

# A First Course on Kinetics and Reaction Engineering

## Example 31.2

### Problem Purpose

This problem will help you determine whether you have mastered the learning objectives for this unit. It illustrates the analysis of a recycle PFR.

### Problem Statement

A recycle PFR is being designed to produce D from A according to reaction (1), but the undesired reaction (2) also takes place. These reactions are irreversible, and their rates are given by equations (3) and (4). The reaction takes place in the liquid phase; the heat capacity of the liquid is constant and equal to  $281 \text{ J L}^{-1} \text{ K}^{-1}$ . The heat of reaction (1) is  $-52 \text{ kJ mol}^{-1}$  and that of reaction (2) is  $-38 \text{ kJ mol}^{-1}$ . The reactor will be adiabatic with a diameter equal to  $0.1 \text{ dm}$ . The feed to the process will flow at a rate of  $150 \text{ L min}^{-1}$  and a temperature of  $420 \text{ K}$ ; the concentration of A is  $4 \text{ mol L}^{-1}$  and there is no D or U in the feed. If the recycle ratio is equal to 4, how long must the reactor be to achieve 50% overall conversion, what will the resulting overall selectivity (mol D per mol U) equal, and what will the outlet temperature be?



$$r_1 = (5.07 \times 10^6 \text{ min}^{-1}) \exp\left\{\frac{-65 \text{ kJ mol}^{-1}}{RT}\right\} C_A \quad (3)$$

$$r_2 = (8.64 \times 10^4 \text{ L mol}^{-1} \text{ min}^{-1}) \exp\left\{\frac{-58.1 \text{ kJ mol}^{-1}}{RT}\right\} C_A^2 \quad (4)$$

### Problem Analysis

In this problem we are given the rate expressions and asked questions about the performance of the reactor, so this is a reaction engineering problem. The reactor is a PFR, and we will be analyzing its steady state performance since the problem does not mention any changes in its operation or things like start-up or shut-down. Insufficient information is provided to write a momentum balance, so we will assume pressure drop is negligible. Similarly, since the reaction takes place in the liquid phase, we will assume the fluid density to be constant. In order to answer the questions posed we will write mole and energy balance design equations for the PFR. It is likely that we will not be able to solve these equations independently. Instead, mole and energy balances will also be generated for the recycle mixing point and they will be solved simultaneously with the PFR design equations.

**Problem Solution**

The system can be represented schematically as shown in Figure 1. Reading through the problem statement we find the following quantities to be specified:  $\tilde{C}_p = 281 \text{ J L}^{-1} \text{ K}^{-1}$ ,  $\Delta H_1(T) = -52 \text{ kJ mol}^{-1}$ ,  $\Delta H_2(T) = -38 \text{ kJ mol}^{-1}$ ,  $D = 0.1 \text{ dm}$ ,  $\dot{V}_a = 150 \text{ L min}^{-1}$ ,  $T_a = 420 \text{ K}$ ,  $C_{A,a} = 4 \text{ mol L}^{-1}$ ,  $C_{D,a} = 0 \text{ mol L}^{-1}$ ,  $C_{U,a} = 0 \text{ mol L}^{-1}$ ,  $R_R = 4$  and  $f_{A,overall} = 0.5$ .



Figure 1. Schematic representation of the recycle PFR system of Example 31.1.

The feed molar flow rates can be calculated from the known feed concentrations and volumetric flow rate according to equations (5) through (7). Since the fluid density is assumed to be constant and the system operates at steady state, the outlet volumetric flow must equal the feed volumetric flow, equation (8). The known recycle ratio then permits the calculation of the volumetric recycle flow, equation (9). A mass balance at the mixing point, together with the assumption of constant density then allows calculation of the volumetric flow rate entering the reactor, equation (10). Again, since the fluid density is constant the volumetric flow leaving the reactor must be the same, equation (11).

$$\dot{n}_{A,a} = \dot{V}_a C_{A,a} \quad (5)$$

$$\dot{n}_{D,a} = \dot{V}_a C_{D,a} \quad (6)$$

$$\dot{n}_{U,a} = \dot{V}_a C_{U,a} \quad (7)$$

$$\dot{V}_d = \dot{V}_a \quad (8)$$

$$\dot{V}_r = R_R \dot{V}_d \quad (9)$$

$$\dot{V}_b = \dot{V}_a + \dot{V}_r \quad (10)$$

$$\dot{V}_c = \dot{V}_b \quad (11)$$

The generalized steady state PFR mole and energy balances are given in equations (12) and (13). In the present case there are two reactions, so the summations over  $j$  expand to give two terms. The reactor is adiabatic, so the heat transfer term in the energy balance equation is set to zero. The volumetric heat capacity of the fluid as a whole is provided, so the summation over the individual species molar heat capacities is replaced. After all these simplifications the mole balances on A, D and U reduce to equations (14) through (16) and the energy balance reduces to equation (17).

$$\frac{d\dot{n}_i}{dz} = \frac{\pi D^2}{4} \sum_{\substack{j=\text{all} \\ \text{reactions}}} \nu_{i,j} r_j \quad (12)$$

$$\pi DU(T_e - T) = \left( \sum_{\substack{i=\text{all} \\ \text{species}}} \dot{n}_i \hat{C}_{pi} \right) \frac{dT}{dz} + \frac{\pi D^2}{4} \sum_{\substack{j=\text{all} \\ \text{reactions}}} (r_j \Delta H_j) \quad (13)$$

$$\frac{d\dot{n}_A}{dz} = f_1(z, \dot{n}_A, \dot{n}_D, \dot{n}_U, T) = -\frac{\pi D^2}{4} (r_1 + r_2) \quad (14)$$

$$\frac{d\dot{n}_D}{dz} = f_2(z, \dot{n}_A, \dot{n}_D, \dot{n}_U, T) = \frac{\pi D^2}{4} r_1 \quad (15)$$

$$\frac{d\dot{n}_U}{dz} = f_3(z, \dot{n}_A, \dot{n}_D, \dot{n}_U, T) = \frac{\pi D^2}{4} r_2 \quad (16)$$

$$\frac{dT}{dz} = f_4(z, \dot{n}_A, \dot{n}_D, \dot{n}_U, T) = -\frac{\pi D^2 (r_1 \Delta H_1(T) + r_2 \Delta H_2(T))}{4 \dot{V} \tilde{C}_p} \quad (17)$$

In order to solve these equations numerically, no matter what software one uses, it will be necessary to provide (a) initial values (at the reactor inlet) of the independent ( $z$ ) and dependent ( $\dot{n}_A$ ,  $\dot{n}_D$ ,  $\dot{n}_U$  and  $T$ ) variables, (b) a final value (at the reactor outlet) of either the independent variable or one of the dependent variables and (c) code that is given values of the independent and dependent variables and uses them to evaluate functions  $f_1$  through  $f_4$  in equations (14) through (17).

The initial values, equations (18) through (21), are not known, and consequently it will not be possible to solve the PFR design equations independently. The final value of the molar flow rate of A,  $\dot{n}_{A,c}$ , can be computed since the overall conversion of A is specified. That is, the outlet molar flow rate of A can be computed from the known feed molar flow rate and overall conversion, equation (22). A mole balance on A at the split, point 2 in Figure 1, then allows the calculation of the molar flow rate of A leaving the reactor, equation (23).

$$\dot{n}_A(z=0) = \dot{n}_{A,b} \quad (18)$$

$$\dot{n}_D(z=0) = \dot{n}_{D,b} \quad (19)$$

$$\dot{n}_U(z=0) = \dot{n}_{U,b} \quad (20)$$

$$T(z=0) = T_b \quad (21)$$

$$\dot{n}_{A,d} = \dot{n}_{A,a} (1 - f_{A,\text{overall}}) \quad (22)$$

$$\dot{n}_A(z=L) = \dot{n}_{A,c} = \dot{n}_{A,r} + \dot{n}_{A,d} = R_R \dot{n}_{A,d} + \dot{n}_{A,d} = (1 + R_R) \dot{n}_{A,d} \quad (23)$$

Looking at functions  $f_1$  through  $f_4$  in equations (14) through (17), the code that evaluates them will be given values of  $z$ ,  $\dot{n}_A$ ,  $\dot{n}_D$ ,  $\dot{n}_U$  and  $T$ , while  $\tilde{C}_p$ ,  $\Delta H_1(T)$ ,  $\Delta H_2(T)$ ,  $D$  and  $\dot{V}_a$  are known constants. Equations (3) and (4) can be used to calculate the reaction rates, but in order to do so the concentration of A must first be calculated. The volumetric flow rate within the reactor will equal  $\dot{V}_b$  (a known constant) since the fluid density is constant, and the code will be given a value for  $\dot{n}_A$ , so the concentration can be computed using equation (24). Therefore, the code that evaluates functions  $f_1$  through  $f_4$  in equations (14) through (17) can be written and used to numerically solve the PFR design equations.

$$C_A = \frac{\dot{n}_A}{\dot{V}_b} \quad (24)$$

To summarize, the only things needed in order to solve the PFR design equations numerically are the initial values, that is  $\dot{n}_{A,b}$ ,  $\dot{n}_{D,b}$ ,  $\dot{n}_{U,b}$  and  $T_b$ . Mole and energy balances on the mixing point, labeled 1 in Figure 1, can be used to find these missing values. In the informational reading for this unit, a general mole balance was written for the mixing point, and then a mole balance at the split point and the definition of the recycle ratio were used to eliminate the molar flow in the recycle stream from the equation, replacing it with the molar flow rate leaving the reactor. That equation is reproduced here as equation (25). An energy balance on the mixing point is given in equation (26) in terms of the molar flow rates of the individual species. Writing the mole balance of equation (25) for each of the three species gives equations (27) through (29). Replacing the molar heat capacities in equation (26) with the volumetric heat capacity of the fluid as a whole leads to the form of the energy balance given in equation (30).

$$\dot{n}_{i,a} + \frac{R_R \dot{n}_{i,c}}{1 + R_R} - \dot{n}_{i,b} = 0 \quad (25)$$

$$\sum_{\substack{i=all \\ species}} \dot{n}_{i,a} \int_{T_a}^{T_b} \hat{C}_{p,i} dT + \sum_{\substack{i=all \\ species}} \dot{n}_{i,r} \int_{T_c(=T_r)}^{T_b} \hat{C}_{p,i} dT = 0 \quad (26)$$

$$\dot{n}_{A,a} + \frac{R_R \dot{n}_{A,c}}{1 + R_R} - \dot{n}_{A,b} = f_5(\dot{n}_{A,b}, \dot{n}_{D,b}, \dot{n}_{U,b}, T_b) = 0 \quad (27)$$

$$\dot{n}_{D,a} + \frac{R_R \dot{n}_{D,c}}{1 + R_R} - \dot{n}_{D,b} = f_6(\dot{n}_{A,b}, \dot{n}_{D,b}, \dot{n}_{U,b}, T_b) = 0 \quad (28)$$

$$\dot{n}_{U,a} + \frac{R_R \dot{n}_{U,c}}{1 + R_R} - \dot{n}_{U,b} = f_7(\dot{n}_{A,b}, \dot{n}_{D,b}, \dot{n}_{U,b}, T_b) = 0 \quad (29)$$

$$\dot{V}_a \tilde{C}_p(T_b - T_a) + \dot{V}_r \tilde{C}_p(T_b - T_c) = f_8(\dot{n}_{A,b}, \dot{n}_{D,b}, \dot{n}_{U,b}, T_b) = 0 \quad (30)$$

Looking at the mixing point design equations, (27) through (30), it can be seen that they contain eight unknown quantities ( $\dot{n}_{A,b}$ ,  $\dot{n}_{D,b}$ ,  $\dot{n}_{U,b}$ ,  $T_b$ ,  $\dot{n}_{A,c}$ ,  $\dot{n}_{D,c}$ ,  $\dot{n}_{U,c}$  and  $T_c$ ) while there are only four equations. Clearly, these equations can only be solved to find the value of four of the unknowns. The key to solving these equations numerically is to choose the stream  $b$  quantities as the unknowns to be solved for, as equations (27) through (30) indicate. In order to solve the equations numerically, it will be necessary to provide (a) guesses for the values of the unknowns, which is easy to do, and (b) code that is given values for the unknowns and that uses them to evaluate the functions  $f_5$  through  $f_8$  in equations (27) through (30).

By choosing  $\dot{n}_{A,b}$ ,  $\dot{n}_{D,b}$ ,  $\dot{n}_{U,b}$  and  $T_b$  as the unknowns to be solved for, writing that code is easy. The code will be given value for  $\dot{n}_{A,b}$ ,  $\dot{n}_{D,b}$ ,  $\dot{n}_{U,b}$  and  $T_b$ . Those values can then be used as the initial conditions for the numerical solution of the PFR design equations. Solving the PFR design equations yields values for  $L$ ,  $\dot{n}_{A,c}$ ,  $\dot{n}_{D,c}$ ,  $\dot{n}_{U,c}$  and  $T_c$ . Once that is done, functions  $f_5$  through  $f_8$  in equations (27) through (30) can be evaluated.

In that way, equations (27) through (30) can be solved numerically to find  $\dot{n}_{A,b}$ ,  $\dot{n}_{D,b}$ ,  $\dot{n}_{U,b}$  and  $T_b$ . Those results can then be used again to solve the PFR design equations to find  $L$ ,  $\dot{n}_{A,c}$ ,  $\dot{n}_{D,c}$ ,  $\dot{n}_{U,c}$  and  $T_c$ . Doing so, one finds that the reactor must have a length of 117 dm to meet the specification of 50% conversion and the outlet temperature will equal 780 K. The molar flow rates of D and U leaving the process can then be calculated using equation (31). Doing so shows that the selectivity of the process will be 8.7 moles of D per mole of U.

$$\dot{n}_{i,c} = \dot{n}_{i,r} + \dot{n}_{i,d} = R_R \dot{n}_{i,d} + \dot{n}_{i,d} \quad \Rightarrow \quad \dot{n}_{i,d} = \frac{\dot{n}_{i,c}}{(1 + R_R)} \quad (31)$$

Using a recycle PFR in this situation has two benefits. First, since the reaction is adiabatic, the recycle stream is hotter than the feed. When the two streams are mixed, the resulting stream entering the PFR has a higher temperature than the feed. As a consequence, the reaction rate inside the reactor is larger than it would have been if there were no recycle. Thus, with recycle we expect the required reactor volume to be smaller. Secondly, the instantaneous selectivity parameter, equation (32), shows that high selectivity is favored by lower concentrations of A. The high recycle ratio in this case dilutes the feed so that the concentration of A entering the reactor is much smaller than the feed concentration of A, and this favors the production of D over the production of U. There are some trade-offs, however. While the higher temperature at the reactor inlet leads to higher rates, it also lowers the selectivity as can be seen from equation (32) noting that  $E_2$  is less than  $E_1$ . In addition, the greater volumetric flow through the reactor means that a larger reactor volume is needed to achieve a given space time compared to a system with no recycle. Still, in this case the use of recycle leads to very high selectivity for the desired product. In a real-world situation, the decision on whether to use recycle and, if so, what recycle ratio to use would be determined on the basis of an economic analysis.

$$S_{D/U} = \frac{r_1}{r_2} = \left( \frac{k_{0,1}}{k_{0,2}} \right) \exp\left( \frac{E_2 - E_1}{RT} \right) C_A^{-1} \quad (32)$$

### Calculation Details Using MATLAB

Two MATLAB functions were written to perform the calculations needed to solve this problem. The first, Example\_31\_2\_pfr, takes the inlet molar flows and temperature as arguments and solves the PFR design equations, returning the reactor length and the outlet stream molar flows and temperature. The second MATLAB function, Example\_31\_2, performs the calculations needed to evaluate the recycle PFR, solving the mixing point 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\_31\_2\_pfr.m; a copy of the final version of that file accompanies this solution. Since the function name must match the filename, I changed the name of the function to Example\_31\_2\_pfr. At the same time, I changed the function declaration so that the inlet molar flows and temperature were passed to the function as arguments, and so that the reactor length, outlet molar flows and 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. Listing 1 shows the first part of the modified file.

The second *required* modification involves entering the code to evaluate functions  $f_1$  through  $f_4$  in equations (14) through (17). The internal function that evaluates the functions is named odeqns; the independent variable is passed into it as the scalar  $t$  and the dependent variables are passed in as the vector  $z$ . I find it useful to start by extract the dependent variables from that vector and assign their values to local variables with the names similar to those used in the problem statement. Once that is done, the concentration of A, the rate coefficients and the reaction rates can be calculated as described in the solution. The functions  $f_1$  through  $f_4$  in equations (14) through (17) then can be evaluated, saving the results in the vector  $dzdt$  which is returned by odeqns. These modifications are shown in Listing 2.

```

% Modified version of the AFCoKaRE MATLAB template file SolvIVDifI.m used
% to model the PFR in Example 31.2 of "A First Course on Kinetics and
% Reaction Engineering."
%
function [nAout,nDout,nUout,Tout,L] = Example_31_2_pfr(nAin,nDin,nUin,Tin)
    % Known quantities and constants (in consistent units)
    k01 = 5.07e6; % /min
    E1 = 65.; % kJ/mol
    k02 = 8.64e4; % dm^3/mol/min
    E2 = 58.1; % kJ/mol
    Cp = 281./1000.; % kJ/dm^3/K
    dH1 = -52.; % kJ/mol
    dH2 = -38.; % kJ/mol
    D = 0.1; % dm
    VFRA = 150.; % dm^3/min
    CAa = 4.0; % mol/dm^3
    RR = 4.0;
    fA = 0.5;
    R = 8.31446/1000.; % kJ/mol/K
    VFRd = VFRA;
    VFRr = RR*VFRd;
    VFRb = VFRr + VFRA;
    nAa = VFRA*CAa;

```

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

```

% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    % t is the value of the independent variable
    % z is a vector containing the values of the dependent variables
    nA = z(1);
    nD = z(2);
    nU = z(3);
    T = z(4);
    CA = nA/VFRb;
    k1 = k01*exp(-E1/R/T);
    k2 = k02*exp(-E2/R/T);
    r1 = k1*CA;
    r2 = k2*CA^2;
    dzdt = [
        -pi()*D^2/4*(r1 + r2);
        pi()*D^2/4*r1;
        pi()*D^2/4*r2;
        -pi()*D^2/4*(r1*dH1 + r2*dH2)/VFRb/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  $z$  in the internal function `odeqns`. The results of performing this modification are shown in Listing 3.

```
% Initial and final values
t0 = 0;
z0 = [
    nAin
    nDin
    nUin
    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,  $z$ ) that is much greater than the actual reactor length. Of course, you don't know what the length is at this point, so all you can do is set  $t_f$  to a large number. The reactor length 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. For this problem, the final condition is given in equation (23). Noting that  $z(1)$  will equal the current value of  $\dot{n}_{A,c}$  the current value of  $\dot{n}_{A,d}$  can be calculated using equation (31). The defining equation for the overall conversion, equation (33), can then be rearranged to give an expression that will equal zero at the specified final conversion; the variable `stop_when` is simply set equal to that expression as shown in Listing 4.

$$f_{A,overall} = \frac{\dot{n}_{A,a} - \dot{n}_{A,d}}{\dot{n}_{A,a}} \quad \Rightarrow \quad f_{A,overall} - \frac{\dot{n}_{A,a} - \dot{n}_{A,d}}{\dot{n}_{A,a}} = 0 \quad (33)$$

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 length and the outlet molar flows and temperature so that they will be returned by the function. This is shown in Listing 5.



```

tf = 1.0E4;
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;

    nAd = z(1)/(1+RR);
    stop_when = fA - (nAa - nAd)/nAa;
end % of internal function stop

```

Listing 4. Modifications to specify the final conditions.

```

% Set the return values
L = t_f;
nAout = z(1);
nDout = z(2);
nUout = z(3);
Tout = z(4);

end % of Example_31_2_pfr.m

```

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

**MATLAB function for evaluating the recycle PFR.** Analysis of the recycle PFR involves the simultaneous solution of the mixing point design equations and the PFR design equations as described in the solution. The mixing point design equations are non-linear, non-differential 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 analyzing the recycle PFR.

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\_31\_2.m; a copy of the final version of that file accompanies this solution. Since the function name must match the filename, I changed the name of the function to Example\_31\_2. At the same time, 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 AFCoKaRE MATLAB template file SolvNonDif.m used
% to model the recycle PFR in Example 31.2 of "A First Course on Kinetics
% and Reaction Engineering."
%
function Example_31_2
    % Known quantities and constants (in consistent units)
    Cp = 281./1000.; % kJ/dm^3/K
    VFRa = 150.; % dm^3/min
    Ta = 420; % K
    CAa = 4.0; % mol/dm^3
    RR = 4.0;
    fA = 0.5;
    R = 8.31446/1000.; % kJ/mol/K
    VFRd = VFRa;
    VFRr = RR*VFRd;
    VFRb = VFRr + VFRa;
    VFRc = VFRb;
    nAa = VFRa*CAa;
    nDa = 0;
    nUa = 0;

```

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

The next required modification is to provide code that evaluates the functions,  $f_5$  through  $f_8$ , in equations (27) through (30). The internal function that evaluates the functions is named `evalEqns`; values for the unknowns are passed in as the vector `z`. I find it useful to start by extract the unknowns from that vector and assign their values to local variables with the names similar to those used in the problem statement. Those unknowns can then be used to solve the PFR design equations by calling the function `Example_31_2_pfr` that was just described. Upon doing so, the molar flows and temperature of stream c are returned. The functions,  $f_5$  through  $f_8$ , in equations (27) through (30) then can be evaluated, saving the results in the vector `f` which is returned by `evalEqns`. Listing 7 shows these modifications.

```

% Function that evaluates the equations
function f = evalEqns(z)
    nAb = z(1);
    nDb = z(2);
    nUb = z(3);
    Tb = z(4);
    [nAc, nDc, nUc, Tc, L] = Example_31_2_pfr(nAb, nDb, nUb, Tb);
    f = [
        nAa + RR*nAc/(1+RR) - nAb;
        nDa + RR*nDc/(1+RR) - nDb;
        nUa + RR*nUc/(1+RR) - nUb;
        VFRa*Cp*(Tb-Ta) + VFRr*Cp*(Tb-Tc);
    ];
end % of internal function evalEqns

```

*Listing 7. Modifications 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`. This is perhaps the trickiest part of the solution because bad guesses may cause the solver to converge to an undesired steady state (multiple steady states are possible), or they may cause the solver to fail to converge to any solution. Listing 8 shows guesses that worked for this particular problem.

```
% guesses for the solution
z_guess = [
    0.5*nAa*(1+RR);
    0.1*nAa*(1+RR);
    0.1*nAa*(1+RR);
    Ta + 20;
];
```

*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, that is, the mixing point design equations, have been solved. The first thing that must be done is to use the results to solve the PFR design equations to find the reactor length and the molar flows and temperature of stream *c*. The problem asks for the outlet temperature, but since point 2 is a simple stream split, the outlet temperature (stream *d*) will be equal to the temperature of stream *c*. The outlet molar flow rates of D and U can be calculated using equation (31), and then the selectivity (moles of D per mole of U) can be calculated using equation (32). These modifications are shown in Listing 9.

```
% Solve the PFR design equations
nAb = z(1);
nDb = z(2);
nUb = z(3);
Tb = z(4);
[nAc, nDc, nUc, Tc, L] = Example_31_2_pfr(nAb, nDb, nUb, Tb);
nDd = nDc/(1+RR);
nUd = nUc/(1+RR);
L
sel_D_per_U = nDd/nUd
Td = Tc

end % of Example_31_2.m
```

*Listing 9. Final modification where the results of solving the mixing point design equations are used to calculate the PFR length, the process selectivity and the product stream temperature.*

At this point, `Example_31_2.m` can be executed by typing `Example_31_2` 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 reactor length 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_31_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 =
  1.0e+03 *

    1.8000
    1.0765
    0.1235
    0.7079

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

f =
  1.0e-10 *

    0
    0.0023
    0
   -0.2365

L =
  116.9273

sel_D_per_U =
    8.7195

Td =
  779.8548
```

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