A First Course on Kinetics and Reaction Engineering
Unit 16. Numerical Data Analysis

Overview

In Units 13 through 15 the models for the kinetics experiments were linearized, after which they were fit to experimental data using linear least squares. This unit describes how a numerical implementation of least squares fitting can be used for the analysis of kinetics data when either the differential mole balance design equation cannot be analytically integrated or the algebraic model equation cannot be linearized. It also discusses the analysis of kinetics data where more than one reaction was taking place or more than one variable related to the extent of reaction was measured.

Learning Objectives

Upon completion of this unit, you should be able to define, in words, the following terms:

- multiple response data
- complete multiple response data set

Upon completion of this unit, you should be able to perform the following specific tasks and be able to recognize when they are needed and apply them correctly in the course of a more complex analysis:

- Distinguish between single response data and multiple response data
- Fit non-linear models to single response kinetics data and assess the accuracy of the resulting model
- State that a simple sum of the squares of the errors for all response variables is not, in most cases, the appropriate function to be minimized when fitting a model to multiple response data
- Fit linear or non-linear models to a complete multiple response kinetics data set

Information

At this point in this course, you hopefully have learned how to generate a rate expression using kinetics data from one of the ideal reactor types, with three restrictions. The first restriction is that, if the mole balance design is a differential equation, the differential mole balance design equation can be integrated analytically to obtain an algebraic equation. The second restriction is that the algebraic form of the model equation can be written in the form of a linear equation. Neither of these restrictions is required, they were imposed because most students taking a course like this are already familiar with linear least squares fitting. Thus, by imposing these restrictions, students can focus more on the analysis of the kinetics data and less on the fitting process. Now, having gained some familiarity with kinetics data analysis, these restrictions can be removed. That is to say, this unit can now focus on the analysis of kinetics data when the model equation cannot be linearized, and even when it cannot be integrated analytically.

In general, the analysis of kinetics data involves four kinds of quantities: constants with known values, parameters (unknown constants), experimental set variables (variables whose values are under the direct control of the person doing the experiments) and response variables (variables whose values

are measured once all the experimental settings have been established). When model equations are linearized, x and y are defined to be specific combinations of the set and response variables that result in a linear model equation. Since the model equation is not being linearized in this unit, x will be used to represent set variables; if there are two or more set variables, subscripts will be used to distinguish the different set variables. Similarly, y will be used to represent the response variable. More specifically, $\hat{y}$ will be used to represent the experimentally measured response variable values and y will be used to represent the response variable values predicted by the model.

With linear least squares, it is possible to derive analytical expressions for the direct calculation of the model parameters (slopes and intercept). For ***some*** non-linear models it is also possible to derive analytical expressions for the calculation of the model parameters, but this approach cannot always be used, so it will not be considered here. Instead, this course will use numerical least squares fitting for all situations where linear least squares cannot be used. You can think of numerical least squares fitting as a trial and error method (though in actuality it is more sophisticated than that). A value is guessed (by you) for each parameter in the model. The computer the calculates the sum of the squares of the errors between the measured values of the response variable, $\hat{y}$, and the values that the model predicts for the response variable, y. It then guesses a new value for each parameter and computes the corresponding sum of the squares of the errors. Whichever guess gave a lower sum of the squares of the errors is taken as the better set of parameter values. The computer then repeats this process, over and over, until no better guesses can be found.

The details of numerical least squares fitting are described in Supplemental Unit S4; it is recommended that you read that supplemental unit now. There are fewer software packages that implement least squares numerically, but as with linear least squares, you should feel free to use a software package of your own choosing. If MATLAB is available to you and you elect to use it, then you will find template files for doing so included in Supplemental Unit S4, and at the end of the examples in this unit there will be a description of how those template files can be used to solve the example. No matter what software you choose to use to perform numerical least squares fitting you will need to provide three things:

- a set of guesses, one for each of the unknown parameters that appears in the model
- code that calculates the value of the response variable, y, for a data point, given the value of each set variable, x_i, for that data point along with the value of each model parameter; in other words, code that uses the model equation to calculate y, given the x_i values and the parameter values
- a set of experimental data points, each of which consists of the experimentally measured value of the response variable (y) and corresponding values for each of the set variables (x_i)

There are three situations that one may encounter when using the model equation to calculate y, given the x_i values and the parameter values (second bullet item above). The first is that the model equation ***can be solved explicitly*** for y. (This includes the situation where the model equation is a

differential equation that can be integrated analytically and the resulting integrated model equation can be solved explicitly for the response variable.) In this case the necessary code will simply evaluate the model equation. The second situation is one where the model equation is *not a differential equation*, but nonetheless it *cannot be solved explicitly* for y. In the third situation, the model equation *is a differential equation*, but *either it cannot be integrated analytically* to obtain an algebraic equation *or the integrated model equation cannot be solved explicitly* for the response variable. In both of these latter cases, the model equation will need to be solved numerically, and (no matter what software you choose to use) that will require you to provide things in addition to those listed in the bullets above (see Supplemental Unit S4). Note, also, that in these latter two situations, the model could actually take the form of a set of algebraic equations (second situation) or a set of coupled differential equations (third situation), and not just a single model equation.

It should also be pointed out that in the third situation, an integral data analysis is being performed, but without the need to analytically integrate the differential model equation. In fact, this numerical approach can be used even when it is possible to analytically integrate the differential model equation and to linearize the resulting integrated model equation. However, there is a trade-off if one chooses to always use numerical least squares with numerical solution of the model equation(s). One avoids having to perform the integration and linearization manually, but one introduces the need to provide guesses for the model parameters as well as the possibility of convergence and other issues associated with numerical solutions.

The third restriction that has been imposed prior to this point in the course is that there is only one response variable in the kinetics data set. When only one reaction is taking place, it is only necessary to measure one response variable that is related to the extent of reaction. If that is done, the data are said to be *single response data*, and every problem encountered prior to this unit has involved single response data. However, one *could* measure two response variables, each of which is related to the extent of reaction. In fact, if there are two or more mathematically independent reactions taking place, then one *must* measure one response variable per mathematically independent reaction. In either of these situations the data are said to be *multiple response data*. The experimental data consist of data points where each data point includes the values of the experimental set variables, x_i, for that experiment along with the corresponding experimentally measured values of the response variables, y_i.

Two things change when one needs to analyze multiple response data. The first is that one needs to decide what objective function to minimize. Recall, with single response data the objective function that is minimized is the sum (over all of the data points) of the squares of the differences between the experimentally measured responses and the responses predicted by the model. Clearly, when there is more than one response measured, a different objective function is needed. It turns out that the proper choice of an objective function is not a simple matter, and this causes the second change that is encountered when analyzing multiple response data: the non-linear least squares fitting routines that are found in most mathematics programs only handle single response data. Supplemental Unit S4 also discusses fitting models to multiple response data; if you haven't already done so, you should read it at this time.

Intuitively, one might guess that the best objective function to use with multiple response data is just the sum of the squares of the errors of each of the response variables. For example, if there were two response variables, y_1 and y_2, one might think that equation (16.1) would be the appropriate objective function to minimize. In fact, it can be shown that equation (16.1) is only appropriate under very specific conditions that are very rarely satisfied experimentally [1]. We will not consider how to determine what is the appropriate objective function to use when fitting multiple response data. However, it has been shown [2, 3] that the determinant in equation (16.2) is an appropriate objective function for the analysis of multiple response data when every response has been measured in every data point (i. e. the responses correspond to a full matrix) and the errors can be assumed to be Normally distributed. The latter assumption is more likely to be satisfied experimentally. The errors, ε_{ij}, appearing in equation (16.2) are defined in equation (16.3) where i denotes one of the response variables, j denotes one of the experimental data points and n denotes the number of different response variables. It should be noted that if there is only one response variable, equation (16.2) reduces to the sum of the squares of the errors (i. e. least squares), as would be expected.

$$\Phi = \sum_{\substack{j=\text{ all}\\ \text{data}\\ \text{points}}} \left[\left(y_{1,\text{model}} - y_{1,\text{expt.}} \right)_j^2 + \left(y_{2,\text{model}} - y_{2,\text{expt.}} \right)_j^2 \right] \tag{16.1}$$

$$\Phi = \begin{vmatrix} \sum\limits_{\text{all j}} \left(\varepsilon_{1j}\right)^2 & \sum\limits_{\text{all j}} \varepsilon_{1j}\varepsilon_{2j} & \cdots & \sum\limits_{\text{all j}} \varepsilon_{1j}\varepsilon_{nj} \\ \sum\limits_{\text{all j}} \varepsilon_{1j}\varepsilon_{2j} & \sum\limits_{\text{all j}} \left(\varepsilon_{2j}\right)^2 & \cdots & \sum\limits_{\text{all j}} \varepsilon_{2j}\varepsilon_{nj} \\ \vdots & \vdots & \ddots & \vdots \\ \sum\limits_{\text{all j}} \varepsilon_{1j}\varepsilon_{nj} & \sum\limits_{\text{all j}} \varepsilon_{2j}\varepsilon_{nj} & \cdots & \sum\limits_{\text{all j}} \left(\varepsilon_{nj}\right)^2 \end{vmatrix} \tag{16.2}$$

$$\varepsilon_{ij} = \left(y_{i,\text{model}} - y_{i,\text{expt.}} \right)_j \tag{16.3}$$

As already noted, most mathematics programs only provide numerical *least squares* fitting routines for single response data. However, they typically do provide routines for minimizing a function with respect to a set of variables. Hence, one can usually fit a model to multiple response data by calling one of these minimization routines instead of a nonlinear least squares fitting routine. The minimization routine will need a user-supplied subroutine that provides the function to be minimized. For fitting complete sets of multiple response data, the user-supplied routine should compute the objective function given in equation (16.2). In order to do so, specifically in order to calculate the responses predicted by the model, that user-supplied subroutine will typically need to call additional routines to solve either a set of nonlinear algebraic equations or a set of coupled ordinary differential equations.

Unfortunately, the general minimization routines provided by mathematics programs will not compute a correlation coefficient or uncertainties in the set of parameters that minimize the objective

function. These quantities can be computed separately, but it is beyond the scope of this course to consider how to do so. A good statistics book or course is recommended.

References Cited

1. W. G. Hunter, *Ind. Eng. Chem. Fundamentals* **6**(3), 461 (1967).
2. G. E. P. Box and N. R. Draper, *Biometrika* **52**, 355 (1965).
3. W. E. Stewart, M. Caracotsios and J. P. Sørensen *AIChE J*. **38**(5), 641 (1992).