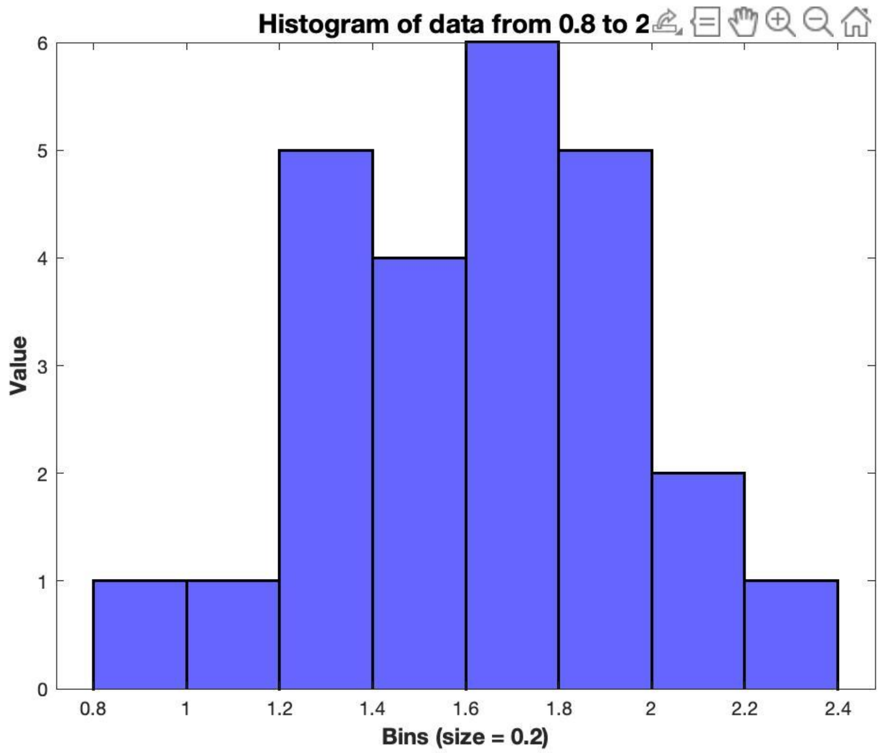
Robert Heeter
BIOE 391 Numerical Methods – Due 6 March 2022

Problem Set 6 Solutions

1. **Figure**

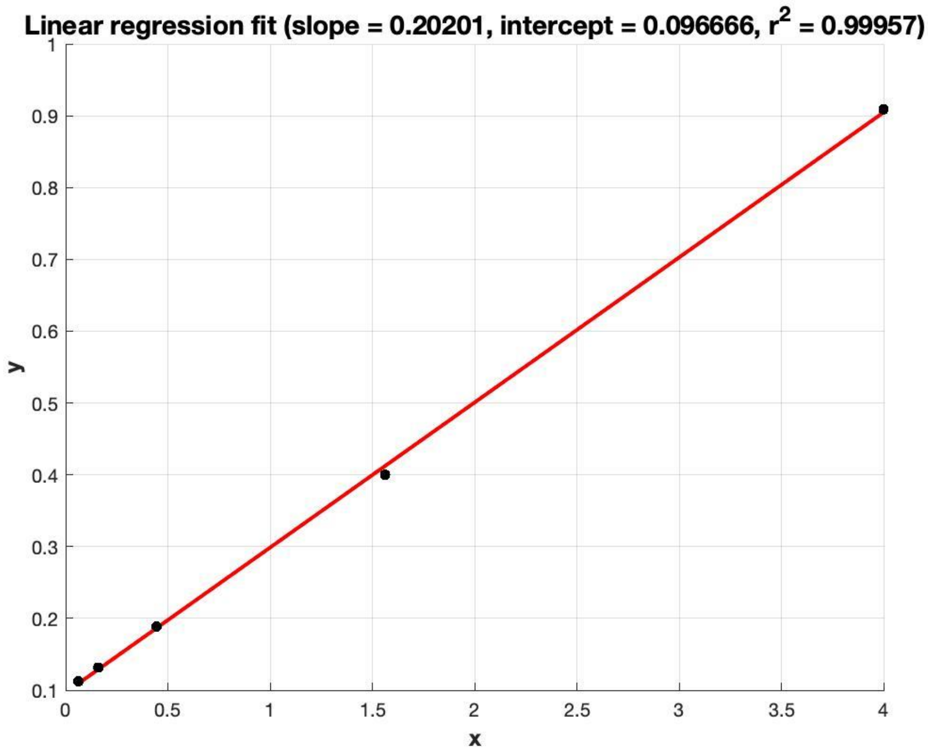


Robert Heeter
BIOE 391 Numerical Methods – Due 6 March 2022

2. **Output**
   Linear least-squares output:
   c_s [mg^2/L^2] = 2.089730
   k_max [1/day] = 10.344931
   r^2 = 0.999569

   Prediction:
   c_p [mg/L] = 2.000000
   k_p given c_p [1/day] = 6.795002

   Figure (from *linregr* function)

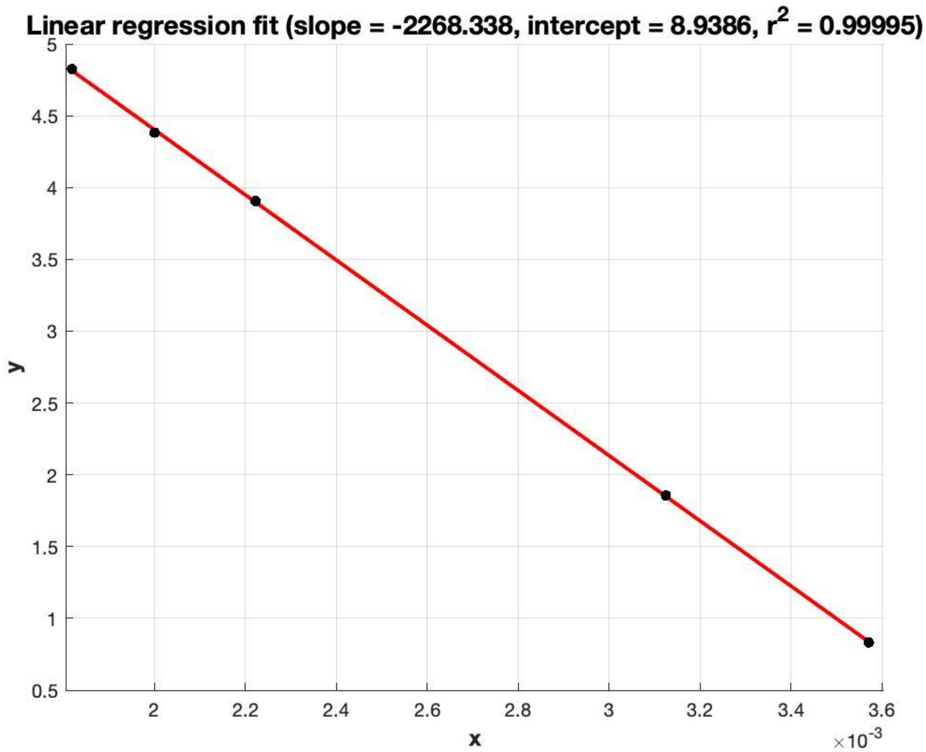


Robert Heeter
BIOE 391 Numerical Methods – Due 6 March 2022

3. Output

Linear least-squares output:
k_01 [1/s] = 7620.498573
E_1 [kcal/mol] = 4.491309
r^2 = 0.999948

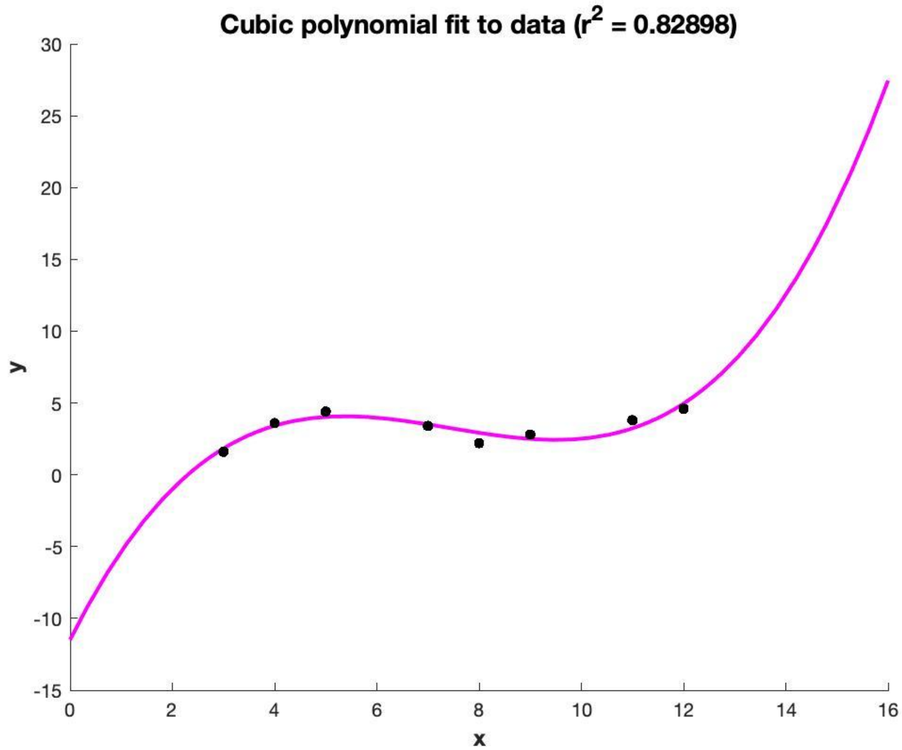
Figure (from *linregr* function)



Robert Heeter
BIOE 391 Numerical Methods – Due 6 March 2022

4. **Output**
   Linear least-squares output:
   Cubic polynomial fit, p(x) = (-11.488707) + (7.143817)x + (-1.041207)x^2 + (0.046676)x^3
   r^2 = 0.828981
   standard error, s_y/x = 0.570031

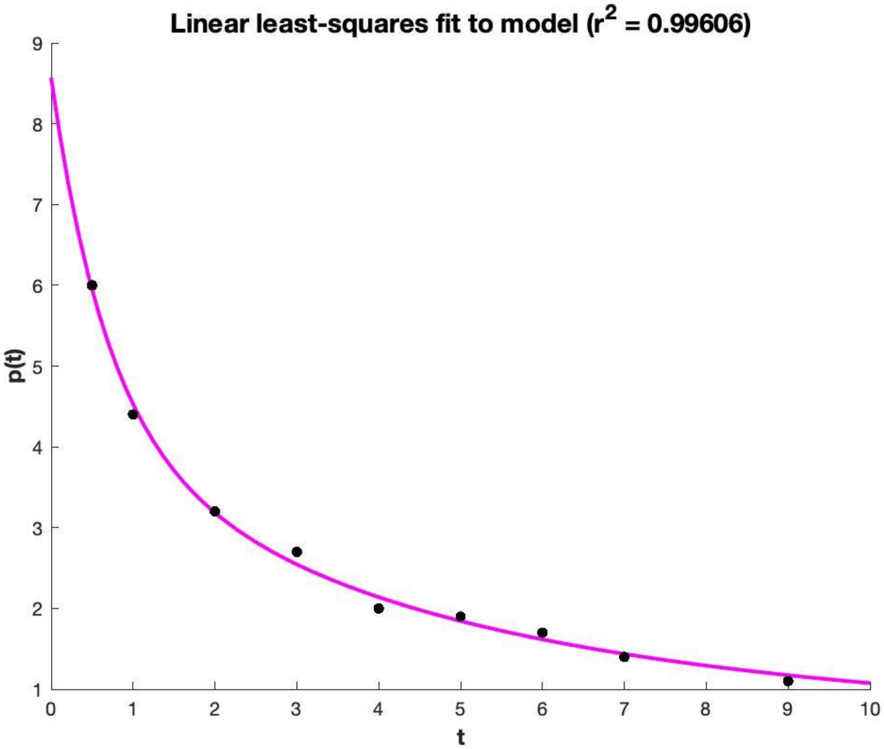
   Figure (to show cubic polynomial fit)



Robert Heeter
BIOE 391 Numerical Methods – Due 6 March 2022

5. **Output**
   Linear least-squares output:
   Model fit, p(t) = (4.137497)e^(-1.5t) + (2.895882)e^(-0.3t) + (1.534920)e^(-0.05t)
   A = 4.137497
   B = 2.895882
   C = 1.534920
   r^2 = 0.996063
   standard error, s_pt/t = 0.115721

   Figure (to show regression model fit)



Robert Heeter

BIOE 391 Numerical Methods – Due 6 March 2022

6.

a. Output

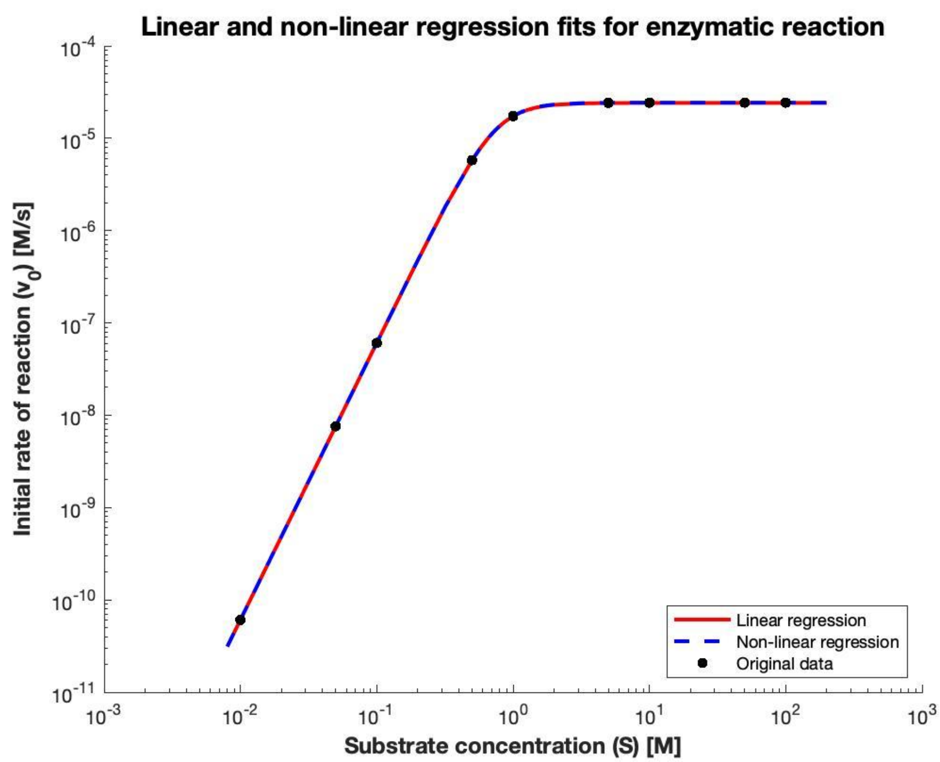
PART A: Linear least-squares output:
K [M^3] = 0.397412
k_m [M/s] = 0.000024
r^2 = 1.000000

Figure – see part B

b. Output

PART B: Non-linear least-squares with fminsearch output:
K [M^3] = 0.399724
k_m [M/s] = 0.000024

Figure



Robert Heeter
BIOE 391 Numerical Methods – Due 6 March 2022

Complete MATLAB Code

```
% Robert Heeter
% BIOE 391 Numerical Methods
% HOMEWORK 6 MATLAB SCRIPT

clc, clf, clear, close all

%% P1. PROBLEM 14.2
disp('P1. PROBLEM 14.2');

data = [0.90 1.42 1.30 1.32 1.35 1.47 1.96 1.47 1.92 1.85 1.74 1.65 2.29 1.82 2.06 1.55 1.63 1.95 1.66
1.35 1.05 1.78 1.71 2.14 1.27];
bins = (0.8:0.2:2.4);
figure
histogram(data,bins,'FaceColor','b','LineWidth',1.5);
xlabel('Bins (size = 0.2)','FontSize',12,'FontWeight','bold');
ylabel('Value','FontSize',12,'FontWeight','bold');
title('Histogram of data from 0.8 to 2.4','FontSize',14,'FontWeight','bold');
fprintf('\n');


%% P2. PROBLEM 14.14
disp('P2. PROBLEM 14.14');

c = [0.5 0.8 1.5 2.5 4]'; % oxygen concentration (mg/L)
k = [1.1 2.5 5.3 7.6 8.9]'; % growth rate of bacteria (per day)

% Linear least-squares
% Linearized equation: (1/k) = (1/k_max) + (c_s/k_max)*(1/c^2)
[a,r2] = linregr((1./c.^2),(1./k)); % use linregr function below with linearized equation
k_max = 1/a(2);
c_s = a(1)*k_max;

% Prediction
c_p = 2; % oxygen concentration (mg/L)
k_p = (k_max*c_p^2)/(c_s+c_p^2); % predicted growth rate at c_p (per day)

% Display results
disp('Linear least-squares output:')
fprintf('c_s [mg^2/L^2] = %f\nk_max [1/day] = %f\nr^2 = %f\n\n',c_s,k_max,r2);

disp('Prediction:')
fprintf('c_p [mg/L] = %f\nk_p given c_p [1/day] = %f\n\n',c_p,k_p);


%% P3. PROBLEM 14.21
disp('P3. PROBLEM 14.21');

dAdt_neg = [460 960 2485 1600 1245]'; % (moles/L/s)
A = [200 150 50 20 10]'; % (moles/L)
T = [280 320 450 500 550]'; % (K)
R = 0.00198; % ideal gas constant (kcal/mol/K)

% Linear least-squares
% Linearized equation: ln[(-dA/dt)/A] = ln(k_01) + (-E_1/R)*(1/T)
[a,r2] = linregr((1./T),log(dAdt_neg./A)); % use linregr function below with linearized equation
k_01 = exp(a(2));
E_1 = -1*R*a(1);

% Display results
disp('Linear least-squares output:')
fprintf('k_01 [1/s] = %f\nE_1 [kcal/mol] = %f\nr^2 = %f\n\n',k_01,E_1,r2);
```

Robert Heeter

BIOE 391 Numerical Methods – Due 6 March 2022

```
%% P4. PROBLEM 15.3
disp('P4. PROBLEM 15.3');

x = [3 4 5 7 8 9 11 12]'; % x-values
y = [1.6 3.6 4.4 3.4 2.2 2.8 3.8 4.6]'; % y-values

% Linear least squares to find coefficients
Z = [ones(size(x)) x x.^2 x.^3];
a = (Z'*Z)\(Z'*y); % vector of coefficients

% Determine standard error and coefficient of determination
st = sum((y-mean(y)).^2);
sr = sum((y-Z*a).^2);
r2 = 1-(sr/st);
s_yx = sqrt(sr/(length(x)-length(a)));

% Plot regression fit
p = @(x) a(1) + a(2).*x + a(3).*x.^2 + a(4).*x.^3;
figure
hold on
fplot(p,[0 16],'-m','LineWidth',2);
plot(x,y,'.k','MarkerSize',15);
xlabel('x','FontSize',12,'FontWeight','bold');
ylabel('y','FontSize',12,'FontWeight','bold');
title(['Cubic polynomial fit to data (r^2 = ',num2str(r2),')'],'FontSize',14,'FontWeight','bold');
hold off

% Display results
disp('Linear least-squares output:')
fprintf('Cubic polynomial fit, p(x) = (%f) + (%f)x + (%f)x^2 + (%f)x^3\nr^2 = %f\nstandard error, s_y/x
= %f\n\n',a(1),a(2),a(3),a(4),r2,s_yx);


%% P5. PROBLEM 15.10
disp('P5. PROBLEM 15.10');

t = [0.5 1 2 3 4 5 6 7 9]'; % time
pt = [6 4.4 3.2 2.7 2 1.9 1.7 1.4 1.1]'; % concentration

% Linear least-squares to find coefficients
Z = [exp(-1.5.*t) exp(-0.3.*t) exp(-0.05.*t)];
a = (Z'*Z)\(Z'*pt); % vector of coefficients

% Determine standard error and coefficient of determination
st = sum((pt-mean(pt)).^2);
sr = sum((pt-Z*a).^2);
r2 = 1-(sr/st);
s_ptt = sqrt(sr/(length(t)-length(a)));

% Plot regression fit
p = @(t) a(1).*exp(-1.5.*t) + a(2).*exp(-0.3.*t) + a(3).*exp(-0.05.*t);
figure
hold on
fplot(p,[0 10],'-m','LineWidth',2);
plot(t,pt,'.k','MarkerSize',15);
xlabel('t','FontSize',12,'FontWeight','bold');
ylabel('p(t)','FontSize',12,'FontWeight','bold');
title(['Linear least-squares fit to model (r^2 = ',num2str(r2),')'],'FontSize',14,'FontWeight','bold');
hold off

% Display results
disp('Linear least-squares output:')
fprintf('Model fit, p(t) = (%f)e^(-1.5t) + (%f)e^(-0.3t) + (%f)e^(-0.05t)\nA = %f\nB = %f\nC = %f\nr^2
= %f\nstandard error, s_pt/t = %f\n\n',a(1),a(2),a(3),a(1),a(2),a(3),r2,s_ptt);
```

Robert Heeter
BIOE 391 Numerical Methods – Due 6 March 2022

```
%% P6. PROBLEM 15.14
disp('P6. PROBLEM 15.14');

S = [0.01 0.05 0.1 0.5 1 5 10 50 100]'; % substrate concentration (M)
v_0 = [6.078e-11 7.595e-9 6.063e-8 5.788e-6 1.737e-5 2.423e-5 2.430e-5 2.431e-5 2.431e-5]'; % initial
rate of reaction (M/s)

% LINEAR LEAST-SQUARES (PART A)
[a,r2] = linregr((1./S.^3),(1./v_0)); % use linregr function below with linearized equation
k_ma = 1/a(2);
Ka = a(1)*k_ma;

% Display results
disp('PART A: Linear least-squares output:')
fprintf('K [M^3] = %f\nk_m [M/s] = %f\nr^2 = %f\n\n',Ka,k_ma,r2);

% NON-LINEAR REGRESSION (PART B)
b = fminsearch(@(a) vssr(a,S,v_0),[Ka k_ma]',[]); % use fminsearch to find constants that minimize sum
of squares of estimate residuals (from function vssr below)
k_mb = b(2);
Kb = b(1);

% Display results
disp('PART B: Non-linear least-squares with fminsearch output:')
fprintf('K [M^3] = %f\nk_m [M/s] = %f\n\n',b(1),b(2));

% Graph results from Parts A and B
v_0a = @(S) (k_ma.*S.^3)./(Ka+S.^3);
v_0b = @(S) (k_mb.*S.^3)./(Kb+S.^3);

figure
hold on
fplot(v_0a,[8e-3,2e2],'-r','LineWidth',2);
fplot(v_0b,[8e-3,2e2],'--b','LineWidth',2);
loglog(S,v_0,'.k','MarkerSize',15);
xlabel('Substrate concentration (S) [M]','FontSize',12,'FontWeight','bold');
ylabel('Initial rate of reaction (v_0) [M/s]','FontSize',12,'FontWeight','bold');
title('Linear and non-linear regression fits for enzymatic
reaction','FontSize',14,'FontWeight','bold');
legend('Linear regression','Non-linear regression','Original data','Location','southeast');
ax = gca;
ax.XScale = 'log';
ax.YScale = 'log';
hold off


%% Additional Functions

function [a, r2] = linregr(x,y)
% ABOUT: Linear regression least squares method, adapted from textbook .m
% file. Uses least squares fit by solving normal equations.
% INPUTS: x = independent variable; y = dependent variable
% OUTPUTS: a = vector of slope a(1) and intercept a(2); r2 = coefficient of
% determination

x = x(:); % set to column vectors
y = y(:);

n = length(x);

% Check inputs are valid
if length(y) ~= n
    error('Input vectors of x and y variables are different lengths.');
end

% Solve normal equations
```

Robert Heeter

BIOE 391 Numerical Methods – Due 6 March 2022

```
sx = sum(x); sy = sum(y);
sx2 = sum(x.*x); sxy = sum(x.*y); sy2 = sum(y.*y);

a(1) = (n*sxy-sx*sy)/(n*sx2-sx^2);
a(2) = sy/n-a(1)*sx/n;

r2 = ((n*sxy-sx*sy)/sqrt(n*sx2-sx^2)/sqrt(n*sy2-sy^2))^2;

% Create plot of data and best fit line
xp = linspace(min(x),max(x),2);
yp = a(1)*xp+a(2);

figure
hold on
plot(xp,yp,'-r','LineWidth',2);
plot(x,y,'.k','MarkerSize',15);
xlabel('x','FontSize',12,'FontWeight','bold');
ylabel('y','FontSize',12,'FontWeight','bold');
title(['Linear regression fit (slope = ',num2str(a(1)),', intercept = ', num2str(a(2)), ', r^2 =
',num2str(r2),')'],'FontSize',14,'FontWeight','bold');
grid on
hold off

end


function v = vssr(a, Sm, vm)
% ABOUT: Non-linear regression residual summation function for problem
% 15.14.
% INPUTS: a = coefficients for function; xm = x-values; ym = y-values
% OUTPUTS: v = sum of squares of estimate residuals

vp = (a(2).*Sm.^3)./(a(1)+Sm.^3);
v = sum((vm-vp).^2);

end
```