A First Course on Kinetics and Reaction Engineering

Example 29.2

Problem Purpose

This problem will help you determine whether you have mastered the learning objectives for this unit. It illustrates the analysis of a simple series reactor network with parallel reactions taking place.

Problem Statement

A 200 L adiabatic CSTR and a 200 L adiabatic PFR are going to be connected in series and used to convert A according to irreversible reactions (1) and (2), where D is the desired product and U is an undesired, low-value byproduct. The feed consists of an aqueous stream containing 2.5 mol A L^{-1}, flowing at a rate of 100 L min^{-1} at 38 °C. The liquid density may be assumed to be constant. The heat of reaction (1) is -21,500 cal mol^{-1} and that of reaction (2) is -24,000 cal mol^{-1}. The heat capacity of the solution is constant and equal to 1.0 cal cm^{-3} K^{-1}. The rate expressions for reactions (1) and (2) are given in equations (3), and (4), respectively. Compare the overall conversion of A and selectivity (mol D per mol U) for the configuration with the CSTR first to those with the PFR first.

$$A \rightarrow D \tag{1}$$

$$A \rightarrow U \tag{2}$$

$$r_1 = \left(3.40\times 10^5 \text{ min}^{-1}\right)\exp\left\{\frac{-9000 \text{ cal mol}^{-1}}{RT}\right\}C_A \tag{3}$$

$$r_2 = \left(1.67\times 10^{14} \text{ L mol}^{-1} \text{ min}^{-1}\right)\exp\left\{\frac{-23{,}000 \text{ cal mol}^{-1}}{RT}\right\}C_A^2 \tag{4}$$

Problem Analysis

In this problem the rate expressions for the reactions are known and we are asked about reactor properties, so this is a reaction engineering problem. It involves both a CSTR and a PFR. Their steady state operation will be analyzed since the problem makes no mention of any changes that would cause transient behavior. For both reactors we will write mole balances on every reactant and product (in this case A, D and U) and an energy balance. The problem does not provide sufficient information to include a mechanical energy balance for the PFR, so we will assume negligible pressure drop. The reactors operate adiabatically, and for the PFR, the reactor diameter and length are not specified, therefore, we'll need to rewrite the PFR balance equations so that the cumulative volume is the independent variable instead of the axial position. In both cases, we will solve the design equations and then use the results to answer the questions posed.

A qualitative analysis shows this to be an interesting problem. The reactions are exothermic and the kinetics are regular. Hence for adiabatic operation the temperature will increase as the reaction proceeds while the concentration of reactant will decrease. As seen in past problems, this typically leads to a rate that initially increases, reaches a maximum value and then decreases. In addition, we have parallel reactions taking place. The instantaneous selectivity parameter, $S_{D/U}$, is the ratio of the desired to undesired reaction rates, as given in equation (5), which shows that the selectivity for the desired product increases as the concentration of the reactant, A, decreases. At the same time, since E2 - E1 is a positive number, the selectivity decreases as the temperature increases. Clearly there will be trade-offs inherent in the system. Having the CSTR last will mean high temperature and low concentration which oppose each other with respect to selectivity and conversion. Having the CSTR first might improve conversion, but the opposing effects of higher reactant concentration and lower temperature make it impossible to qualitatively assess how the selectivity will be affected. It will simply be necessary to simulate both systems quantitatively in order to find out.

$$S_{D/U} = \frac{r_D}{r_U} = \frac{k_{0,1}\exp\left[\frac{-E_1}{RT}\right]C_A}{k_{0,2}\exp\left[\frac{-E_2}{RT}\right]C_A^2} = \frac{k_{0,1}}{k_{0,2}}\exp\left[\frac{E_2 - E_1}{RT}\right]\frac{1}{C_A} \tag{5}$$

Problem Solution

Before analyzing the two specified configurations, we can write the design equations for the two reactors since they will be the same in either configuration. The general steady state CSTR mole and energy balance design equations are given in equations (6) and (7). There are two reactions taking place, so the summation over the reactions in both equations will expand to give two terms. The reactors operate adiabatically and perform negligible work through moving shafts and boundaries, so the heat input and work terms will equal zero. Finally, 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 replaced by a single sensible heat term. With those simplifications, steady state CSTR mole balances for A, D and U and a steady state CSTR energy balance will take the forms given in equations (8) through (11), where the subscripts “in” and “out” designate the inlet and outlet flow streams.

$$0 = \dot{n}_i^0 - \dot{n}_i + V \sum_{\substack{j=all\\reactions}} \nu_{i,j} r_j \tag{6}$$

$$0 = \sum_{\substack{i=all\\species}} \left(\dot{n}_i^0 \int_{T^0}^{T} \hat{C}_{pi}\, dT \right) + V \sum_{\substack{j=all\\reactions}} r_j \Delta H_j\left(T\right) - \dot{Q} + \dot{W} \tag{7}$$

$$0 = \dot{n}_{A,in} - \dot{n}_{A,out} - V\left(r_{1,out} + r_{2,out}\right) \tag{8}$$

$$0 = \dot{n}_{D,in} - \dot{n}_{D,out} + Vr_{1,out} \tag{9}$$

$$0 = \dot{n}_{U,in} - \dot{n}_{U,out} + Vr_{2,out} \tag{10}$$

$$0 = \dot{V}_{in}\tilde{C}_p\left(T_{out} - T_{in}\right) + V\left(r_{1,out}\Delta H_1\left(T_{out}\right) + r_{2,out}\Delta H_2\left(T_{out}\right)\right) \tag{11}$$

The problem statement gives values for V (200 L), $\dot{V}$ (100 L min^{-1}, which is constant since the fluid density is constant), ΔH_1 (-21,500 cal mol^{-1}, independent of T), ΔH_2 (-24,000 cal mol^{-1}, independent of T) and $\tilde{C}_p$ (1.0 cal cm^{-3} K^{-1}), and, of course, the ideal gas constant is known. The remaining quantities, (inlet and outlet molar flow rate of each species and inlet and outlet temperature) are either unknown, or their value depends upon whether the CSTR precedes the PFR or vice versa.

Similarly, the general steady state PFR mole balance is given by equation (12), and the general steady state PFR energy balance, by equation (13). 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 (14) needs to be used to replace dz in equations (12) and (13). As was the case for the CSTR, there are two reactions taking place, so the summation over the reactions in both equations will expand to give two terms. The PFR also operates adiabatically, so the heat input term equals zero, and again, 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 (15) through (18), where again the subscripts "in" and "out" designate the inlet and outlet flow streams.

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

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

$$dV = \frac{\pi D^2}{4}dz \tag{14}$$

$$\frac{d\dot{n}_A}{dV} = -r_1 - r_2; \quad \dot{n}_A\left(0\right) = \dot{n}_{A,in} \tag{15}$$

$$\frac{d\dot{n}_D}{dV} = r_1; \quad \dot{n}_D\left(0\right) = \dot{n}_{D,in} \tag{16}$$

$$\frac{d\dot{n}_U}{dV} = r_2; \quad \dot{n}_U\left(0\right) = \dot{n}_{U,in} \tag{17}$$

$$\frac{dT}{dV} = -\frac{r_1\Delta H_1\left(T\right) + r_2\Delta H_2\left(T\right)}{\dot{V}\tilde{C}_p}; \quad T\left(0\right) = T_{in} \tag{18}$$

Case 1, CSTR followed by PFR. Figure 1 shows a schematic representation of this case with the three flow streams labeled. Thus, in the CSTR design equations, “in” is equivalent to “a” and “out” is equivalent to “b”. Similarly, in the PFR design equations, “in” is equivalent to “b” and “out” is equivalent to “c”. Making these substitutions leads to the design equations given in equations (19) through (26).

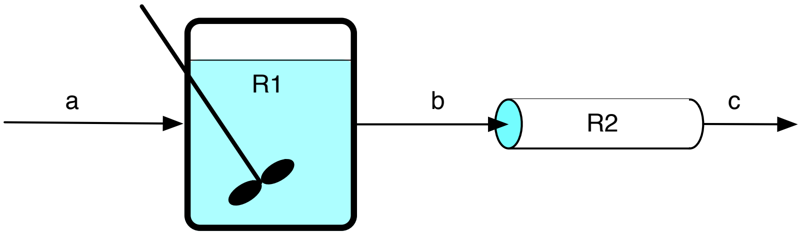

Figure 1. Schematic representation of the case with the CSTR followed by the PFR.

$$0 = f_1\left(\dot{n}_{A,b},\dot{n}_{D,b},\dot{n}_{U,b},T_b\right) = \dot{n}_{A,b} - \dot{n}_{A,a} - Vr_{1,b} \tag{19}$$

$$0 = f_2\left(\dot{n}_{A,b},\dot{n}_{D,b},\dot{n}_{U,b},T_b\right) = \dot{n}_{D,b} - \dot{n}_{D,a} + V\left(r_{1,b} - r_{2,b}\right) \tag{20}$$

$$0 = f_3\left(\dot{n}_{A,b},\dot{n}_{D,b},\dot{n}_{U,b},T_b\right) = \dot{n}_{U,b} - \dot{n}_{U,a} + Vr_{2,b} \tag{21}$$

$$0 = f_4\left(\dot{n}_{A,1},\dot{n}_{D,1},\dot{n}_{U,1},T_1\right) = \dot{V}_0\tilde{C}_p\left(T_1 - T_0\right) + V\left(r_1\Delta H_1\left(T_1\right) + r_2\Delta H_2\left(T_1\right)\right) \tag{22}$$

$$\frac{d\dot{n}_A}{dV} = f_5\left(V,\dot{n}_A,\dot{n}_D,\dot{n}_U,T\right) = -r_1; \qquad \dot{n}_A\left(0\right) = \dot{n}_{A,b} \tag{23}$$

$$\frac{d\dot{n}_D}{dV} = f_6\left(V,\dot{n}_A,\dot{n}_D,\dot{n}_U,T\right) = r_1 - r_2; \qquad \dot{n}_D\left(0\right) = \dot{n}_{D,b} \tag{24}$$

$$\frac{d\dot{n}_U}{dV} = f_7\left(V,\dot{n}_A,\dot{n}_D,\dot{n}_U,T\right) = r_2; \qquad \dot{n}_U\left(0\right) = \dot{n}_{U,b} \tag{25}$$

$$\frac{dT}{dV} = f_8\left(V,\dot{n}_A,\dot{n}_D,\dot{n}_U,T\right) = -\frac{r_1\Delta H_1\left(T\right) + r_2\Delta H_2\left(T\right)}{\dot{V}\tilde{C}_p}; \qquad T\left(0\right) = T_b \tag{26}$$

The CSTR design equations, (19) through (22), are non-linear, non-differential equations. In order to solve them numerically one must (a) identify a set of unknown quantities, equal in number to the number of equations, (b) provide a guess for the value of each unknown, and (c) provide code that when given values for the unknowns, evaluates the equations. You can see in the equations above that I have identified $\dot{n}_{A,b}$, $\dot{n}_{D,b}$, $\dot{n}_{U,b}$ and T_b as the unknowns, and I can guess a value for each. The code I provide will need to evaluate the functions f_1 through f_4, given values for $\dot{n}_{A,b}$, $\dot{n}_{D,b}$, $\dot{n}_{U,b}$ and T_b. To do that, I’ll need to be able to provide or calculate a value for every other quantity that appears in those functions. We've already seen that V, $\dot{V}_a = \dot{V}_b = \dot{V}_c$, ΔH_1, ΔH_2 and $\tilde{C}_p$ are known constants. The inlet molar flow rate of A is also a constant that can be calculated from the inlet volumetric flow rate and the inlet concentration of A ($C_{A,a}$ = 1.3 mol L^{-1}) given in the problem statement using equation (27). The inlet molar flow rates of D and U are zero since the problem does not mention that any D or U are present in the feed.

$$\dot{n}_{A,a} = C_{A,a}\dot{V}_a \tag{27}$$

The only other quantities, apart from the unknowns, appearing in functions f_1 through f_4 are the rates, $r_{1,b}$ and $r_{2,b}$. The rates can be calculated using equations (3) and (4). In order to do so, however, the concentration of A in stream b first must be computed using equation (28). The rate coefficients in equations (3) and (4) will be evaluated at T_b.

$$C_{A,b} = \frac{\dot{n}_{A,b}}{\dot{V}_b} = \frac{\dot{n}_{A,b}}{\dot{V}_a} \tag{28}$$

With that information, code can be written to evaluate the functions f_1 through f_4, given values for $\dot{n}_{A,b}$, $\dot{n}_{D,b}$, $\dot{n}_{U,b}$ and T_b, and the design equations (8) through (10) can be solved numerically. Upon doing so, one finds $\dot{n}_{A,b}$ = 60.7 mol min^{-1}, $\dot{n}_{D,b}$ = 113.5 mol min^{-1}, $\dot{n}_{U,b}$ = 75.7 mol min^{-1} and T_b = 354 K.

The PFR design equations, (23) through (26), are initial value ODEs because the value of each of the 4 dependent variables is known at the same value ($V = 0$) of the independent variable. They can be solved numerically to obtain the values of all of the variables at some other point of interest, here the outlet. To do so you will need to provide (a) the initial values of the independent and dependent variables, (b) the final value of either the independent variable or one of the dependent variables at the outlet and (c) code that, given values for the independent and dependent variables, evaluates the functions f_5 through f_8 in equations (23) through (26), where the ODEs have been written in the standard form with one derivative on the left of each equals sign and a function of the independent and dependent variables on the right.

Looking at the schematic diagram, the initial values of the molar flow rates and the temperature (at $V = 0$) are the values for stream b, which were just calculated. The final value of the independent variable (V = 200 L) is known. In order to evaluate the functions, f_5 through f_8, we'll need to calculate every quantity that appears in them except for the independent and dependent variables (V, $\dot{n}_A$, $\dot{n}_D$, $\dot{n}_U$ and T), which the code will be given. Looking at the functions, we see that they contain the same quantities that the CSTR equations contained, and all of those except the rates are known constants. The rates are again calculated using equations (3) and (4). To do so, it is necessary to first calculate the concentration of A. In this case, that concentration will be calculated using the given values for the independent and dependent variables as in equation (29). The rate coefficients in equations (3) and (4) will be evaluated at the given value of T.

$$C_A = \frac{\dot{n}_A}{\dot{V}_b} = \frac{\dot{n}_A}{\dot{V}_a} \tag{29}$$

That is all the information needed to numerically solve the PFR design equations to determine the molar flow rates and the temperature at the outlet of the PFR. Doing so, one finds that $\dot{n}_{A,c}$ = 2.2 mol min^{-1}, $\dot{n}_{D,c}$ = 154.5 mol min^{-1}, $\dot{n}_{U,c}$ = 93.3 mol min^{-1} and T_c = 367 K. Knowing those quantities, the overall conversion and the selectivity can be calculated using equations (30) and (31) to find the conversion is 99.1% and the selectivity is 1.66 mol D per mol U.

$$f_A = \frac{\dot{n}_{A,a} - \dot{n}_{A,c}}{\dot{n}_{A,a}} \tag{30}$$

$$S_{D/U} = \frac{\dot{n}_{D,c}}{\dot{n}_{U,c}} \tag{31}$$

Case 2, PFR followed by CSTR. Figure 2 presents a schematic representation of this configuration. The analysis of this configuration is nearly the same as the first configuration, so a detailed description will not be presented. The differences are that first, the PFR equations are written using stream a as the initial values, and they are solved first to find the molar flow rates and temperature for stream b: $\dot{n}_{A,b}$ = 96.0 mol min^{-1}, $\dot{n}_{D,b}$, = 110.5 mol min^{-1}, $\dot{n}_{U,b}$ = 43.5 mol min^{-1}, and T_b = 345 K. The CSTR design equations can then be solved using the molar flows and temperature for stream b as the “in” values and the molar flows and temperature for stream c as the “out” values. Since the values for stream b are known upon solving the PFR design equations, the CSTR design equations can be solved to find the unknown molar flows and temperature for stream c: $\dot{n}_{A,c}$ = 21.7 mol min^{-1}, $\dot{n}_{D,c}$, = 164.6 mol min^{-1}, $\dot{n}_{U,c}$ = 63.7 mol min^{-1}, and T_c = 362 K. These results correspond to a conversion of 91.3% and a selectivity of 2.59 mol D per mol U.

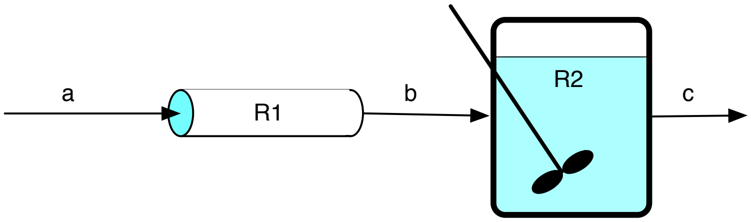

Figure 2. Schematic representation of the case with the PFR followed by the CSTR.

As noted in the problem analysis, this particular reaction/reactor combination involved several trade-offs that made it impossible to qualitatively predict the better configuration. The results show that while placing the CSTR second did adversely affect the conversion (91.3% versus 99.1%), it had a much larger impact upon the selectivity (2.59 vs. 1.66 mol D per mol U).

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 CSTR design equations and a second MATLAB function for solving the PFR design equations. In each of these functions, the inlet stream properties were passed into the function as arguments, and after numerically solving the design equations, the outlet stream properties were returned. Once these functions were written, a third MATLAB function was written to solve each of the two cases solving the reactor design equations with the appropriate inlet stream properties for each case. This approach reduces the amount of code duplication, but providing guesses for the solution of the CSTR equations can become challenging since the CSTRs operate at very different feed conditions in the two configurations. It wasn’t necessary here, but one solution if this becomes a problem, is to pass both the inlet stream properties and a guess for the outlet stream properties as arguments to the function that solves the CSTR design equations. In that way, different guesses can be used depending upon the feed to the reactor.

MATLAB function for modeling the CSTR. The CSTR design equations, (8) through (11), are non-linear, non-differential equations. Supplemental Unit S2 describes how to solve sets of non-differential equations numerically using MATLAB, and it provides a template file named SolvNonDif.m for doing so. Before it can be used to solve a problem, that template file must be modified in four places, each indicated by a comment that begins “% EDIT HERE”. In addition to those required modifications I made a few additional modifications that will be described here along with the required modifications.

I recommend that you work with a copy of the file that has been given a more meaningful name; I made a copy of SolvNonDif.m and saved it as Example_29_2_cstr.m. Since the function name must match the filename, I changed the name accordingly. At the same time, I changed the function declaration to indicate that the inlet molar flow rates of A, D and U and the inlet temperature would be passed into the function as arguments. I also changed the declaration to indicate that the outlet molar flow rates of A, D and U and the outlet temperature would be returned. (These items could have been passed as a single inlet vector and an outlet vector, but for clarity I used individual variables.) 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 at the point indicated. At the same time these are entered, they should be converted to a consistent set of units. I also entered the ideal gas constant. Listing 1 shows the relevant part of Example_29_2_cstr.m.

```
% Modified version of the AFCoKaRE MATLAB Template file SolvNonDif.m used
% in the solution of Example29.2 of "A First Course on Kinetics and
% Reaction Engineering."
%
function [nAout,nDout,nUout,Tout]=Example_29_2_cstr(nAin, nDin, nUin, Tin)
    % Known quantities and constants
    V = 200.0; % L
    VFR = 100.0; % L/min
    dH1 = -21500.0; % cal/mol
    dH2 = -24000.0; % cal/mol
    cP = 1.0*1000; % cal/L/K
    k01 = 3.4E5; % /min
    k02 = 1.67E14; % /min
    E1 = 9000.0; % cal/mol
    E2 = 23000.0; % cal/mol
    % Gas constant
    R = 1.987; % cal/mol/K
```

Listing 1. Changes to the first part of SolvNonDif.m for use in modeling the CSTR.

The second *required* modification involves entering the code to evaluate the right hand sides of equations (7) through (10) which have been written in the form f(unknowns) = 0. This occurs within an internal function named `evalEqns` where the unknowns and the equations are entered as vector quantities named `z` and `f`, respectively. Thus, it is necessary to map the unknowns in this problem solution ($\dot{n}_{A,out}$, $\dot{n}_{D,out}$, $\dot{n}_{U,out}$ and T_{out}) to a vector `z`, and to return the values of the functions in the vector `f`. I find it useful at the start of the internal function that will evaluate the functions, 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 solution variables to the vector `z`.

Recall from the solution that in order to evaluate the functions it is first necessary to calculate the rates using equations (3) and (4). Before than can be done, it is first necessary to calculate the concentration of A and then evaluate the rate, all at the outlet conditions. Once the rate has been calculated the functions can next be evaluated. The code containing all these modifications is shown in Listing 2.

```
% Function that evaluates the equations
function f = evalEqns(z)
    nA = z(1);
    nD = z(2);
    nU = z(3);
    T = z(4);
    CA = nA/VFR;
    r1 = k01*exp(-E1/R/T)*CA;
    r2 = k02*exp(-E2/R/T)*CA^2;
    f = [
        nAin - nA - V*(r1 + r2)
        nDin - nD + V*r1
        nUin - nU + V*r2
        VFR*cP*(T - Tin) + V*(r1*dH1 + r2*dH2)
    ];
end % of internal function evalEqns
```

Listing 2. Modified version of the internal function evalEqns in the function for modeling the CSTR.

The third *required* modification is where guesses for the unknowns are provided. The guesses are entered in the array named z_guess. They must be entered using the same mapping of the unknowns to the vector z as was used in the internal function `evalEqns`. The previous modification, where variables with more meaningful names were defined, serves as a key to remind you which variable is `z(1)`, which is `z(2)`, and so on. Listing 3 shows the guesses I used; there is nothing special about them, they are just guesses.

```
% guesses for the solution
z_guess = [
    0.5*nAin
    nDin + 0.5*nAin
    nUin + 0.01*nAin
    Tin + 10
];
```

Listing 3. Guesses for the values of the unknowns.

The 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. If you look carefully at Listing 4, you'll see that I also increased the maximum number of function evaluations used by the equation solver, `fsolve`. If I hadn't done this, the routine would not have converged to a solution in both reactor network configurations; different guesses would have been needed. In this case, allowing for significantly more function evaluations allowed the method to find solutions for both configurations using the same guess. A change like this won't always work, but in this case it did.

```
    % Solve the set of algebraic equations
    options = optimoptions(@fsolve,'MaxFunEvals',2000);
    z = fsolve(@evalEqns, z_guess, options);
    display('The values of the functions being solved are as follows:');
    f = evalEqns(z)
    nAout = z(1);
    nDout = z(2);
    nUout = z(3);
    Tout = z(4);
```

Listing 4. Final modification to the function for modeling the CSTR.

MATLAB function for modeling the PFR. The PFR design equations, (15) through (18), 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_2_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 inlet molar flow rates of A, D and U and the inlet temperature would be passed into the function as arguments and the outlet molar flow rates of A, D and U and the outlet temperature would be returned. I replaced the long introductory comment with a brief comment stating the purpose of the modified version. None of these modifications were required. The first *required* modification involves entering all the known quantities from the problem statement along with universal constants that will be needed (from handbooks or other reference sources). As these were entered, they were converted to a consistent set of units. The result of making all these modifications is shown in Listing 5.

```
% Modified version of the AFCoKaRE MATLAB Template file SolvIVDifI.m used
% in the solution of Example 29.2 of "A First Course on Kinetics and
% Reaction Engineering."
%
function [nAout,nDout,nUout,Tout] = Example_29_2_pfr(nAin,nDin,nUin,Tin)
    % Known quantities and constants
    V = 200.0; % L
    VFR = 100.0; % L/min
    dH1 = -21500.0; % cal/mol
    dH2 = -24000.0; % cal/mol
    cP = 1.0*1000; % cal/L/K
    k01 = 3.4E5; % /min
    k02 = 1.67E14; % /min
    E1 = 9000.0; % cal/mol
    E2 = 23000.0; % cal/mol
    % Gas constant
    R = 1.987; % cal/mol/K
```

Listing 5. 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 (15) through (18). Notice that the equations are provided as a vector quantity. Thus, it is necessary to map the dependent variables used in the problem statement ($\dot{n}_A$, $\dot{n}_D$, $\dot{n}_U$, 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 reactant concentration according to equations (29). Following that, the rates are calculated (at the outlet temperature) using equations (3) and (4). Finally, equations (15) through (18) 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 6.

```
% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    nA = z(1);
    nD = z(2);
    nU = z(3);
    T = z(4);
    CA = nA/VFR;
    r1 = k01*exp(-E1/R/T)*CA;
    r2 = k02*exp(-E2/R/T)*CA^2;
    dzdt = [
        -r1 - r2
        r1
        r2
        -(r1*dH1 + r2*dH2)/VFR/cP
    ];
end % of internal function odeqns
```

Listing 6. Results of the second required modification to the code for modeling the PFR.

The third *required* modification involves providing the initial values of the independent (V = 0 L) and dependent variables (passed in as arguments) and the final value of the independent variable (V = 200 L). The results of performing this modification are shown in Listing 7.

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

Listing 7. Results of the third required modification.

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 8.

```
    % 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);
    nDout = z(2);
    nUout = z(3);
    Tout = z(4);
```

Listing 8. Results of the fourth required modification.

MATLAB function for analyzing the reactor networks. Once the functions for modeling the CSTR and the PFR were available, a third MATLAB function, named Example_29_2, was created to analyze the two reactor networks. That function is shown in Listing 9. The function begins by entering or calculating the molar flow rates and the temperature of the feed stream. For the first reactor network, those quantities are passed to the function that models the CSTR, while for the second network they are passed to the function that models the PFR. The molar flow rates and temperature for the outlet stream from the first reactor are returned. Those returned values are then simply passed to the function that evaluates the second reactor as the inlet values, and the outlet values are returned. Once the outlet from the second reactor has been obtained, the conversion and selectivity are calculated according to equations (30) and (31). Listing 9 shows the code. That code can then be executed by simply typing Example_29_2 at the MATLAB command prompt. Doing so produces the output shown in Listings 10 and 11.

```matlab
% MATLAB file used in the solution of Example_29_2 of "A First Course on
% Kinetics and Reaction Engineering."
%
function Example_29_2
    % Known Constants
    CAa = 2.5; % mol/L
    VFR = 100.0; % L/min
    Ta = 38.0 + 273.15; % K
    nDa = 0.0; % mol/min
    nUa = 0.0; % mol/min
    % Calculated constants
    nAa = CAa*VFR;

    % CSTR first
    display('Case with CSTR first');
    % calculate the CSTR outlet values
    [nAb, nDb, nUb, Tb] = Example_29_2_cstr(nAa,nDa,nUa,Ta);
    % calculate the PFR outlet values
    [nAc, nDc, nUc, Tc] = Example_29_2_pfr(nAb, nDb, nUb, Tb);
    % calculate and report overall conversion and yield
    percent_conversion = 100*(nAa - nAc)/nAa
    sel_D_per_U = nDc/nUc

    % PFR first
    display('Case with PFR first');
    % calculate the PFR outlet values
    [nAb, nDb, nUb, Tb] = Example_29_2_pfr(nAa,nDa,nUa,Ta);
    % calculate the CSTR outlet values
    [nAc, nDc, nUc, Tc] = Example_29_2_cstr(nAb, nDb, nUb, Tb);
    % calculate overall conversion and yield
    percent_conversion = 100*(nAa - nAc)/nAa
    sel_D_per_U = nDc/nUc

end % of Example_29_2.m
```

Listing 9. MATLAB function for evaluating the two reactor networks.

```
>> Example_29_2
Case with CSTR first

Equation solved, fsolve stalled.

fsolve stopped because the relative size of the current step is less than the
default value of the step size tolerance squared and the vector of function
values
is near zero as measured by the default value of the function tolerance.

<stopping criteria details>

The values of the functions being solved are as follows:

f =
   1.0e-08 *

   -0.0000
    0.0000
   -0.0000
   -0.2794

percent_conversion =
   99.1236

sel_D_per_U =
    1.6571
```

Listing 10. First half of the output from the execution of Example_29_2.

```
Case with PFR first

Equation solved, fsolve stalled.

fsolve stopped because the relative size of the current step is less than the
default value of the step size tolerance squared and the vector of function
values
is near zero as measured by the default value of the function tolerance.

<stopping criteria details>

The values of the functions being solved are as follows:

f =
   1.0e-09 *

   -0.0001
    0.0000
    0.0000
   -0.2328


percent_conversion =
   91.3044


sel_D_per_U =
    2.5855
```

Listing 10. Second half of the output from the execution of Example_29_2.