# CHEN26210 – Transport I Fall 2011

# Chemical Engineering Department MATLAB nlinfit Example

**T.D. Placek Auburn University**

**MATLAB nlinfit Example**

This example demonstrates how to use MATLAB’s “nlinfit” to perform a nonlinear regression of data to a user defined function.

In this case, a set of x and y data is to be fit to the function:

$$y=\frac{1}{b\_{1}}+\frac{b\_{2}}{x}$$

Note: this is similar to fitting data to a polynomial function using “polyfit” except the user must provide the function whose coefficients are to be determined.

b0 represents the “initial guesses” for the parameters b1 and b2. In this case, they are both assumed to be 0.5.

bhat (or in math notation, $\hat{b}$ ) are the parameter estimates (that is, the answers)

In the graph that is produced, black symbols are the original data and red symbols are the results of the regression.

% nlinfit example (simplified)

% Tim Placek - placetd@auburn.edu

% CHEN 2610 - Fall 2011

clear

clc

clf

format compact

% the data

 x=2:16;

 y=[6.42,8.2,9.58,9.5,9.7,10,9.93,9.99,...

 10.49,10.59,10.6,10.8,10.6,10.9,10.76];

% the proposed functionality (fh is a handle to the function)

 fh=@(b,x) 1./(b(1)+b(2)./x);

% guess values for parameters (beta0)

 b0=[0.5,0.5];

% plot the raw data

 plot(x,y,'s','markersize',5,'color',[0,0,0]);

 hold on

% determine best fit values for coefficient (bhat)

 bhat=nlinfit(x,y,fh,b0);

% plot the fit

 xf = linspace(x(1), x(length(x)));

 plot(xf,fh(bhat,xf),'linewidth',1,'color',[1,0,0]);

 legend('original data','fit data','location','Best') % the result

 bhat(1)

 bhat(2)

Output:

ans =

 0.0845

ans =

 0.1152

Graph:

