CHBE 553: Assignment 3 (02/04/2019)

Non-linear regression:
Objective: To fit a model equation D = a.t^b for the given data file

Levenberg Marquardt algorithm was developed to solve non-linear least square problems. Least square problems arise in the context of fitting a parameterized function to a set of measured data points by minimizing the sum of the squares of the errors between the data points and the function [1]. Non-linear squares methods iteratively reduce the sum of the square between the function and the measured data points through a sequence of updates to parameter values.
The Levenberg algorithm combines two minimization methods: the gradient descent method and the Gauss-Newton method. This method acts more like gradient decent method when the parameters are far from their optimal value and acts more like Gauss-Newton method when the parameters are close to their optimal value.
Algorithm:

$$y = y(x,a1,a2,\ldots..aM) \quad (1)$$

Considering the data points to follow chi-square:

$$\chi^2 = \sum_{i=1}^{N} \frac{(y - yi)^2}{\sigma i^2} \quad (2)$$

$$\beta = \sum_{i=1}^{N} \frac{((y - yi)^2)}{\sigma i^2} \frac{\partial y}{\partial ak} \quad (3)$$

$$\alpha kl = \sum_{i=1}^{N} \frac{1}{\sigma i^2} \frac{\partial y}{\partial al} \frac{\partial y}{\partial ak} \quad (4)$$

$\alpha kl'$ in the Levenberg algorithm is αkl with diagonal elements set as (1+λ) $\alpha kl, k = l$

$$\delta a = inv(\alpha kl') * \beta \quad (5)$$

$$anew = aold + \delta a \quad (6)$$

- Guess the parameters: a and b and compute $\chi^2(a)$ at the initial guess values.
- A small λ value approximately 10-3 is chosen and the $\alpha kl'$ and β matrices are evaluated.
- δa is evaluated from () and the χ^2 is evaluated at the new value (anew), $\chi^2(a + \delta a)$
- Both the χ^2 values are compared and if
  - $\chi^2(a + \delta a) < \chi^2(a)$, λ is reduced by a factor of 10 and the new χ^2 is evaluated using the new λ
  - $\chi^2(a + \delta a) > \chi^2(a)$, λ is increased by a factor of 10 and the new χ^2 is evaluated using the new λ.
- The iteration is stopped when absolute of $\chi^2(a + \delta a)$ - $\chi^2(a)$ is insignificant(~machine epsilon)

By following this algorithm, using an initial guess of **a = 5** and **b = 1**, the model equation $D = a.t^b$ was fitted to the given data. The final value of a and b after the iterative Levenberg algorithm was found to be **10.0955** and **0.4979** respectively.
The final plot was obtained as follows:

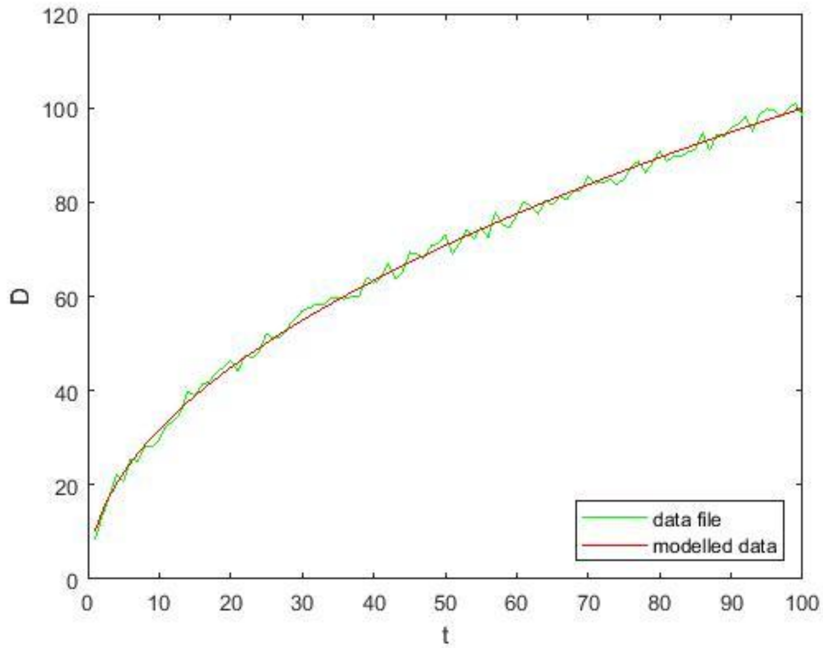

Fig 1: Non-linear regression model

Linear Regression:

For the same data, a linear equation was tested to fit the data.
Model equation: $D = a.t^b$
For linear regression : $\log D = \log(a) + b.\log(t)$
The following plot was obtained using linear regression algorithm:

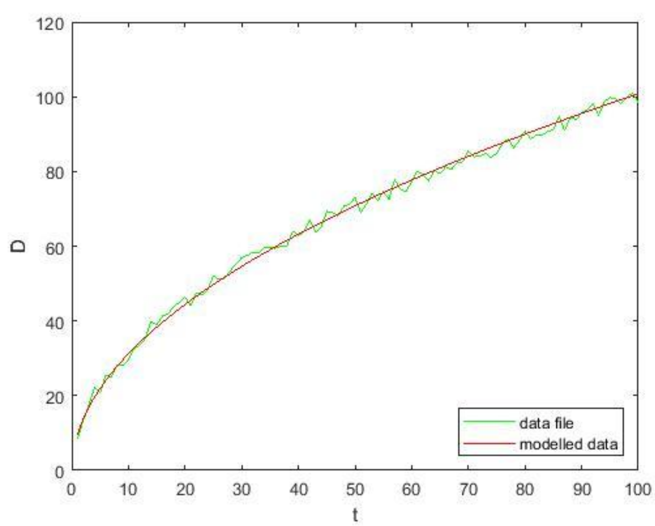

Fig 2: Linear Regression model

From fig2 it can be seen that the linear regression model is an approximate fit for the given data. The parameters a and b were evaluated to be **9.65** and **0.5093** respectively.
MATLAB CODE:
The parameters were evaluated in MATLAB using both the direct minimization and general minimization algorithm.

CONCLUSION:
It can be seen from fig1 and fig2 that non-linear and linear regression, $D = a.t^{b}$ results in a similar model for the given dataset.
Model equation becomes: (Non-linear regression) $D = 10.t^{0.4979}$
(linear regression) $D = 9.65.t^{0.5093}$

MATLAB CODE:
PART 1 : NON-LINEAR REGRESSION

```
%Non-linear regression D = at^b
%import data
data = xlsread ('datafile.xlsx');
t = data(:,1);
D = data(:,2);
%initial guess
a = 5;
b = 1;
lambda = 10^-2;
chisq = [0;0];
%evaluating chi square using the initial parameters
chisq = chi_square(D,t,a,b,lambda);
deltaa = [0;0];
a_new = a;
b_new = b;
%evaluating delta_a using the initial parameters
deltaa = delta_a(D,t,a,b,lambda);
D_new = zeros(size(D));
for i = 1:100
    if chisq(2) < chisq(1) %if condition
    lambda = lambda / 10;
    %evaluate delta at new lambda values
    deltaa = delta_a(D,t,a_new,b_new,lambda);
    a_new = a_new + deltaa(1);
    b_new = b_new + deltaa(2);
    %evaluating chi_square at new parameters a,b and lambda
    chisq = chi_square(D,t,a_new,b_new,lambda)
    else
    lambda = lambda * 10;
    %evaluate delta at new lambda values
    deltaa = delta_a(D,t,a_new,b_new,lambda);
    a_new = a_new + deltaa(1);
    b_new = b_new + deltaa(2);
    %evaluating chi_square at new parameters a,b and lambda
    chisq = chi_square(D,t,a_new,b_new,lambda)
    end
    err = abs(chisq(1)-chisq(2));
    %stoppage criteria
    if(err<eps)
        break;
```

```
    end
end
%evaluating D using the estimated a and b values
for i = 1:100
    D_new(i) = a_new*t(i)^b_new;
end
%plotting data set and model data
plot(t,D,'g')
xlabel('t');
ylabel('D');
hold on
plot(t,D_new,'r-')
legend({'data file','modelled data'},'location','southeast')


function alpha_prime = alpha_func(D,t,a,b,lambda) %function alpha_func to
evaluate alpha prime
J = [0 0];
alpha = zeros(2);
Hessian = zeros(2);
alpha_prime = zeros(2);
for i = 1:100
    %derivatives to form the jacobi dD/da and dD/db
    dD_da (i) = t(i)^b;
    dD_db (i) = a*t(i)^b*log(t(i));
    J = [dD_da(i) dD_db(i)];
    %Hessian matrix is also equal to J'*J
    Hessian = J' * J;
    alpha = Hessian;
    %setting the diagnal elements to (1+lambda)*diagonal element
    alpha(1,1) = Hessian(1,1)*(1+lambda);
    alpha(2,2) = Hessian(2,2)*(1+lambda);
    alpha_prime = alpha_prime + alpha;
end
end

function beta = betafunc(D,t,a,b) %function betafunc to evaluate beta
beta = [0;0];
for i = 1:size(t)
    %Evaluating D_new at a and b parameters
    D_new(i) = a*t(i)^b;
    %derivatives of D with respect to a and b
    dD_da = t(i)^b;
    dD_db = a*t(i)^b *log(t(i));
    %evaluating beta
    beta(1) = beta(1) + (D(i) - D_new(i))* dD_da;
    beta(2) = beta(2) + (D(i) - D_new(i))* dD_db;
end
end



function chisq = chi_square(D,t,a,b,lambda) %function to evaluate chisquare
at new and old parameters
D_old = zeros(size(D));
chisq = [0;0];
```

```
chisq_old = 0;
chisq_new = 0;
deltaa = [0;0];
for i = 1:100
    %calculating D at the old values of a and b
    D_old(i) = a * t(i)^b;
    %evaluating chi square at the old values of a and b
    chisq_old = chisq_old + ((D(i) - D_old(i))^2);
    %function call to estimate delta_a
    deltaa = delta_a(D,t,a,b,lambda);
    %Obtaining new parameter values by adding dek_a to the old values
    a_new = a + deltaa(1);
    b_new = b + deltaa(2);
    %evaluating D at the new values of a and b
    D_new(i) = a_new *t(i)^(b_new);
    %evaluating chi square at the new values of a and b
    chisq_new = chisq_new + ((D(i) - D_new(i))^2);
end
chisq = [chisq_old;chisq_new]; %return value
end

function del_a = delta_a(D,t,a,b,lambda) % function to estimate delta a
alpha_prime = alpha_func(D,t,a,b,lambda); %function call to evaluate alpha
prime
beta = betafunc(D,t,a,b);  %function call to evaluate beta
del_a = inv(alpha_prime)*beta; %estimate del a = inv(alpha prime)*beta
end
```

PART 2: LINEAR REGRESSION

DIRECT METHOD

```
%linear regression using direct method
%initialization
s = 0;
sx = 0;
sy = 0;
sxx = 0;
sxy = 0;
%importing data set
data = xlsread('datafile.xlsx');
t = data(:,1);
D = data(:,2);
% model function logD = log(a) + b log(t)
x = size(t);
x = log10(t);
y = size(D);
y = log10(D);
for i = 1:100
    s = s+1;
    sx = sx + x(i);
    sy = sy + y(i);
    sxx =  sxx + x(i)^2;
    sxy = sxy + x(i)*y(i);
end
del = s*sxx-sx*sx;
log_a = (sy*sxx - sx*sxy)/del;
a = 10^(log_a)
```

```
b = (s*sxy - sx*sy)/del;
%parameters of the linear model
%for loop for evaluating D_new with the estimated parameters
D_new = zeros(size(D));
for i = 1:100
    D_new(i) = a*t(i)^b;
end
%plotting data set and the model data
plot(t,D,'g')
xlabel('t');
ylabel('D');
hold on
plot(t,D_new,'r-')
legend({'data file','modelled data'},'location','southeast')
```

GENERALIZED LINEAR REGRESSION

```
%linear regression model using generalized linear regression algorithm
%importing data
data = xlsread('datafile.xlsx');
t = data(:,1);
D = data(:,2);
% model function logD = log(a) + b log(t)
A = size(t);
A = log10(t);
[m,n] = size(t);
% insert the first column in A as 1
A = [ones(m,1) A];
b = size(D);
b = log10(D);
Const = [0;0];
%Generalized linear regression
%(A'*A)*a = A'*b
Const = inv(A'*A)*(A'*b);
log_a = Const(1);
b = Const(2);
%Parameters of the linear model
a = 10^log_a;

%For loop to evaluate  D_new at the evaluated parameters a and b
for i = 1:100;
    D_new(i) = a * t(i)^b;
end
%plotting both data set and the model values for comparison
plot(t,D,'g')
xlabel('t');
ylabel('D');
hold on
plot(t,D_new,'r-')
legend({'data file','modelled data'},'location','southeast')
```