Experiment 7

Constrained Optimization using the method of Lagrange's Multipliers

I. Aim

Finding and Visualizing Maxima-Minima of a function of several variables under a given constraint using Lagrange’s Multipliers

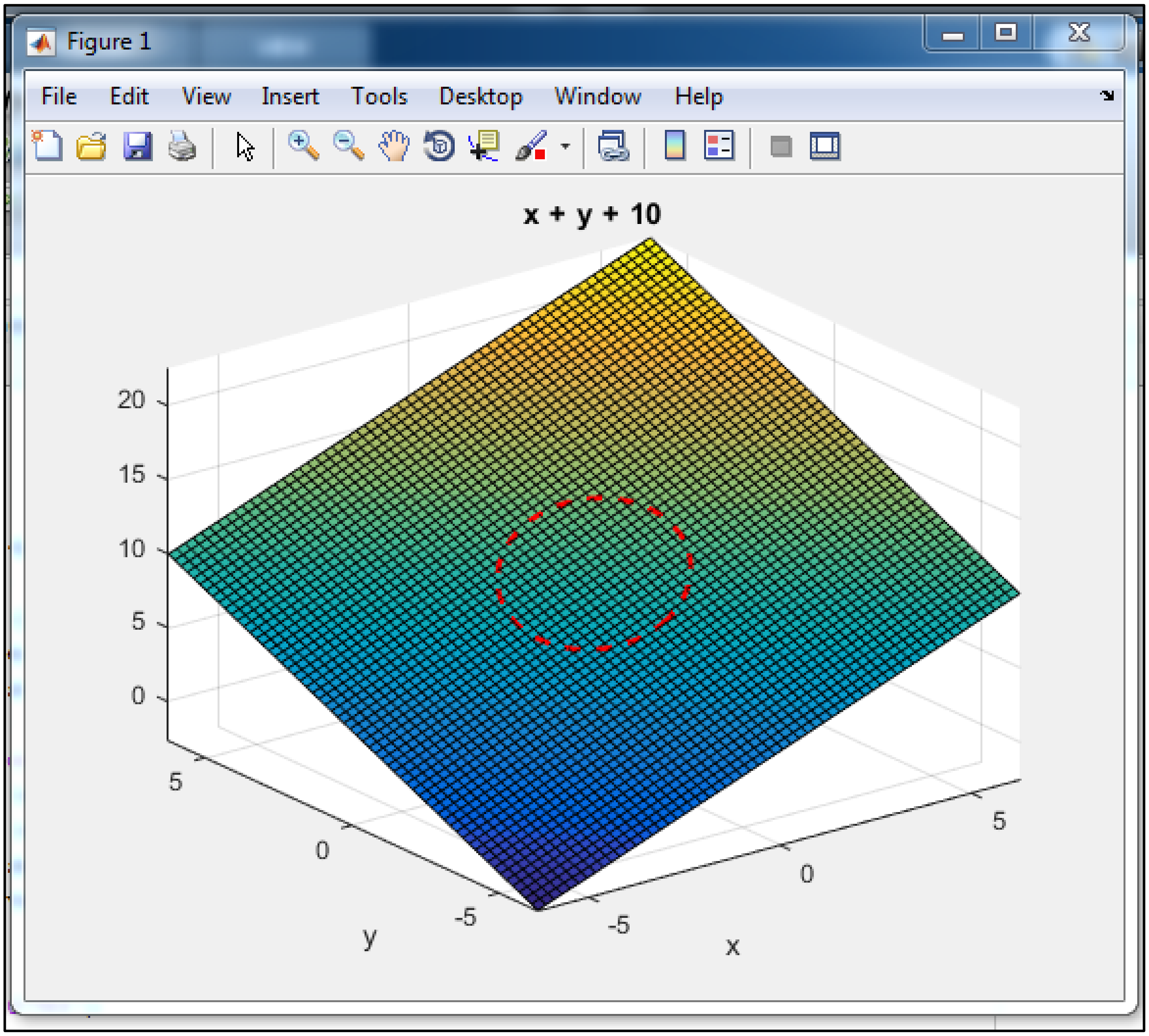
II. Mathematical Background

In many real-life (or practical) applications, we are required to find the maxima and minima of a function of more than 1 variable, where the variables are connected by some relation or condition known as constraint. Here, f(x,y,z) is a function of three variables where x,y,z are related by a known constraint g(x,y,z) = c (say).

We know, gradient(f)=lambda*gradient(g) and hence, gradient(f-lambda*(g)) = 0. Therefore, we will solve gradient(f-lambda*(g)) = 0 for lamda, x and y.

III. MATLAB Code for constrained Maxima Minima (2 Variables)

```
clc
close all
syms x y L
f = x+y+10;
g = x^2+y^2-4;
j = f-(L*g);
j = jacobian(j,[x,y]);
[L,X,Y]=solve([j,g]);
z=double(subs(f,{x,y},{X,Y}));
h=ezplot(g);
val=get(h,'contourMatrix');
xval=val(1,2:end);
yval=val(2,2:end);
zval=double(subs(f,{x,y},{xval,yval}));
plot3(xval,yval,zval,'r--','LineWidth',2);
hold on
ezplot(g)
ezsurf(f)
```



IV. MATLAB Code for constrained Maxima Minima (3 Variables)

Editor - D:\VIT\MATLAB\new\matlab\lag_3.m

lag_3.m

```
clc
clc
clearvars
syms x y z L
f = input('Enter the function f(x,y,z): ');
g = input('Enter the constraint function g(x,y,z): ');
F = f + L*g;
gradF = jacobian(F,[x,y,z]);
[L,x1,y1,z1] = solve(g,gradF(1),gradF(2),gradF(3));
Z = [x1 y1 z1];
disp('[x y z]=')
disp(Z)
```

Command Window

```
Enter the function f(x,y,z):  x^2+y^2+z^2
Enter the constraint function g(x,y,z): 3*x^2+4*x*y+6*y^2-140
[x y z]=
[          -2,         -4, 0]
[           2,          4, 0]
[ -2*14^(1/2),  14^(1/2), 0]
[  2*14^(1/2), -14^(1/2), 0]

>>
```

V. Question – Answers

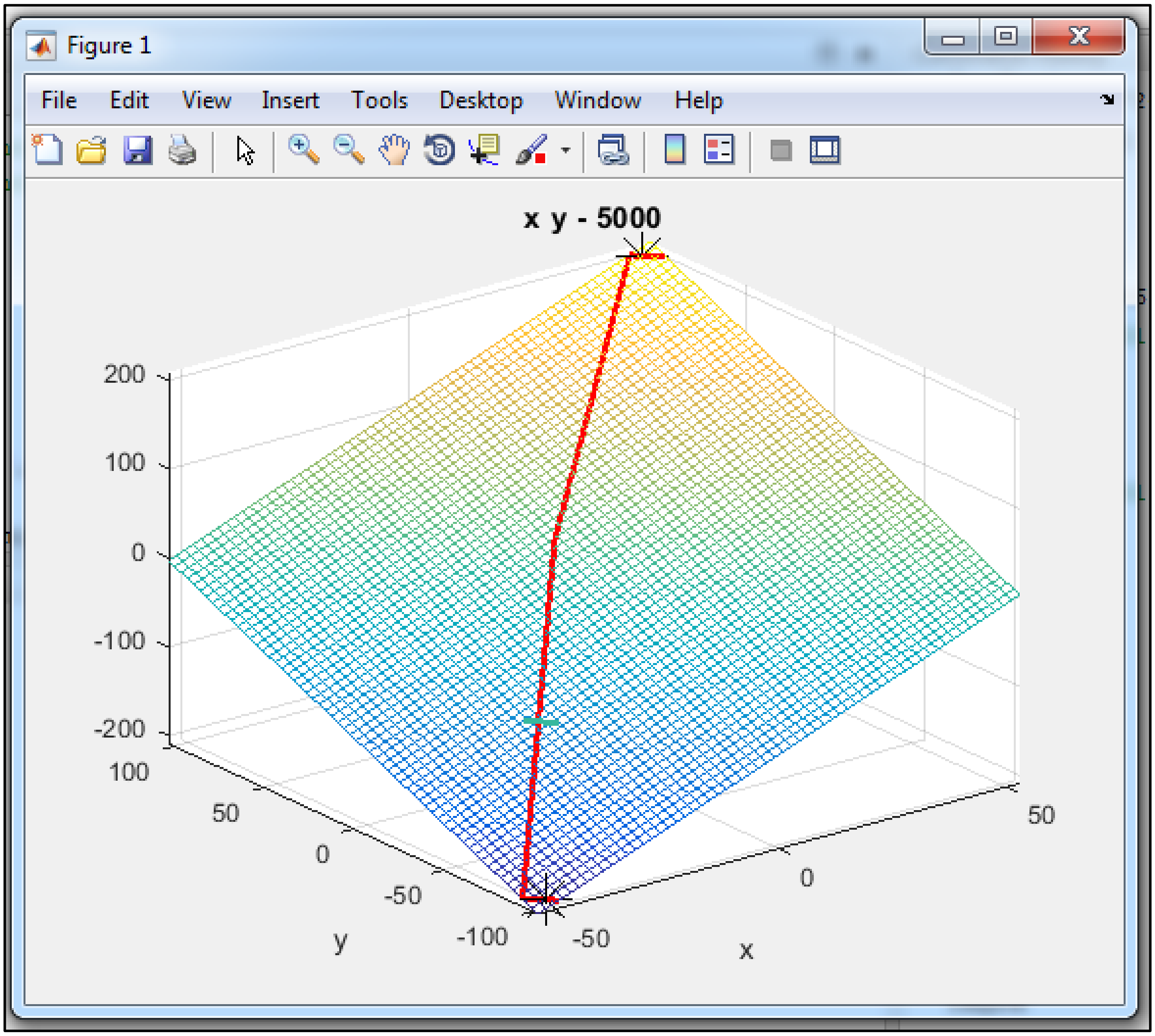
Q1 Answer

```
clc
clear all
syms x y L
f = input('Enter the function z = ');
g = input('Enter the constraint g(x,y): ');
F = f + L*g;
gra = jacobian(F,[x,y]);
[L,x1,y1] = solve(g,gra(1),gra(2),'Real',true); % Solving only for Real x and y
x1 = double(x1); y1 = double(y1);
xmx = max(x1);
xmn = min(x1); % Finding max and min of x-coordinates for plot range
ymx = max(y1);
ymn = min(y1); % Finding max and min of y-coordinates for plot range
range = [xmn-3 xmx+3 ymn-3 ymx+3]; % Setting plot range
ezmesh(f,range);
hold on;
```

```
grid on;
h = ezplot(g,range);
set(h,'LineWidth',2);
M = get(h,'contourMatrix');
xdt = M(1,2:end); % Avoiding first x-data point
ydt = M(2,2:end); % Avoiding first y-data point
zdt = double(subs(f,{x,y},{xdt,ydt}));
plot3(xdt,ydt,zdt,'-r','LineWidth',2);
axis(range);
a=[];
 for i = 1:numel(x1)
    fxy(i) = subs(f,[x,y],[x1(i),y1(i)]);
    a(end+1)=fxy(i);
    plot3(x1(i),y1(i),fxy(i),'*k','MarkerSize',20);
 end
 disp(['Minimum amount of Fencing =',num2str(min(a))]);
```

Command Window

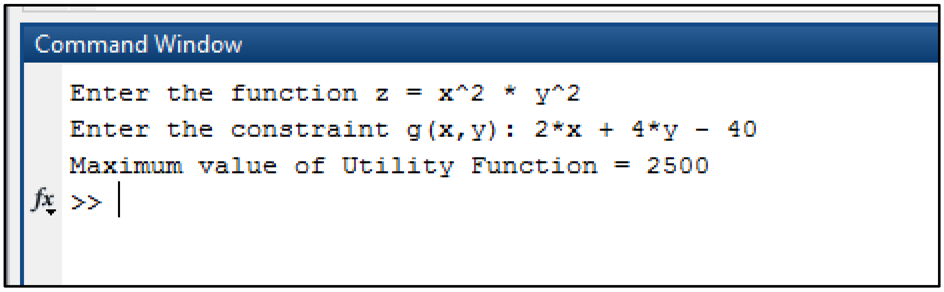
```
Enter the function z = 2*x + y
Enter the constraint g(x,y): x*y - 5000
Minimum amount of Fencing =-200
>>
```



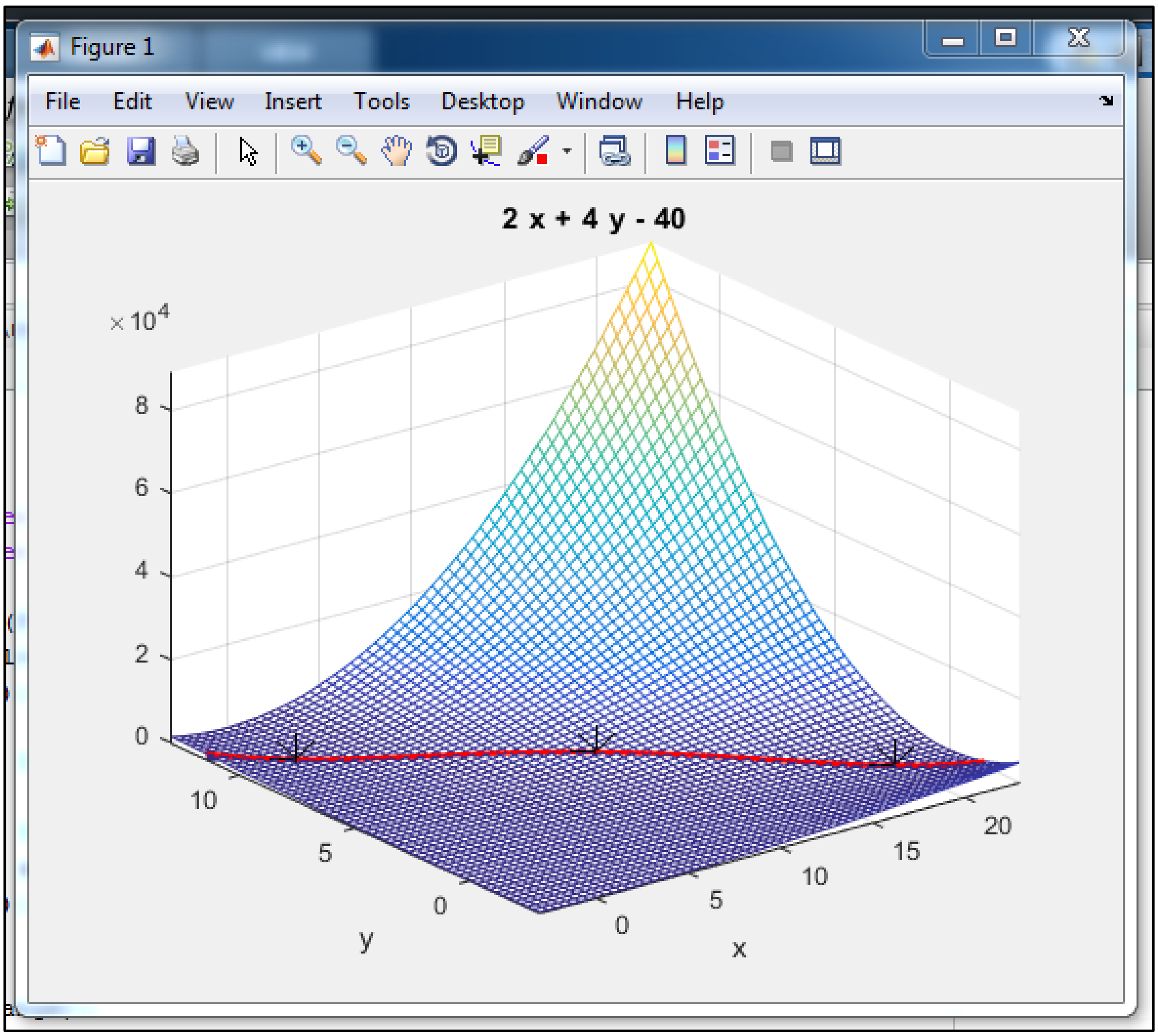
Q2 Answer

```
clc
clear all
syms x y L
f = input('Enter the function z = ');
g = input('Enter the constraint g(x,y): ');
F = f + L*g;
gra = jacobian(F,[x,y]);
[L,x1,y1] = solve(g,gra(1),gra(2),'Real',true); % Solving only for Real x and y
x1 = double(x1); y1 = double(y1);
xmx = max(x1);
xmn = min(x1); % Finding max and min of x-coordinates for plot range
ymx = max(y1);
ymn = min(y1); % Finding max and min of y-coordinates for plot range
range = [xmn-3 xmx+3 ymn-3 ymx+3]; % Setting plot range
ezmesh(f,range);
hold on;
grid on;
h = ezplot(g,range);
set(h,'LineWidth',2);
M = get(h,'contourMatrix');
xdt = M(1,2:end); % Avoiding first x-data point
ydt = M(2,2:end); % Avoiding first y-data point
zdt = double(subs(f,{x,y},{xdt,ydt}));
plot3(xdt,ydt,zdt,'-r','LineWidth',2);
axis(range);
```

```
a=[];
 for i = 1:numel(x1)
    fxy(i) = subs(f,[x,y],[x1(i),y1(i)]);
    a(end+1)=fxy(i);
    plot3(x1(i),y1(i),fxy(i),'*k','MarkerSize',20);
 end
 disp(['Maximum value of Utility Function = ',num2str(min(a))]);
```



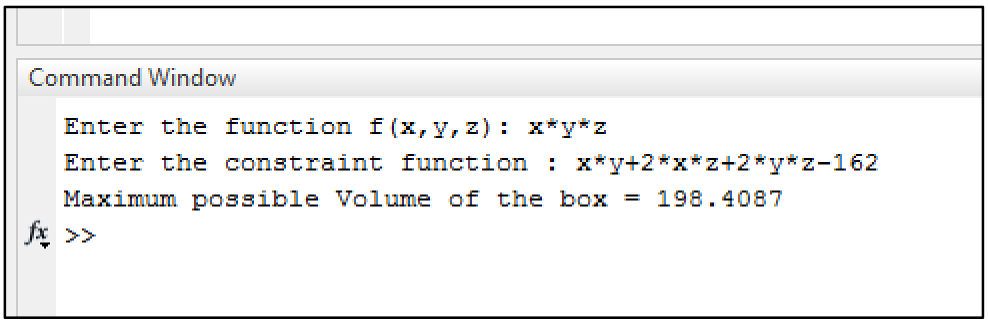
```
Command Window
Enter the function z = x^2 * y^2
Enter the constraint g(x,y): 2*x + 4*y - 40
Maximum value of Utility Function = 2500
>>
```



Q3 Answer

```
clc
```

```
clear all
syms x y z L
f = input('Enter the function f(x,y,z): ');
g = input('Enter the constraint function : ');
F = f + L*g;
gra = jacobian(F,[x,y,z]);
[L,x1,y1,z1] = solve(g,gra(1),gra(2),gra(3));
a=[];
for i = 1:numel(x1)
    fxy(i) = subs(f,[x,y,z],[x1(i),y1(i),z1(i)]);
    a(end+1)=fxy(i);
end
display(['Maximum possible Volume of the box = ',num2str(max(a))])
```



Command Window

```
Enter the function f(x,y,z): x*y*z
Enter the constraint function : x*y+2*x*z+2*y*z-162
Maximum possible Volume of the box = 198.4087
>>
```

Q4 Answer

```
clc
clear all
syms x y L
f = input('Enter the function z = ');
g = input('Enter the constraint function g(x,y): ');
F = f + L*g;
gra = jacobian(F,[x,y]);
[L,x1,y1] = solve(g,gra(1),gra(2),'Real',true);
j=0;
a=[];
 for i = 1:numel(x1)
fxy(i) = subs(f,[x,y],[x1(i),y1(i)]);
 a=[a;fxy(i) x1(i) y1(i)];
 j=j+1;
 end
 max=0;
 max_index=0;
 for i=1:j
     if a(i)>max
         max=a(i);
         max_index=i;
         sx=double(a(1+numel(x1)));
         sy=double(a(1+2*numel(x1)));
     end
 end
 disp('For Maximum Sales: ');
 disp(['Amount needed to be spent for Development (in $) = ',num2str(sx)]);
 disp(['Amount needed to be spent for Promotion (in $) = ',num2str(sy)]);
```

```
    disp(['Amount needed to be spent for Development (in $) = ',num2str(sx)]);
    disp(['Amount needed to be spent for Promotion (in $) = ',num2str(sy)]);
```

Command Window

```
Enter the function z = 20 * (x^1.5) * y
Enter the constraint function g(x,y): x+y-60000
For Maximum Sales:
Amount needed to be spent for Development (in $) = 36000
Amount needed to be spent for Promotion (in $) = 24000
>>
```

--X--