

## ECE133A Hw7:

### Matlab Code:

```
%%
% File: hw7.m
%
% Author: Thomas Kost
%
% Date: 7 March 2021
%
% @brief homework 7 implementations
%

clear,clc,close all;
%% 13.2 a: Least Squares
[u,v] = robappr;
A = [ones(length(u),1),u];
alpha_beta = pinv(A)*v;

%plotting

fig1 = figure();
line = linspace(min(u),max(u),1000);
linear_fit = alpha_beta(1) + alpha_beta(2)*line;
hold on;
scatter(u,v);
plot(line,linear_fit);

%% 13.2 b: Newtons method

% least squares loss function
% sqrt((A*alpha_beta -v)'*(A*alpha_beta-v) + 25)
cutoff_error= 10e-6;
error = inf;
itter =0;
while(error >= cutoff_error)
    h_in = A*alpha_beta - v;
    mag = h_in'*h_in +25;
    grad_coeff = 1/sqrt(mag);
    hess_coeff = -(grad_coeff)^3;
    gradient = A'*grad_coeff *h_in;
    hess_first_term = A'*hess_coeff*(h_in*h_in')*A;
    hess = hess_first_term + eye(length(hess_first_term))*grad_coeff;
    alpha_beta = alpha_beta - pinv(hess)*gradient;
    error = gradient'*gradient;
    itter = itter + 1;
end
linear_fit = alpha_beta(1) + alpha_beta(2)*line;
plot(line,linear_fit);
hold off;
xlabel("U");
```

```

ylabel("V")
legend("data", "ls", "newton");
saveas(fig1, "p13_2_plot.jpg");

%%18.4: exponential fitting
x = [1,2,3,4,5];
y = [5.2,4.5,2.7,2.5,2.1,1.9];

error = inf;
theta = [0;0];
lambda = 10;
while(error >= cutoff_error)
    % calculate jacobian
    J = [1,theta(2)]' * exp(x*theta(2));
    theta_new = theta - pinv(J'*J + lambda*I)*J'*(theta(1)*exp(x'*theta(2)));
    if (theta(1)*exp(x'*theta(2)) < theta_new(1)*exp(x'*theta_new(2)))
        lambda = 0.8*lambda;
        theta = theta_new;
    else
        lambda = 2*lambda;
    end
end

end

## Result:

```

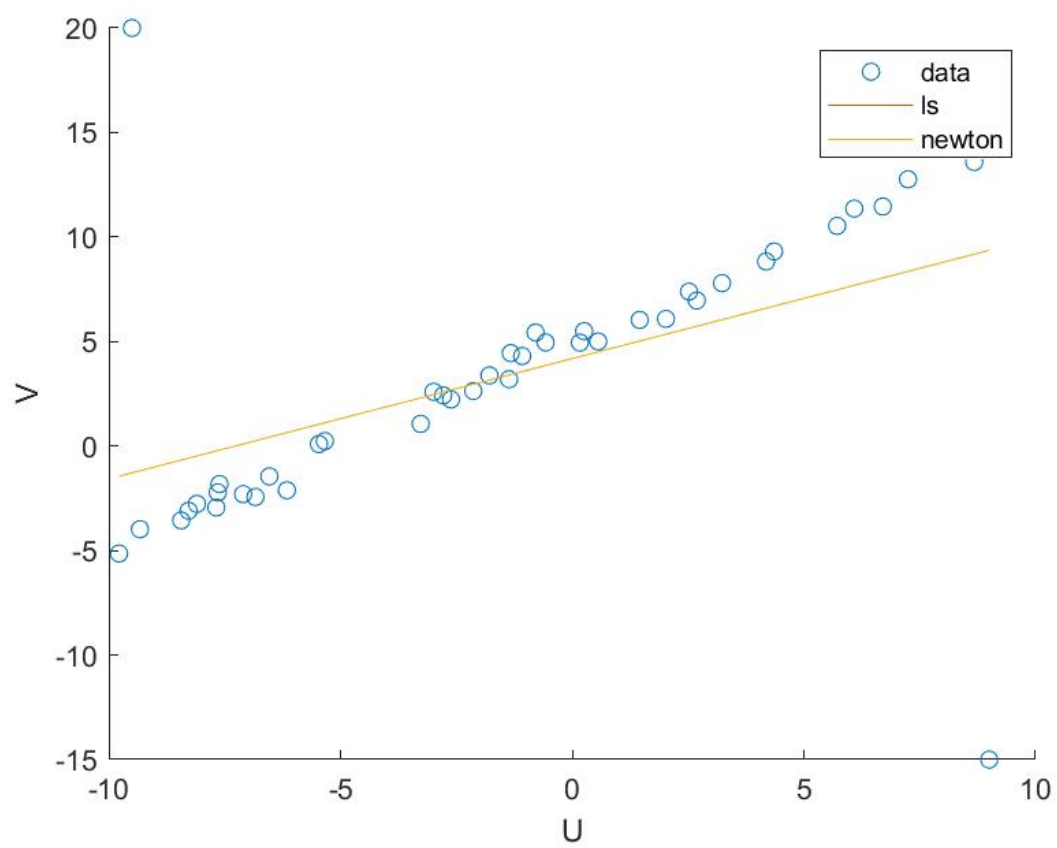


Figure 1: figure