

# A First Course on Kinetics and Reaction Engineering

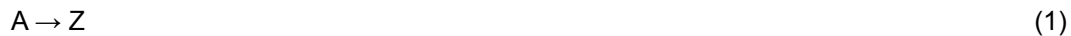
## Example 30.1

### 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

Reaction (1) takes place in an adiabatic, steady state PFR using  $750 \text{ dm}^3 \text{ min}^{-1}$  of a constant-density solution containing the reactant at a concentration of  $3.8 \text{ mol dm}^{-3}$  and a temperature of  $25 \text{ }^\circ\text{C}$ . The heat of reaction is  $-79.8 \text{ kJ mol}^{-1}$ , and may be assumed to be constant over the range of temperatures where this reactor operates. The heat capacity of the solution is equal to  $987 \text{ cal L}^{-1} \text{ K}^{-1}$ , independent of composition and temperature. For a conversion of 80%, compare the required PFR volume without thermal back-mixing to that with thermal back-mixing assuming a simple counter-current heat exchanger that can be described using an arithmetic mean temperature difference with  $AU_{AM} = 1500 \text{ kJ min}^{-1} \text{ K}^{-1}$ . The rate expression is given in equation (2).



$$r_1 = \left(3.38 \times 10^6 \text{ min}^{-1}\right) \exp\left\{\frac{-50 \text{ kJ mol}^{-1}}{RT}\right\} C_A \quad (2)$$

### 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 requested, 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 does not provide sufficient information to write a momentum balance, so it will be necessary to assume that the pressure drop is negligible. 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

Reading through the problem, we find that we are given the values of the following constant quantities:  $\dot{V} = 750 \text{ dm}^3 \text{ min}^{-1}$ ,  $C_{A,feed} = 3.8 \text{ mol dm}^{-3}$ ,  $T_{feed} = 25 \text{ }^\circ\text{C}$ ,  $\Delta H_1(T) = -79.8 \text{ kJ mol}^{-1}$ ,  $\tilde{C}_p = 987 \text{ cal L}^{-1} \text{ K}^{-1}$ ,  $f_A = 0.8$ ,  $k_0 = 3.38 \times 10^6 \text{ min}^{-1}$ ,  $E = 50 \text{ kJ mol}^{-1}$  and  $AU_{AM} = 1500 \text{ kJ min}^{-1} \text{ K}^{-1}$ . Since the problem states that the solution density is constant and the stream is never split, the volumetric flow rate will be

constant and it isn't necessary to specify it separately for each stream. The molar flow rate of A into the process is also constant and can be calculated using equation (3). Finally, the problem does not indicate that any product Z is present in the feed to the process, so  $\dot{n}_{Z,feed} = 0$ .

$$\dot{n}_{A,feed} = \dot{V}C_{A,feed} \quad (3)$$

The model for the PFR will be the same whether the PFR is integrated with a heat exchanger or isolated, but the identity of the streams entering and leaving the reactor will differ. For that reason, when writing the PFR design equations, "in" and "out" will be used to denote the stream entering the PFR and the stream leaving the PFR, respectively. The general steady state PFR mole and energy balances are given by equations (4) and (5). In the present case there is only one reaction, so the summations over  $j$  contain only one term. In addition, a volumetric heat capacity for the fluid as a whole is given, allowing the summation over the individual species heat capacities to also be replaced by a single 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 (6), where  $V$  denotes the cumulative reactor volume. After all these simplifications and substitution of the rate expression, the mole balances on A and Z reduce to equations (7) and (8) and the energy balance reduces to equation (9).

$$\frac{d\dot{n}_i}{dz} = \frac{\pi D^2}{4} \sum_{\substack{j=all \\ reactions}} v_{i,j} r_j \quad (4)$$

$$\pi D U (T_e - T) = \left( \sum_{\substack{i=all \\ species}} \dot{n}_i \hat{C}_{pi} \right) \frac{dT}{dz} + \frac{\pi D^2}{4} \sum_{\substack{j=all \\ reactions}} (r_j \Delta H_j) \quad (5)$$

$$\frac{1}{\pi D^2} \frac{d}{dz} = \frac{d}{dV} \quad (6)$$

$$\frac{d\dot{n}_A}{dV} = f_1(V, \dot{n}_A, \dot{n}_Z, T) = -\left(3.38 \times 10^6 \text{ min}^{-1}\right) \exp\left\{\frac{-50000 \text{ kJ mol}^{-1}}{RT}\right\} C_A \quad (7)$$

$$\frac{d\dot{n}_Z}{dV} = f_2(V, \dot{n}_A, \dot{n}_Z, T) = \left(3.38 \times 10^6 \text{ min}^{-1}\right) \exp\left\{\frac{-50000 \text{ kJ mol}^{-1}}{RT}\right\} C_A \quad (8)$$

$$\frac{dT}{dV} = f_3(V, \dot{n}_A, \dot{n}_Z, T) = \frac{-\Delta H_1 \left(3.38 \times 10^6 \text{ min}^{-1}\right) \exp\left\{\frac{-50000 \text{ kJ mol}^{-1}}{RT}\right\} C_A}{\dot{V} \tilde{C}_p} \quad (9)$$

Equations (7) through (9) are ordinary differential equations that can be solved numerically. To do so, no matter what software is used, it will be necessary to provide (a) initial values for the independent ( $V$ ) and dependent ( $\dot{n}_A$ ,  $\dot{n}_Z$  and  $T$ ) variables, (b) the final value of either the independent variable or one of the dependent variables and (c) code that is given values for the independent and dependent variables and uses those values to evaluate the functions  $f_1$  through  $f_3$  in equations (7) through (9).

The initial conditions needed to solve the design equations for a PFR correspond to the reactor inlet. At the inlet to the reactor, the cumulative volume will equal zero. The initial values of the dependent variables are constants, equations (10) through (12). In this problem, the conversion of A is specified, so that the outlet molar flow rate of A becomes the final condition needed to solve the design equations for the PFR. Since all conversion takes place in the reactor, the final condition can be calculated using equation (13).

$$\dot{n}_A(V=0) = \dot{n}_{A,in} \quad (10)$$

$$\dot{n}_Z(V=0) = \dot{n}_{Z,in} \quad (11)$$

$$T(V=0) = T_{in} \quad (12)$$

$$\dot{n}_A(V) = \dot{n}_{A,in}(1 - f_A) \quad (13)$$

The code needed to solve the design equations numerically will be given values for  $V$ ,  $\dot{n}_A$ ,  $\dot{n}_Z$  and  $T$ . It must use those to evaluate  $f_1$  through  $f_3$  in equations (7) through (9). Going through those equations variable by variable, the only quantity that is not known is the concentration of A. By definition, the concentration of A can be calculated using equation (14). Once that is done, the design equations can be solved numerically.

$$C_A = \frac{\dot{n}_A}{\dot{V}} \quad (14)$$

Now that we have a model for the PFR, we can consider the two reactor systems described in the problem statement. The first system is an isolated PFR, Figure 1.



Figure 1. Schematic representation of the isolated PFR.

In this configuration, the stream entering the PFR is the feed stream. The initial conditions, that is, the inlet molar flow rates of A and Z and the inlet temperature are known, equations (15) through (17). The final condition is found using equation (13). With that, everything needed to solve the PFR design

equations is available, and the design equations can be solved numerically to find that a reactor volume of 37400 L is required to reach the specified conversion.

$$\dot{n}_{A,in} = \dot{n}_{A,feed} \quad (15)$$

$$\dot{n}_{Z,in} = \dot{n}_{Z,feed} = 0 \quad (16)$$

$$T_{in} = T_{feed} \quad (17)$$

The second system to be considered consists of an integrated counter-current heat exchanger and a PFR, as shown schematically in Figure 2. In this configuration, the inlet stream is stream *b*, and the outlet stream is stream *c*. However, since no reaction takes place within the heat exchanger,  $\dot{n}_{i,a} = \dot{n}_{i,b}$ , and the initial values of the molar flow rates are the same as before, equations (15) and (16). The final condition is also the same as before and is calculated using equation (13). However for this configuration the initial value of the temperature, equation (18), is not known.

$$T_{in} = T_b \quad (18)$$

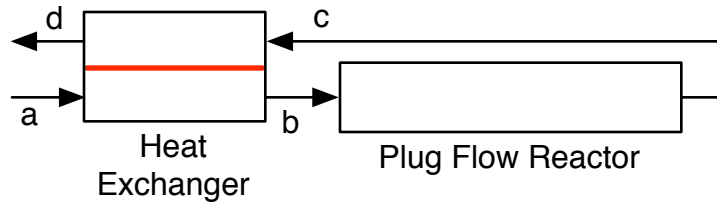


Figure 2. Schematic representation of the integrated heat exchanger and PFR.

Thus, the PFR design equations cannot be solved independently of the heat exchanger design equations. Put differently, the heat exchanger design equations are needed in order to calculate the value of  $T_b$ . The heat exchanger design equations consist of the requirement that the heat lost by one stream must equal the heat gained by the other stream, equation (19), and the heat transfer equation. In this case, the problem specifies that the heat transfer can be described using an arithmetic mean temperature difference, equations (20) and (21).

$$0 = \sum_{\substack{i=all \\ species}} \dot{n}_{i,c} \int_{T_c}^{T_d} \hat{C}_{p,i} dT + \sum_{\substack{i=all \\ species}} \dot{n}_{i,a} \int_{T_a}^{T_b} \hat{C}_{p,i} dT \quad (19)$$

$$- \sum_{\substack{i=all \\ species}} \dot{n}_{i,c} \int_{T_c}^{T_d} \hat{C}_{p,i} dT = U_{AM} A \Delta T_{AM} \quad (20)$$

$$\Delta T_{AM} = \frac{T_c + T_d}{2} - \frac{T_a + T_b}{2} \quad (21)$$

In the present problem, we know the volume specific heat capacity of the fluid as a whole, so the summations in equations (19) and (20) can be replaced with single terms. In addition, since the heat capacity is constant, the integral can be evaluated, leading to equations (22) and (23). We have only two heat exchanger design equations; in order to solve them numerically, we must identify two unknowns that appear in those design equations. As you may have noticed from the equations, and as recommended in the informational reading for this unit, I have selected  $T_b$  and  $T_d$  to be the two unknowns. Importantly, even though the value of  $T_c$  is unknown, I did not choose it as one of the unknowns. The reason for this choice should become apparent upon considering the numerical solution of equations (22) and (23).

$$0 = f_4(T_b, T_d) = \dot{V}\tilde{C}_p(T_d - T_c) + \dot{V}\tilde{C}_p(T_b - T_a) \quad (22)$$

$$0 = f_5(T_b, T_d) = \dot{V}\tilde{C}_p(T_d - T_c) + AU_{AM}\Delta T_{AM} \quad (23)$$

To solve equations (22) and (23) numerically it is necessary to provide (a) a guess for the unknowns and (b) code that is given values for the unknowns and uses them to evaluate the two functions,  $f_4$  and  $f_5$ , in equations (22) and (23). Providing a guess for the unknowns is straightforward. The code that is needed will be given values for  $T_b$  and  $T_d$ . In order to evaluate the functions,  $f_4$  and  $f_5$ , it will be necessary to know or calculate every other quantity that appears in equations (22) and (23). Since the code will be given a value for  $T_b$ , the remaining initial value needed in order to solve the PFR design equations will be known. Therefore, the PFR design equations can be solved, and that will yield the value of  $T_c$ . In this configuration,  $T_a$  is equal to  $T_{feed}$ . With that value, the arithmetic mean temperature difference can be calculated using equation (21). The remaining quantities in equations (22) and (23), namely,  $\dot{V}$ ,  $\tilde{C}_p$  and  $AU_{AM}$  are known constants. Therefore, the heat exchanger design equations can be solved numerically to find the values of  $T_b$  and  $T_d$ . Once  $T_b$  is known, the PFR design equations can be solved to find that the required volume is only 8870 L when thermal back-mixing is used.

### Calculation Details Using MATLAB

Two MATLAB functions were written to perform the calculations needed to solve this problem. The first, `Example_30_1_pfr`, takes the inlet temperature as an argument and solves the PFR design equations, returning the reactor volume and the outlet temperature. The second MATLAB function, `Example_30_1`, performs the calculations needed to evaluate the two reactor configurations, solving the heat exchanger design equations and calling the first function as necessary.

**MATLAB function for modeling the PFR.** The PFR design equations are initial value ODEs. Supplemental Unit S5 provides template files that can be used to solve them. In this problem, the final value of a dependent variable is provided, so the appropriate template file is `SolvIVDifD.m`. Before that file can be used, you must make six required modifications.

To begin, I made a copy of the template file and saved it as Example\_30\_1\_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\_1\_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 reactor volume 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 are entered, they should be converted to a consistent set of units. For this problem, I decided to use units of [L, mol, min, K, and kJ]. Listing 1 shows the first part of the modified file.

```
% Modified version of the MATLAB template file SolvIVDifD.m used to model
% the pfr in Example 30.1 of "A First Course on Kinetics and Reaction
% Engineering."
%
function [V,Tout] = Example_30_1_pfr(Tin)
    % Known quantities and constants (in consistent units)
    VFR = 750.0; % L/min
    CAa = 3.8; % mol/L
    dH = -79.8; % kJ/mol
    cP = 987.0 * 4.184e-3; % kJ/L/K
    fA = 0.8;
    k0 = 3.38e6; % /min
    E = 50.0; % kJ/mol
    R = 8.31446e-3; % kJ/mol/K
    nAin = CAa*VFR;
    nZin = 0;
```

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

The second *required* modification involves entering the code to evaluate functions  $f_1$  through  $f_3$  in equations (7) through (9). Notice that the equations are provided as a vector quantity. Thus, it is necessary to map the dependent variables used in the problem statement ( $\dot{n}_A$ ,  $\dot{n}_Z$  and  $T$ ) 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, 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  $z$ . Given the values of the independent and dependent variables, the concentration of A and the rate can be calculated using equations (14) and (2), respectively. Following that, the functions  $f_1$  through  $f_3$  in equations (7) through (9) can be evaluated, saving the results in the vector  $dzdt$ . These modifications all occur within the internal function `odeqns`, which is shown in Listing 2.

```

% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    nA = z(1);
    nZ = z(2);
    T = z(3);
    CA = nA/VFR;
    r = k0*exp(-E/R/T)*CA;
    dzdt = [
        -r
        r
        -dH*r/VFR/cP;
    ];
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. 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 to the vector  $z_0$  as was used previously for  $z$ . The results of performing this modification are shown in Listing 3.

```

% Initial values
t0 = 0;
z0 = [
    nAin
    nZin
    Tin
];

```

Listing 3. Specification of the initial values.

The fourth and fifth *required* modifications involve providing the final value of one dependent variable, in this problem  $\dot{n}_{A,out}$ . The first step (fourth modification) is to select a value for  $t_f$  (the final value of the independent variable,  $V$ ) that is much greater than the actual cumulative reactor volume. Of course, you don't know what the volume is at this point, so all you can do is set  $t_f$  to a large number. The cumulative volume should never reach this value. If it does, the code will print a warning and instruct you to set  $t_f$  to a larger value.

The second step (fifth modification) is to provide the final condition. Basically, this involves adding the code to calculate the value of a variable named `stop_when`. The variable, `stop_when`, should equal zero when the final condition is reached. Here the final condition we want to reach is given in equation (13). Noting that  $z(1)$  will equal the current value of  $\dot{n}_{A,out}$  and re-arranging equation (14) yields equation (24). Clearly, the final condition will be reached when the right hand side of equation (24) equals zero, so

the variable `stop_when` is set equal to the right hand side of equation (24). The results of these two modifications are shown in Listing 4.

$$0 = \dot{n}_{A,out} - \dot{n}_{A,in}(1 - f_A) = z(1) - \dot{n}_{A,in}(1 - f_A) \quad (24)$$

```
tf = 1E6; % This value should NOT be reached
options = odeset('Events',@stop);
[t, zz, te, ze, ie] = ode45(@odeqns,[t0, tf],z0,options);

% Function that provides the integration stopping criterion
function [stop_when, isterminal, direction] = stop(t,z)
    isterminal = 1;
    direction = 0;

    stop_when = z(1) - nAin*(1-fA);
end % of internal function stop
```

Listing 4. Modifications to specify the final conditions.

The sixth and 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 reactor volume and the outlet temperature so that they will be returned by the function. This is shown in Listing 5.

```
% set the return value
V = t_f;
Tout = z(3);
end % of Example_30_1_pfr.m
```

Listing 5. Final modification where the return variables are set to the proper values.

**MATLAB function for evaluating the two reactor configurations.** In the configuration featuring an integrated heat exchanger and PFR, the solution showed that a set of non-linear (non-differential) equations must be solved (the heat exchanger design 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 for the function for evaluating the two reactor configurations.

Before it can be used to solve a problem, `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_1.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_1`. 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 6 shows the results of all these modifications.

```
% Modified version of the MATLAB template file SolvNonDif.m used in the
% solution of Example 30.1 of "A First Course on Kinetics and Reaction
% Engineering."
%
function Example_30_1
    % Known quantities and constants (in consistent units)
    VFR = 750.0; % L/min
    CAa = 3.8; % mol/L
    Ta = 25 + 273.15; % K
    UA = 1500.0; % kJ/min/K
    dH = -79.8; % kJ/mol
    cP = 987.0 * 4.184e-3; % kJ/L/K
    fA = 0.8;
    nAa = VFR*CAa;
```

*Listing 6. Initial modifications to the template file SolvNonDif.m.*

The next required modification is to provide code that evaluates the functions,  $f_4$  and  $f_5$ , in equations (22) and (23). 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 where, for example, you use  $z(2)$  when it you should use  $z(3)$ . Listing 7 shows how this was done in Example\_30\_1.m and how code to evaluate the functions was entered so that the vector named  $f$  contains the value of  $f_4$  as its first element, the value of  $f_5$  as its second element.

```
% Function that evaluates the equations
function f = evalEqns(z)
    Tb = z(1);
    Td = z(2);
    [V,Tc] = Example_30_1_pfr(Tb);
    AMTD = 0.5*((Td-Ta) + (Tc-Tb));
    f = [
        VFR*cP*(Tb-Ta) - VFR*cP*(Tc-Td)
        VFR*cP*(Tb-Ta) - UA*AMTD
    ];
end % of internal function evalEqns
```

*Listing 7. 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  $z\_guess$ . They must be entered in

order with  $z_1$  first and  $z_n$  last. If you follow my recommendation in the previous modification, the statements where variables with more meaningful names were defined will serve as a key to remind you which variable is  $z_1$ , which is  $z_2$ , and so on. Listing 8 shows that I simply guessed that  $T_b$  would be 25 K larger than  $T_a$  and  $T_d$  would be 50 K larger than  $T_a$ .

```
% guesses for the solution
z_guess = [
    Ta + 25
    Ta + 50
];
```

*Listing 8. 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 required PFR volume for the integrated heat exchanger and PFR. This value is displayed as `V_backmix`. After that, the reactor configuration without a heat exchanger is evaluated by again calling the MATLAB function for solving the PFR design equations, but using the feed temperature,  $T_a$ , as the PFR inlet temperature. The volume of the PFR is returned and displayed as `V_no_backmix`. These modifications are shown in Listing 9.

```
Tb = z(1);
[V,T] = Example_30_1_pfr(Tb);
V_backmix = V

% Case with no backmix, feed to reactor is at Ta
[V,T]=Example_30_1_pfr(Ta);
V_no_backmix = V
```

*Listing 9. Final modification where the results of solving the heat exchanger design equations are used to calculate the PFR volume.*

At this point, `Example_30_1.m` can be executed by typing `Example_30_1` at the MATLAB command prompt. The resulting output is shown in Listing 10. It is important to check first, that the PRF code stopped because the specified conversion was reached and second, that the non-linear equation solver converged. In the former case, a warning would have been printed if the specified volume was reached instead of the specified conversion. No such warning appears, so the proper final condition was reached. In the latter case, the output reports that the equations were solved. Therefore, the results produced by the calculations can be accepted.

```
Example_30_1
```

```
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 =  
 324.3706  
 356.8946
```

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

```
f =  
 1.0e-09 *  
-0.1746  
 0.0041
```

```
V_backmix =  
 8.8656e+03
```

```
V_no_backmix =  
 3.7398e+04
```

*Listing 10. Output from the execution of Example\_30\_1.*