**EAST WEST UNIVERSITY**

**LAB – 5**

**Steepest Ascent Method**

**Course Code: ICE470**

**Course Title: Applied Numerical Methods**

**Section – 01**

**Submitted To:**

Muhammad Suhail Najeeb  
Lecturer

Dept. of ECE

East West University

**Submitted By:**

MD Imran Ahmed

ID: 2015-2-50-018

**Date of Submission**: 15-07-2019

**Objective**: Use Steepest Ascent Method to find the maximum of

F(x,y) = 2xy + 2x –x2 -2y2

Visualize the function using MATLAB to find the initial guess:

**Code:**

clc;

clear all;

close all;

%declare function

func = @(x,y) 2.\*x.\*y + 2.\*x - x.^2 - 2.\*y.^2;

x = linspace(-5,5,100); %create a linspace in x direction

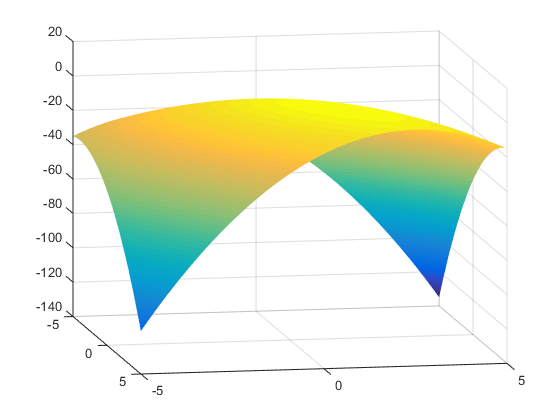
y = linspace(-5,5,100); %create a linspace in y direction

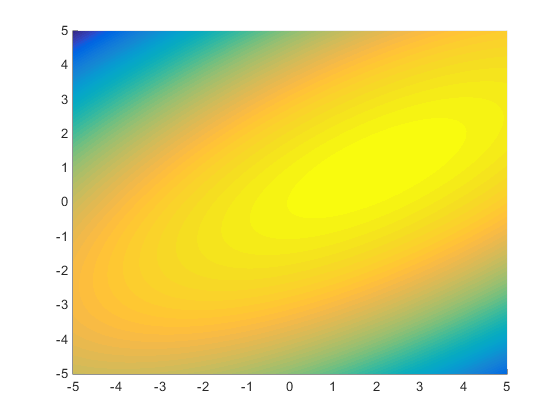
[X,Y] = meshgrid(y,x); %now create a plane or mesh using those x,y values,that is actually a matrix

F = func(X,Y); %now pass that [X,Y] matrix into our f(x,y) function

surf(x,y,F');

shading interp;





**Steepest ascent algorithm on the function**

F(x,y) = 2xy + 2x –x2 -2y2

**Code:**

clc;

clear all;

close all;

%declare function

func = @(x,y) 2.\*x.\*y + 2.\*x - x.^2 - 2.\*y.^2;

%dfdx = @(x,y) 2\*y +2 - 2\*x;

%dfdy = @(x,y) 2\*x - 4\*y;

%initial guess

x0 = 1;

y0 = -1;

%algorithm parameters

dx = 0.001;

dy = 0.001;

alpha = 0.1;

tol = 1e-3;

g = [inf,inf];

while norm(g) > tol

% norm(g) = sqrt(gx^2 +gy^2)

%clculate gradients

f1 = func(x0-dx/2,y0);

f2 = func(x0+dx/2,y0);

gx = (f2-f1)/dx;

f1 = func(x0,y0-dy/2);

f2 = func(x0,y0+dy/2);

gy = (f2-f1)/dy;

g = [gx;gy];

%update position of guess

x0 = x0 + alpha\*gx;

y0 = y0 + alpha\*gy;

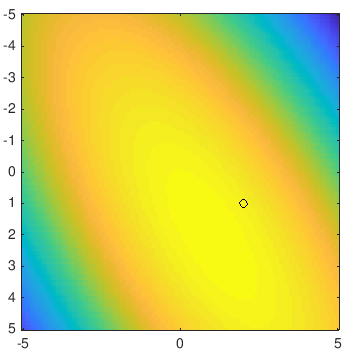
end

%Report the answer

[x0, y0]

**Output:**

ans = 1.9989 0.9993



**Discussion:**

Alpha is the step size towards the gradient, it takes us local maximum to finally global maximum. Choosing a precise value for alpha is must while choosing too much small or too much bigger value can cause divergence.

For example:

Keeping the initial guess fixed that is x0 = 1, y0 = -1

For alpha = 0.01 it takes 2 minute and 19 sec to reach the global maximum for our optimizer and takes to the maxima where x = 1.9989 and y = 0.9993.

But for alpha = 0.001, I had to wait 11 minute 13 sec without getting any point near to the maxima.

For alpha = 0.1 it takes only 13 sec to terminate optimization but solution is slightly different from what we got for the alpha = 0.01.

Now,

What if we change the value of our initial guess? We know that to choose the initial value we should plot the function and choose a value near to the maximum of that visualized plot. So that we can easily reach the global maxima without taking much time or terminating too many iteration.

For our above function it we choose x0 = 1, y0 = -1.

But if we change these value to x0 = 5, y0 = -3 it took 2 minute and 22 second to reach a maxima which was (2.0011, 1.0007)!! But this is not what we were expecting as our optimum value. Which is far away from what we visualized from that contour plot.