A First Course on Kinetics and Reaction Engineering

Activity 9.2

Problem Purpose

This problem illustrates the use of a Lineweaver-Burk type of plot to determine the values of the constants in a Michaelis-Menten rate expression *with inhibition*.

Problem Statement

Suppose the enzyme-catalyzed reaction (1) is believed to obey Michaelis-Menten kinetics with inhibition, equation (2). This is the rate expression derived in Example 9.3. To test this, the rate of production of P was measured as a function of the concentrations of S and I using a 500 cm^3 chemostat and 10.0 mg of enzyme. The temperature, pressure and solution volume were all constant over the course of the experiments. On the basis of the resulting data, presented in Table 1, does equation (2) offer an acceptable description of the reaction rate? If so, what are the best values of V_{max}, K_I and K_m?

Table 1. Data for Activity 9.2

C_S (M)	C_I (M)	r_P (M/min)
0.100	0.100	0.000798
0.086	0.050	0.000838
0.080	0.005	0.000912
0.075	0.001	0.000915
0.070	0.100	0.000745
0.063	0.050	0.000819
0.056	0.005	0.000898
0.048	0.001	0.000901
0.047	0.100	0.000689
0.041	0.050	0.000771
0.036	0.005	0.000896
0.030	0.001	0.000890
0.025	0.100	0.000563
0.021	0.050	0.000664
0.015	0.005	0.000846
0.010	0.001	0.000855

$$S \rightarrow P \tag{1}$$

$$r_{P,1} = \frac{V_{\max}\left[S\right]}{K_m + K_I\left[I\right] + \left[S\right]} \tag{2}$$

Problem Analysis

We will follow the same approach as was used in Example 9.4. Specifically, the rate expression will be linearized by taking its reciprocal. The experimental data will then be used to compute values for the new independent and dependent variables. A linear model (this time with 2 independent variables) will be fit to those values, and the accuracy of the fit will be assessed. If the fit is found to be sufficiently accurate, the best values of the three model parameters, and their uncertainties will be calculated.

Problem Solution

Taking the reciprocal of rate expression (2) yields equation (3). V_{max}, K_I and K_m are constants, so this equation has a linear form like that shown in equation (4) if y, x_1 and x_2 are defined as shown in equations (5), (6) and (7). The slopes, m_1 and m_2, and the y-intercept, b, are related to the original kinetic parameters according to equations (8), (9) and (10).

$$\frac{1}{r_{P,1}} = \left(\frac{K_m}{V_{\max}}\right)\frac{1}{C_S} + \left(\frac{K_I}{V_{\max}}\right)\frac{C_I}{C_S} + \frac{1}{V_{\max}} \tag{3}$$

$$y = m_1 x_1 + m_2 x_2 + b \tag{4}$$

$$y = \frac{1}{r_{P,1}} \tag{5}$$

$$x_1 = \frac{1}{C_S} \tag{6}$$

$$x_2 = \frac{C_I}{C_S} \tag{7}$$

$$m_1 = \left(\frac{K_m}{V_{\max}}\right) \tag{8}$$

$$m_2 = \left(\frac{K_I}{V_{\max}}\right) \tag{9}$$

$$b = \frac{1}{V_{\max}} \tag{10}$$

In this problem, we are asked to test whether a mathematical model offers an accurate representation of experimental data. Problems of this type can be solved by fitting the model equation to the data and then assessing the quality of the fit statistically and visually. In this particular problem, the model equation is linear, so linear least squares can be used to fit it to the experimental data (see Supplemental Unit S3). Linear least squares fitting can be performed manually, using a calculator, using a spreadsheet or using mathematics software. No matter which tool one chooses to use, it will be necessary to provide the following information and input data:

- the number of independent (x) variables
- whether or not the model includes an intercept (b)
- a set of experimental data points, each of which consists of a value for the dependent variable (y) and corresponding values for each of the independent variables (x_i)

In this particular problem, the model, equation (4), has two independent variables, x_1 and x_2, as well as a y-intercept, b. Each row in Table 1 represents one data point. For each of these data points, values of x_1, x_2 and y can be calculated using the values of C_S, C_I and r_P for that data point and equations (5), (6) and (7). When this information and input data are provided to whichever linear least squares fitting tool one chooses to employ, the resulting output shows that the correlation coefficient, r^2, is equal to 0.999, the best value of the slope, m_1, is equal to 0.67 ± 0.16 min, the best value of the slope, m_2, is equal to 166 ± 3 min M^{-1} and the best value of the y-intercept, b, is equal to 1084 ± 7 min M^{-1}. (Here the uncertainties are the 95% confidence limits, other least squares fitting tools might provide different measures of the uncertainties in the fitted parameter values.) In most cases, a parity plot like that shown in Figure 1 and residuals plots like those shown in Figures 2 and 3 are also provided, but if such plots are not provided, they can be generated easily.

Before the best values for m_1, m_2 and b can be accepted, one must decide whether the final model is sufficiently accurate. In this case, the accuracy of the model can be assessed using the correlation coefficient, the parity plot and the residuals plots. The closer the correlation coefficient is to a value of 1.0, the better the fit of the model to the data. In this case, the correlation coefficient of 0.999 indicates a very good fit. Additionally, if the fit is accurate, then the deviation of the data points in the parity plot from the diagonal line should be small. This can be seen to be the case for this problem by examination of Figure 1. Finally, if the fit is accurate, the residuals should scatter randomly about zero in the residuals plots; there should not be any apparent trends in the scatter of the data about the line in the figures. Examination of Figures 2 and 3 shows this to be true for the present problem. Thus, the model does appear to offer a sufficiently accurate representation of the experimental data, and the values of the slopes and intercept can accepted.

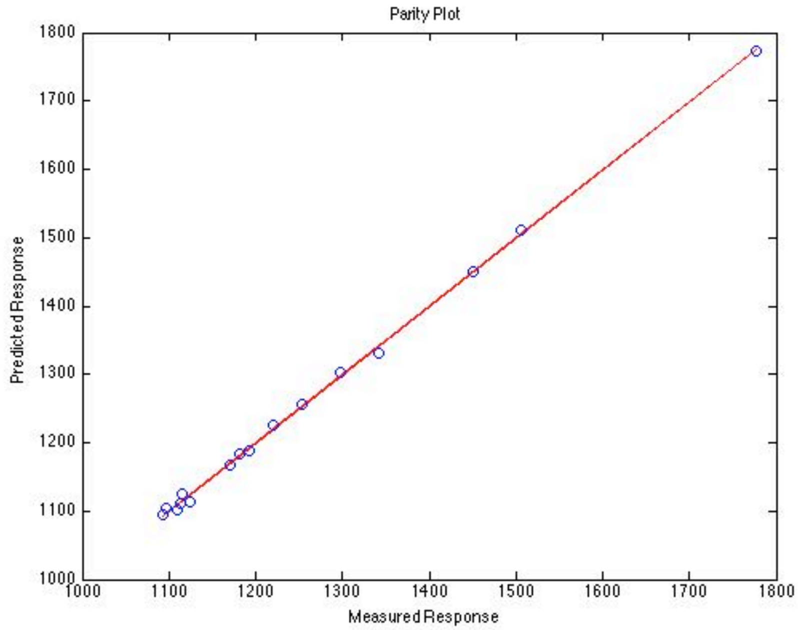

Figure 1. Parity plot comparing the experimental value of y to the value predicted by the fitted model.

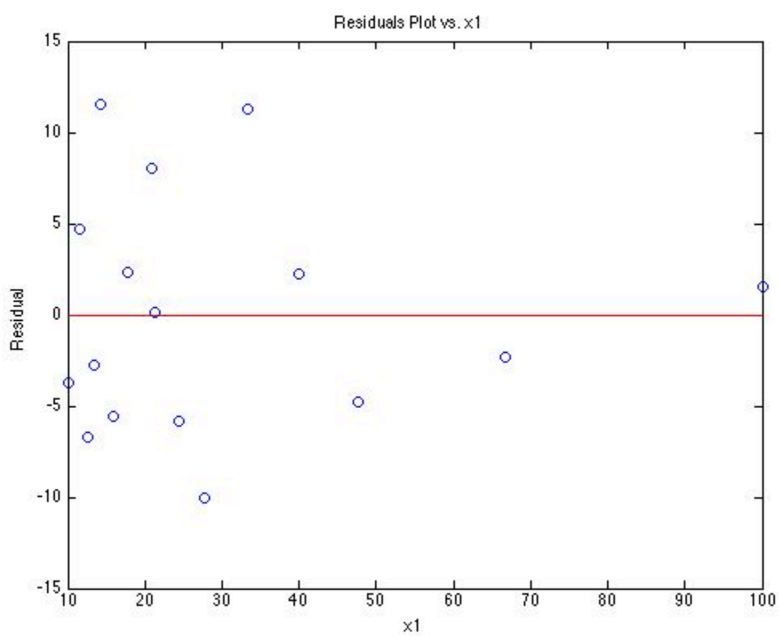

Figure 2. Residuals plot showing the difference between the experimental and model-predicted values of y as a function of x_1.

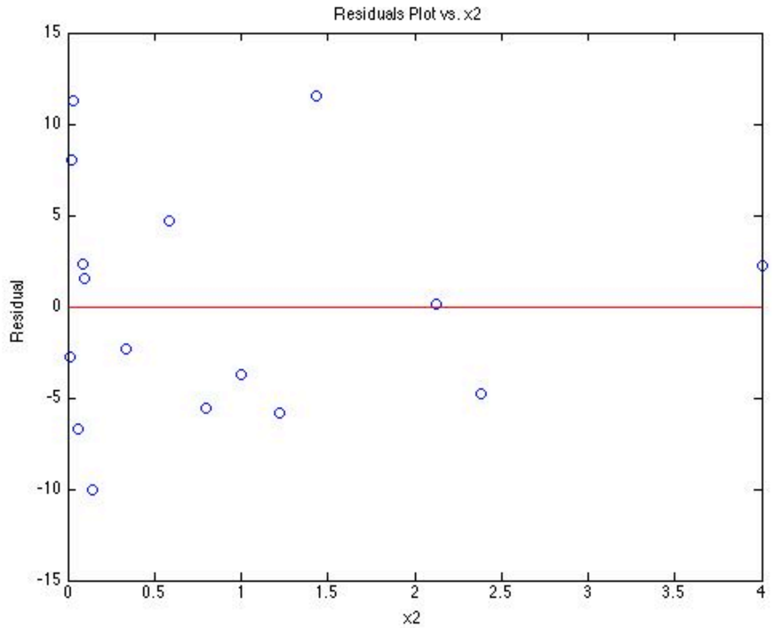

Figure 3. Residuals plot showing the difference between the experimental and model-predicted values of y as a function of x_2.

Having found the model to be acceptable, the problem additionally asks for the best values of V_{max}, K_I and K_m. Equations (8) through (10) can be rearranged to get equations (11), (12) and (13), and these can be used to calculate V_{max}, K_I and K_m from the values of the slopes, m_1 and m_2, and the intercept, b.

$$V_{\max} = \frac{1}{b} \tag{11}$$

$$K_m = m_1 V_{\max} = \frac{m_1}{b} \tag{12}$$

$$K_I = m_2 V_{\max} = \frac{m_2}{b} \tag{13}$$

A differential error analysis shows that if a model parameter, p, is related to the slopes, m_i, and intercept, b, of a linearized form of the model, as in equation (14), then the uncertainty in that parameter, λ_p, is related to the slope, intercept and their uncertainties, λ_{m_i} and λ_b, according to equation (15). Applying that relationship to the present problem shows that the uncertainties in V_{max}, K_I and K_m should be calculated using equations (16) through (18).

$$p = f\left(\underline{m}, b\right) \tag{14}$$

$$\lambda_p = \sqrt{\sum_i \left(\frac{\partial f}{\partial m_i}\right)^2 \lambda_{m_i}^2 + \left(\frac{\partial f}{\partial b}\right)^2 \lambda_b^2} \tag{15}$$

$$\lambda_{V_{max}} = \frac{\lambda_b}{b^2} \tag{16}$$

$$\lambda_{K_m} = \sqrt{\left(\frac{\lambda_{m_1}}{b}\right)^2 + \left(\frac{m_1 \lambda_b}{b^2}\right)^2} \tag{17}$$

$$\lambda_{K_I} = \sqrt{\left(\frac{\lambda_{m_2}}{b}\right)^2 + \left(\frac{m_2 \lambda_b}{b^2}\right)^2} \tag{18}$$

Using the values of m_1, λ_{m_1}, m_2, λ_{m_2}, b and λ_b resulting from fitting the model to the data and equations (11) through (13) and (16) through (18), the best values of the parameters can be calculated to be V_{max} = (9.22 ± 0.06) x 10^{-4} M min^{-1}, K_m = (6.18 ± 1.47) x 10^{-4} M and K_I = 0.153 ± 0.003.

Calculation Details Using MATLAB

Three MATLAB script files are provided with Supplemental Unit S3. The file names indicate the number of independent variables and whether or not the model has an intercept. The script named FitLinmSR is used when the model has one independent variable (x) and *does not* include the intercept (b). FitLinmbSR is used when the model has one independent variable and *does* include the intercept, and FitLinSR is used when the model has two or more independent variables. (With MATLAB, when the model has two or more independent variables, it must have an intercept; Supplemental Unit S3 describes how to convert a model without an intercept into a model that has an intercept.) In this problem the model has two independent variables, so the script file named FitLinSR will be used. To do so, the script file must be located in the current MATLAB working directory or in the MATLAB search path.

Before executing FitLinSR, the experimental values of x_1 and x_2 must be stored in a matrix named `x`, and the experimental values of y must be stored in a vector named `y_hat`. More specifically, the first column in matrix `x` must be a vector containing the values of x_1, the second column in matrix `x` must be a vector containing the values of x_2 and the third column in matrix `x` must contain the value 1.0 in every row, as described in Supplemental Unit S3. Upon execution of the script, it will return the correlation coefficient as `r_squared`, the slopes as a vector named `m`, the 95% confidence limits on the slopes as a vector named `m_u`, the intercept as `b` and the 95% confidence limits on the intercept as `b_u`. It will also generate a parity plot and two residuals plots.

Therefore the following steps must be performed in order to solve this problem using MATLAB:

- enter the values of C_S as a vector (here I named it `CS`)
- enter the values of C_I as a vector (here I named it `CI`)

- enter the values of r_P as a vector (here I named it `rP`)
- calculate the values of x_1 and x_2 and store them, along with a vector of 1.0's, in a matrix named x
- calculate the values of y and store them in a vector named `y_hat`
- execute the MATLAB script file named FitLinmb
- use the returned values of `m`, `m_u`, `b` and `b_u` to calculate V_{max} (here I named it `Vmax`), K_m (here I named it `Km`), K_I (here I named it `KI`), $\lambda_{V_{\max}}$ (here I named it `lambda_Vmax`), λ_{K_m} (here I named it `lambda_Km`) and λ_{K_I} (here I named it `lambda_KI`) according to equations (11) through (13) and (16) through (18)

Listing 1 shows how the values of C_S were saved as a vector named `CS`. The values of C_I and r_P were saved in an analogous manner. Listing 2 shows how the values of x_1, x_2 and y are calculated as vectors, how the matrix x is created, and how the parameters V_{max}, K_I and K_m and their uncertainties are calculated after executing FitLinSR. The code shown in Listings 1 and 2 could be performed at the MATLAB command prompt. Instead, I put them in a MATLAB script file named Activity_9_2.m which accompanies this example. The output generated upon execution of Activity_9_2 in MATLAB is shown in Listing 3.

```
CS = [0.100
0.086
0.080
0.075
0.070
0.063
0.056
0.048
0.047
0.041
0.036
0.030
0.025
0.021
0.015
0.010];
```

Listing 1. MATLAB code used to store the values of C_S from Table 1 in a vector named `CS`.

```
% Calculate the x and y_hat values
x1 = 1./CS;
x2 = CI./CS;
y_hat = 1./rP;

% Form the x matrix
x = [x1 x2 ones(length(x1),1)];

% Use the MATLAB script file "FitLinmbSR.m" from "A First Course on
% Kinetics and Reaction Engineering" to fit a straight line with slope and
% intercept to the data
FitLinSR

% Calculate Vmax and its 95% confidence limits
Vmax = 1/b
lambda_Vmax = b_u/b^2

% Calculate Km and its 95% confidence limits
Km = m(1)*Vmax
lambda_Km = sqrt(m_u(1)^2/b^2 + (m(1)*b_u/b^2)^2)

% Calculate KI and its 95% confidence limits
KI = m(2)*Vmax
lambda_KI = sqrt(m_u(2)^2/b^2 + (m(2)*b_u/b^2)^2)
```

Listing 2. MATLAB code that prepares the data, executes FitLinSR and uses the results to calculate the values of the model parameters and their uncertainties.

```
>> Activity_9_2

r_squared =
    0.9988

m =
    0.6697
  165.6725

m_u =
    0.1594
    3.4234

b =
   1.0845e+03

b_u =
    6.7188

Vmax =
   9.2209e-04

lambda_Vmax =
   5.7127e-06

Km =
   6.1753e-04

lambda_Km =
   1.4701e-04

KI =
    0.1528

lambda_KI =
    0.0033
```

Listing 3. MATLAB command window output upon running Activity_9_2.m