A First Course on Kinetics and Reaction Engineering

Example 29.3

Problem Purpose

This problem will help you determine whether you have mastered the learning objectives for this unit. It illustrates the analysis of a parallel reactor network and the effect of mixing streams of unequal conversion.

Problem Statement

A liquid phase process that uses a 60 dm^3 PFR is going to be shut down so that a second 40 dm^3 PFR can be added to increase throughput. Space constraints dictate that the second reactor will be added in parallel with the first. The design calls for processing a feed solution containing 1 mol L^{-1} of A at 60 °C and flowing at a rate of 0.55 dm^3 min^{-1}. The solution density may be assumed to be constant. The reactors will operate adiabatically and at steady state. The reaction is irreversible, and the reaction rate is given in equation (2) where k_0 = 2.63 x 10^7 L mol^{-1} min^{-1} and E = 62 kJ mol^{-1}. The heat of reaction and the heat capacity of the solution may be assumed to be constant and equal to -35 kJ mol^{-1} and 800 J L^{-1} K^{-1}, respectively. Compare the conversion when the feed is split equally between the two reactors to the conversion when the feed is split so that the space times are equal in the two reactors.

$$2\,\text{A} \rightarrow \text{Y} + \text{Z} \qquad (1)$$

$$r = k_0 \exp\left\{\frac{-E}{RT}\right\} C_A^2 \qquad (2)$$

Problem Analysis

We are given the rate expression and asked to find reactor variables, so this is a reaction engineering problem involving PFRs. The problem states that the reactors operate adiabatically and at steady state, so to answer the questions posed, a mole balance on every reactant and product and an energy balance will be written and solved for each reactor in the network. Negligible pressure drop will be assumed because insufficient information is provided to include a mechanical energy balance. The reactor length and diameter are not given, but since the reactors are adiabatic, the design equations can be written in a form where the cumulative volume is the dependent variable instead of the axial position.

Problem Solution

It is good practice, when solving reactor network problems, to draw a simple schematic diagram and label each of the flow streams. This problem involves a simple parallel network of two PFRs as shown in Figure 1 where the original reactor is designated as R1 and the new reactor as R2. Notice that the flow streams have been labeled a through f.

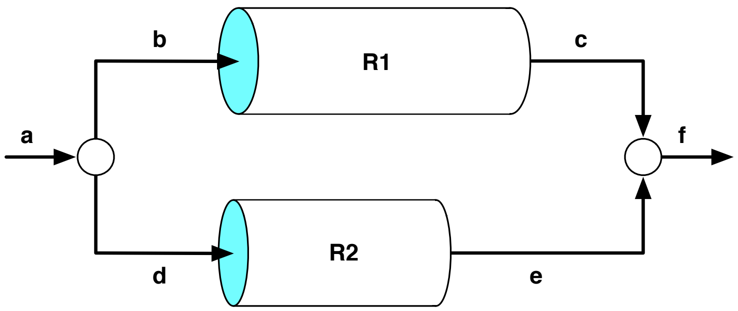

Figure 1. Schematic representation of the reactor network.

The design equations are the same for each reactor, all that differ are the reactor volumes and the volumetric flow rates into the reactors. Therefore we'll start by writing the design equations that will apply to either of the reactors. The general steady state PFR mole balance is given by equation (3), and the general steady state PFR energy balance, by equation (4). As noted in the problem analysis and described in Unit 26, these equations will need to be re-written to use the cumulative volume as the dependent variable. That is, equation (5) needs to be used to replace dz in equations (3) and (4). There is only one reaction taking place, so the summation over the reactions in both equations will reduce to a single term. The PFR also operates adiabatically, so the heat input term equals zero. In addition, we are given a volume-specific heat capacity for the fluid as a whole, so the summation over the individual heat capacities of the species can be replace by a single sensible heat term. Implementing these changes leads to the PFR design equations given by equations (6) through (9), where the subscripts "in" and "out" designate the inlet and outlet flow streams.

$$\frac{d\dot{n}_i}{dz} = \frac{\pi D^2}{4} \sum_{\substack{j=all\\reactions}} \nu_{i,j} r_j \tag{3}$$

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

$$dV = \frac{\pi D^2}{4} dz \tag{5}$$

$$\frac{d\dot{n}_A}{dV} = f_1\left(V, \dot{n}_A, \dot{n}_Y, \dot{n}_Z, T\right) = -2r \ ; \quad \dot{n}_A\left(0\right) = \dot{n}_{A,in} \tag{6}$$

$$\frac{d\dot{n}_Y}{dV} = f_2\left(V, \dot{n}_A, \dot{n}_Y, \dot{n}_Z, T\right) = r \ ; \quad \dot{n}_Y\left(0\right) = \dot{n}_{Y,in} \tag{7}$$

$$\frac{d\dot{n}_Z}{dV} = f_3\left(V,\dot{n}_A,\dot{n}_Y,\dot{n}_Z,T\right) = r \ ; \quad \dot{n}_Z\left(0\right) = \dot{n}_{Z,in} \tag{8}$$

$$\frac{dT}{dV} = f_4\left(V,\dot{n}_A,\dot{n}_Y,\dot{n}_Z,T\right) = \frac{-r\Delta H\left(T\right)}{\dot{V}_{in}\tilde{C}_p}; \quad T\left(0\right) = T_{in} \tag{9}$$

Equations (6) through (9) are a set of coupled, initial value ordinary differential equations where the independent variable is V and the dependent variables are $\dot{n}_A$, $\dot{n}_Y$, $\dot{n}_Z$ and T. They can be solved numerically, but no matter what software one uses, it will be necessary to provide (a) the initial values (here the inlet values) of the independent and dependent variables, (b) the final value (here the outlet value) of either the independent variable or one of the dependent variables and (c) code that is given the values of the independent and dependent variables and uses them to evaluate the four derivatives in equations (6) through (9); that is, the functions f_1 through f_4 in those equations.

In all cases, the initial value of the independent variable is V = 0 dm^3, the initial values of $\dot{n}_Y$ and $\dot{n}_Z$ are 0 (since no Y or Z are present in the feed) and the initial value of T is 60 °C. The initial value of $\dot{n}_A$ will depend how the feed is split, but in all cases it can be calculated using equation (10). This is true because splitting the feed does not change the concentration: $C_{A,a}$ = $C_{A,b}$ = $C_{A,d}$. Also, in all cases the final value is the volume of the reactor being modeled, i. e. V_{R1} or V_{R2}.

$$\dot{n}_{A,in} = C_{A,a}\dot{V}_{in} \tag{10}$$

At this point we can consider the two cases, starting with the one where the feed is split evenly between the two reactors. In this case the inlet volumetric flow rate for both reactors is found using equation (11). The volumetric flow rate of stream a is given as 0.55 dm^3 min^{-1}.

$$\dot{V}_b = \dot{V}_d = \frac{\dot{V}_a}{2} \tag{11}$$

Knowing the (constant) inlet volumetric flow rate for each reactor, the initial conditions can be specified as already described, and the final values are just the reactor volumes. It remains to provide the code that evaluates the functions f_1 through f_4. The gas constant (R), the heat capacity ($\tilde{C}_p$ = 800 J L^{-1} K^{-1}), and the heat of reaction ($\Delta H(T)$ = -35 kJ mol^{-1}) and inlet volumetric flow rate (equation (11)) are known constants. The only other quantity appearing in the design equations that needs to be calculated is the rate, and that can be done using equation (2) by substituting k_0 and E given in the problem statement along with the temperature (one of the dependent variables given to the code) and the concentration of A, calculated using equation (12) or equation (13), depending on the reactor being modeled (the molar flow rate of A is also one of the dependent variables that will be given to the code). The inlet volumetric flow rates are used in equations (12) and (13) because the fluid density is constant causing the volumetric flow rate to also be constant.

$$C_{A,R1} = \frac{\dot{n}_A}{\dot{V}_b} \tag{12}$$

$$C_{A,R2} = \frac{\dot{n}_A}{\dot{V}_d} \tag{13}$$

At this point, the design equations, (6) through (9), can be solved for each of the two reactors. The molar flow rate of A in stream *f* will equal the sum of the molar flow rates of A in streams *c* and *e*, equation (14). Then the overall conversion can be calculated using equation (15). Doing so one finds the conversion to equal 79.4%.

$$\dot{n}_{A,f} = \dot{n}_{A,c} + \dot{n}_{A,e} \tag{14}$$

$$f_A = \frac{\dot{n}_{A,a} - \dot{n}_{A,f}}{\dot{n}_{A,a}} \tag{15}$$

The only difference in the second case is that the feed is split so that the space times for the two reactors are equal, equation (16). Noting that the sum of the volumetric flow rates of streams *b* and *d* are equal to the known (constant) volumetric flow rate of stream *a* (the feed) allows the calculation of the inlet volumetric flow rate for each reactor, equations (17) and (18).

$$\frac{V_{R1}}{\dot{V}_b} = \frac{V_{R2}}{\dot{V}_d} \tag{16}$$

$$\frac{V_{R1}}{\dot{V}_b} = \frac{V_{R2}}{\dot{V}_d} = \frac{V_{R2}}{\dot{V}_a - \dot{V}_b} \qquad \Rightarrow \qquad \dot{V}_b = \frac{\dot{V}_a V_{R1}}{\left(V_{R2} + V_{R1}\right)} \tag{17}$$

$$\dot{V}_d = \dot{V}_a - \dot{V}_b \tag{18}$$

The remainder of the solution is the same as the first case. Performing the calculations one find the conversion to equal 80.3%. While the difference in conversions in this example is small (79.4% vs. 80.3%), this result agrees with the caveat presented in the informational reading that one should avoid mixing stream of unequal conversion in a parallel reaction network. The reason is easy to understand from a qualitative analysis. For a typical reaction, the rate gets smaller as the conversion increases. As a consequence, each additional increment of conversion requires a larger volume that the increment before it. When you mix two streams of unequal conversion, the resulting stream will have a conversion intermediate between the two. If the stream with the higher conversion had only been converted to the after-mixing conversion, the reactor volume that would be saved, would be more than is needed to convert the stream with the lower conversion to the after-mixing conversion. Thus, the after-mixing conversion could have been attained using less total reactor volume, leaving some extra left-over volume. By mixing the two streams, you have effectively wasted that extra volume.

Calculation Details Using MATLAB

In this problem, it was possible to solve the design equations for each of the reactors individually. Therefore I elected to write a MATLAB function for solving the PFR design equations, and a second function to split the feed and call the function that evaluates the PFRs. Since the reactor volume and the inlet volumetric flow rate will vary depending on the case and reactor being considered, those quantities were passed to the function that solves the PFR design equations as arguments. The PFR design equations, (6) through (9), are coupled ordinary differential equations (ODEs). Supplemental Unit S5 provides template files that can be used to solve such equations. 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 SolvIVDifI.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_29_3_pfr.m; a copy of that file accompanies this solution. As required, I changed the function name to match the filename. I also changed the function declaration to indicate that the reactor volume and inlet volumetric flow rate would be passed into the function as arguments and the outlet molar flow rates of A, Y and Z and the outlet temperature would be returned. I replaced the long introductory comment with a brief comment stating the purpose of the modified version. 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. Since the inlet volumetric flow rate was passed into the function, the constant inlet molar flow rates could also be calculated at this point. The result of making all these modifications is shown in Listing 1.

```
% Modified version of the AFCoKaRE MATLAB template file SolvIVDifD.m used
% for the solution of Example 29.3 of "A First Course on Kinetics and
% Reaction Engineering."
function [nAout,nYout,nZout,Tout] = Example_29_3_pfr(V,VFRin)
    % Known quantities and constants
    CAin = 1.0; % mol/L
    CYin = 0; % mol/L
    CZin = 0; % mol/L
    Tin = 60 + 273.15; % K
    k0 = 2.63E7; % L/mol/min
    E = 62.; % kJ/mol
    dH = -35.; % kJ/mol
    cP = 0.8; % kJ/L/K
    % Gas constant
    R = 8.3145/1000; % kJ/mol/K
    % Calculated constants
    nAin = VFRin*CAin;
    nYin = VFRin*CYin;
    nZin = VFRin*CZin;
```

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

The second *required* modification involves entering code that, given values for the independent and dependent variables, evaluates the derivatives when the ODEs are written in the standard form as in equations (6) through (9). Notice that the dependent variables are provided as a vector quantity and the evaluated functions are returned as a vector quantity. Thus, it is necessary to map the dependent variables used in the problem statement ($\dot{n}_A$, $\dot{n}_Y$, $\dot{n}_X$, T) to a vector z, and the corresponding derivatives are mapped to a vector dzdt. I find it useful at the start of the internal function that will evaluate the derivatives (odeqns), 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 concentration of A according to equation (12) or (13). Following that, the rates are calculated using equation (2). Finally, equations (6) through (9) are evaluated, saving the results in the vector dzdt using the same mapping that was used to map the dependent variables to the vector z. The resulting code is shown in Listing 2.

```
% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    nA = z(1);
    nY = z(2);
    nZ = z(3);
    T = z(4);
    CA = nA/VFRin;
    r = k0*exp(-E/R/T)*CA^2;
    dzdt = [
        -2*r
        r
        r
        -r*dH/VFRin/cP;
    ];
end % of internal function odeqns
```

Listing 2. Results of the second required modification.

The third *required* modification involves providing the initial values of the independent (V = 0 L) and dependent variables (calculated where the constants were entered) and the final value of the independent variable (V which is passed in as an argument). The results of performing this modification are shown in Listing 3. The fourth and final *required* modification is to calculate any additional quantities that the problem asked for. Since this function returns the outlet molar flows and the outlet temperature as individual values, and not as a vector, all I had to do here was to extract those individual values from the solution vector, z as shown in Listing 4.

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

Listing 3. Results of the third required modification.

```
    % Solve the ODEs
    [t,zz] = ode45(@odeqns,[t0 tf],z0);

    % Return the values of the independent and dependent variables when the
    % integration stopping condition was satisfied
    last_value = length(t);
    t_f = t(last_value);
    z = zz(last_value,:);
    nAout = z(1);
    nYout = z(2);
    nZout = z(3);
    Tout = z(4);
```

Listing 4. Results of the fourth required modification.

MATLAB function for analyzing the reactor networks. Once the function for modeling the PFR was available, a second MATLAB function, named Example_29_3, was created to analyze the two reactor networks. That function is shown in Listing 5. The known reactor volumes, feed volumetric flow rate and feed concentration of A are first entered. Next the case where the feed is equally split is evaluated. The inlet flow rate for each of the two reactors is calculated using equation (11), the outlet molar flows and temperature for each reactor are then calculated by calling the function that models the PFR. After that, the overall conversion is calculated using equation (15). Finally, the case where the reactor space times are equal is evaluated. The inlet flow rate for each reactor is calculated using equations (17) and (18). The outlet molar flows and temperature for each reactor are then calculated by calling the function that models the PFR, and the conversion is calculated using equation (15). The function is executed by typing Example_29_3 at the MATLAB command prompt, and the output shown in Listing 6 is generated.

```
% Function used to analyze the two parallel reactor networks in Example
% 29.3 of "A First Course on Kinetics and Reaction Engineering."
function Example_29_3
    VR1 = 60.0; % L
    VR2 = 40.0; % L
    VFRa = 0.55; % L/min
    CAa = 1.0; % mol/L

    % Case 1 where the feed is split equally
    display('Case 1: Equally split feed')
    VFRin = VFRa/2.;
    % Calculate the outlet molar flows and temperature for the reactors
    [nAoutR1,nYoutR1,nZoutR1,ToutR1] = Example_29_3_pfr(VR1,VFRin);
    [nAoutR2,nYoutR2,nZoutR2,ToutR2] = Example_29_3_pfr(VR2,VFRin);
    conversion = (VFRa*CAa - nAoutR1 - nAoutR2)/VFRa/CAa

    % Case 2 where the space times are equal
    display('Case 2: Equal space times')
    VFRb = VFRa*VR1/(VR1 + VR2);
    VFRd = VFRa - VFRb;
    % Calculate the outlet molar flows and temperature for the reactors
    [nAoutR1,nYoutR1,nZoutR1,ToutR1] = Example_29_3_pfr(VR1,VFRb);
    [nAoutR2,nYoutR2,nZoutR2,ToutR2] = Example_29_3_pfr(VR2,VFRd);
    conversion = (VFRa*CAa - nAoutR1 - nAoutR2)/VFRa/CAa

end % of Example_29_3.m
```

Listing 5. MATLAB function for evaluating the two reactor networks.

```
>> Example_29_3
Case 1: Equally split feed
conversion =
    0.7944

Case 2: Equal space times
conversion =
    0.8029
```

Listing 6. Output from the execution of Example_29_3.