

A First Course on Kinetics and Reaction Engineering

Example 26.2

Problem Purpose

This problem will help you determine whether you have mastered the learning objectives for this unit. It illustrates the analysis of a system where more than one reaction is taking place. It also shows one situation where the standard problem solving approach provided in the reading needs to be modified slightly.

Problem Statement

An adiabatic plug flow reactor with a diameter of 10 cm and a length of 5 m will be fed an aqueous solution containing A and B at 1.0 and 1.2 M concentrations, respectively. The feed will be supplied at 30 °C. Series-parallel reactions (1) and (2) below will take place; with the rates given in equations (3) and (4). The heat of reaction (1) is $-11,300 \text{ cal mol}^{-1}$ and that of reaction (2) is $-11,700 \text{ cal mol}^{-1}$; both heats of reaction may be assumed to be constant. The heat capacity of the solution and the density of the solution may be taken to be constant and equal to those of water ($1.0 \text{ cal g}^{-1} \text{ K}^{-1}$ and 1.0 g cm^{-3}). What volumetric flow rate will maximize the outlet concentration of the desired product D? (Pressure drop in the reactor may be neglected.)



$$r_1 = \left(8.93 \times 10^2 \text{ L mol}^{-1} \text{ min}^{-1}\right) \exp\left\{\frac{-6.7 \text{ kcal mol}^{-1}}{RT}\right\} C_A C_B \quad (3)$$

$$r_2 = \left(9.24 \times 10^2 \text{ L mol}^{-1} \text{ min}^{-1}\right) \exp\left\{\frac{-5.1 \text{ kcal mol}^{-1}}{RT}\right\} C_A C_D \quad (4)$$

Problem Analysis

The reactor in this problem is a steady state PFR. The rate expression is known, so this is not a kinetics problem. To answer the reaction engineering questions, the mole and energy balance design equations will be solved and used to calculate the outlet concentration of D. A mechanical energy (momentum) balance will not be included in the design equations because the problem states that pressure drop may be neglected. The equations will be solved repeatedly using a range of values of the inlet volumetric flow rate, and the results will be plotted. The flow rate corresponding to the maximum concentration of D can then be determined by examining the plot.

Problem Solution

The following quantities are provided in the problem statement or can be inferred: $D = 10$ cm, $L = 5$ m, $C_A^0 = 1.0$ mol L⁻¹, $C_B^0 = 1.2$ mol L⁻¹, $\dot{n}_D^0 = 0$, $\dot{n}_S^0 = 0$, $\dot{n}_U^0 = 0$, $T^0 = 30$ °C = 303 K, $k_{01} = 8.93 \times 10^2$ L mol⁻¹ min⁻¹, $E_1 = 6.7$ kcal mol⁻¹, $k_{02} = 9.24 \times 10^2$ L mol⁻¹ min⁻¹, $E_2 = 5.1$ kcal mol⁻¹, $\Delta H_1(T) = -11300$ cal mol⁻¹, $\Delta H_2(T) = -11700$ cal mol⁻¹, $\tilde{C}_p = 1.0$ cal g⁻¹ K⁻¹ and $\rho = 1.0$ g cm⁻³. Since the problem states that the heats of the reactions are constant, the values listed here apply at any temperature. The values listed here do not have consistent units (e. g. cm, m and L are used as measures of length, both cal and kcal are used for energy, etc.), so they should be converted to consistent units before using them in the design equations. The system can be represented schematically as shown in Figure 1.

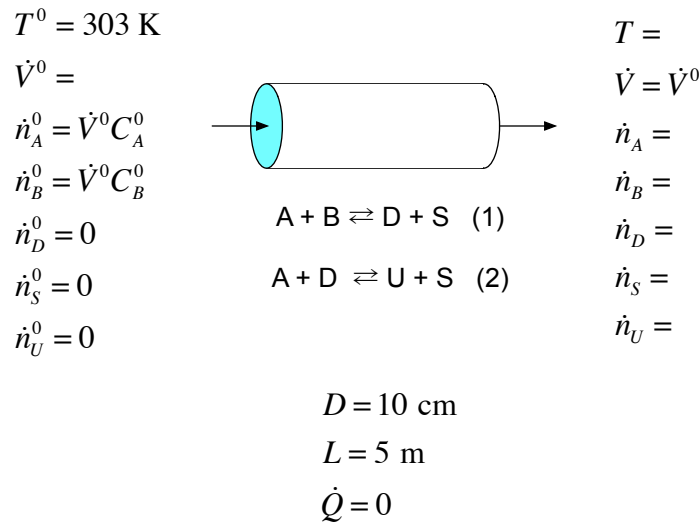


Figure 1. Schematic representation of the reactor.

The generalized, steady-state mole balance equation for a PFR is given in equation (5). It can be used to generate the mole balances for each of the reactants and products present in the system as given in equations (6) through (10), where the equations are written in the preferred form with a single derivative on the left of the equals sign and a function of the independent and dependent variables on the right side of the equals sign. Since there are two reactions taking place, the summation expands to give two terms unless one of the stoichiometric coefficients is zero because that species does not participate in one of the reactions.

$$\frac{d\dot{n}_i}{dz} = \frac{\pi D^2}{4} \left(\sum_{j=1}^{N_{\text{rxns}}} \nu_{i,j} r_j \right) \quad (5)$$

$$\frac{d\dot{n}_A}{dz} = f_1(z, \dot{n}_A, \dot{n}_B, \dot{n}_D, \dot{n}_S, \dot{n}_U, T) = \frac{\pi D^2}{4} (-r_1 - r_2) \quad (6)$$

$$\frac{d\dot{n}_B}{dz} = f_2(z, \dot{n}_A, \dot{n}_B, \dot{n}_D, \dot{n}_S, \dot{n}_U, T) = -\frac{\pi D^2}{4} r_1 \quad (7)$$

$$\frac{d\dot{n}_D}{dz} = f_3(z, \dot{n}_A, \dot{n}_B, \dot{n}_D, \dot{n}_S, \dot{n}_U, T) = \frac{\pi D^2}{4} (r_1 - r_2) \quad (8)$$

$$\frac{d\dot{n}_S}{dz} = f_4(z, \dot{n}_A, \dot{n}_B, \dot{n}_D, \dot{n}_S, \dot{n}_U, T) = \frac{\pi D^2}{4} (r_1 + r_2) \quad (9)$$

$$\frac{d\dot{n}_U}{dz} = f_5(z, \dot{n}_A, \dot{n}_B, \dot{n}_D, \dot{n}_S, \dot{n}_U, T) = \frac{\pi D^2}{4} r_2 \quad (10)$$

The general steady-state energy balance equation for a PFR is given in equation (11). In simplifying equation (11) the summation in the denominator was replaced by the heat capacity of the solution as a whole, the heat transfer term in the numerator was eliminated since the reactor is adiabatic, and the summation in the numerator was expanded over the two reactions taking place. This leads to equation (12) which is again written in the preferred form. Since pressure drop can be neglected, a mechanical energy (momentum) balance is not needed because the pressure drop is specified to be insignificant.

$$\frac{dT}{dz} = \frac{\pi D U (T_e - T) - \frac{\pi D^2}{4} \left(\sum_{j=1}^{N_{\text{rxn}}} r_j \Delta H_j(T) \right)}{\left(\sum_{i=1}^{N_{\text{species}}} \dot{n}_i \hat{C}_{p,i} \right)} \quad (11)$$

$$\frac{dT}{dz} = f_6(z, \dot{n}_A, \dot{n}_B, \dot{n}_D, \dot{n}_S, \dot{n}_U, T) = \frac{-\frac{\pi D^2}{4} (r_1 \Delta H_1^0(T) + r_2 \Delta H_2^0(T))}{\dot{V} \rho \tilde{C}_p} \quad (12)$$

Design equations (6) through (10) and (12) represent a set of 6 ordinary differential equations (ODEs) with 6 dependent variables (\dot{n}_A , \dot{n}_B , \dot{n}_D , \dot{n}_S , \dot{n}_U and T) and one independent variable (z). They are initial value ODEs because the value of each of the 6 dependent variables is known at the same value ($z_0 = 0$) of the independent variable. A set of n initial value ODEs containing n dependent variables and one independent variable can be solved numerically to obtain the values of all of the variables at some other point of interest. If you are not familiar with the numerical solution of initial value ODEs, Supplemental Unit S5 of “A First Course on Kinetics and Reaction Engineering” offers a brief introduction to the topic and provides MATLAB template files named SolvIVDifI.m and SolvIVDifD.m that can be used as a starting point for solving equations of this kind if one elects to use MATLAB for that purpose.

Generally, no matter what software package is used, the numerical solution of these equations will require you to provide (a) the initial values (z_0 , \dot{n}_A^0 , \dot{n}_B^0 , \dot{n}_D^0 , \dot{n}_S^0 , \dot{n}_U^0 and T^0), (b) the final value of either the independent variable or one of the dependent variables at the point of interest and (c) code that, given

values for the independent and dependent variables, evaluates the functions f_1 through f_6 in equations (6) through (10) and (12).

As described in the problem analysis above, we will choose a value for the inlet volumetric flow rate, solve the design equations, calculate the outlet concentration of D, plot that concentration versus the chosen inlet volumetric flow rate and repeat. Noting that the inlet concentrations of A and B are given (all other inlet concentrations are presumed to equal zero), the inlet molar flow rates can be calculated as indicated in Figure 1. This gives the initial values (\dot{n}_A^0 , \dot{n}_B^0 , \dot{n}_D^0 , \dot{n}_S^0 , \dot{n}_U^0 and T^0 at $z = z_0 = 0$) needed to solve the design equations. The final value is that the reactor is 5 m long, that is, $z = z_f = L = 5$ m.

Looking at the functions f_1 through f_6 , the only quantities that have not already been specified are r_1 , r_2 and \dot{V} , but since the liquid density is constant, \dot{V} will equal the inlet volumetric flow rate, \dot{V}^0 . The code that will evaluate functions f_1 through f_6 will be given values of z , \dot{n}_A , \dot{n}_B , \dot{n}_D , \dot{n}_S , \dot{n}_U and T , so all it will need to calculate are the two rates. These can be computed using equations (3) and (4), but in order to do so, concentrations are needed. By definition, these can be computed using equations (13) through (15). It is important to remember that in a flow system, the concentration is the molar flow rate divided by the volumetric flow rate, not the molar flow rate divided by the volume.

$$C_A = \frac{\dot{n}_A}{\dot{V}} \quad (13)$$

$$C_B = \frac{\dot{n}_B}{\dot{V}} \quad (14)$$

$$C_D = \frac{\dot{n}_D}{\dot{V}} \quad (15)$$

This gives all the information needed to solve the design equations, so at this point we can proceed as follows:

- Choose a volumetric flow rate
- Solve the design equations, obtaining outlet values of \dot{n}_A , \dot{n}_B , \dot{n}_D , \dot{n}_S , \dot{n}_U and T
- Calculate the corresponding outlet molar concentration of D using equation (15)
- Repeat over a range of volumetric flow rates, plotting the outlet concentration of D as a function of the volumetric flow rate
- Identify the volumetric flow rate that maximizes the outlet concentration of D

The resulting plot of the outlet concentration of D versus volumetric flow rate is shown in Figure 2. It can be seen that a volumetric flow rate of $2100 \text{ cm}^3 \text{ min}^{-1}$ maximizes the final concentration of D at $6.85 \times 10^{-5} \text{ mol cm}^{-3}$.

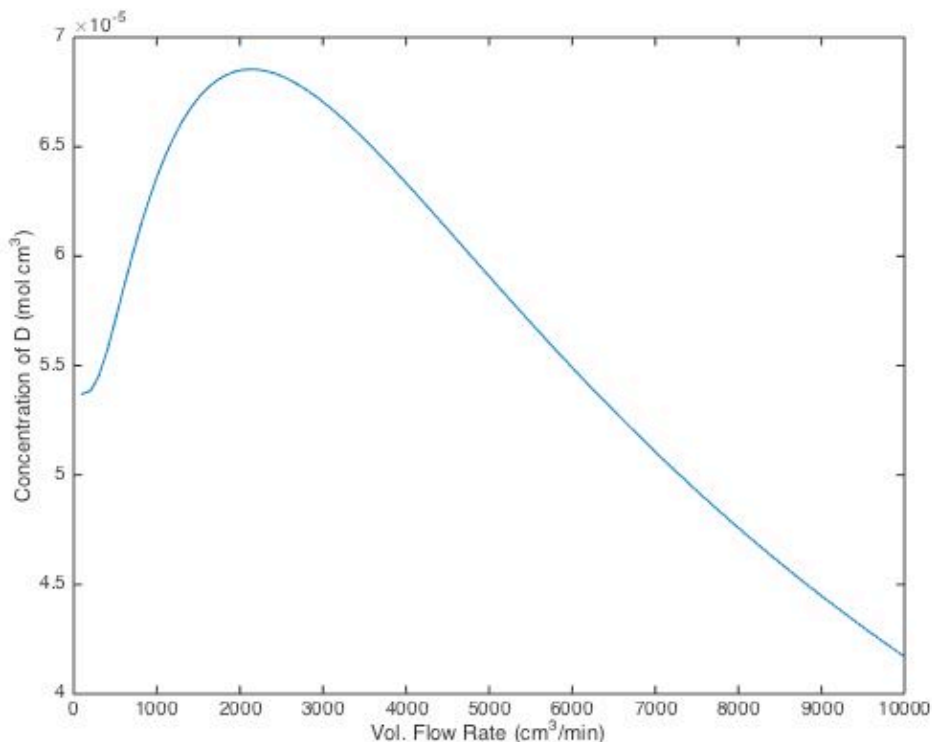


Figure 2. Outlet concentration of D as a function of volumetric flow rate.

Calculation Details Using MATLAB

If you elect to use MATLAB to solve the design equations, Supplemental Unit S5 provides template files that can be used. In this problem, the equations are initial value ODEs and the final value of the independent variable is provided, so the appropriate template file is SolvIVDifl.m. Before that file can be used, you must make four required modifications. Here I will also describe a few non-required modifications that you might want to consider when solving problems of this type.

To begin, I made a copy of the template file and saved it as Example_26_2.m; a copy of that file accompanies this solution. Since the function name must match the filename, I changed the name of the function to Example_26_2. 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 involves entering all the known quantities from the problem statement along with universal constants that will be needed (from handbooks or other reference sources). As these were entered, they were converted to a consistent set of units. The result of making all these modifications is shown in Listing 1.

```

% Modified version of the MATLAB template file SolvIVDifI.m used in the
% solution of Example 26.2 of "A First Course on Kinetics and Reaction
% Engineering."
%
function Example_26_2
    % Known quantities and constants
    D = 10; % cm
    L = 500; % cm
    CA0 = 1.0/1000.0; % mol/cm^3
    CB0 = 1.2/1000.0; % mol/cm^3
    nD0 = 0;
    nS0 = 0;
    nU0 = 0;
    T0 = 30 + 273.15; % K
    k01 = 8.93E2*1000.; % cm^3/(mol min)
    E1 = 6700; % cal/mol
    k02 = 9.24E2*1000.; % cm^3/(mol min)
    E2 = 5100; % cal/mol
    dH1 = -11300; % cal/mol
    dH2 = -11700; % cal/mol
    Cp = 1.0; % cal/(g K)
    rho = 1.0; % g/cm^3
    % Universal Constants
    R = 1.987; % cal/(mol K)

```

Listing 1. Initial comment, function declaration and known constants after modification of SolvIVDifI.m

I wanted to solve the set of ODEs many times for a range of different values of the volumetric flow rate. Therefore, immediately following the code shown in Listing 1, I set up storage for 100 different volumetric flow rates and the corresponding 100 outlet concentrations of D. Since I'm looking for the maximum concentration of D, I created a variable to store it and a variable to store the corresponding volumetric flow rate. I then created a for loop that would repeat the calculations for each of the 100 volumetric flow rates. This is shown in Listing 2. The internal function that evaluates the ODEs (odeqns) cannot be located within this for loop, so I moved it to the very end of the file (this change is not shown here, but can be seen in full function file, Example_26_2.m, which accompanies this solution.

```

% Create 100 element vectors to hold the data for later plotting and
% set the range of values of VFR to be used
VFR = linspace(0,1.0E4);
concD = zeros(100,1);
% Create variables to hold the maximum CD and corresponding VFR
VFR_at_max = 0;
concD_max = 0;
% Solve the design equations for the chosen range of VFRs
for i = 1:100

```

Listing 2. Code for looping over a range of inlet volumetric flow rates.

There are two additional quantities that are constant each time the design equations are solved. These are the inlet molar flow rates of A and B. They couldn't be entered along with all the previous constants because they depend upon the value of the volumetric flow rate. Within the for loop that began in Listing 2, the volumetric flow rate is constant, so the first thing I do within that loop is to calculate these quantities. The code added to do so is shown in Listing 3.

```
% Additional Calculated Constants
nA0 = VFR(i) * CA0;
nB0 = VFR(i) * CB0;
```

Listing 3. Code within the for loop that calculates the inlet molar flow rates of A and B.

The second *required* modification involves entering code that, given values for the independent and dependent variables, evaluates the functions f_1 through f_6 in equations (6) through (10) and (12). This takes place within an internal function named `odeeqns`, and as noted earlier, that internal function cannot be inside the for loop, so I moved it to the end of `Example_26_2` before modifying it. Notice that within that internal function, the equations are provided as a vector quantity. Thus, it is necessary to map the dependent variables used in the problem statement (\dot{n}_A , \dot{n}_B , \dot{n}_D , \dot{n}_S , \dot{n}_U and T) to a vector z , and the corresponding derivatives are mapped to a vector $dzdt$. (Don't be confused: in the template file z represents the vector of dependent variables and t represents the independent variable whereas in our problem, z is the variable used for the independent variable. In other words, the independent variable, z , used in the problem statement maps to the variable `t` in the MATLAB code.) I find it useful at the start of the internal function that will evaluate the derivatives, 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 resulting list of variables serves as a reminder of the mapping of the problem statement variables to the vector z . The required code first calculates the three concentrations that appear in the rate expressions according to equations (13) through (15). Following that, the rates are calculated using equations (3) and (4). Finally, the functions f_1 through f_6 are evaluated using equations (6) through (10) and (12), saving the results in the vector $dzdt$. The resulting code is shown in Listing 3.

```

% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    nA = z(1);
    nB = z(2);
    nD = z(3);
    nS = z(4);
    nU = z(5);
    T = z(6);
    CA = nA/VFR(i);
    CB = nB/VFR(i);
    CD = nD/VFR(i);
    r1 = k01*exp(-E1/R/T)*CA*CB;
    r2 = k02*exp(-E2/R/T)*CA*CD;
    dzdt = [
        pi()*D^2/4*(-r1-r2)
        pi()*D^2/4*(-r1)
        pi()*D^2/4*(r1-r2)
        pi()*D^2/4*(r1+r2)
        pi()*D^2/4*(r2)
        -pi()*D^2/4*(r1*dH1+r2*dH2)/VFR(i)/rho/Cp
    ];
end % of internal function odeqns

```

Listing 3. Results of the second required modification.

The third required modification involves providing the initial values of the independent and dependent variables and the final value of the independent variable. Recall that the independent variable, z , from the problem statement maps to the variable t in the code. Thus t_0 in the code is set equal to the initial value of the axial position, which is denoted as $z_0 = 0$ in the problem statement. The initial values of the dependent variables are entered as a vector named z_0 , and they must use the same mapping of the problem variables (\dot{n}_A^0 , \dot{n}_B^0 , \dot{n}_D^0 , \dot{n}_S^0 , \dot{n}_U^0 and T^0) to the vector z_0 as was used previously to map (\dot{n}_A , \dot{n}_B , \dot{n}_D , \dot{n}_S , \dot{n}_U and T) to z . As with the initial value, the final value, z_f in the problem statement, maps to t_f in the code. This code appears at the beginning of the for loop and is shown in Listing 4.

```

% Initial and final values
t0 = 0;
z0 = [
    nA0
    nB0
    nD0
    nS0
    nU0
    T0
];
tf = L;

```

Listing 4. Results of the third required modification.

The fourth and final *required* modification is to use the results from solving the ODEs to calculate whatever the problem requested. In this case, once the design equations have been solved, the concentration of D is calculated and saved for later plotting. Then, if it is a new maximum, that value and the corresponding volumetric flow rate are saved. Once the looping has finished and the design equations have been solved for all of the values of the volumetric flow rate, the plot is generated and the maximum concentration of D is reported along with the corresponding volumetric flow rate as shown in Listing 5.

```

% Calculate the concentration of D
concd(i) = z(3)./VFR(i);
if (concd(i) > concD_max)
    concD_max = concd(i);
    VFR_at_max = VFR(i);
end
end % of loop through range of values of VFR

% Generate the plot
figure
plot(VFR,concd)
xlabel('Vol. Flow Rate (cm^3/min)')
ylabel('Concentration of D (mol cm^3)')

% Report the VFR and CD_max
concd_max
VFR_at_max

```

Listing 5. Results of the fourth required modification.

Once the file containing all the modifications had been saved, it was executed by typing Example_26_2 at the MATLAB command prompt. Doing so generated Figure 2 and the output shown in Listing 6.

```

>> Example_26_2

concd_max =
    6.8543e-05

VFR_at_max =
    2.1212e+03

```

Listing 6. Output produced upon execution of the modified file.