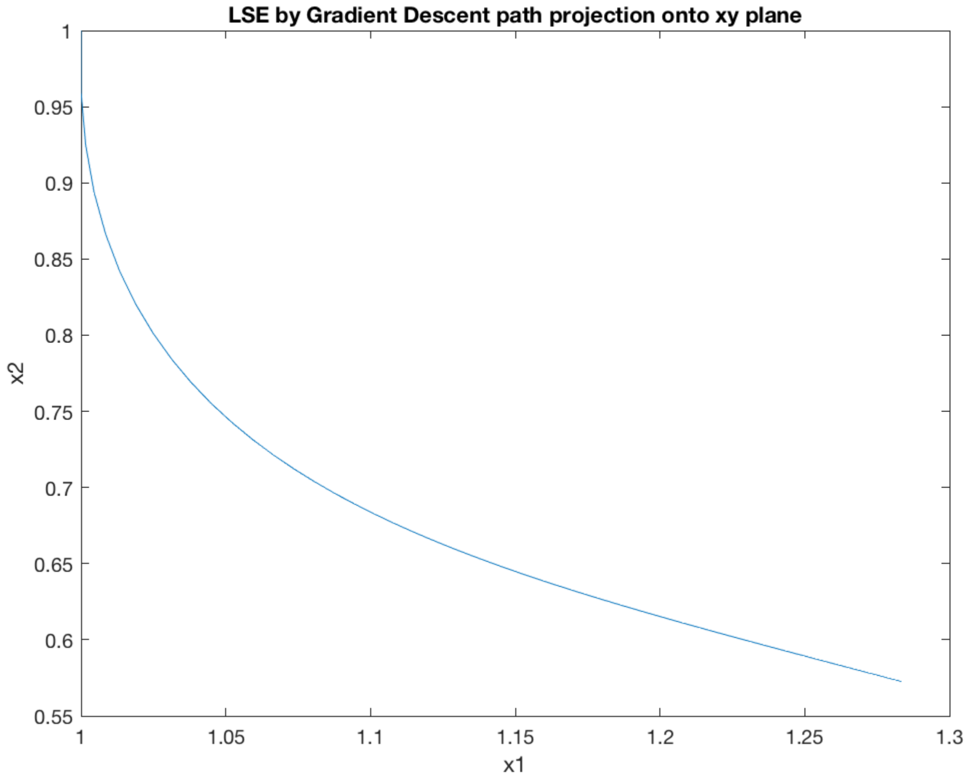
```
%GI07 Homework 1 Question 1 Part 2
%Klaudia Ludwisiak

close all;
clear all;

%===========Least Squares Error estimation by
 Gradient=Descent============

%======PART=A ==== function written called myLSE which computes the
 least
%squares solution by gradient descent. Code in separate .m file

% function takes inputs defined below and computes Least Square error
% solution by gradient descent. Solution is that combination of X's
% which minimises sum of error squared, where error is defined as:
 e=Ax-b
%
%------INPUTS-----
%A is coefficients matrix of size m*n
%b is the corresponding solutions vector of size m*1
%guess is a vector of initial guess of values of x (size m by 1)
%step is the step size of the gradient descent
%tol is the tolerance and dictates when function will stop iterating
%i.e. what diff. between predicted and observed values are we ok with?

%=======PART=B ==== test myLSE.m by solving system of equatuons
 provided.
% The system can be represented in matrix form as:
A=[1 -1; 1 1; 1 2];
b=[1;1;3];
%setting the remaining inputs for myLSE:
guess=[1;1];
tol=0.01;%needed reduction to converge at solution to 2 decimal points
step=0.01;

[x,path]=myLSE(A,b,guess,step,tol);


%Additional test
Aa=[1 -1 2; 1 2 1; 1 2 2];
bb=[1;1;3];
%setting the remaining inputs for myLSE:
guess2=[-2;0;1];
step2=0.05;
tol2=0.05; %needs decreasing so that solution converges

[x2,path2]=myLSE(Aa,bb,guess2,step2,tol2);
%solution produced is equivalent to matlab Aa\bb therefore correct
 code

%=======PART=C==== PLOT solution
```

```
figure;
plot(path(:,1),path(:,2))
xlabel('x1');
ylabel('x2');
title('LSE by Gradient Descent path projection onto xy plane');
```



Published with MATLAB® R2016b