A First Course on Kinetics and Reaction Engineering

Example 22.3

Problem Purpose

This problem illustrates a reactor design problem involving an isothermal, steady state CSTR.

Problem Statement

A certain kind of living cell grows according to the Monod equation as given in equation (1). The concentrations in equation (1) are mass concentrations. At the final conditions specified here (for which equation (1) is accurate), 2.2 grams of substrate are consumed for every one gram of cells that is produced. An isothermal CSTR is needed to process a liquid feed stream flowing at 60 cm^3 min^{-1} and containing 0.04 g cm^{-3} of substrate and 8.0 x 10^{-5} g cm^{-3} of cells at steady state. What reaction volume is required to increase the cell mass to one hundred times the feed value?

$$r=\frac{\left(0.014\text{ min}^{-1}\right)\left[S\right]\left[X\right]}{\left(0.001\text{ g cm}^{-3}\right)+\left[S\right]} \tag{1}$$

Problem Analysis

In this problem, a reaction is taking place in a CSTR and the kinetics of the reaction are known. Therefore it is a CSTR reaction engineering problem. We are asked to size the reactor, making it a reactor design problem. The problem specifies that the reactor is operating isothermally at steady state. To model the system, a mass balance will be written for each reactant and product. These equations can be solved independently of the energy balance, and, in this case, there isn't sufficient information to write an energy balance. Thus the mole balances represent a complete set of design equations for modeling the system.

Problem Solution

Letting S represent the substrate and X, the cells, and noting the effective stoichiometry given in the problem statement, the reaction can be written as equation (2).

$$2.2\text{ S}\rightarrow\text{X} \tag{2}$$

Figure 1 shows a schematic representation of the reactor. The quantities provided in the problem statement are given in a consistent set of units, and they have been entered in the diagram. The reacting fluid is stated to be a liquid, and it will be assumed that the liquid density is constant. With this assumption, the outlet volumetric flow rate will equal the inlet volumetric flow rate as indicated in the schematic diagram.

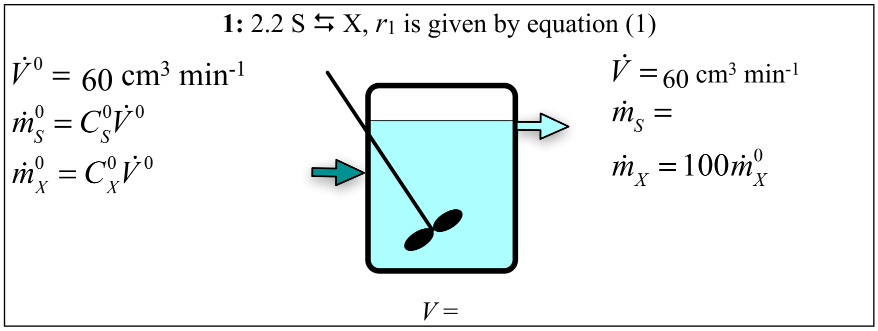
1: 2.2 S ⇆ X, r_1 is given by equation (1)
$\dot{V}^0$ = 60 cm^3 min^{-1}
$\dot{m}_S^0 = C_S^0\dot{V}^0$
$\dot{m}_X^0 = C_X^0\dot{V}^0$
$\dot{V}$ = 60 cm^3 min^{-1}
$\dot{m}_S =$
$\dot{m}_X = 100\dot{m}_X^0$
$V =$

Figure 1. Schematic representation of the CSTR.

Mass balances can be written for each species present in the system. The generalized, steady state mass balance equation is given by equation (3), where the rate is given in mass units. In writing the balance equations, I'll use the equation numbering in Figure 1 (i. e. r_1) and not the numbering in the problem statement. Writing equation (3) once for the substrate and a second time for the cell mass leads to equations (4) and (5).

$$0 = \dot{m}_i^0 - \dot{m}_i + V \sum_{\substack{j=all\\reactions}} \nu_{i,j} r_j = \dot{m}_i^0 - \dot{m}_i + V\nu_{i,1} r_1 \tag{3}$$

$$0 = \dot{m}_S^0 - \dot{m}_S - 2.2Vr_1 \tag{4}$$

$$0 = \dot{m}_X^0 - \dot{m}_X + Vr_1 \tag{5}$$

This reactor is isothermal. As noted in the problem analysis, the mass balances can be solved independently of the energy balance equation. This is true because the rate coefficients are constants when the reactor is isothermal. Furthermore, in this particular problem, the question can be answered without using the energy balance, so it won't be written.

In order for the design equations (4) and (5) to be solved, the number of unknowns must equal the number of design equations, in this case two. Figure 1 indicates that the outlet mass flow of substrate and the reaction volume are unknown, so they will be chosen as the unknown variables. Thus, the design equations have the form shown in equations (6) and (7).

$$0 = f_1\left(\dot{m}_S, V\right) = \dot{m}_S^0 - \dot{m}_S - 2.2Vr_1 \tag{6}$$

$$0 = f_2\left(\dot{m}_S, V\right) = \dot{m}_X^0 - \dot{m}_X + Vr_1 \tag{7}$$

Equations (6) and (7) do not contain integrals or derivatives. As such, one could attempt to solve them manually, using symbolic algebra software or using numerical methods software. A variety of software packages are available for doing so, and you should use the one you feel most comfortable with. Supplemental Unit S2 presents a brief introduction to the numerical solution of sets of non-differential equations like these using numerical methods software. No matter what software package you use, you typically will need to provide two things as input in order to solve the equations:

- Code that evaluates the functions, f_1 and f_2 in equations (6) and (7), given values of the unknown variables, $\dot{m}_S$ and V
- A guess for the solution, that is, a guess for the values of $\dot{m}_S$ and V that cause the functions, f_1 and f_2, to equal zero

The functions to be evaluated contain quantities other than the four unknown variables, $\dot{m}_S$ and V. In order to evaluate the functions, values will be needed for each of those other quantities. Here the inlet mass flow rates ($\dot{m}_S^0$, $\dot{m}_X^0$) and the outlet mass flow of cells ($\dot{m}_X$) are constants whose values are given in the problem statement or calculated directly from information given in the problem statement. The rate of the reaction (r_1) is a variable quantity whose, and its value will need to be computed in order to evaluate the functions, f_1 and f_2.

The rate can be computed using equation (1), which introduces the mass concentrations of S and M. Those concentrations can be calculated by expressing them in terms of the mass flow rates and the volumetric flow rate as in equations (8) and (9). The code that evaluates the functions, f_1 and f_2, will be given $\dot{m}_S$, while $\dot{m}_X$ and the volumetric flow rate are known constants.

$$\left[S\right]=\frac{\dot{m}_S}{\dot{V}} \tag{8}$$

$$\left[X\right]=\frac{\dot{m}_X}{\dot{V}} \tag{9}$$

That is everything that is needed in order to evaluate the functions, f_1 and f_2. The only other thing that will be needed in order to solve the equations numerically is a guess for the values of the unknowns. Since all that is needed is a guess, one possibility is to use the inlet substrate mass flow rate and a reaction volume of 100 cm^3. With that, equations (6) and (7) can be solved numerically to find $\dot{m}_S$ and V. Upon doing so one finds the outlet mass flow of substrate to equal 1.35 g min^{-1} and the necessary reaction volume to be 4430 cm^3.

Calculation Details Using MATLAB

Supplemental Unit S2 describes how to solve sets of non-differential equations numerically using MATLAB, and it provides a template file named SolvNonDif.m for doing so. Before it can be used to solve a problem, that template file must be modified in four places, each indicated by a comment that begins "% EDIT HERE". In addition to those required modifications I made a few additional modifications that will be described here along with the required modifications.

I recommend that you work with a copy of the file that has been given a more meaningful name; I used Example_22_3.m. Since the function name must match the filename, I changed the name of the function to Example_22_3. At the same time, knowing that I won't need to use the results from these calculations in subsequent calculations, I changed the function so that it does not return any values. The template file begins with a long set of comments describing what it does and how to use it; I replaced these comments with a brief comment stating the purpose of the modified version. None of these modifications were required. The first *__required__* modification is to enter the values of all universal and problem specific constants at the point indicated. At the same time these are entered, they should be converted to a consistent set of units. At this point I also calculated $\dot{m}_S{}^0$, $\dot{m}_X{}^0$ and $\dot{m}_X$, since they, too, are constants. Listing 1 shows the section of Example_22_3.m where these modifications were made.

```
% Modified version of the MATLAB template file SolvNonDif.m used in the
% solution of Example 22.3 of "A First Course on Kinetics and Reaction
% Engineering."
%
function Example_22_3
    % Known quantities and constants (in consistent units)
    VFR = 60; % cm3/min
    CS0 = 0.04; % g/cm3
    CX0 = 8e-5; % g/cm3
    mS0 = VFR*CS0; % g/min
    mX0 = VFR*CX0; % g/min
    mX = 100*mX0; % g/min
```

Listing 1. Portion of the modified template file SolvNonDif.m where problem-specific and universal constants were entered in consistent units.

The second *__required__* modification involves entering the code to evaluate functions, f_1 and f_2, equations (6) and (7). In the code, this occurs within an internal function named evalEqns; within evalEqns, both the unknowns and the equations are provided as vector quantities named `z` and `f`, respectively. Thus, it is necessary to map the variables used in the problem solution to represent the unknowns ($\dot{m}_S$ and V) to a vector `z`, and to return the values of the functions in the vector `f`. I find it useful at the start of the internal function that will evaluate the functions, to define local variables with the names used in the problem statement. This modification is not required, but in my opinion, it makes the code more readable and easier to debug. In addition, the list of variables here serves as a reminder of the mapping of the problem statement variables to the vector `z`.

Recall from the solution that the functions contained variable quantities (r_1, [S] and [X]) that depend upon the unknowns. In the code being written here to evaluate the functions, we are given values of the unknowns as just described. Therefore, these variable quantities can be evaluated here using the equations given in the problem statement; specifically equations (1), (8) and (9). The constants from the problem statement have already been entered and are available at this point in the MATLAB file, so the functions, f_1 and f_2, can next be evaluated. The code containing all these modifications is shown in Listing 2.

```
% Function that evaluates the equations
   function f = evalEqns(z)
    % mapping of unknowns into vector z
    mS = z(1);
    V = z(2);
    % calculate variables quantities
    CS = mS/VFR;
    CX = mX/VFR;
    r2 = 0.014*CS*CX/(0.001 + CS);
    % evaluate the functions
    f = [
        mS0 - mS - 2.2*V*r2
        mX0 - mX + V*r2
    ];
end % of internal function evalEqns
```

Listing 2. Portion of the modified template file SolvNonDif.m showing the entry of the code to evaluate the functions being solved.

The third *required* modification is where guesses for the unknowns are provided. The guesses are entered in the array named z_guess. They must be entered using the same mapping of the unknowns to the vector z as was used above. The previous modification, where variables with more meaningful names were defined, serves as a key to remind you which variable is z_1, which is z_2, and so on. As noted in the problem solution, one possibility for a guess is to simply use the inlet substrate mass flow rate and a reaction volume of 100 cm^3, and that is what is done in Listing 3.

```
% guesses for the solution
z_guess = [
    mS0
    100.0
];
```

Listing 3. Portion of the modified template file SolvNonDif.m where guesses are provided for the unknowns.

The final *required* modification only applies if you need to use the results from solving the set of equations to calculate other quantities. In this case, the answer we seek (V) is found directly by solving the design equations. Nonetheless, the code shown in Listing 4 was added to make it easier to identify the results.

```
% Report the results
substrate_mass_flow = z(1)
reaction_volume = z(2)
```

Listing 4. Final modification to the template file SolvNonDif.m.

At this point, the modified template file can be used to solve the equations. To do so, the function name is typed at the MATLAB command prompt. Numerical methods for solving non-differential equations don't always converge to an acceptably accurate final answer (see Supplemental Unit S2 for more reasons why and what to do if this happens). Therefore, you should always examine the output from

MATLAB closely; it should indicate that the equations were solved. If it doesn't, then you need to determine why a solution was not obtained. In addition, you should check the values of the functions that are calculated using the final solution and printed. These should all be close to zero if the solution is accurate. In the present problem, an accurate solution was found as can be seen in Listing 5.

```
>> Example_22_3

Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>

The solver found the following values for the unknowns:

z =
   1.0e+03 *

    0.0014
    4.4308

The corresponding values of the functions being solved are as follows:

f =
   1.0e-10 *

    0.3083
    0.0237

substrate_mass_flow =
    1.3546

reaction_volume =
   4.4308e+03
```

Listing 5. Output from the execution of Example_22_3.m.