A First Course on Kinetics and Reaction Engineering
Supplemental Unit S3. Fitting Linear Models to Data

Defining the Problem

This supplemental unit describes how to use linear least squares to fit a mathematical model to a set of experimental data in which each data point consists of the value of a single, experimentally measured response variable, $\hat{y}$, and one or more experimentally set variables, x_i. The model must have a linear form wherein the predicted value of the response variable, y, depends on the set variables as given in equation (1), where n_{set} represents the number of set variables per experimental data point. The parameters m_i and b in equation (1) must be unknown constants and n_{set} must be equal to or greater than 1. The parameter b in equation (1) is optional. The fitting process finds the best values for the unknown parameters, m_i and b. This supplemental unit also describes how to assess the accuracy of the resulting model and decide whether or not it is acceptable.

$$y = b + \sum_{i=1}^{n_{set}} m_i x_i \tag{1}$$

In some instances in the analysis of kinetics data, the model takes the form given in equation (2) which *does not have the required form*. Specifically, in equation (2) the slope is a *known* constant (1 in this case), not an unknown constant (m). The methods discussed in this supplemental unit cannot be used to fit equation (2) to experimental data. Fortunately, numerical methods are not necessary in this case. Since x and y are known for each experimental data point, a corresponding value of b can be calculated. The best value for b is simply the average of the values calculated using each data point. The standard deviation of the values calculated using each data point can be used as a measure of the uncertainty in the best value of b. If the model is an accurate representation of the data, then the standard deviation will be a very small fraction of the best value.

$$y = x + b \tag{2}$$

Information and Data Required for Numerical Solution

This supplemental unit considers the use of computer software to fit a linear model of the form given in equation (1) to a corresponding set of data where each data point includes a value for the experimentally measured response variable, $\hat{y}$, and each of the experimentally set variables, x_i. There are many different software packages that can be used to accomplish this task, but irrespective of the particular brand of software you use, you will need to provide three things as input to that software:

- the number, n_{set}, of set variables (also called independent variables), x_i
- whether or not the model includes an intercept, b

- a set of experimental data points, each of which consists of a value for the response variable (also called the dependent variable), $\hat{y}$, and corresponding values for each of the set variables x_1 through $x_{n_{set}}$.

The software will then compute and return the best value for each parameter (b and/or m_i) in the model. It will also typically return the value of the correlation coefficient, r^2, and some measure of the uncertainty in the value of each parameter (typically either the standard deviation or the 95% confidence interval). If the software does not generate them, the results can be used to generate a model plot (when there is only one set variable, x) or a parity plot and residuals plots (when there are two or more set variables) that can be used to assess how accurately the model represents the data.

Overview of the Numerical Method

For the purpose of describing what the software does, equation (1) will be re-written as shown in equation (3) which simply uses θ_i to represent both the slopes and the intercept. The objective is to fit that model to a set of experimental data points, where each data point data point l takes the form $\left(x_{1,l}, x_{2,l}, \cdots, x_{n_s,l}, \hat{y}_l\right)$. The "^" over the response variable designates that it is the experimentally measured value; the absence of a "^" indicates a value predicted by the model, as in equation (3).

$$y = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_{n_s} x_{n_s} + \theta_{n_s+1} \tag{3}$$

The first task in fitting equation (3) to experimental data is to find the best values of the parameters, θ_1 through θ_{n_s+1}. Experimental measurements are subject to small random fluctuations, often called experimental noise. As a consequence, if the set variable values for data point l are substituted into the model equation, it will not predict a response value, y_l, that is exactly equal to the experimentally measured response, $\hat{y}_l$. Instead, there will be an error, or residual, ε_l, associated with each data point l, as defined in equation (4). The "best" values for the parameters in the model equation are those parameter values that minimize the aggregate of these errors over all of the data points. For single response data, the overall aggregate error, Φ, is taken to equal the sum of the squares of the errors for each of the data points, equation (5), where n_e represents the number of experimental data points. The squares of the errors are used instead of the errors themselves so that a large positive error for one data point is not offset by a large negative error for a different data point. The function, Φ, in equation (5) is called the objective function, and the "best" values of the parameters in the model equation are those parameter values that minimize Φ.[1]

$$\varepsilon_l = \hat{y}_l - y_l \tag{4}$$

[1] Note that in equation (5) every data point is weighted equally. If each data point does not have the same variance, and if the variance associated with each data point is known or can be estimated, an appropriately weighted sum of the squares of the errors is a better objective function.

$$\Phi = \sum_{l=1}^{n_e} \varepsilon_l^2 = \sum_{l=1}^{n_e} \left(\hat{y}_l - y_l \right)^2 = \sum_{l=1}^{n_e} \left(\hat{y}_l - \theta_1 x_{1,l} - \theta_2 x_{2,l} - \cdots - \theta_{n_s} x_{n_s,l} - \theta_{n_s+1} \right)^2 \tag{5}$$

If the parameters are to be chosen so that the resulting value of the error function, equation (5), is a minimum, then the partial derivative of Φ with respect to each of the parameters should be equal to zero. Equation (6) expresses this requirement for any one parameter θ_k. The derivative in equation (6) can be evaluated by differentiating equation (5). The result is given in equation (7) for k equal to 1 through n_s, and in equation (8) for $k = n_s + 1$. Equations (7) and (8) represent a set of $n_s + 1$ equations, and those equations can be solved simultaneously to find the "best" values of the $n_s + 1$ parameters.

$$\frac{\partial \Phi}{\partial \theta_k} = 0 \tag{6}$$

$$0 = \sum_{l=1}^{n_e} \left(\hat{y}_l - \theta_1 x_{1,l} - \theta_2 x_{2,l} - \cdots - \theta_{n_s} x_{n_s,l} - \theta_{n_s+1} \right) x_{k,l}; \quad k = 1 \text{ through } n_s \tag{7}$$

$$0 = \sum_{l=1}^{n_e} \left(\hat{y}_l - \theta_1 x_{1,l} - \theta_2 x_{2,l} - \cdots - \theta_{n_s} x_{n_s,l} - \theta_{n_s+1} \right); \quad k = n_s + 1 \tag{8}$$

For example, consider the application of equation (6) to the simplified model given in equation (9). In that case n_s is equal to 1 with, θ_1 is equal to m and θ_2 equal to b. Applying equations (7) and (8) gives two equations containing m and b as unknowns. Those two equations can be solved to obtain expressions for the "best" values of m and b as given in equations (10) and (11).

$$y = mx + b \tag{9}$$

$$m = \frac{\displaystyle\sum_{l=1}^{n_e} \left\{ \left(x_l - \frac{\displaystyle\sum_{i=l}^{n_e} x_l}{n_e} \right) \left(\hat{y}_l - \frac{\displaystyle\sum_{l=1}^{n_e} \hat{y}_l}{n_e} \right) \right\}}{\displaystyle\sum_{l=1}^{n_e} \left\{ \left(x_l - \frac{\displaystyle\sum_{l=1}^{n_e} x_l}{n_e} \right)^2 \right\}} \tag{10}$$

$$b = \frac{\displaystyle\sum_{l=1}^{n_e} \hat{y}_l}{n_e} - m \left(\frac{\displaystyle\sum_{l=1}^{n_e} x_l}{n_e} \right) \tag{11}$$

Equations analogous to equations (10) and (11) can be derived for the parameters θ_k in model equation (3). When you use mathematics software to fit equation (3), or equivalently equation (1), to experimental data, the software is evaluating those equations and returning the best values of the parameters to you.

It is important to realize that the model parameters that minimize the objective function are not precisely known. That is, due to the statistical nature of the whole process, there is uncertainty in each of the model parameter values. For this reason, each parameter should be reported as the best value along with an estimate of the uncertainty in that value. Two common measures used to describe the uncertainty are the standard error of the parameter and the 95% confidence interval. The latter is a range of values chosen such that there is a 95% probability that the true value of the parameter lies within that range. There are standard statistical methods for estimating either the standard error or the 95% confidence intervals for the parameters, but they won't be discussed here (consult a good statistics textbook). Most software for fitting linear equations to experimental data also performs these statistical calculations and returns either the standard error or the 95% confidence interval for each parameter.

Just because the parameters found in a regression analysis minimize the objective function (i. e. the overall error), this does not guarantee that the fit is a good one. For example if one had a set of data points that fell perfectly on a parabola, one could still use regression analysis to fit a straight line to the data. The result would be the slope and intercept of the straight line that best fit the data. However, as seen in Figure S3.1, the "best" straight line still wouldn't fit the data very well. Normally one expects that there will be some "scatter" of the data about the line. That is, the experimental data points will not fall exactly on the plot of the model. However, this scatter should be random; it should not be systematic. That is, there should not be any apparent trend in the scatter and it should not correlate with any aspect of the experimental work. The data in Figure S3.1 show a systematic scatter about the linear function.

There are statistical measures of how well a model fits the data. One such computed measure of the "goodness of fit" is known as a correlation coefficient which is defined in equation (12). In equation (12), a bar over a quantity indicates the average of that quantity taken over all the data points. The correlation coefficient is often denoted by the variable r, and it, or its square, offer a measure of how well the model describes the data. When r^2 is nearly equal to 1.0, the model offers a very good representation of the experimental data whereas an r^2 value near zero indicates a model that does a poor job of describing the experimental results. Most software for fitting linear equations to experimental data will calculate and return the value of r or r^2.

$$r = \frac{\sum_{l=1}^{n_e}\left\{\left(x_l - \overline{x}\right)\left(y_l - \overline{y}\right)\right\}}{\sqrt{\left[\sum_{l=1}^{n_e}\left\{\left(x_l - \overline{x}\right)^2\right\}\right]\left[\sum_{l=1}^{n_e}\left\{\left(y_l - \overline{y}\right)^2\right\}\right]}} \qquad (12)$$

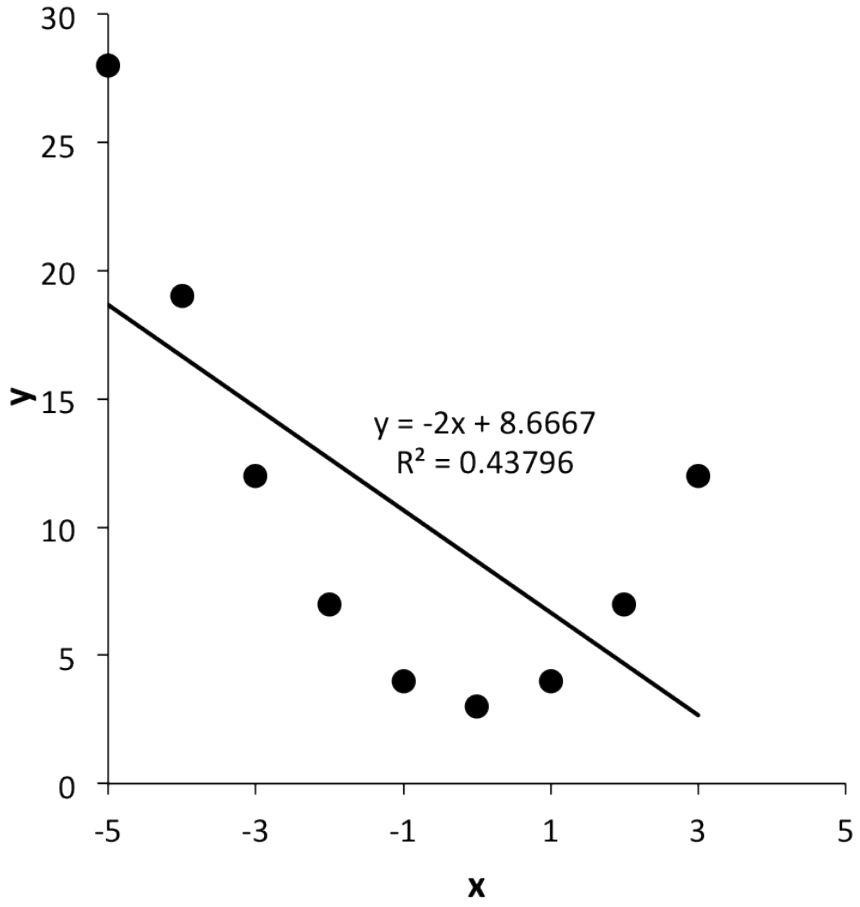

Figure S3.1. Straight line fit to a set of data that fall exactly on a parabola. While the "best" slope of the straight line is -2 and the "best" intercept is 8.6667, the line does not fit the data well.

A plot of the experimental data as points and the model function as a line on the same set of coordinate axes, as in Figure S3.1, is called a model plot. When the fit is good, the deviations of the points from the line should be small and random, but a plot like this can only be constructed if the model and data only have one set variable. If the model has two or more set variables, there are two other types of plots that can be used to assess how well the equation fits the data. One is called a *parity plot* or a *rectifying plot*. In it, one simply plots the experimentally measured value, $\hat{y}_l$, for each data point versus the value, y_l, for the same data point that is predicted by the best-fit model. If the fit is perfect, all the data points will lie on a diagonal line passing through the origin. The quality of the fit is indicated by the magnitude of the deviations away from the diagonal line.

A parity plot will show the magnitude of deviations between the experimental data and the model's predictions, but it may not reveal systematic trends in the deviations. For this, a second type of plot, called a *residuals plot*, can be used. To construct a residuals plot, one uses the fitted equation and the data to calculate the error, ε_l, for each data point. Then the error can be plotted against each of the set

variables. The errors can also be plotted against other properties that might affect the quality of the fit or data. The expected behavior would be for the errors to scatter randomly about zero; if they exhibit systematic deviations from zero, this suggests that the model has failed to capture the full functional dependence of the response variable upon the abscissa variable. As noted, one can also plot the errors against, for example, the technician who performed the experiment or the supplier of the reactants to see if the error correlates with these factors.

Some software for fitting linear equations to experimental data will generate a model plot or a parity plot and residuals plots. However, if the software does not do this automatically, it is easily done after the fitting has been completed.

MATLAB Script Files

MATLAB provides a number of built-in functions that can be used to perform the tasks described in the preceding section. It takes very few lines of MATLAB code to set up a file that will fit a single response linear model to appropriate data, calculate the regression coefficient and 95% uncertainty limits, and make appropriate model, parity and residuals plots. However, to allow you to focus more on kinetics and reaction engineering and less on computer programming, three MATLAB script files have been created for your use. These files are named “FitLinSR.m”, “FitLinmbSR.m” and “FitLinmSR.m”; these filenames are meant to convey the purpose of the script. Thus FitLinSR.m **fit**s a general **lin**ear model, equation (1), to **s**ingle **r**esponse data, FitLinmbSR.m fits a **lin**ear equation with slope **m** and intercept **b,** equation (9), to **s**ingle **r**esponse data and FitLinmSR.m fits a **lin**ear equation with slope **m**, but no intercept, equation (13), to **s**ingle **r**esponse data. Each of these scripts will perform the fitting (i. e. calculate the best parameter values), calculate the 95% confidence limits for each parameter, calculate the correlation coefficient, r^2, and generate either a model plot or a parity plot and residuals plots for each set variable.

$$y = mx \tag{13}$$

The scripts require no modification in order to be used. They simply call the standard MATLAB functions mentioned in the previous section to perform the calculations described earlier, including the statistical calculations that were not described. They also process the results to make plots that can be used in assessing the quality of the fit. You are encouraged to read through the script files and the related MATLAB documentation in order to obtain a greater appreciation of how they work. Three “How-To” files are provided as part of this supplemental unit. These files give step by step instructions for using each of the MATLAB script files to fit the corresponding model to appropriate single response data. Consequently, step by step instructions for using the script files will not be presented here.

There is ***one very important requirement for the use of the script file FitLinSR.m*** to fit a general linear model, equation (1) to experimental data. The MATLAB function “regress” that is used within FitLinSR.m *requires* that there be an intercept. If the model equation does not have an intercept, it must be converted into a form that does have one. Otherwise, as noted in the MATLAB documentation, the 95% confidence limits returned by the MATLAB script will be wrong. Fortunately, it is trivially easy to convert a linear equation without an intercept into a linear equation with an intercept.

To illustrate, suppose one had experimental data that involved four set variables, x_1 through x_4, and a linear model without an intercept, equation (14). Dividing equation (14) by x_4 gives equation (15). Four new variables can be defined as in equations (16) through (19). Substitution of these new variables in equation (15) leads to equation (20) which is linear ***and has an intercept***. Thus, instead of fitting equation (14) to the original (y, x_1, x_2, x_3, x_4) data, you would calculate y', x'_1, x'_2, and x'_3 for each data point and then fit equation (20) to those data. In this way, MATLAB will not return erroneous 95% confidence intervals for the parameters.

$$y = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 \tag{14}$$

$$\frac{y}{x_4} = \theta_1 \frac{x_1}{x_4} + \theta_2 \frac{x_2}{x_4} + \theta_3 \frac{x_3}{x_4} + \theta_4 \tag{15}$$

$$y' = \frac{y}{x_4} \tag{16}$$

$$x'_1 = \frac{x_1}{x_4} \tag{17}$$

$$x'_2 = \frac{x_2}{x_4} \tag{18}$$

$$x'_3 = \frac{x_3}{x_4} \tag{19}$$

$$y' = \theta_1 x'_1 + \theta_2 x'_2 + \theta_3 x'_3 + \theta_4 \tag{20}$$