

**THE ESTIMATION PROCEEDURE**

We have seen in Parameter Estimation how to compare the predicted concentration of a reactant or product against experimental data. This was illustrated with a batch reactor example involving a single reaction. In solving this problem, a rate expression was guessed, an analytical solution for the concentration of one of the components was sought

$$c_j = f(t, k, V_r, \text{ reaction order, initial conditions, stoichiometry}) \quad (1)$$

and the objective function

$$\Phi = \sum_{i=1}^n [c_j(t_i) - \hat{c}_j(t_i)]^2 \quad (2)$$

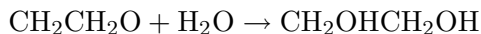
was minimized using a nonlinear optimization routine. A Matlab program `parameter_estimation.m` did this by starting with a guessed value for the rate constant and used this with Equation 1 to predict the concentration of HCN. These values were compared against the experimental values to minimize the objective function defined by Equation 2 using the Matlab function `fminsearch`.

This method will not work for multiple reactions since, in general, you cannot integrate the differential equations analytically. A more general approach is required. This (more general) approach uses the defining design equations, varies the unknown model parameters, numerically integrates the equations, and again minimizes an objective function.

To illustrate the latter approach, both are employed to solve a different reaction, the hydrolysis of ethylene oxide. Two programs are used in the solution. The first, `parameter_1_est.m`, uses Equations 1 and 2 to solve the problem. The second, `parameter_2_est.m`, works with the differential equations that describe how concentration changes with time in a batch reactor.

**EXAMPLE**

The hydrolysis of ethylene oxide in the presence of  $\text{H}_2\text{SO}_4$  to ethylene glycol has been studied at  $55^\circ\text{C}$  by mixing 500 ml of a 2M solution of ethylene oxide in water with 500 ml of water containing 0.9 wt%  $\text{H}_2\text{SO}_4$  in a batch reactor. The initial concentration of ethylene oxide is 1 gmole/liter and the initial concentration of water is 52.8 gmoles/liter. ( $\text{H}_2\text{SO}_4$  is a catalyst and is not consumed during the reaction.)



The concentration versus time data are shown in the table and you are to use these data to determine the reaction order for ethylene oxide and the value of the rate constant.

Time (min)	Ethylene Glycol Concentration (gmole/liter)
0.0	0.000
0.5	0.155
1.0	0.279
1.5	0.366
2.0	0.467
3.0	0.619
4.0	0.706
6.0	0.856
10.0	0.962

We will assume the reaction is  $n$ th-order in ethylene oxide (EO) since the water is present in excess and its concentration does not change appreciably.

$$r = kc_{EO}^n$$

At this point you would need to make a guess for the value of  $n$  and it is reasonable to guess  $n = 1$  or  $n = 2$ . Note that each guessed value for  $n$  requires you to solve the material balance to generate the appropriate Equation 1

The program `parameter_1_est.m` employs the integrated forms for a first and second order reaction, *i.e.*

$$c_{EO} = c_{EO,o} \exp(-kt) \text{ for the first order form of Equation 1}$$

$$c_{EO} = \frac{1}{\left[kt + \frac{1}{c_{EO,o}}\right]} \text{ for the second order form of Equation 1}$$

Run the program and notice how well the data fit the two expressions and how different the residuals are for each case.

The second approach in program `parameter_2_est.m` employs two functions. One lists the design equations to be integrated with  $k$  and  $n$  being unknown. This is function `raten`. The second function `f2` contains the objective function  $\Phi$  in which the values of concentration are found by calling `raten` and are then compared against the experimental data. The program contains two more “second functions”, `f3` and `f4`, in which the reaction order is fixed as first and second order, respectively. This will permit a comparison of the results from letting  $n$  be variable versus fixed. Notice that when `fminsearch` is asked to solve `f2` it uses the initial guesses for  $k$  and  $n$  and integrates the design equations the first time. `fminsearch` then varies  $k$  and  $n$  until the objective function is minimized. There are issues of sufficient data and efficient searching that enter into a fitting procedure such as this. If the initial guesses are too far off, the optimization routine wanders away from a productive region and no solution is found.

After optimum parameters are found the program integrates the design equations with these values so the plot of  $c_{EO}$  versus time can be generated.

You should study the program and run it to see how your initial guess for  $k$  and  $n$  affects the time to reach an optimum fit.