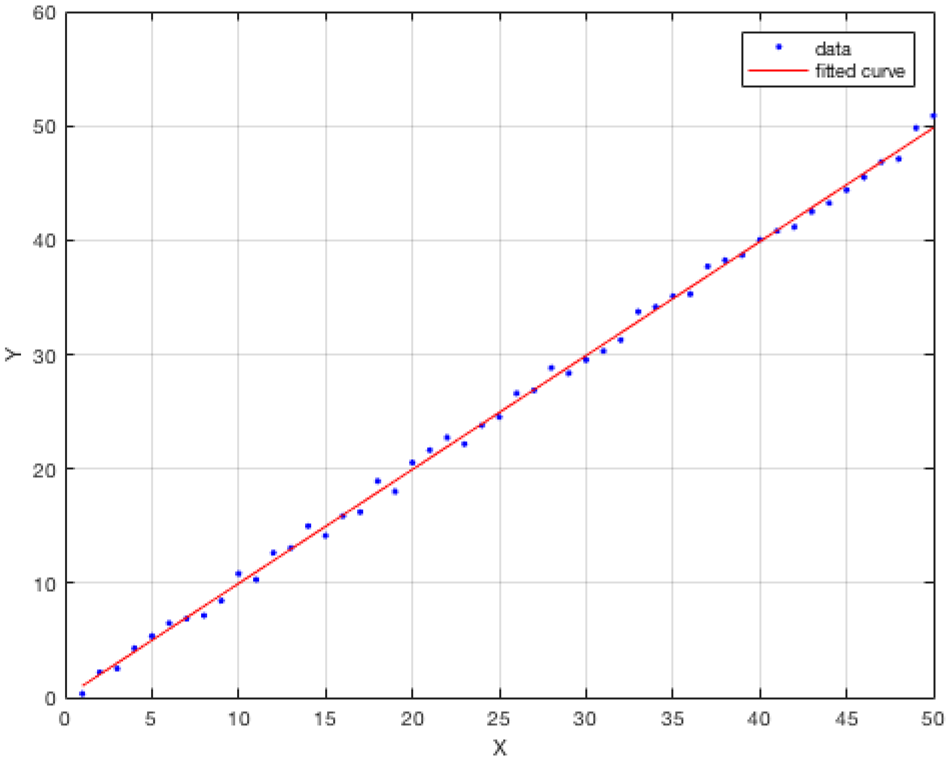
```
% Neural Network ECE 559 - Fall 2018
% Homework 3
% Montagna Marco
% exercise 3

clear all
close all
clc

xi = zeros(1, 50);
for i=1:1:50
    xi(1,i)=i;
end

ui = rand(1,50)*2-1;
yi = zeros(1, 50);
for i=1:1:50
    yi(1,i)=i+ui(1,i);
end

t=fittype('w0 + w1*x','coeff',{'w0','w1'});
y=fit(xi',yi',t)
plot(y,xi',yi');
grid on;
xlabel('X');
ylabel('Y');
```

```
Warning: Start point not provided, choosing random start point.

y =

     General model:
     y(x) = w0 + w1*x
     Coefficients (with 95% confidence bounds):
       w0 =     0.01756  (-0.3428, 0.3779)
       w1 =      0.9962  (0.9839, 1.009)
```

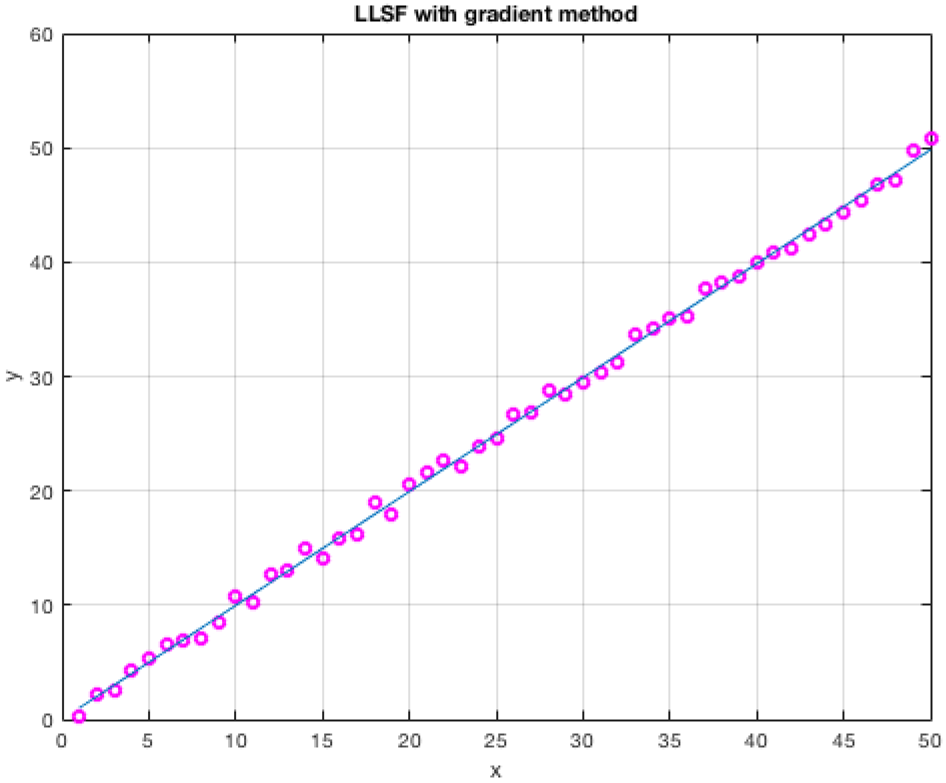


gradient method

```
eta = 0.00001;
% initial point
w_grad = [0.1 0.1]
i_grad = 1; % number of iterations
df_dw = [0,0];
    %update grad
    for k=1:50
        df_dw(1) = df_dw(1) + (yi(k)-(w_grad(1)+w_grad(2)*xi(k)));
        df_dw(2) = df_dw(2) + (yi(k)-
(w_grad(1)+w_grad(2)*xi(k)))*xi(k);
    end
grad = -2.*df_dw;


% gradient descend algorithm
while (abs(grad) > 0)
    % every time that there is another loop grad has to be computed
 again
    df_dw = [0,0];
    % update points
    w_grad = w_grad-eta*grad';
    %update grad
    for k=1:50
```

```
        df_dw(1) = df_dw(1) + (yi(k)-(w_grad(1)+w_grad(2)*xi(k)));
        df_dw(2) = df_dw(2) + (yi(k)-
(w_grad(1)+w_grad(2)*xi(k)))*xi(k);
    end
    grad = -2.*df_dw;

end

figure(2);
plot(xi, yi, 'mo')
hold on
grid on
% plot the line
final_y = w_grad(1)+w_grad(2)*xi;
plot(xi, final_y);
xlabel('x')
ylabel('y')
title('LLSF with gradient method')
disp('Final values of w0 and w1: ')
disp(num2str(w_grad(1)));
disp('    ');
disp(num2str(w_grad(2)));
```

```
w_grad =

    0.1000    0.1000

Final values of w0 and w1:
0.017558

0.99624
```



Published with MATLAB® R2017b