THE UNIVERSITY OF TEXAS AT ARLINGTON, TEXAS
DEPARTMENT OF ELECTRICAL ENGINEERING

EE 5321 - 001

OPTIMAL CONTROL

HW # 2
ASSIGNMENT

by

SOUTRIK MAITI

1001569883

Presented to

Prof. Michael Niestroy

Feb 8,2018

Problem 1:

a) Computing the first derivative of the given function:
MATLAB CODE:

```
clc
clear all
close all
%Equation for first problem
syms y x
y=(x^4)-(4*x^3)+(8*x)+1;

dy=diff(y,x); %first derivative of y

cpy=double(solve(dy==0)) %critical points of y
```

Result: The critical points of the function are as follows-

```
cpy =

    1.0000
   -0.7321
    2.7321
```

b) Computing the second derivative of the given function and determining the nature of the critical points:
MATLAB CODE:

```
ysol=(cpy.^4)-(4.*cpy.^3)+(8.*cpy)+1 %value of y at the critical points

ddy=diff(dy,x); %second derivative of y
ynat=(12.*cpy.^2-24.*cpy) %determining the nature of critical points
```

Result:

```
ysol =

    6.0000
   -3.0000
   -3.0000
```

```
ynat =

   -12
    24
    24
```

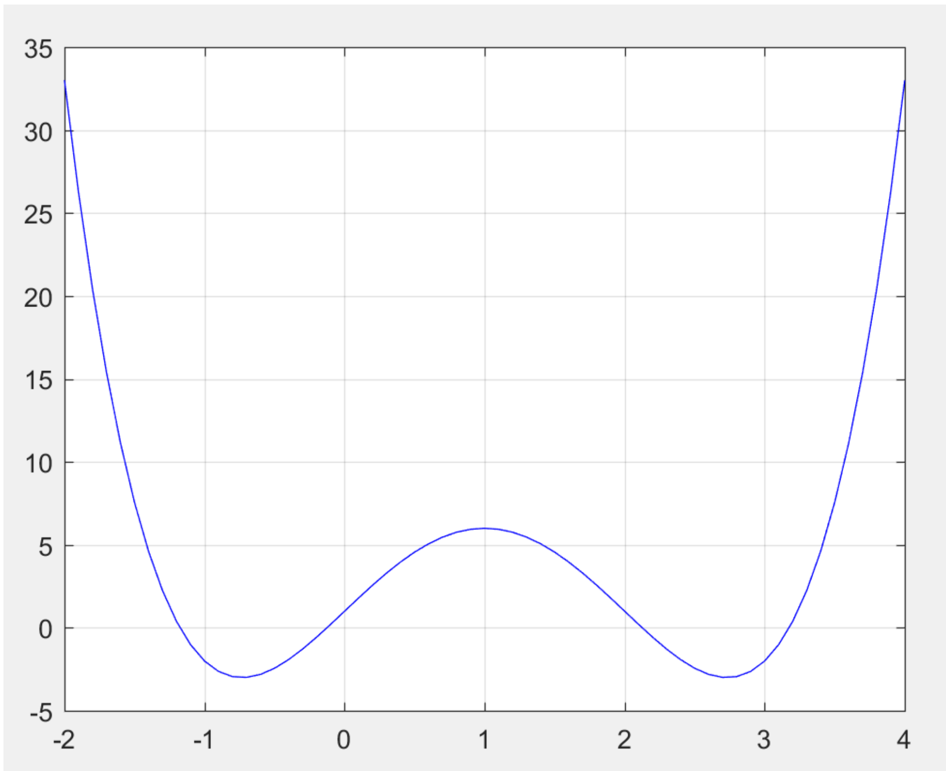
From the above results we can say that the maxima occur at 1 and the minima occur at –0.7321.
The **global minima** from the second derivative is determined to be at -0.7321.
The local maxima occur at 1.

c) Plotting the function:

MATLAB CODE:

```
x = [-2:0.1:4];
y = (x.^4)-(4.*x.^3)+(8.*x)+1;
plot(x,y,'b-');
grid
```

Result:



d) MATLAB CODE:

```
x0 = -1.5;
options = optimoptions('fminunc','Display','iter','Algorithm','quasi-newton');
[xoptimal, optimal_cost] = fminunc('hw2p1cf',x0,options)
```

Cost function:

```
function cost = hw2p1cf(x)

cost = (x^4)-(4*x^3)+(8*x)+1;

end
```

Result:

Iteration	Func-count	f(x)	Step-size	First-order optimality
0	2	7.5625		32.5
1	4	-2.4375	0.0307692	4.5
2	6	-2.86285	1	2.4
3	8	-2.98985	1	0.71
4	10	-2.9999	1	0.0698
5	12	-3	1	0.00172
6	14	-3	1	4.35e-06

```
Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>


xoptimal =

   -0.7321


optimal_cost =

   -3.0000
```

The number of iterations required to calculate is 5 and the **global minima** is -0.732 from the previous part of the problem and it matches with the results from this code. Therefore, our algorithm worked correctly, and the optimal cost of the function is -3.0000.

e) When x0=1.1

MATLAB CODE:

```
x0 = 1.1;
options = optimoptions('fminunc','Display','iter','Algorithm','quasi-newton');
[xoptimal, optimal_cost] = fminunc('hw2p1cf',x0,options)
```

Result: The number of iterations decrease as the value of x is selected near the local maxima. The same code with x=1.1 took 3 iterations.

Iteration	Func-count	f(x)	Step-size	First-order optimality
0	2	5.9401		1.2
1	8	-2.99971	1.36049	0.117
2	14	-3	0.0277423	1.57e-06

```
Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>


xoptimal =

    2.7321


optimal_cost =

   -3.0000
```

f) When x=1.0, using the same code as above
Result: The number of iterations is just 1 as the value of x is exactly the critical point which is the local maxima.

Iteration	Func-count	f(x)	Step-size	First-order optimality
0	2	6		1.19e-07

```
Initial point is a local minimum.

Optimization completed because the size of the gradient at the initial point
is less than the default value of the optimality tolerance.

<stopping criteria details>


xoptimal =

     1


optimal_cost =

     6
```

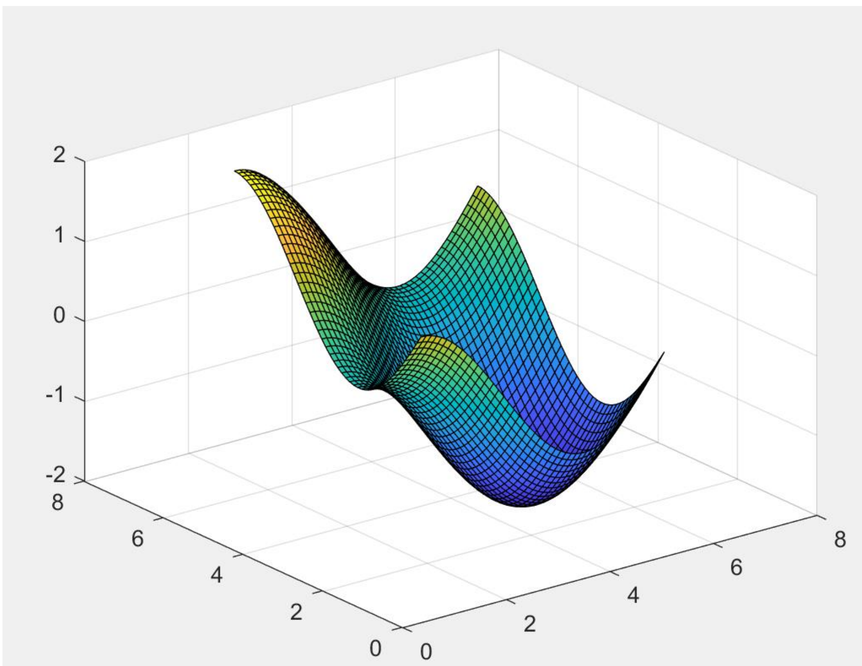
Problem 2:

a) MATLAB CODE:

```
x=[pi/2 :0.1: 2*pi];
y=[pi/2 :0.1: 2*pi];

for i=1:length(x)
    for j=1:length(y)
        z(j,i)=sin(x(i))+cos(y(j));
    end
end
surf(x,y,z);
grid on;
```

Result:



b) MATLAB CODE:

```
x0=[5.5,5.5]';
options = optimoptions('fminunc','Display','iter','Algorithm','quasi-newton');
[xopt,optimal_cost] = fminunc('hw2p2cf',x0,options)
```

Result:

```
                                                        First-order
 Iteration  Func-count       f(x)        Step-size      optimality
     0           3        0.00312945                        0.709
     1           6         -0.914907             1          0.997
     2          12          -1.54316      0.134752          0.769
     3          15           -1.7267             1          0.684
     4          18          -1.87794             1           0.45
     5          21          -1.97915             1          0.177
     6          24          -1.99875             1         0.0499
     7          27                -2             1         0.0023
     8          30                -2             1       0.000101
     9          33                -2             1       8.63e-07

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>


xopt =

    4.7124
    3.1416


optimal_cost =

   -2.0000
```

Problem 3:

a) MATLAB CODE:

Cost function:

```
function cost = hw2p3cf(x)
    cost = sin(x(1))+ cos(x(2));
end
```

Constraint function:

```
function [cineq, ceq] = hw2p3c(x)
    ceq = [];
    cineq = [];
    if x(2) < 4
        cineq = -1*(x(2)-4);
    else
        cineq = 0;
    end
end
```

```
x=[pi/2 :0.1: 2*pi];
y=[pi/2 :0.1: 2*pi];

for i=1:length(x)
    for j=1:length(y)
        z(j,i)=sin(x(i))+cos(y(j));
    end
end
x0=[5.5,5.5;2,2];
for i=1:2

    init=x0(i,:);
    % Finding minimum(unconstrained)
options = optimoptions('fminunc','Display','iter','Algorithm','quasi-newton');
[xoptimal, optimal_cost] = fminunc(@hw2p3cf,init,options)


% Finding minimum (constrained)
options = optimoptions('fmincon','Display','iter','Algorithm','sqp');
[xoptimalc, optimal_costc] = fmincon(@hw2p3cf,init,[],[],[],[],[],[],@hw2p3c,options)


% Contour plot
figure;
contour(x,y,z);
title('Contour plot for intial point (5.5,5.5)');
hold on;
plot3(xoptimal(1),xoptimal(2),optimal_cost,'b*')
plot3(xoptimalc(1),xoptimalc(2),optimal_costc,'r*')
grid on;
%plotting line y=4
line([x(1) x(end)],[4 4],'Color','green','linestyle','--','lineWidth',1.5)
legend('Contour plot','Unconstrained','Constrained','Inequality');
hold off
end
```

Results:

When x0=[5.5,5.5]

The unconstrained minimum

Iteration	Func-count	f(x)	Step-size	First-order optimality
0	3	0.00312945		0.709
1	6	-0.914907	1	0.997
2	12	-1.54316	0.134752	0.769
3	15	-1.7267	1	0.684
4	18	-1.87794	1	0.45
5	21	-1.97915	1	0.177
6	24	-1.99875	1	0.0499
7	27	-2	1	0.0023
8	30	-2	1	0.000101
9	33	-2	1	8.63e-07

Local minimum found.

Optimization completed because the size of the gradient is less than the default value of the optimality tolerance.

<stopping criteria details>

```
xoptimal =

    4.7124    3.1416


optimal_cost =

   -2.0000
```

The constrained minimum

Iter	Func-count	Fval	Feasibility	Step Length	Norm of step	First-order optimality
0	3	3.129449e-03	0.000e+00	1.000e+00	0.000e+00	7.087e-01
1	6	-9.149071e-01	0.000e+00	1.000e+00	1.000e+00	9.966e-01
2	15	-1.689978e+00	2.974e-01	1.176e-01	1.269e+00	5.375e-01
3	18	-1.623247e+00	0.000e+00	1.000e+00	4.371e-01	7.568e-01
4	21	-1.823687e+00	2.615e-01	1.000e+00	3.092e-01	5.621e-01
5	24	-1.652609e+00	0.000e+00	1.000e+00	2.640e-01	7.568e-01
6	55	-1.652609e+00	0.000e+00	1.578e-05	3.236e-06	7.568e-01

Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

```
xoptimalc =

    4.6669    4.0000


optimal_costc =

   -1.6526
```

When x0=[2,2]

The unconstrained minima

Iteration	Func-count	f(x)	Step-size	First-order optimality
0	3	0.493151		0.909
1	6	-0.309672	1	0.748
2	9	-1.02945	1	0.994
3	12	-1.53793	1	0.684
4	15	-1.68507	1	0.727
5	18	-1.99817	1	0.057
6	21	-1.99988	1	0.0156
7	24	-2	1	0.000325
8	27	-2	1	4.27e-05
9	30	-2	1	3.16e-08

Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.

<stopping criteria details>

```
xoptimal =

    4.7124    3.1416


optimal_cost =

   -2.0000
```

The constrained minima

Iter	Func-count	Fval	Feasibility	Step Length	Norm of step	First-order optimality
0	3	4.931506e-01	2.000e+00	1.000e+00	0.000e+00	9.093e-01
1	6	9.825438e-03	0.000e+00	1.000e+00	2.043e+00	7.568e-01
2	9	-7.813812e-01	3.604e-01	1.000e+00	7.242e-01	9.953e-01
3	13	-1.271350e+00	1.081e-01	7.000e-01	2.681e+00	8.418e-01
4	16	-1.629200e+00	0.000e+00	1.000e+00	1.227e+00	7.568e-01
5	50	-1.629200e+00	0.000e+00	5.412e-06	3.610e-06	7.568e-01

Local minimum possible. Constraints satisfied.

fmincon stopped because the size of the current step is less than
the default value of the step size tolerance and constraints are
satisfied to within the default value of the constraint tolerance.

<stopping criteria details>

```
xoptimalc =

    4.4908    4.0000


optimal_costc =

   -1.6292
```

Contour Plot: