Ayush Sharma
(150123046)

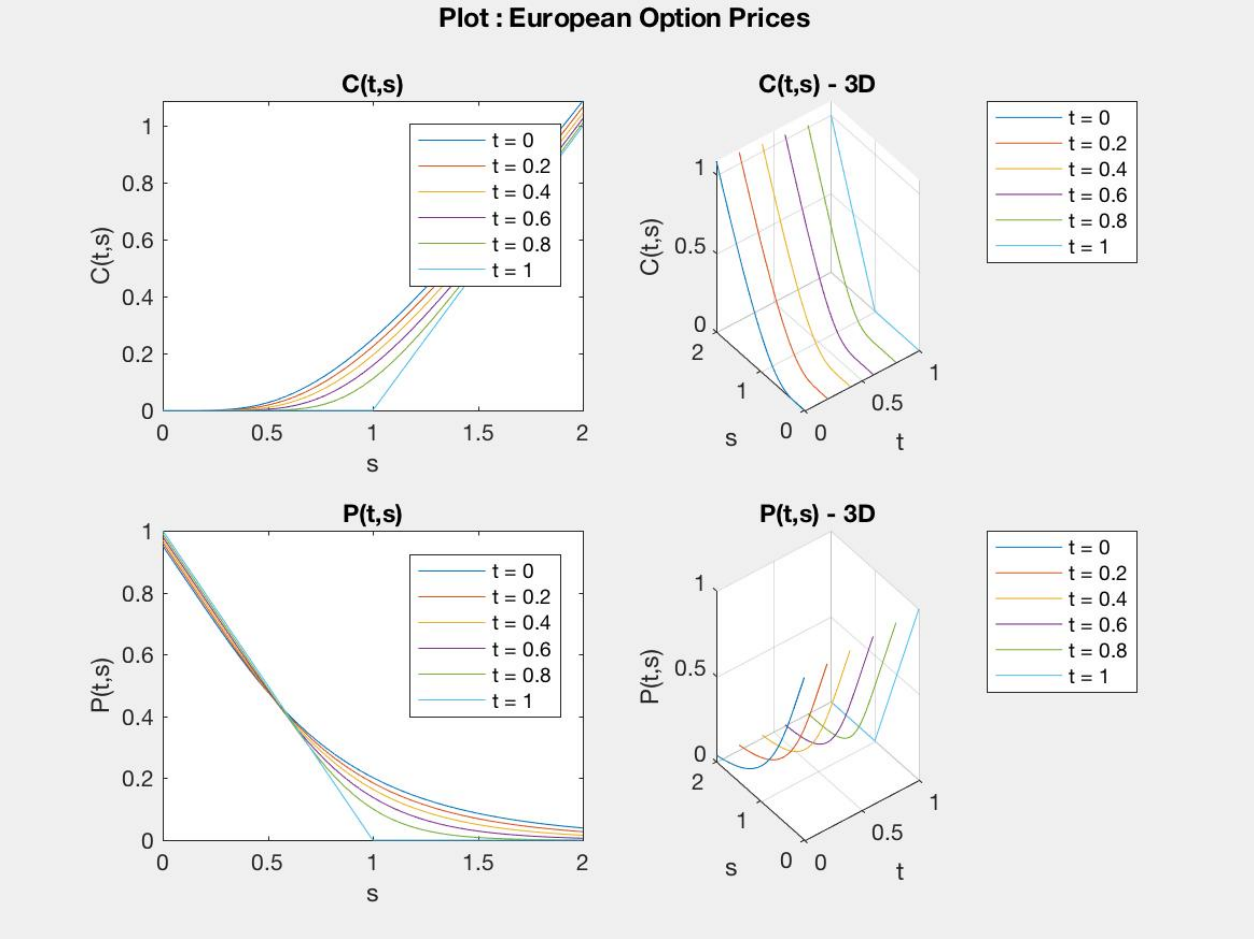
18 March 2018

QUESTION 1.

Write a single program to compute the prices of European call and put options at time t for $0 \leq t \leq T$ in the classical BSM framework. Denote the call and put prices by $C(t, s)$ and $P(t, s)$ respectively, with s being the price of an underlying asset.
{Code included at the end.}

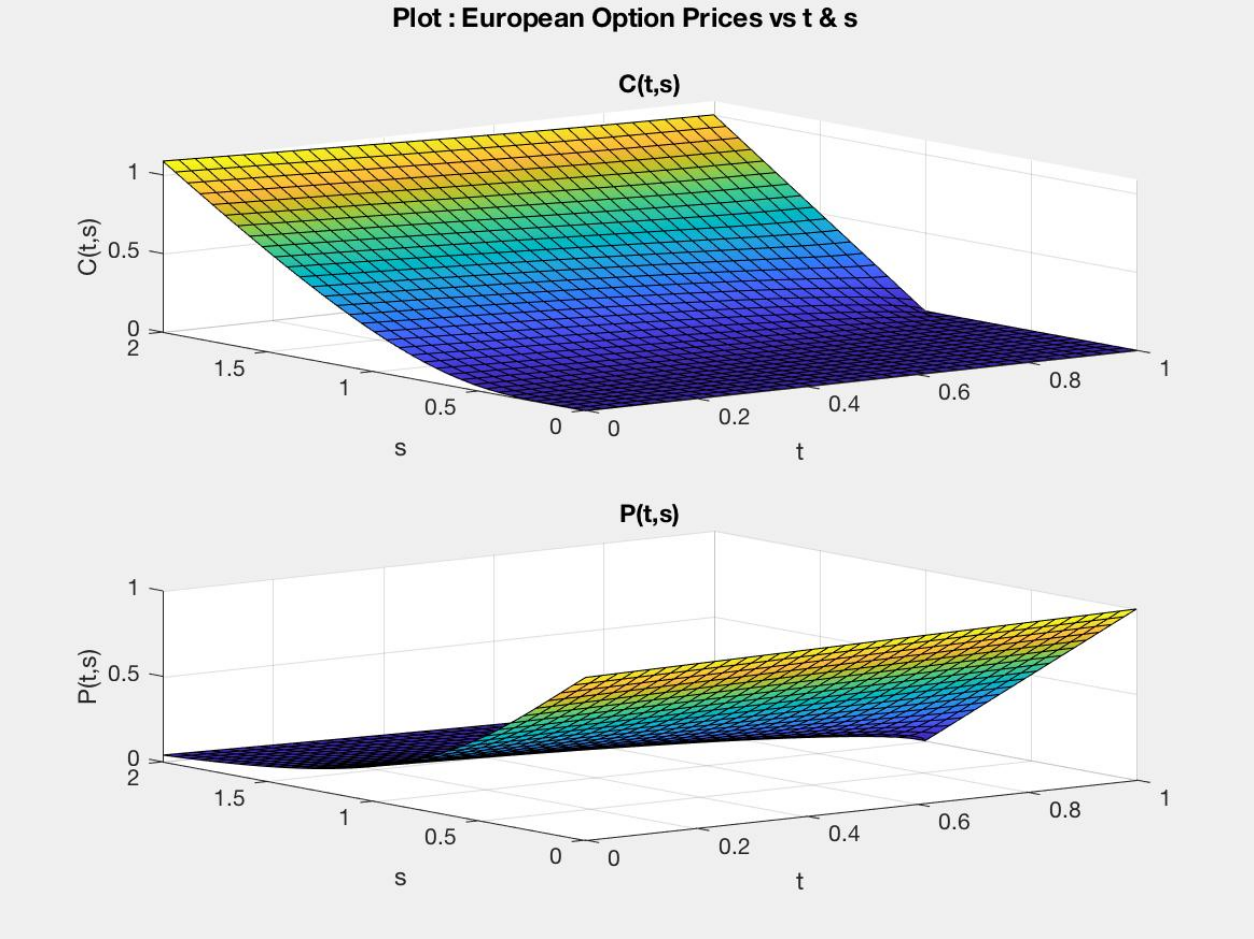
QUESTION 2.

Assume $T = 1, K = 1, r = 0.05, \sigma = 0.6$. Plot, in a single graph, $C(t, s)$ as a function of s alone for $t = 0,0.2,0.4,0.6,0.8,1$. Do a similar plot for $P(t, s)$ as a function of s. Now, show the same information in a 3-dimensional form, i.e., as a function both t and s.



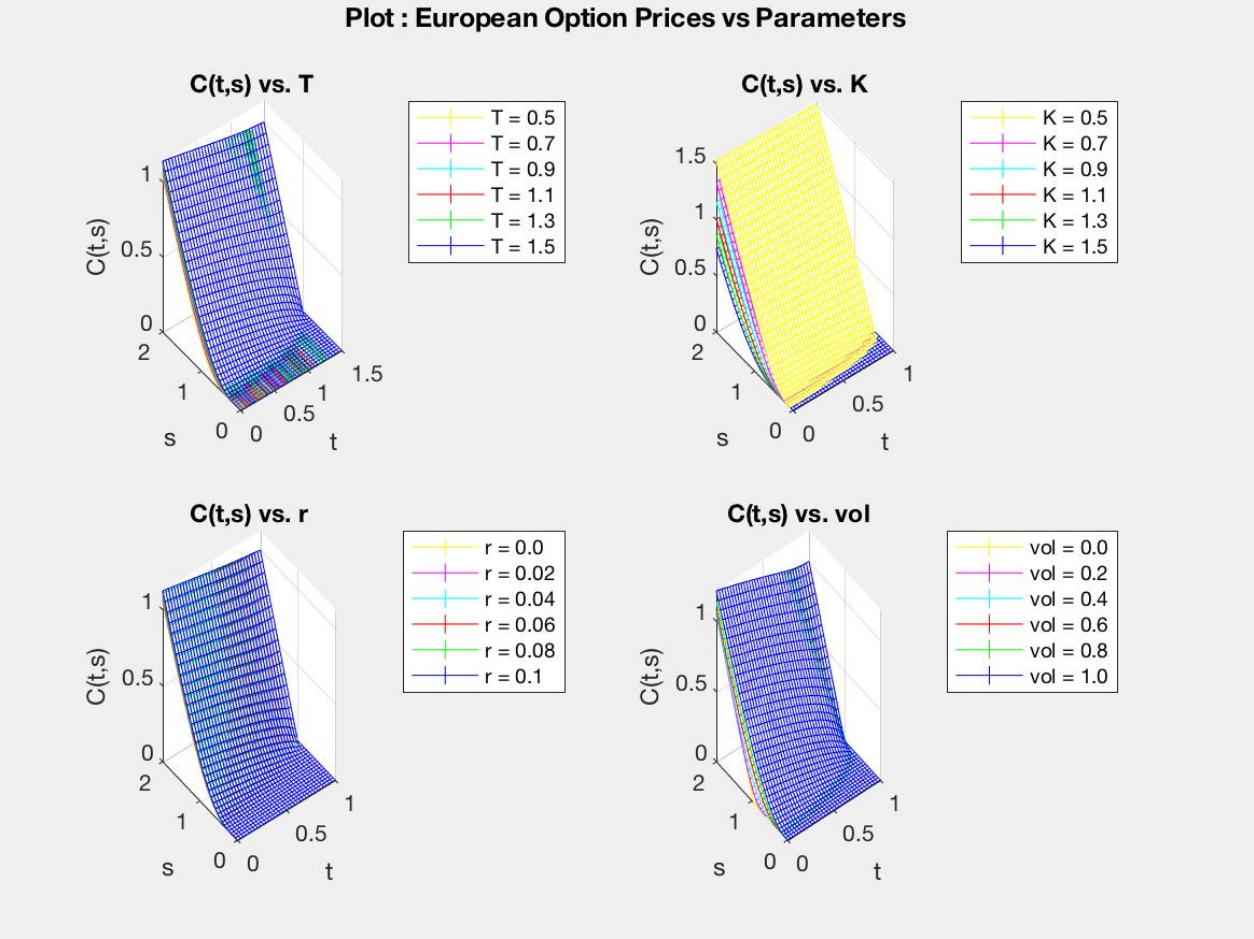
QUESTION 3.

Plot $C(t, s)$ and $P(t, s)$ as a smooth surface above the (t, s)-plane.



QUESTION 4.

Study the sensitivity of both the functions C and P as a function of model parameters. If required, you may assume different parameter values as opposed to the one given above. Present your results in the form of tables and graphs (both in two and three dimensional).



CODE (MATLAB)

###FUNCTION FOR "BSM PRICING ALGORITHM" FOR OPTIONS

```
function [ OptionValue ] = bsmopt( S, t, T, K, r, vol, Flag )
%BSMOPT Summary of this function goes here
%   Detailed explanation goes here

d1 = (log(S./K) + (r + (vol./2))*(T-t))./(vol .* sqrt(T-t));
d2 = d1 - (vol .* sqrt(T-t));

%Flag = 1 for a call option, or Flag = 0 for a put option.
if (Flag == 1)
    OptionValue = S.*normcdf(d1, 0, 1) - K.*exp(-r.*(T-t)).*normcdf(d2, 0, 1);
else
    OptionValue = K.*exp(-r.*(T-t)).*normcdf(-d2, 0, 1) - S.*normcdf(-d1, 0, 1);
end

end
```

SCRIPT FOR ASSIGNMENT PROBLEMS

```
% % Part-1-2

T = 1;
K = 1;
r = 0.05;
vol = 0.6;

% t = 0; S = 1;

C = @(t,s) bsmopt( s, t, T, K, r, vol, 1 );
P = @(t,s) bsmopt( s, t, T, K, r, vol, 0 );

% % Part-2

F = figure('Color','white');
p = uipanel('Parent',F,'BorderType','none');
p.Title = 'Plot : European Option Prices vs t & s';
p.TitlePosition = 'centertop';
p.FontSize = 12;
p.FontWeight = 'bold';

subplot(2,2,1, 'Parent',p);
for t = 0:0.2:1
    fplot(@(s) C(t, s), [0 2]);
    hold on;
end
hold off;
xlabel('s');
ylabel('C(t,s)');
legend({'t = 0' 't = 0.2' 't = 0.4' 't = 0.6' 't = 0.8' 't = 1'});
title('C(t,s)');

subplot(2,2,2, 'Parent',p);
for t_ = 0:0.2:1
    fplot3(@(s) t_, @(s) s, @(s) C(t_, s), [0 2]);
    hold on;
end
hold off;
xlabel('t');
ylabel('s');
zlabel('C(t,s)');
legend({'t = 0' 't = 0.2' 't = 0.4' 't = 0.6' 't = 0.8' 't = 1'});
title('C(t,s) - 3D');

subplot(2,2,3, 'Parent',p);
for t = 0:0.2:1
    fplot(@(s) P(t, s), [0 2]);
    hold on;
end
hold off;
xlabel('s');
ylabel('P(t,s)');
legend({'t = 0' 't = 0.2' 't = 0.4' 't = 0.6' 't = 0.8' 't = 1'});
title('P(t,s)');

subplot(2,2,4, 'Parent',p);
for t_ = 0:0.2:1
    fplot3(@(s) t_, @(s) s, @(s) P(t_, s), [0 2]);
    hold on;
end
hold off;
```

```
xlabel('t');
ylabel('s');
zlabel('P(t,s)');
legend({'t = 0' 't = 0.2' 't = 0.4' 't = 0.6' 't = 0.8' 't = 1'});
title('P(t,s) - 3D');

saveas(F,'2.jpg');
clear('F');

% % Part-3

F = figure('Color','white');
p = uipanel('Parent',F,'BorderType','none');
p.Title = 'Plot : European Option Prices vs t & s';
p.TitlePosition = 'centertop';
p.FontSize = 12;
p.FontWeight = 'bold';

subplot(2,1,1, 'Parent',p);
fsurf(C, [0 0.99999 0 2]);
xlabel('t');
ylabel('s');
zlabel('C(t,s)');
title('C(t,s)');

subplot(2,1,2, 'Parent',p);
fsurf(P, [0 0.99999 0 2]);
xlabel('t');
ylabel('s');
zlabel('P(t,s)');
title('P(t,s)');

saveas(F,'3.jpg');
clear('F');

% % Part-4

F = figure('Color','white');
p = uipanel('Parent',F,'BorderType','none');
p.Title = 'Plot : European Option Prices vs Parameters';
p.TitlePosition = 'centertop';
p.FontSize = 12;
p.FontWeight = 'bold';

color = ['y' 'm' 'c' 'r' 'g' 'b'];

subplot(2,2,1, 'Parent',p);
for T = 0.5:0.2:1.5
       c = color(int8((T-0.5)/0.2 + 1));
    fmesh(@(t,s) bsmopt( s, t, T, K, r, vol, 1 ), [0 0.99999*T 0 2], 'EdgeColor', c);
    hold on;
end
hold off;
xlabel('t');
ylabel('s');
zlabel('C(t,s)');
legend({'T = 0.5' 'T = 0.7' 'T = 0.9' 'T = 1.1' 'T = 1.3' 'T = 1.5'});
title('C(t,s) vs. T');

T = 1;

subplot(2,2,2, 'Parent',p);
for K = 0.5:0.2:1.5
```

```
      c = color(int8((K-0.5)/0.2 + 1));
    fmesh(@(t,s) bsmopt( s, t, T, K, r, vol, 1 ), [0 0.99999 0 2], 'EdgeColor', c);
    hold on;
end
hold off;
xlabel('t');
ylabel('s');
zlabel('C(t,s)');
legend({'K = 0.5' 'K = 0.7' 'K = 0.9' 'K = 1.1' 'K = 1.3' 'K = 1.5'});
title('C(t,s) vs. K');

K = 1;

subplot(2,2,3, 'Parent',p);
for r = 0.0:0.02:0.1
      c = color(int8((r-0.0)/0.02 + 1));
    fmesh(@(t,s) bsmopt( s, t, T, K, r, vol, 1 ), [0 0.99999 0 2], 'EdgeColor', c);
    hold on;
end
hold off;
xlabel('t');
ylabel('s');
zlabel('C(t,s)');
legend({'r = 0.0' 'r = 0.02' 'r = 0.04' 'r = 0.06' 'r = 0.08' 'r = 0.1'});
title('C(t,s) vs. r');

r = 0.05;

subplot(2,2,4, 'Parent',p);
for vol = 0.0:0.2:1.0
      c = color(int8((vol-0.0)/0.2 + 1));
    fmesh(@(t,s) bsmopt( s, t, T, K, r, vol, 1 ), [0 0.99999 0 2], 'EdgeColor', c);
    hold on;
end
hold off;
xlabel('t');
ylabel('s');
zlabel('C(t,s)');
legend({'vol = 0.0' 'vol = 0.2' 'vol = 0.4' 'vol = 0.6' 'vol = 0.8' 'vol = 1.0'});
title('C(t,s) vs. vol');

vol = 0.6;

saveas(F,'4.jpg');
clear('F');
```