A First Course on Kinetics and Reaction Engineering

Example 30.2

Problem Purpose

This problem will help you determine whether you have mastered the learning objectives for this unit. It illustrates the general approach to solving the design equations for a PFR with feed pre-heating by the effluent stream.

Problem Statement

An adiabatic PFR with a volume of 3 m^3 will be used to process 1.25 mol s^{-1} of an equimolar gas phase mixture of A and B at 300 K and 2.5 atm. This feed will be pre-heated using the product stream from the reactor. The heat of reaction is constant and equal to -9,800 cal mol^{-1}. The heat capacity of the gas is constant and equal to 25.8 cal mol^{-1} K^{-1}. Pressure drop in the reactor is negligible. Heat transfer in the heat exchanger can be described using an arithmetic mean temperature difference with AU_{AM} = 13.6 cal K^{-1} s^{-1}. What will the final conversion equal if reaction (1) takes place with a rate given by equation (2)?

$$\text{A + B} \rightarrow \text{Y + Z} \tag{1}$$

$$r_1 = \left(2.83\times 10^4\ \text{s}^{-1}\right)\exp\left\{\frac{-12900\ \text{cal mol}^{-1}}{RT}\right\}C_A \tag{2}$$

Problem Analysis

This is a reaction engineering problem, not a kinetics problem, because we are given the rate expression along with other information about the operation of a reactor and asked to find reactor properties. The reactor is a PFR, and since nothing in the problem statement indicates transient behavior, we can model the reactor using the steady state design equations. The reactor volume is given, not its diameter and length, so we will need to use the cumulative reactor volume as the independent variable in the design equations. The problem specifies that there is no pressure drop, so a momentum balance will not be needed. The reactor inlet and outlet are integrated with a heat exchanger, so we will most likely need to write design equations for the heat exchanger, too.

Problem Solution

A schematic representation of the integrated heat exchanger and PFR is presented in Figure 1. The four flow streams have been designated as a through d. Reading through the problem statement, one finds that the following quantities are specified: V = 3 m^3, $\dot{n}_{total,a}$ = 1.25 mol s^{-1}, $y_{A,a}$ = 0.5, $y_{B,a}$ = 0.5, T_a = 300 K, P = 2.5 atm, $\Delta H_1(T)$ = -9.8 kcal mol^{-1}, $\hat{C}p$ = 25.8 cal mol^{-1} K^{-1}, k_0 = 2.83 x 10^4 s^{-1}, E = 12.9 kcal mol^{-1} and AU_{AM} = 13.6 cal s^{-1} K^{-1}. The feed molar flow rates of the reagents in stream a can be calculated according to equations (3) through (5), and since no reaction takes place in the heat exchanger, these are also the inlet molar flow rates in stream b.

$$\dot{n}_{A,a} = \dot{n}_{total,a} y_{A,a} = \dot{n}_{A,b} \tag{3}$$

$$\dot{n}_{B,a} = \dot{n}_{total,a} y_{B,a} = \dot{n}_{B,b} \tag{4}$$

$$\dot{n}_{Y,a} = \dot{n}_{Z,a} = \dot{n}_{Y,b} = \dot{n}_{Z,b} = 0 \tag{5}$$

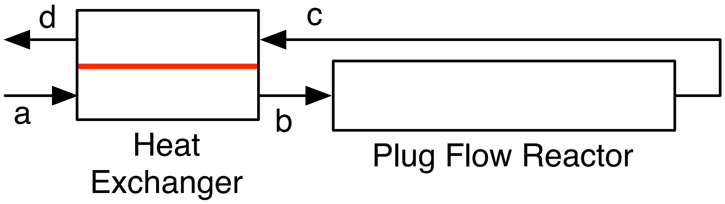

Figure 1. Schematic diagram of the heat exchanger and reactor.

First consider the PFR. The general steady state PFR mole and energy balances are given by equations (6) and (7). In the present case there is only one reaction, so the summations over j contain only one term. The reactor is adiabatic, so the heat transfer term in the energy balance equation is set to zero. Finally, the cross sectional area can be moved inside the derivatives as indicated in equation (8), where V denotes the cumulative reactor volume. After all these simplifications, the mole balances on A, B, Y and Z reduce to equations (9) through (12) and the energy balance reduces to equation (13).

$$\frac{d\dot{n}_i}{dz} = \frac{\pi D^2}{4} \sum_{\substack{j=all\\reactions}} \nu_{i,j} r_j \tag{6}$$

$$\pi D U \left(T_e - T\right) = \left(\sum_{\substack{i=all\\species}} \dot{n}_i \hat{C}_{pi} \right) \frac{dT}{dz} + \frac{\pi D^2}{4} \sum_{\substack{j=all\\reactions}} \left(r_j \Delta H_j\right) \tag{7}$$

$$\frac{1}{\frac{\pi D^2}{4}} \frac{d}{dz} = \frac{d}{dV} \tag{8}$$

$$\frac{d\dot{n}_A}{dV} = f_1\left(V, \dot{n}_A, \dot{n}_B, \dot{n}_Y, \dot{n}_Z, T\right) = -\left(2.83\times 10^5 \text{ s}^{-1}\right) \exp\left\{\frac{-9930 \text{ cal mol}^{-1}}{RT}\right\} C_A \tag{9}$$

$$\frac{d\dot{n}_B}{dV} = f_2\left(V, \dot{n}_A, \dot{n}_B, \dot{n}_Y, \dot{n}_Z, T\right) = -\left(2.83\times 10^5 \text{ s}^{-1}\right) \exp\left\{\frac{-9930 \text{ cal mol}^{-1}}{RT}\right\} C_A \tag{10}$$

$$\frac{d\dot{n}_Y}{dV} = f_3\left(V,\dot{n}_A,\dot{n}_B,\dot{n}_Y,\dot{n}_Z,T\right) = \left(2.83\times10^5\text{ s}^{-1}\right)\exp\left\{\frac{-9930\text{ cal mol}^{-1}}{RT}\right\}C_A \tag{11}$$

$$\frac{d\dot{n}_Z}{dV} = f_4\left(V,\dot{n}_A,\dot{n}_B,\dot{n}_Y,\dot{n}_Z,T\right) = \left(2.83\times10^5\text{ s}^{-1}\right)\exp\left\{\frac{-9930\text{ cal mol}^{-1}}{RT}\right\}C_A \tag{12}$$

$$\frac{dT}{dV} = f_5\left(V,\dot{n}_A,\dot{n}_B,\dot{n}_Y,\dot{n}_Z,T\right) = \frac{-\Delta H_1\left(2.83\times10^5\text{ s}^{-1}\right)\exp\left\{\frac{-9930\text{ cal mol}^{-1}}{RT}\right\}C_A}{\left(\dot{n}_A+\dot{n}_B+\dot{n}_Y+\dot{n}_Z\right)\hat{C}_p} \tag{13}$$

The PFR design equations, equations (9) through (13), are a set of initial-value ordinary differential equations, and they can be solved numerically. To do so, no matter what software is used, it is necessary to provide (a) the initial values, here the PFR inlet values, of the independent variable (V) and the dependent variables ($\dot{n}_A$, $\dot{n}_B$, $\dot{n}_Y$, $\dot{n}_Z$ and T), (b) the final, that is outlet, value of either the independent variable or one of the dependent variables and (c) code that is given values of the independent dependent variables and that evaluates the functions f_1 through f_5 in equations (9) through (13).

Considering first the initial values, at the inlet to the reactor, the cumulative volume is equal to zero. The inlet stream to the PFR is stream b; the molar flow rates of A, B, Y and Z in stream b have already been listed, so those initial values are known. However the initial value of the temperature, T_b, is not known. The problem specifies that the total reactor volume is 3 m^3, so the final value is known.

Looking at the PFR design equations, equations (9) through (13), the only quantities that appear, other than the independent and dependent variables are the ideal gas constant (known), the heat capacity (known) and the concentration of A. The concentration of A can be calculated, as in equation (14). With that, code can be written that evaluates the functions f_1 through f_5, given values of the independent dependent variables.

$$C_A = \frac{\dot{n}_A}{\dot{V}} = \frac{\dot{n}_A}{\frac{\dot{n}_{total}RT}{P}} = \frac{\dot{n}_A P}{\left(\dot{n}_A+\dot{n}_B+\dot{n}_Y+\dot{n}_Z\right)RT} \tag{14}$$

The PFR design equations cannot be solved at this point because the initial value of the temperature, T_b, is not known. Instead, it will be necessary to generate the design equations for the heat exchanger and solve them simultaneously with the PFR design equations following the approach given in the informational reading for this unit. There are two heat exchanger design equations; the first is an energy balance, which takes the general form shown in equation (15). In the present problem, the molar heat capacity for the gas as a whole is given instead of the species' individual molar heat capacities. In addition, the heat capacity is independent of temperature, so the integral in equation (15) can be evaluated trivially. As a result, the heat exchanger design equation takes the form shown in equation (16).

$$\sum_{\substack{i=all\\species}} \dot{n}_{i,c}\int_{T_c}^{T_d}\hat{C}_{p,i}\,dT + \sum_{\substack{i=all\\species}} \dot{n}_{i,a}\int_{T_a}^{T_b}\hat{C}_{p,i}\,dT = 0 \tag{15}$$

$$0 = f_6\left(T_b,T_d\right) = \dot{n}_{total,c}\hat{C}_p\left(T_d - T_c\right) + \dot{n}_{total,a}\hat{C}_p\left(T_b - T_a\right) \tag{16}$$

The other heat exchanger design equation is a heat transfer equation. In this case, we are given an overall heat transfer coefficient that is based upon the arithmetic mean temperature difference, so the general form of the equation is that given in equation (17). Making the same substitutions for the heat capacities leads to equation (18).

$$\sum_{\substack{i=all\\species}} \dot{n}_{i,c}\int_{T_c}^{T_d}\hat{C}_{p,i}\,dT + U_{AM}A\Delta T_{AM} = 0 \tag{17}$$

$$0 = f_7\left(T_b,T_d\right) = \left(\dot{n}_{total,c}\right)\hat{C}_p\left(T_d - T_c\right) + U_{AM}A\Delta T_{AM} \tag{18}$$

There are two heat exchanger design equations, equations (16) and (18), but in those equations there are four unknown quantities: $\dot{n}_{total,c}$, T_b, T_c and T_d. (By examination of the reaction stoichiometry, you might note that $\dot{n}_{total,a}$ = $\dot{n}_{total,c}$, but even so that still leaves three unknowns in two equations.) As discussed in the informational reading for this unit, the key to solving these non-differential, non-linear equations numerically is to choose T_b as one of the unknowns and to *not* choose T_c as the other. Here, as indicated in equations (16) and (18), the unknowns are taken to be T_b, and T_d.

In order to solve the heat exchanger design equations numerically, it will be necessary to (a) provide guesses for the values of T_b and T_d and (b) provide code that is given values for T_b and T_d and that uses those values to evaluate the equations being solved, that is the functions f_6 and f_7 in equations (16) and (18). Providing guesses is straightforward. Looking at the functions that need to be evaluated, it can be seen that the unknown quantities in those equations, other than those that will be given, are $\dot{n}_{total,c}$, T_c and ΔT_{AM}. Since our code will be given a value for T_b, it will have all the information needed to solve the PFR design equations, and doing so will yield $\dot{n}_{total,c}$ and T_c. With that, the arithmetic mean temperature difference can be computed using equation (19). At that point, the functions f_6 and f_7 in equations (16) and (18) can be evaluated.

$$\Delta T_{AM} = \frac{T_c + T_d}{2} - \frac{T_a + T_b}{2} \tag{19}$$

Thus, everything that is needed to solve the heat exchanger design equations numerically is available. Once that is done, T_b will be known and can be used to solve the PFR design equations numerically. Doing so will yield the molar flow rate of A in stream c among other quantities. Then, noting that the molar flow rate of A is the same in streams c and d, the conversion can be computed according to equation (20). Doing so, one finds that the conversion is 97%.

$$f_A = \frac{\dot{n}_{A,a} - \dot{n}_{A,d}}{\dot{n}_{A,a}} \tag{20}$$

The problem did not ask for a comparison to a PFR without an integrated heat exchanger, but it should be noted that if the feed to the process was admitted directly to the PFR without pre-heating, the conversion would be less than 1%. The reason is that the feed temperature is so low that the initial reaction rate is insufficient to cause the temperature rise sufficiently. By preheating, the initial rate is high enough that the reaction can generate substantial heat and thereby drive the reaction to high conversion.

Calculation Details Using MATLAB

Two MATLAB functions were written to perform the calculations needed to solve this problem. The first, Example_30_2_pfr, takes the inlet temperature as an argument and solves the PFR design equations, returning the outlet molar flow rates and the outlet temperature. The second MATLAB function, Example_30_2, analyzes the integrated heat exchanger and PRF by solving the heat exchanger design equations, using the first function to solve the PFR design equations as necessary.

MATLAB function for modeling the PFR. The PFR design equations are initial value ODEs. Supplemental Unit S5 describes how such equations are solved numerically and provides template files that can be used to solve them. In this problem, the final value of the independent variable (the reactor volume) is provided, so the appropriate template file is SolvIVDifI.m. Before that file can be used, it must be modified in four places.

To begin, I made a copy of the template file and saved it as Example_30_2_pfr.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_30_s_pfr. At the same time, I changed the function declaration so that the inlet temperature was passed to the function as an argument and so that the outlet molar flow rates and the outlet temperature were returned. 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 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 results of all these changes are shown in Listing 1.

The second *required* modification involves entering the code to evaluate functions f_1 through f_5 in equations (9) through (13). This occurs within the internal function `odeqns` which receives the independent variables as the scalar, `t`, and the dependent variables as a vector, `z`. Variables using the same name as in the problem statement ($\dot{n}_A$, $\dot{n}_Z$ and T) were extracted from the vector `z`, because, in my opinion, it makes the code more readable and easier to debug. In addition, the list of variables then serves as a reminder of the mapping of the problem statement variables to the vector `z`. Given the values of the independent and dependent variables, the concentration of A and the rate can be calculated using the ideal gas law and the rate expression given in the problem statement. Following that, the functions f_1

through f_5 in equations (9) through (13) can be evaluated, saving the results in the vector dzdt. These modifications are shown in Listing 2.

```
% Modified version of the MATLAB template file SolvIVDifI.m used in the
% solution of Example 30.2 of "A First Course on Kinetics and Reaction
% Engineering."
%
function [nAout,nBout,nYout,nZout,Tout] = Example_30_2_pfr(Tin)
    % Known quantities and constants (in mol m^3 s atm cal K)
    V = 3.0; % m^3
    ntot_in = 1.25; % mol/s
    yAin = 0.5;
    yBin = 0.5;
    P = 2.5; % atm
    cPi = 25.8; % cal/mol/K
    dH = -9800; % cal/mol
    k0 = 2.83e4; % /s
    E = 12900.0; % cal/mol
    R_energy = 1.987; % cal/mol/K
    R_pv = 0.08206 * 1e-3; % m^3*atm/mol/K
    nAin = yAin*ntot_in;
    nBin = yBin*ntot_in;
    nYin = 0;
    nZin = 0;
```

Listing 1. Non-required modifications and entry of known constants.

```
    % Function that evaluates the ODEs
    function dzdt = odeqns(t,z)
        nA = z(1);
        nB = z(2);
        nY = z(3);
        nZ = z(4);
        T = z(5);
        VFR = (nA+nB+nY+nZ)*R_pv*T/P;
        CA = nA/VFR;
        r = k0*exp(-E/R_energy/T)*CA;
        dzdt = [
            -r
            -r
            r
            r
            -dH*r/(nA+nB+nY+nZ)/cPi
        ];
    end % of internal function odeqns
```

Listing 2. Internal function odeqns after required modifications have been made.

The third *required* modification involves providing the initial values of the independent and dependent variables and the final value of the independent variable. The initial values of the dependent variables are entered as a vector named `z0`, and they must use the same mapping of the problem variables to the vector `z0` as was used previously for `z`. The results of performing this modification are shown in Listing 3.

```
    % Initial and final values
    t0 = 0;
    z0 = [
        nAin
        nBin
        nYin
        nZin
        Tin
    ];
    tf = V;
```

Listing 3. Specification of the initial values.

The final *required* modification is to use the results from solving the ODEs to calculate whatever the problem requested. In this case, all that needs to be done is to set the values of the outlet molar flow rates and the outlet temperature so that they will be returned by the function. This is shown in Listing 4.

```
    % set the values of the return variables
    nAout = z(1);
    nBout = z(2);
    nYout = z(3);
    nZout = z(4);
    Tout = z(5);
end % of Example_30_2_pfr.m
```

Listing 4. Final modification where the return variables are set to the proper values.

MATLAB function for evaluating the integrated heat exchanger and PFR. The heat exchanger design equations must be solved in order to analyze the integrated heat exchanger and PFR. The heat exchanger design equations are non-differential, non-linear equations. Supplemental Unit S2 describes how to solve such equations numerically using MATLAB, and it provides a template file named SolvNonDif.m for doing so. That template file was used as the starting point here. Before it can be used, SolvNonDif.m must be modified in four places, each indicated by a comment that begins “% EDIT HERE”.

In addition to those required modifications, I recommend that you work with a copy of the file that has been given a more meaningful name. In this case, I made a copy of the file and saved it as Example_30_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_30_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 is to enter the values of all universal and problem specific constants, converting them to a consistent set of units as they are entered. Listing 5 shows the results of all these modifications.

```
% Modified version of the MATLAB template file SolvNonDif.m used in the
% solution of Example 30.2 of "A First Course on Kinetics and Reaction
% Engineering."
%
function Equation_30_2
    % Known quantities and constants (in consistent units)
    Ta = 300; % K
    ntot_in = 1.25; % mol/s
    yAin = 0.5;
    yBin = 0.5;
    cPi = 25.8; % cal/mol/K
    nAa = yAin*ntot_in;
    nBa = yBin*ntot_in;
    nYa = 0;
    nZa = 0;
    UA = 13.6;
```

Listing 5. Initial modifications to the template file SolvNonDif.m.

The next required modification is to provide code that evaluates the functions f_6 and f_7 in equations (16) and (18). At this point, you need to decide which of the unknowns in the equations being solved is going to be represented as z_1, which as z_2, and so on. I recommend actually defining variables with names similar to those used in your solution and setting them equal to the corresponding z_i. Doing so will make your code a little less efficient, but it may also reduce the chances of coding errors. Listing 6 shows how this was done in Example_30_2.m and how code to evaluate the functions was entered so that the vector named `f` contains the value of f_6 as its first element and the value of f_7 as its second element. Notice that before the functions could be evaluated, it was necessary to call the function that solves the PFR design equations in order to obtain the value of T_c.

```
    % Function that evaluates the equations
    function f = evalEqns(z)
        Tb = z(1);
        Td = z(2);
        [nAc,nBc,nYc,nZc,Tc] = Example_30_2_pfr(Tb);
        AMTD = 0.5*((Td-Ta) + (Tc-Tb));
        f = [
            (nAa+nBa+nYa+nZa)*cPi*(Tb-Ta) - (nAc+nBc+nYc+nZc)*cPi*(Tc-Td)
            (nAa+nBa+nYa+nZa)*cPi*(Tb-Ta) - UA*AMTD
        ];
    end % of internal function evalEqns
```

Listing 6. Modifications to evaluate the functions being solved.

The third required modification is where guesses for the unknowns in the heat exchanger design equations are provided. The guesses are entered in the array named z_guess. Listing 7 shows that I simply guessed that T_b would be 50 K larger than T_a and T_d would be 100 K larger than T_a.

```
    % guesses for the solution
    z_guess = [
        Ta + 50
        Ta + 100
    ];
```

Listing 7. Modification where guesses are provided for the unknowns.

The final modification is where any additional calculations are performed after the non-linear equations have been solved. Solving the heat exchanger design equations yields T_b, the PFR inlet temperature. Therefore, after the heat exchanger design equations had been solved, the MATLAB function for solving the PFR design equations was called, using that value of T_b. Doing so returns the outlet molar flow rates and the outlet temperature. With that result, the conversion can be calculated and displayed. I chose to also display all four stream temperatures. While the problem did not request it, I again called the MATLAB function for solving the PFR design equations, but using the feed temperature, T_a, as the PFR inlet temperature. That allowed me to calculate the conversion that would be observed using a stand-alone PFR. These changes are shown in Listing 8.

```
    % Use the value of Tb found by solving the heat exchanger design
    % equations to calculate the outlet molar flow of A and then the
    % conversion
    Tb = z(1);
    Td = z(2);
    [nAc,nBc,nYc,nZc,Tc]=Example_30_2_pfr(Tb);
    display('With thermal backmix:')
    Ta
    Tb
    Tc
    Td
    percent_conversion = 100*(nAa-nAc)/nAa

    % Case with no backmix, feed to reactor is at Ta
    [nAb,nBb,nYb,nZb,Tb]=Example_30_2_pfr(Ta);
    display('Without thermal backmix:')
    Tb
    percent_conversion = 100*(nAa-nAb)/nAa
```

Listing 8. Final modification where the fractional conversion is calculated.

At this point, Example_30_2.m can be executed by by typing Example_30_2 at the MATLAB command prompt. The resulting output is shown in Listing 9. It is important to check that the non-linear equation solver converged. In this case it did, and therefore, the results produced by the calculations can be accepted.

```
>> Example_30_2

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 =
  377.4979
  483.7725

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

f =
   1.0e-10 *

   -0.8276
   -0.1774

With thermal backmix:

Ta =
   300

Tb =
  377.4979

Tc =
  561.2704

Td =
  483.7725

percent_conversion =
   96.7618

Without thermal backmix:

Tb =
  300.5323

percent_conversion =
    0.2803
```

Listing 9. Output from the execution of Example_30_2.