Financial Engineering II
Lab Assignment 1

Kumar Harsha, 11012318

January 11, 2014

Contents

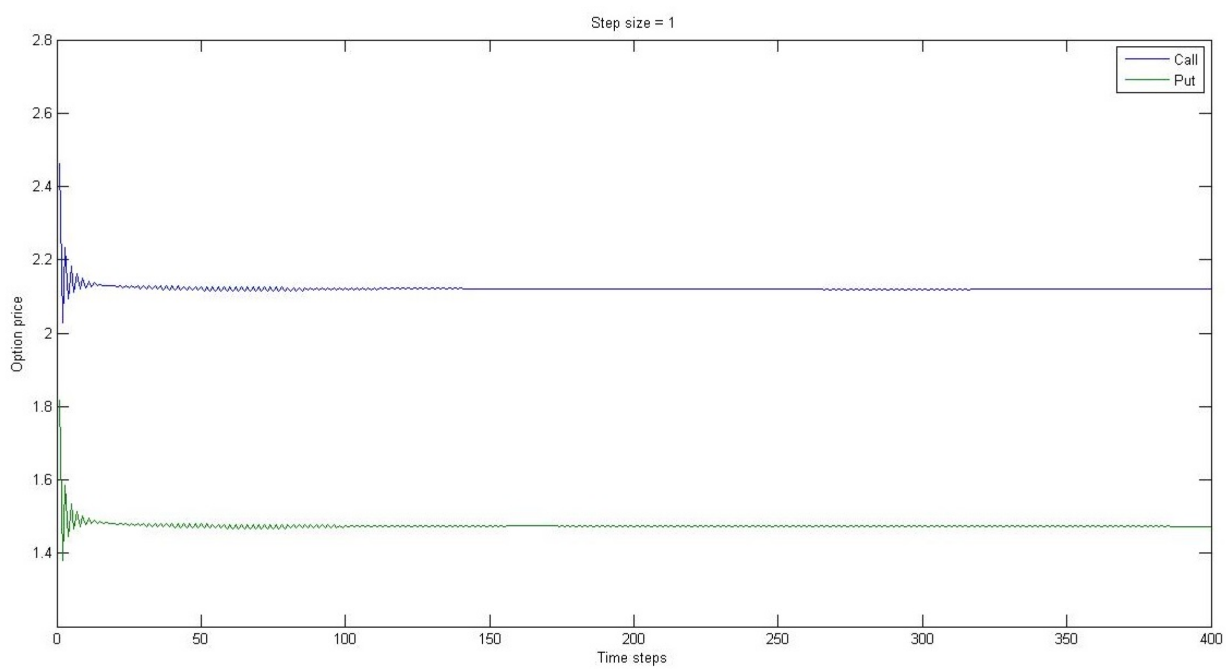
1 Question 1

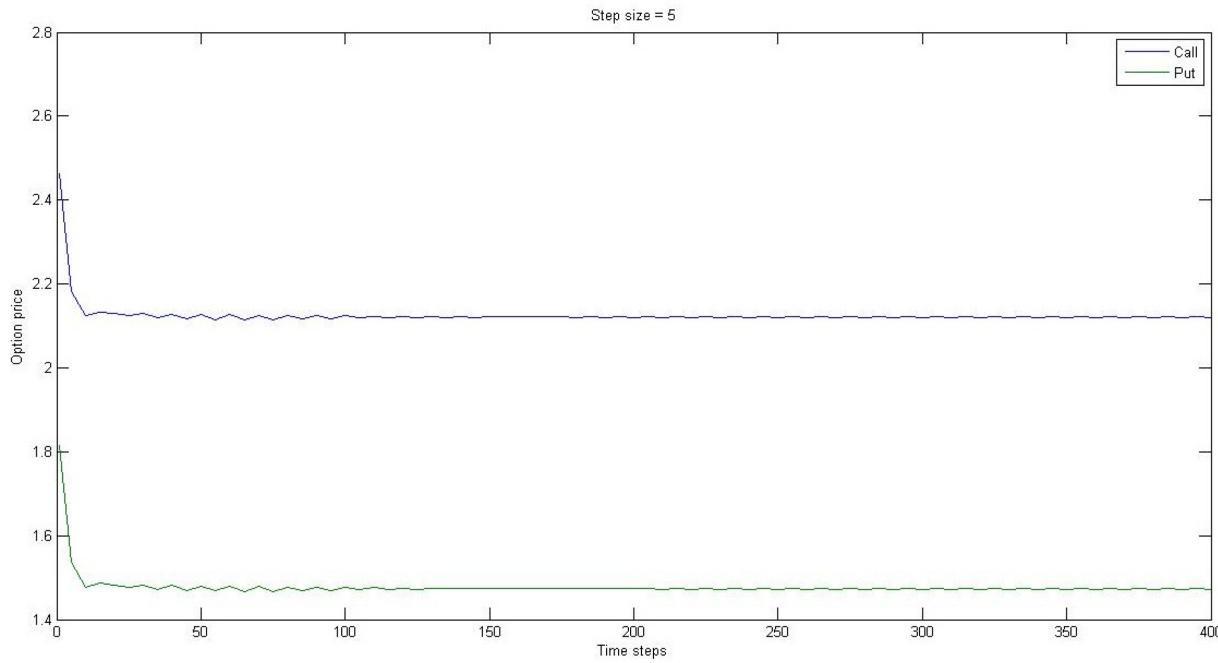
The initial option prices for various time steps are tabulated below:

Steps	Call option	Put option
1	2.4634	1.8161
5	2.1832	1.5359
10	2.1248	1.4775
20	2.1296	1.4823
50	2.1272	1.4799
100	2.1237	1.4764
200	2.1202	1.4729
400	2.1201	1.4728

2 Question 2

The plots for varying time steps are as follows:





The option prices converge after the number of time steps goes beyond 100, then closely following the continuous model.

3 Question 3

The values of call option at the specified time intervals are tabulated below:

0	0.30	0.75	1.50	2.70
2.1296	3.7248	7.7407	20.2796	66.0571
0	1.9992	4.6752	14.1865	50.3233
0	0.9492	2.5443	9.3937	37.8520
0	0	1.2110	5.7170	27.9667
0	0	0.4877	3.0718	20.1311
0	0	0.1599	1.3855	13.9202
0	0	0	0.4934	8.9973
0	0	0	0.1279	5.0951
0	0	0	0.0212	2.1458
0	0	0	0.0017	0.4610
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

Following is the table for the values of the put option:

0	0.30	0.75	1.50	2.70
1.4823	0.8236	0.2062	0.0005	0
0	1.4629	0.5147	0.0074	0
0	2.2876	1.0582	0.0497	0
0	0	1.8447	0.2057	0
0	0	2.8017	0.5983	0
0	0	3.8058	1.3200	0
0	0	0	2.3365	0
0	0	0	3.4839	0
0	0	0	4.5764	0.1437
0	0	0	5.5074	0.9106
0	0	0	6.2591	2.3929
0	0	0	0	3.9333
0	0	0	0	5.1542
0	0	0	0	6.1220
0	0	0	0	6.8891
0	0	0	0	7.4972
0	0	0	0	7.9792
0	0	0	0	8.3612
0	0	0	0	8.6640
0	0	0	0	0
0	0	0	0	0

4 Code

The following are the Matlab code snippets for the assignment.

Function to evaluate option price at t=0

```
function [ price ] = european( start, strike, rate, steps, sigma,
  T, callorput )
%compute value of european option
%   start = price of asset at t = 0
%   strike = strike price for the option
%   rate = risk free interest rate
%   steps = number of time steps
%   sigma = volatility
%   T = total time
%   callorput = 0 for call, 1 for put

optionprices = zeros(steps+1);
```

```
dt = T/steps;
up = exp(sigma*sqrt(dt) + (rate-sigma*sigma*0.5)*dt);
down = exp(-sigma*sqrt(dt) + (rate-sigma*sigma*0.5)*dt);
p = (exp(rate*dt) - down)/(up - down);
for i = 1:steps+1
    if callorput == 0 %call option
        optionprices(i,steps+1) = max(start*(up^(steps-i+1))*(down
           ^(i-1)) - strike, 0);
    elseif callorput == 1 %put option
        optionprices(i,steps+1) = max(strike - start*(up^(steps-i
           +1))*(down^(i-1)), 0);
end
for i = 1:steps
    for j = 1:steps+1-i
        optionprices(j,steps+1-i) = (optionprices(j,steps+2-i)*p +
            optionprices(j+1,steps+2-i)*(1-p))*exp(-rate*dt);
    end
end

price = optionprices(1,1);

end
```

Function to compute option price at different times

```
function [ pricetable ] = dispoptionprices( start, strike, rate,
   steps, sigma, T, t, callorput )
%compute value of european option
%   start = price of asset at t = 0
%   strike = strike price for the option
%   rate = risk free interest rate
%   steps = number of time steps
%   sigma = volatility
%   T = total time
%   t = array of times at which option prices are required
%   callorput = 0 for call, 1 for put

optionprices = zeros(steps+1);
dt = T/steps;
t = int8(t./(T/steps));
up = exp(sigma*sqrt(dt) + (rate-sigma*sigma*0.5)*dt);
down = exp(-sigma*sqrt(dt) + (rate-sigma*sigma*0.5)*dt);
p = (exp(rate*dt) - down)/(up - down);
for i = 1:steps+1
    if callorput == 0 %call option
```

```
        optionprices(i,steps+1) = max(start*(up^(steps-i+1))*(down
           ^(i-1)) - strike, 0);
    elseif callorput == 1 %put option
        optionprices(i,steps+1) = max(strike - start*(up^(steps-i
           +1))*(down^(i-1)), 0);
end
for i = 1:steps
    for j = 1:steps+1-i
        optionprices(j,steps+1-i) = (optionprices(j,steps+2-i)*p +
            optionprices(j+1,steps+2-i)*(1-p))*exp(-rate*dt);
    end
end

n = size(t);
n = max(n(1), n(2));
pricetable = zeros(steps+1,n);
for i=1:n
    pricetable(:,i) = optionprices(:,t(i)+1);
end

end
```

Final code

```
format long; clc; clear all;
%initial values
s0 = 9;
strike = 10;
rate = 0.06;
sigma = 0.3;
T = 3;
steps = [1,5,10,20,50,100,200,400];
n = size(steps);
n = n(2);

%question 1
%valuate option for call and put
cnp = zeros(n,3);
cnp(:,1) = steps';
for i=1:n
    cnp(i,2) = (european(s0, strike,rate,steps(i),sigma,T,0));
    cnp(i,3) = (european(s0, strike,rate,steps(i),sigma,T,1));
end

uitable('Data', cnp, 'ColumnName', {'Steps', 'Call', 'Put'});
```

```
%question 2
%compare option prices for various step sizes
n = 400;
stepsof1 = zeros(n,3);
stepsof1(:,1) = [1:n]';
for i=1:n
    stepsof1(i,2) = (european(s0, strike,rate,stepsof1(i,1),sigma,
       T,0));
    stepsof1(i,3) = (european(s0, strike,rate,stepsof1(i,1),sigma,
       T,1));
end

stepsof5 = zeros(1+n/5,3);
stepsof5(1,:) = stepsof1(1,:);
for i=1:n/5
    stepsof5(i+1,:) = stepsof1(5*i,:);
end

figure
plot(stepsof1(:,1), stepsof1(:,2), stepsof1(:,1), stepsof1(:,3));
legend('Call', 'Put'); xlabel('Time steps'); ylabel('Option price'
   ); title('Step size = 1');
figure
plot(stepsof5(:,1), stepsof5(:,2), stepsof5(:,1), stepsof5(:,3));
legend('Call', 'Put'); xlabel('Time steps'); ylabel('Option price'
   ); title('Step size = 5');


%question 3
%option prices at different times for 20 time steps
t = [0, 0.30, 0.75, 1.50, 2.70];
n = 20;
callprices = dispoptionprices(s0,strike,rate,n,sigma,T,t,0);
putprices = dispoptionprices(s0,strike,rate,n,sigma,T,t,1);

figure
uitable('Data', callprices);
figure
uitable('Data', putprices);
```