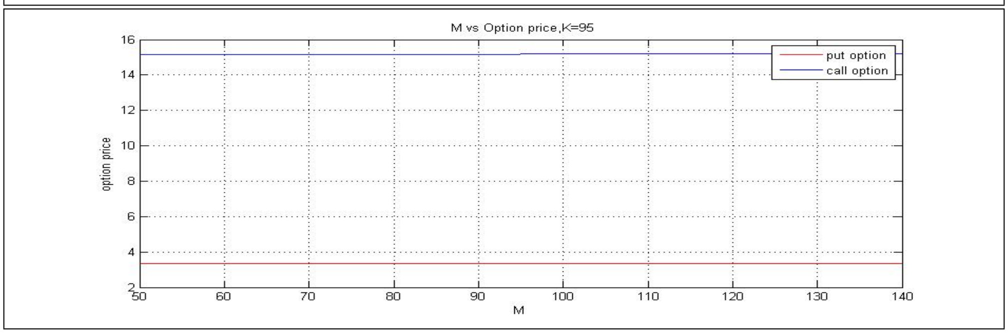
(a) Set1: $u = e^{\sigma\sqrt{\Delta t}}; d = e^{-\sigma\sqrt{\Delta t}};$




K vs Option price
put option
call option
Option price
K

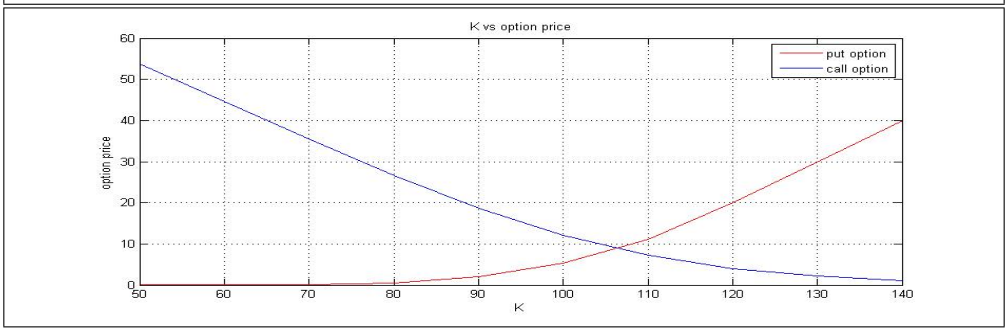

rate vs option price
put option
call option
option price
rate


sigma vs option price
put option
call option
option price
sigma

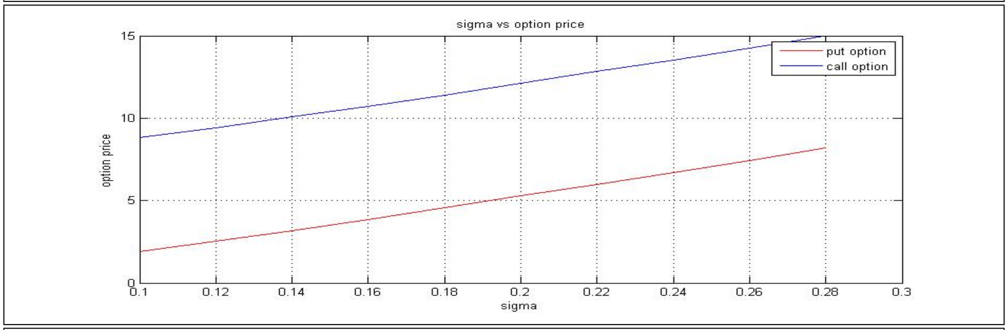

M vs Option price,K=95
put option
call option
option price
M

1.3 Set 2

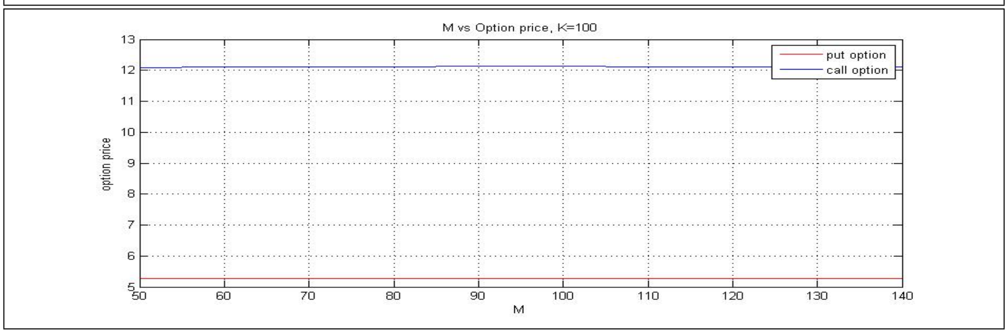
(b) Set2: $u = e^{\sigma\sqrt{\Delta t}+(r-\frac{1}{2}\sigma^2)}$; $d = e^{-\sigma\sqrt{\Delta t}+(r-\frac{1}{2}\sigma^2)}$;




rate vs option price
put option
call option
option price
rate
16
14
12
10
8
6
4
0.04
0.05
0.06
0.07
0.08
0.09
0.1
0.11
0.12
0.13


sigma vs option price
put option
call option
option price
sigma
15
10
5
0
0.1
0.12
0.14
0.16
0.18
0.2
0.22
0.24
0.26
0.28
0.3


M vs Option price, K=95
put option
call option
option price
M
16
14
12
10
8
6
4
2
50
60
70
80
90
100
110
120
130
140


M vs Option price, K=100
put option
call option
option price
M
13
12
11
10
9
8
7
6
5
50
60
70
80
90
100
110
120
130
140

2 Question 2

Here we have to calculate the value of look-back options.

2.1 (Put/Call Option Prices at t=0

Option/M	5	10	15
Call	13.12301638	11.3237088	10.9163330209
Put	1.761724358	2.576634182	3.132208488

2.2 Comparison of prices for varying M

The value of the look back option price converges to $\tilde{1}0$ at t =0.

2.3 The option value tree at M=5

13.123016380101753	24.389042832688094	33.754451890269870	43.998836657225297	55.299778147246116	66.065747876824730
0	2.278984839813039	15.809289559294520	24.597739518736827	34.115918120583643	46.316455617996667
0	0	4.631207439337448	21.792472666301542	30.722517951295142	38.864459138769206
0	0	0	10.335389034543775	19.265735975637796	30.467365950501801
0	0	0	9.411243976477726	30.722517951295142	38.864459138769206
0	0	0	0	13.564371592427966	23.570691986538520
0	0	0	0	12.540695197213665	21.322671560876586
0	0	0	0	8.463232085674232	17.830054009036004
0	0	0	0	19.124928940228930	38.864459138769206
0	0	0	0	0	23.570691986538520
0	0	0	0	0	16.118695507311060
0	0	0	0	0	11.447319946051095
0	0	0	0	0	16.118695507311060
0	0	0	0	0	9.366821923361798
0	0	0	0	0	9.366821923361798
0	0	0	0	0	7.832552328620935
0	0	0	0	0	38.864459138769178
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0

3 Question 3

4 Matlab Codes

4.1 Question 1

4.1.1 Driver Script

```
%driver to run the entire process
i=1;
S=100;K=100;T=1;M=100;r=0.08;sigma=0.2;%fixed parameters
display('for appropriate set of u and d:');
display('displaying put option value');
americanput(S,K,T,M,r,sigma)
display('displaying call option value');
americancall(S,K,T,M,r,sigma)
V=10;% varying upto V values each parameters
X=zeros(1,V);
X1=zeros(1,V);
Y=zeros(1,V);
%varying S
s1=50;
while(i<=V)
    X(i)=americanput(s1,K,T,M,r,sigma);
    X1(i)=americancall(s1,K,T,M,r,sigma);
    Y(i)=s1;
    s1=s1+10;
    i=i+1;
end
plot(Y,X,'r');
hold on
plot(Y,X1);
figure
%varying K
k1=50;i=1;
while(i<=V)
    X(i)=americanput(S,k1,T,M,r,sigma);
    X1(i)=americancall(S,k1,T,M,r,sigma);
    Y(i)=k1;
    k1=k1+10;
    i=i+1;
end
plot(Y,X,'r');
hold on
```

```
plot(Y,X1);
figure
%varying r
r1=0.04;i=1;
while(i<=V)
    X(i)=americanput(S,K,T,M,r1,sigma);
    X1(i)=americancall(S,K,T,M,r1,sigma);
    Y(i)=r1;
    r1=r1+0.01;
    i=i+1;
end
plot(Y,X,'r');
hold on
plot(Y,X1);
figure
%varying sigma
sigma1=0.1;i=1;
while(i<=V)
    X(i)=americanput(S,K,T,M,r,sigma1);
    X1(i)=americancall(S,K,T,M,r,sigma1);
    Y(i)=sigma1;
    sigma1=sigma1+0.02;
    i=i+1;
end
plot(Y,X,'r');
hold on
plot(Y,X1);
figure
%varying M, k=100
M1=50;i=1;
while(i<=V)
    X(i)=americanput(S,K,T,M1,r,sigma);
    X1(i)=americancall(S,K,T,M1,r,sigma);
    Y(i)=M1;
    M1=M1+10;
    i=i+1;
end
plot(Y,X,'r');
hold on
plot(Y,X1);
figure
%varying M, k=95
M1=50;i=1;K=95;
```

```
while(i<=V)
    X(i)=americanput(S,K,T,M1,r,sigma);
    X1(i)=americancall(S,K,T,M1,r,sigma);
    Y(i)=M1;
    M1=M1+10;
    i=i+1;
end
plot(Y,X,'r');
hold on
plot(Y,X1);
figure
%varying M, k=105
K=105;
M1=50;i=1;
while(i<=V)
    X(i)=americanput(S,K,T,M1,r,sigma);
    X1(i)=americancall(S,K,T,M1,r,sigma);
    Y(i)=M1;
    M1=M1+10;
    i=i+1;
end
plot(Y,X,'r');
hold on
plot(Y,X1);
```

4.2 Member Function 1

```
function[P]=americanput(S,K,T,M,r,sigma)
%S is the initial stock price
%K is the strike price of the option
%T is the expiry time period
%M refers to the number of time steps in the binomial model
%r is the risk free rate
%sigma is the volatility factor
%t=T/M;
%set1
%u=exp(sigma*sqrt(t));
%d=1/u;
%set2
%u=exp(sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
%d=exp(-1*sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
t=T/M;
%u=exp(sigma*sqrt(t));
%d=1/u;
```

```
u=exp(sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
d=exp(-1*sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
% create the price tree
U=zeros(M+1,M+1);
U(1,1)=S;% starting price
i=M+1;
while(i>=1)
     j=1;
     while(j<=i)
          U(j,i)=S*(u^(i-j))*(d^(j-1));
          j=j+1;
     end
     i=i-1;
end
%U%price tree
%calculate the risk neutral measure
p=(exp(r*t)-d)/(u-d);
%calculate the option prices
i=M+1;j=1;
while(j<=i)
     U(j,M+1)=max(0,(K-U(j,M+1)));%put formula applied
     j=j+1;
end
i=M;
while(i>=1)
     j=1;
     while(j<=i)
          A=(p*(U(j,i+1))+(1-p)*U(j+1,i+1))*(exp(-1*r*t));
          G=max(max((K-U(j,i)),0),A);
          U(j,i)=G;
          j=j+1;
     end
     i=i-1;
end
%U%option price tree
P=U(1,1);
end
```

4.3 Member Function 2

```
function[P]=americancall(S,K,T,M,r,sigma)
%S is the initial stock price
%K is the strike price of the option
%T is the expiry time period
```

```
%M refers to the number of time steps in the binomial model
%r is the risk free rate
%sigma is the volatility factor
%t=T/M;
%set1
%u=exp(sigma*sqrt(t));
%d=1/u;
%set2
%u=exp(sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
%d=exp(-1*sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
t=T/M;
%u=exp(sigma*sqrt(t));
%d=1/u;
u=exp(sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
d=exp(-1*sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
% create the price tree
U=zeros(M+1,M+1);
U(1,1)=S;% starting price
i=M+1;
while(i>=1)
    j=1;
    while(j<=i)
        U(j,i)=S*(u^(i-j))*(d^(j-1));
        j=j+1;
    end
    i=i-1;
end
%calculate the risk neutral measure
p=(exp(r*t)-d)/(u-d);
%calculate the option prices
i=M+1;j=1;
while(j<=i)
    U(j,M+1)=max(0,(-K+U(j,M+1)));%put formula applied
    j=j+1;
end
i=M;
while(i>=1)
    j=1;
    while(j<=i)
        A=(p*(U(j,i+1))+(1-p)*U(j+1,i+1))*(exp(-1*r*t));
        G=max(max((-K+U(j,i)),0),A);
        U(j,i)=G;
        j=j+1;
```

```
    end
    i=i-1;
end
P=U(1,1);
end
```

4.4 Question 2

4.4.1 Driver Script

```
function[P]=lookbackcall(S,T,M,r,sigma)
    %S is the initial stock price
    %T is the expiry time
    %M is the number of time steps
    %r is the risk free rate
    %sigma is the volatility factor
    %Here we will calculate the price of a lookback call option at t=0;
    %calculating all paths as this is path dependent is the first step we
    %will take to calculate the option price payoff.
    t=T/M;
    %u=exp(sigma*sqrt(t));
    %d=1/u;
    u=exp(sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
    d=exp(-1*sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
    U=zeros(2^(M),M+1);%tree storage matrix
    %calculate the paths 2^N paths
    i=0;
    while(i<=2^(M-1))
        stock=S;%starting value of the stock path
        min_stock=S;
        X=de2bi(i,M);%one path calculated
        %use this path to calculate the final payoff
        j=1;
        while(j<=M)
            if(X(j)==0)
                stock=stock*u;
            end
            if(X(j)==1)
                stock=stock*d;
            end
            min_stock=min(min_stock,stock);%maintains the min value of the pa
            j=j+1;
        end
        U(i+1,M+1)=stock-min_stock;%payoff calculated for call option
        i=i+1;
```

```
    end
    %calculate the risk neutral measure
    p=(exp(r*t)-d)/(u-d);
    %backtracking the option tree to get the remaining values
    i=M;
    while(i>=1)
        j=1;
        while(j<=2^(i-1))
            A=(p*(U(1+(2*(j-1)),i+1))+(1-p)*U(2+(2*(j-1)),i+1))*(exp(-1*r*t))
            U(j,i)=A;
            j=j+1;
        end
        i=i-1;
    end
    U%the option tree
    P=U(1,1);
end
```

4.5 Member Function 1

```
function[P]=lookbackcall(S,T,M,r,sigma)
    %S is the initial stock price
    %T is the expiry time
    %M is the number of time steps
    %r is the risk free rate
    %sigma is the volatility factor
    %Here we will calculate the price of a lookback call option at t=0;
    %calculating all paths as this is path dependent is the first step we
    %will take to calculate the option price payoff.
    t=T/M;
    %u=exp(sigma*sqrt(t));
    %d=1/u;
    u=exp(sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
    d=exp(-1*sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
    U=zeros(2^(M),M+1);%tree storage matrix
    %calculate the paths 2^N paths
    i=0;
    while(i<=2^(M-1))
        stock=S;%starting value of the stock path
        min_stock=S;
        X=de2bi(i,M);%one path calculated
        %use this path to calculate the final payoff
        j=1;
        while(j<=M)
```

```
            if(X(j)==0)
                stock=stock*u;
            end
            if(X(j)==1)
                stock=stock*d;
            end
            min_stock=min(min_stock,stock);%maintains the min value of the pa
            j=j+1;
        end
        U(i+1,M+1)=stock-min_stock;%payoff calculated for call option
        i=i+1;
    end
    %calculate the risk neutral measure
    p=(exp(r*t)-d)/(u-d);
    %backtracking the option tree to get the remaining values
    i=M;
    while(i>=1)
        j=1;
        while(j<=2^(i-1))
            A=(p*(U(1+(2*(j-1)),i+1))+(1-p)*U(2+(2*(j-1)),i+1))*(exp(-1*r*t))
            U(j,i)=A;
            j=j+1;
        end
        i=i-1;
    end
    U%the option tree
    P=U(1,1);
end
```

4.6 Member Function 2

```
function[P]=lookbackput(S,T,M,r,sigma)
    %S is the initial stock price
    %T is the expiry time
    %M is the number of time steps
    %r is the risk free rate
    %sigma is the volatility factor
    %Here we will calculate the price of a lookback call option at t=0;
    %calculating all paths as this is path dependent is the first step we
    %will take to calculate the option price payoff.
    t=T/M;
    %u=exp(sigma*sqrt(t));
    %d=1/u;
    u=exp(sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
```

```
    d=exp(-1*sigma*sqrt(t)+t*(r-(sigma*sigma)/2));
    U=zeros(2^(M),M+1);%tree storage matrix
    %calculate the paths 2^N paths
    i=0;
    while(i<=2^(M-1))
        stock=S;%starting value of the stock path
        max_stock=S;
        X=de2bi(i,M);%one path calculated
        %use this path to calculate the final payoff
        j=1;
        while(j<=M)
            if(X(j)==0)
                stock=stock*u;
            end
            if(X(j)==1)
                stock=stock*d;
            end
            max_stock=max(max_stock,stock);%maintains the max value of the pa
            j=j+1;
        end
        U(i+1,M+1)=max_stock-stock;%payoff calculated for put option
        i=i+1;
    end
    %calculate the risk neutral measure
    p=(exp(r*t)-d)/(u-d);
    %backtracking the option tree to get the remaining values
    i=M;
    while(i>=1)
        j=1;
        while(j<=2^(i-1))
            A=(p*(U(1+(2*(j-1)),i+1))+(1-p)*U(2+(2*(j-1)),i+1))*(exp(-1*r*t))
            U(j,i)=A;
            j=j+1;
        end
        i=i-1;
    end
    %U%the option tree
    P=U(1,1);
end
```