

A First Course on Kinetics and Reaction Engineering

Example 32.2

Problem Purpose

This problem will help you determine whether you have mastered the learning objectives for this unit. It illustrates the analysis of a semi-batch reactor where a product boils off during processing. It also illustrates a situation where the semi-batch reactor design equations must be formulated and solved separately for two different stages of processing.

Problem Statement

When a solution containing A and B is allowed to react, it produces Y and Z according to reaction (1). The product Z is only slightly soluble in the solution, and consequently it tends to volatilize into the vapor phase. The other reagents, A, B and Y have negligible vapor pressures at process conditions (which are essentially isothermal due to a large excess of solvent and a low heat of reaction). At the process temperature, the concentration equilibrium constant, K , is equal to 8×10^{-5} , and the Henry's law constant for Z is equal to $8 \times 10^{-4} \text{ mol L}^{-1} \text{ atm}^{-1}$. The rate expression is given in equation (2). The initial charge to the reactor consists of 100 gal of a liquid solution containing A and B at equal concentrations of 4 mol L^{-1} at 80°C . The reactor holds 100 gal of liquid with a headspace of 8 gal. At the start of processing, the headspace is evacuated. Calculate the equilibrium conversion of A for a closed reactor and then plot the conversion of A as a function of time for a process wherein a pressure relief valve is used to maintain a maximum headspace pressure of 1.5 atm. Z is a small diatomic molecule, and its vaporization from the liquid phase does not cause an appreciable change in the liquid volume.



$$r_1 = \left(5 \times 10^{-4} \text{ L mol}^{-1} \text{ min}^{-1}\right) C_A C_B \left\{ 1 - \frac{C_Y C_Z}{K C_A C_B} \right\} \quad (2)$$

Problem Analysis

This is not a kinetics problem: we are given the rate expression and asked to analyze the performance of reactors. First, we are asked for the equilibrium conversion for a closed reactor. There are two ways to approach this problem: write the appropriate reactor design equations and solve them for the conversion as the time approaches infinity or write and solve the equilibrium equations (since there are two phases in the reactor, there will be two equilibrium expressions that must be solved simultaneously). We will use the latter approach here. The second question is a reaction engineering problem and will be solved by formulating and solving the reactor design equations. The problem specifies that the process is isothermal, so the mole balance design equations can be solved independently of the energy balance design equations. This particular problem does not ask any questions related to heat transfer or temperature, so we will not need to formulate an energy balance. In formulating the mole balance design

equations, we will assume that the vapor and liquid phases are always equilibrated. Initially the pressure relief valve will be closed, but once the pressure builds to 1.5 atm, it will open. As a consequence, the semi-batch operation will need to be evaluated in two stages, one stage corresponding to the time when the pressure valve is closed and the second stage corresponding to the time during which it is open.

Problem Solution

First consider the equilibration of a closed reactor. As indicated in the schematic diagram of Figure 1, there are a liquid phase and a vapor phase (head space) present in the reactor. At thermodynamic equilibrium, reaction (2) must be at equilibrium in each of the two phases and the two phases must be in equilibrium with each other. In this case, however, the problem statement indicates that the only species present in the vapor phase is Z. In that situation, there can be no reaction in the vapor phase, so the only requirements are that the reaction is at equilibrium in the liquid phase and the liquid and vapor phases are equilibrated.

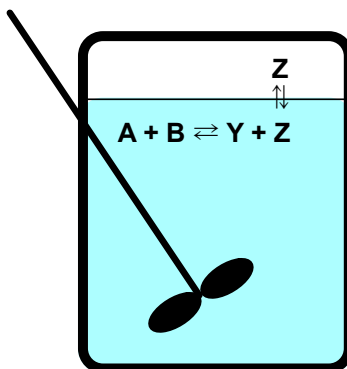


Figure 1. Schematic representation of the equilibrated closed reactor.

After making a schematic, it is a good idea to read through the problem statement and assign every quantity that is specified to an appropriate variable symbol. In this problem, the following quantities are specified: $P_A = P_B = P_Y = 0$ atm, $K = 8 \times 10^{-5}$, $h_Z = 8 \times 10^{-4}$ mol L⁻¹ atm⁻¹, $k = 5.4 \times 10^{-4}$ L mol⁻¹ min⁻¹, $V_{liq,0} = V_{liq}(t) = 100$ gal, $C_{A,0} = C_{B,0} = 4$ mol L⁻¹, $C_{Y,0} = C_{Z,0} = 0$ mol L⁻¹, $T_0 = 80$ °C, $V_{vap} = 8$ gal, $P_0 = 0$ atm and $P_{max} = 1.5$ atm. The maximum pressure only applies in the case where the reactor is fitted with a pressure relief valve. Knowing the initial liquid volume and the initial liquid concentrations, the initial moles of each species can be calculated according to equation (3) where i is equal to A, B, Y and Z.

$$n_{i,0} = V_{liq,0} C_{i,0} \quad (3)$$

To answer the first question, we need a reaction equilibrium expression, equation (4) and a phase equilibrium expression, equation (5). We also could have obtained equation (4) by noting that the reaction rate will equal zero at equilibrium. These are the two equations that must be solved in order to calculate the equilibrium composition of the closed system.

$$K = \frac{C_{Y,eq} C_{Z,eq}}{C_{A,eq} C_{B,eq}} \quad (4)$$

$$P_{Z,eq} h_Z = C_{Z,eq} \quad (5)$$

We have two equations, so they can be solved to find values for two unknowns. Here, I will choose $C_{A,eq}$ and $C_{Z,eq}$ as the two unknowns to be found by solving equations (4) and (5). Having identified the unknowns to be solved for, the equations can be written in the standard form with a zeroes on one side of the equals sign and functions of the unknowns on the other side, as shown in equations (6) and (7). At the same time, both sides of equation (4) were multiplied by $C_{A,eq}C_{B,eq}$ to eliminate the denominator since this often improves convergence during numerical solution.

$$0 = f_1(C_{A,eq}, C_{Z,eq}) = KC_{A,eq} C_{B,eq} - C_{Y,eq} C_{Z,eq} \quad (6)$$

$$0 = f_2(C_{A,eq}, C_{Z,eq}) = P_{Z,eq} h_Z - C_{Z,eq} \quad (7)$$

In order to solve these equations numerically, no matter what software is used, it will be necessary to provide a guess for each of the two unknowns and computer code. That code will receive values for the two unknowns, $C_{A,eq}$ and $C_{Z,eq}$, and it must evaluate and return the corresponding values of the two functions, f_1 and f_2 , in equations (6) and (7). In order to evaluate those functions, the code must first compute values for every quantity appearing in the functions. Looking at the two functions, the values of the Henry's law constant, h_Z , and the equilibrium constant, K , have already been specified. Thus, in order to evaluate functions f_1 and f_2 , the supplied computer code will first need to calculate values for $C_{B,eq}$, $C_{Y,eq}$, and $P_{Z,eq}$. It should be noted that since equation (7) is one of the equations being solved, it cannot be used to compute the value of $P_{Z,eq}$.

The code will be given values for $C_{A,eq}$ and $C_{Z,eq}$. Having the value of $C_{A,eq}$, the moles of A, $n_{A,eq}$, can be calculated using equation (8). Then, knowing the initial moles of A, the equilibrium extent of reaction, ξ_{eq} , can be calculated, equation (9). That, in turn, permits calculation of the moles of B and Y, equations (10) and (11), and then their concentrations, equations (12) and (13). The total moles of Z can also be calculated from the extent of reaction, equation (14), but those moles of Z will be split between the liquid and vapor phases. Since the code will be given the liquid phase concentration of Z, $C_{Z,eq}$, the liquid phase moles of Z can be calculated using equation (15). Subtracting the moles of Z in the liquid phase from the total moles of Z gives the moles of Z in the vapor phase, equation (16). Finally, assuming ideal gas behavior, the pressure of Z in the vapor phase can be calculated, equation (17).

$$n_{A,eq} = V_{liq} C_{A,eq} \quad (8)$$

$$\xi_{eq} = n_{A,0} - n_{A,eq} \quad (9)$$

$$n_{B,eq} = n_{B,0} - \xi_{eq} \quad (10)$$

$$n_{Y,eq} = n_{Y,0} + \xi_{eq} \quad (11)$$

$$C_{B,eq} = \frac{n_{B,eq}}{V_{liq}} \quad (12)$$

$$C_{Y,eq} = \frac{n_{Y,eq}}{V_{liq}} \quad (13)$$

$$n_{Z,tot,eq} = n_{Z,0} + \xi_{eq} \quad (14)$$

$$n_{Z,liq,eq} = V_{liq} C_{Z,eq} \quad (15)$$

$$n_{Z,vap,eq} = n_{Z,tot,eq} - n_{Z,liq,eq} \quad (16)$$

$$P_{Z,eq} = \frac{n_{Z,vap,eq} RT}{V_{vap}} \quad (17)$$

At this point, all the quantities needed to evaluate the functions, f_1 and f_2 in equations (6) and (7) can be found, and therefore the code necessary to solve equations (6) and (7) numerically can be written and the equations solved. Upon doing so, one finds $C_{A,eq} = 3.92 \text{ mol L}^{-1}$ and $C_{Z,eq} = 0.017 \text{ mol L}^{-1}$. The moles of A can again be calculated using equation (8), and then the fractional conversion of A can be calculated using equation (18). Performing the calculations, one finds $f_{A,eq} = 1.85\%$. Additional calculations show that the final pressure is 20.8 atm at equilibrium.

$$f_{A,eq} = \frac{n_{A,0} - n_{A,eq}}{n_{A,0}} \quad (18)$$

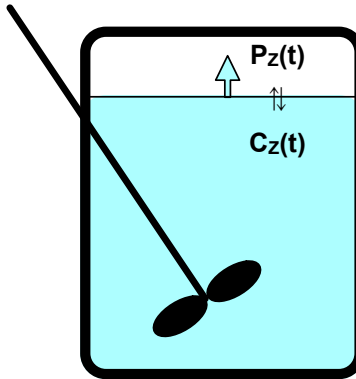


Figure 2. Schematic representation of the first stage of semi-batch operation.

Now we can consider the second question from the problem statement where the reactor is fitted with the pressure relief valve. During the first stage, the headspace is not vented because the pressure is not sufficiently large to open the relief valve. Thus, as indicated in the schematic diagram, Figure 2, the pressure in the headspace will increase over time, due to the continual generation of Z in the liquid phase by reaction and the evaporation of Z into the headspace. We will assume that phase equilibrium exists at all times. While nothing is leaving the reactor during this stage of processing, it is still a semi-batch operation. This is so because the reactor design equations are balances on the reaction volume. In this case, the reaction only takes place in the liquid phase, and the balances are written for that phase only. From the perspective of the liquid phase, Z is leaving the reaction volume when it evaporates, making this a semi-batch process.

The quantities provided in the problem statement have already been assigned to appropriate variable symbols, so that need not be repeated here. As noted in the problem analysis, the reactor is isothermal, so the mole balance design equations can be solved independently from the energy balance. Indeed, in this problem, an energy balance is not needed in order to answer the questions posed. The general form of the mole balance for a semi-batch reactor is given in equation (19). There is only one reaction taking place in this system, so the summation in equation (19) will expand to a single term. Using equation (19), mole balances can be written for each of the four reagents, leading to equations (20) through (23). Notice that since only Z evaporates from the reacting liquid, \dot{n}_A , \dot{n}_B and \dot{n}_Y are all equal to zero. It should also be noted that since \dot{n}_Z represents the net molar flow of Z into the reactor, it will have a negative value.

$$\frac{dn_i}{dt} = \dot{n}_i + V \sum_{\substack{j=\text{all} \\ \text{reactions}}} \nu_{i,j} r_j \quad (19)$$

$$\frac{dn_A}{dt} = f_3(t, n_A, n_B, n_Y, n_Z) = -V_{liq} r_1 \quad (20)$$

$$\frac{dn_B}{dt} = f_4(t, n_A, n_B, n_Y, n_Z) = -V_{liq} r_1 \quad (21)$$

$$\frac{dn_Y}{dt} = f_5(t, n_A, n_B, n_Y, n_Z) = V_{liq} r_1 \quad (22)$$

$$\frac{dn_Z}{dt} = f_6(t, n_A, n_B, n_Y, n_Z) = \dot{n}_Z + V_{liq} r_1 \quad (23)$$

The design equations, (20) through (23), are a set of initial value ordinary differential equations where the independent variable is time and the dependent variables are the moles of A, B, Y and Z in the reactor. To solve the semi-batch design equations numerically, no matter what software is used, it will be necessary to provide initial values (i. e. values of n_A , n_B , n_Y and n_Z at $t = 0$), a final value of either t , n_A , n_B , n_Y or n_Z , and computer code that is given values for t , n_A , n_B , n_Y and n_Z and uses them to evaluate and

return the values of functions f_3 through f_6 in equations (20) through (23). For this stage of the process, the initial values are known as seen in equations (24) through (27).

$$n_A(t=0) = C_{A,0} V_{liq} \quad (24)$$

$$n_B(t=0) = C_{B,0} V_{liq} \quad (25)$$

$$n_Y(t=0) = C_{Y,0} V_{liq} = 0 \quad (26)$$

$$n_Z(t=0) = C_{Z,0} V_{liq} = 0 \quad (27)$$

The first stage ends when the pressure in the headspace reaches 1.5 atm. The time when this occurs can be denoted as $t_{f,1}$ (final time for stage 1). Since Z will be the only species present in the vapor phase, that means the partial pressure of Z will equal 1.5 atm. Substituting that value into equation (4) along with the known Henry's law constant allows calculation of the final liquid phase concentration of Z. Multiplying that concentration by the liquid volume gives the moles of Z at the end of the first stage, equation (28). This is the final condition needed to solve the design equations for the first stage.

$$n_Z(t = t_{f,1}) = V_{liq} C_Z = V_{liq} P_Z h_Z = (1.5 \text{ atm}) h_Z V_{liq} \quad (28)$$

The code that must be provided to numerically solve equations (20) through (23) will be given values for t , n_A , n_B , n_Y and n_Z , and it must use them to evaluate the functions, f_3 through f_6 , in those equations. Looking at functions f_3 through f_6 it can be seen that they contain quantities whose values are not known, specifically r and \dot{n}_Z . These must be calculated before the functions can be evaluated.

Equation (2) can be used to compute the rate; the equilibrium constant is known but the reagent concentrations are not. The concentrations of A, B, Y and Z can be calculated by definition (i. e. using equations (12) and (13) and equivalent equations for Y and Z), after which the rate can be computed.

The molar flow rate of Z into the reactor, \dot{n}_Z , is the only other quantity needed in order to evaluate the functions f_3 through f_6 . To find \dot{n}_Z , a mole balance is written on the headspace. The molar flow of Z into the headspace must equal the molar flow of Z from the liquid (which is the negative of \dot{n}_Z , the net molar flow of Z into the liquid). Since nothing leaves the headspace and no reaction takes place in the headspace, a mole balance on Z in the headspace requires the molar flow of Z in to equal the rate of accumulation of Z, equation (29). Again, \dot{n}_Z will be negative, so the flow into the headspace in equation (29) will be positive, as expected. Re-writing the moles of z in the headspace in terms of the ideal gas law, substituting Henry's law, equation (5), and then re-writing the liquid phase concentration of Z in terms of the moles of Z and the liquid volume gives equation (30). Finally, substituting equation (23) into equation (30) and rearranging gives equation (31) for the molar flow rate of Z leaving the liquid. All of the quantities appearing in equation (31) are known, and so a value for \dot{n}_Z can be calculated. At this point, all quantities needed in order to evaluate the functions, f_3 through f_6 , in equations (20) through (23) are known.

$$-\dot{n}_Z = \frac{d}{dt} n_{Z,vap} \quad (29)$$

$$-\dot{n}_Z = \frac{d}{dt} n_{Z,vap} = \frac{d}{dt} \left(\frac{P_Z V_{head}}{RT} \right) = \frac{d}{dt} \left(\frac{C_Z V_{head}}{h_Z RT} \right) = \frac{d}{dt} \left(\frac{n_Z V_{head}}{V_{liq} h_Z RT} \right) = \frac{V_{head}}{V_{liq} h_Z RT} \frac{dn_Z}{dt} \quad (30)$$

$$-\dot{n}_Z = \frac{V_{head}}{V_{liq} h_Z RT} \frac{dn_Z}{dt} = \frac{V_{head}}{V_{liq} h_Z RT} (\dot{n}_Z + V_{liq} r_1) \Rightarrow \dot{n}_Z = \frac{-V_{liq} r_1}{1 + \frac{V_{head}}{V_{liq} h_Z RT}} \quad (31)$$

Upon writing the code to evaluate the functions, f_3 through f_6 , and solving the design equations numerically, one finds that the pressure in the headspace reaches 1.5 atm after 0.67 min. Solving the design equations also gives the values of n_A , n_B , n_Y and n_Z at that time. These are used as the initial conditions for the second stage, shown schematically in Figure 3. Knowing the first stage lasts 0.67 min, we can go back and solve the design equations for a range of times between 0 and 0.67 min, and each time we solve them, we can calculate the conversion. This will generate the data needed for the requested plot for times up to 0.67 min.

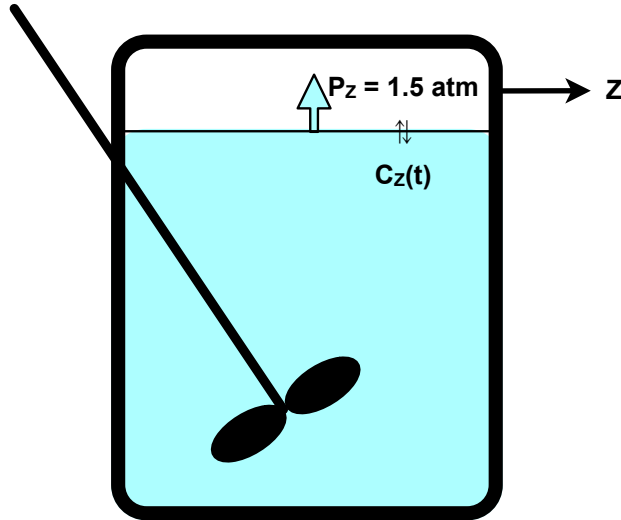


Figure 3. Schematic representation of the second stage of semi-batch processing.

During the second stage, the concentration of Z in the liquid phase remains constant because any additional Z that is formed evaporates into the vapor phase and is immediately vented so that the pressure in the headspace remains constant. That is, the liquid phase concentration of Z during the second stage is constant, being in equilibrium with the Z in the headspace at a pressure of 1.5 atm, as expressed in equation (32).

$$C_Z = (1.5 \text{ atm})h_Z \quad (32)$$

Since the liquid phase concentration of Z is not changing, we do not need the mole balance design equation for Z, equation (23). The other three equations remain unchanged, and as already noted, the initial conditions are simply the ending conditions from stage 1. Thus, the three remaining design equations can be solved over a range of final times, and at each time the corresponding conversion of A can be calculated via equation (33). Since we want the overall conversion, not that just for stage 2, we use the original moles of A in equation (33).

$$f_A = \frac{n_{A,0} - n_A}{n_{A,0}} \quad (28)$$

The resulting plot is shown in Figure 4. From it we can see that by venting the headspace the final conversion is approximately 18% as compared to the non-vented equilibrium conversion of 1.85%. This result illustrates a situation where a semi-batch reactor offers a performance advantage. In the closed system the reaction reached equilibrium at a low conversion, but by allowing one component to boil off and be vented, it was possible to reach a much higher conversion using essentially the same equipment.

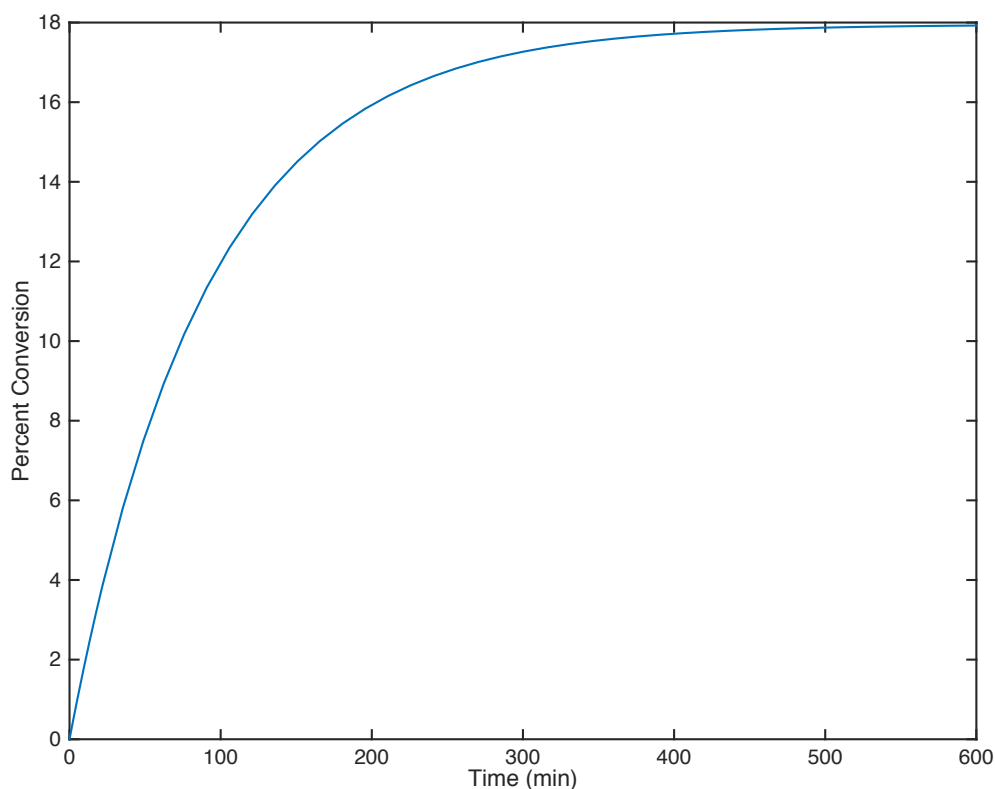


Figure 4. Conversion versus time during semi-batch processing.

Calculation Details Using MATLAB

The equilibrium equations used to answer the first question in this problem are (non-differential, non-integral) non-linear equations. The numerical solution of equations of this type is discussed in Supplemental Unit S2. That supplemental unit provides a template file, SolvNonDif.m, that served as the basis of the code presented here. The semi-batch reactor design equations used to answer the second question in this problem are initial value ordinary differential equations (ODEs). The numerical solution of sets of initial value ODEs is discussed in Supplemental Unit S5, which also provides two template files: SolvIVDifl.m and SolvIVDifD.m. The former template file is used when the final value of the independent variable is specified, and it served as the basis of the code used to solve the design equations for the second stage of semi-batch operation. The other template file is used when the final value of one dependent variable is specified; SolvIVDifD served as the basis of the code used to solve the design equations for the first stage of semi-batch operation. A fourth MATLAB file was created so that all parts of the problem could be solved by execution of a single MATLAB function. That file calls the others as needed to perform all of the calculations.

Solution of the equilibrium equations for the closed reactor. The template file SolvNonDif.m was copied and saved as Example_32_2_equil.m; a copy accompanies this solution. The first modification, shown in Listing 1, entailed changing the introductory comment, renaming the function so that it returned the equilibrium conversion and the final pressure, and entering all the known constant quantities specified in the problem statement. Additional constant quantities that could be computed at this point were also calculated.

```
% Modified version of the AFCoKaRE MATLAB template file SolvNonDif.m used
% in the solution of Example 32.2 of "A First Course on Kinetics and
% Reaction Engineering. More specifically, this file performs the
% equilibrium calculations for the semi-batch reactor without a relief
% valve.
%
function [fA_equil,Pfinal] = Example_32_2_equil
    % Known quantities and constants (in min, mol, L, atm and K)
    K = 8e-5;
    hZ = 8e-4; % mol/L/atm
    k = 5e-4; % L/mol/min
    V = 100*3.785; % L
    CA0 = 4.; % mol/L
    CB0 = 4.; % mol/L
    T = 80 + 273.15; % K
    Vhead = 8*3.785; % L
    Pmax = 1.5; % atm
    % Constants found in handbooks and other reference sources
    R = 0.08206; % L*atm/mol/K
    % Calculated constants
    nA0 = CA0*V;
    nB0 = CB0*V;
    nY0 = 0.0;
    nZ0 = 0.0;
```

Listing 1. Initial modifications to SolvNonDif.m.

The second modifications, shown in Listing 2, occurred within the internal function `evalEqns`. This is the code to evaluate the functions, f_1 and f_2 in equations (6) and (7). Values for the unknowns are passed into `evalEqns` as a vector named `z`. The unknowns, $C_{A,eq}$ and $C_{Z,eq}$, are first extracted from that vector, after which $C_{B,eq}$, $C_{Y,eq}$, and $P_{Z,eq}$ are computed according to equations (8) through (17). The functions are then evaluated and stored in the vector `f`, which is returned by `evalEqns`.

```
% Function that evaluates the equations
function f = evalEqns(z)
    CA = z(1);
    CZ = z(2);
    nA = CA*V;
    extent = nA0 - nA;
    nB = nB0 - extent;
    nY = nY0 + extent;
    CB = nB/V;
    CY = nY/V;
    nZtot = nZ0 + extent;
    nZliq = CZ*V;
    nZhead = nZtot - nZliq;
    PZ = nZhead*R*T/Vhead;
    f = [
        K*CA*CB - CY*CZ
        PZ*hZ - CZ
    ];
end % of internal function evalEqns
```

Listing 2. Code that evaluates the equilibrium expressions.

Guesses for the unknowns, $C_{A,eq}$ and $C_{Z,eq}$, were next entered, after which the equilibrium expressions were solved numerically, Listing 3.

```
% guesses for the solution
z_guess = [
    0.9*CA0
    0.1*CA0
];

% Solve the set of algebraic equations
z = fsolve(@evalEqns, z_guess);
display('The solver found the following values for the unknowns:');
z
display('The corresponding values of the functions being solved are as follows:');
f = evalEqns(z)
```

Listing 3. Code that provides guesses and then solves the equilibrium expressions.

The final modification occurs after the equations have been solved, Listing 4. It entails calculating the equilibrium conversion and the final pressure so they will be properly returned.

```

% calculate the conversion and set the return variable values
CA = z(1);
CZ = z(2);
nA = CA*V;
extent = nA0 - nA;
nZtot = nZ0 + extent;
nZliq = CZ*V;
nZhead = nZtot - nZliq;
fA_equil = (nA0 - nA)/nA0;
Pfinal = nZhead*R*T/Vhead;

end % of Example_32_2_equil.m

```

Listing 4. Calculation of the equilibrium conversion and the final pressure.

First stage of semi-batch operation. The template file SolvIVDifD.m was copied and saved as Example_32_2_semi_a.m; a copy accompanies this solution. The first modification, shown in Listing 5, entailed changing the introductory comment, renaming the function so that it returned the duration of the first stage and the final moles of each of the reagents, and entering all the known constant quantities specified in the problem statement. Additional constant quantities that could be computed at this point were also calculated.

```

% Modified version of the AFCoKaRE MATLAB template file SolvIVDifD.m used
% in the solution of Example 32.2 of "A First Course on Kinetics and
% Reaction Engineering. More specifically, this file performs the
% calculations for the first stage of semibatch operation where the
% pressure relief valve is closed.
%
function [ts1,nAs1,nBs1,nYs1,nZs1] = Example_32_2_semi_a
    % Known quantities and constants (in min, mol, L, atm and K)
    K = 8e-5;
    hZ = 8e-4; % mol/L/atm
    k = 5e-4; % L/mol/min
    V = 100*3.785; % L
    CA0 = 4.; % mol/L
    CB0 = 4.; % mol/L
    T = 80 + 273.15; % K
    Vhead = 8*3.785; % L
    Pmax = 1.5; % atm
    % Constants found in handbooks and other reference sources
    R = 0.08206; % L*atm/mol/K
    % Calculated constants
    nA0 = CA0*V;
    nB0 = CB0*V;
    nY0 = 0.0;
    nZ0 = 0.0;

```

Listing 5. Initial modifications to SolvIVDifD.m.

The second modifications, shown in Listing 6, occurred within the internal function `odeqns`. This is the code to evaluate the functions, f_3 through f_6 in equations (20) through (23). Values for the unknowns are passed into `odeqns` as a vector named `z`. The dependent variables, n_A , n_B , n_Y and n_Z , are first extracted from that vector, after which the rate and molar flow of Z are computed as described in the solution and equations (29) through (31). The functions are then evaluated and stored in the vector `dzdt`, which is returned by `odeqns`.

```
% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    nA = z(1);
    nB = z(2);
    nY = z(3);
    nZ = z(4);
    CA = nA/V;
    CB = nB/V;
    CY = nY/V;
    CZ = nZ/V;
    r = k*CA*CB*(1 - CY*CZ/K/CA/CB);
    nZdot = -V*r/(1 + V*hZ*R*T/Vhead);
    dzdt = [
        -V*r
        -V*r
        V*r
        nZdot + V*r
    ];
end % of internal function odeqns
```

Listing 6. Code that evaluates the ODEs.

The initial values were then entered, along with a large value for the final value of the time. This time should not be reached; the pressure in the headspace should reach 1.5 atm first. (If this final value of the time is reached, the code will print a warning and the file will need to be edited, making this value larger.) The final value of the dependent variable, n_Z , is set within the internal function named `stop`. The value is not entered directly, but instead, the variables `stop_when` is defined so that it will equal zero when n_Z reaches its desired final value. Here the partial pressure of Z is computed and the maximum pressure is subtracted. If the partial pressure equals the maximum pressure, the difference will be zero, so `stop_when` is set equal to this expression. These modifications are shown in Listing 7.

```

% Initial values
t0 = 0.0;
z0 = [
    nA0
    nB0
    nY0
    nZ0
];

tf = 100.; % should never reach this
options = odeset('Events',@stop);
[t, zz, te, ze, ie] = ode45(@odeqns,[t0, tf],z0,options);
% NOTE: For stiff equations replace ode45 with ode15s

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

    nZ = z(4);
    CZ = nZ/V;
    PZ = CZ/hZ;
    stop_when = PZ - Pmax;
end % of internal function stop

```

Listing 7. Setting the initial and final conditions.

The final modification simply involves extracting the results so they will be properly returned, Listing

8.

```

% Set the return values
ts1 = t;
nAs1 = zz(:,1);
nBs1 = zz(:,2);
nYs1 = zz(:,3);
nZs1 = zz(:,4);

end % of Example_32_2_semi_a.m

```

Listing 8. Setting the return values for the first stage of semi-batch processing.

Second stage of semi-batch operation. The template file SolvIVDifl.m was copied and saved as Example_32_2_semi_b.m; a copy accompanies this solution. The first modification, shown in Listing 9, entailed changing the introductory comment, renaming the function so that it received the duration of the first stage, the final moles of each of the reagents and the desired total processing time as arguments, and entering all the known constant quantities specified in the problem statement.

```

% Modified version of the AFCoKaRE MATLAB template file SolvIVDifD.m used
% in the solution of Example 32.2 of "A First Course on Kinetics and
% Reaction Engineering. More specifically, this file performs the
% calculations for the second stage of semibatch operation where the
% pressure relief valve is open.
%
function [ts2,nAs2,nBs2,nYs2,nZs2] =
Example_32_2_semi_b(t0,nA0,nB0,nY0,nZ0,tf)
    % Known quantities and constants (in min, mol, L, atm and K)
    K = 8e-5;
    hZ = 8e-4; % mol/L/atm
    k = 5e-4; % L/mol/min
    V = 100*3.785; % L
    CA0 = 4.; % mol/L
    CB0 = 4.; % mol/L
    T = 80 + 273.15; % K
    Vhead = 8*3.785; % L
    Pmax = 1.5; % atm
    % Constants found in handbooks and other reference sources
    R = 0.08206; % L*atm/mol/K

```

Listing 9. Initial modifications to SolvIVDifD.m.

The second modifications, shown in Listing 10, occurred within the internal function odeqns. This is the code to evaluate the functions, f_3 through f_5 in equations (20) through (22). This is the same as above, except f_6 does not need to be evaluated, and so the molar flow of Z does not need to be calculated.

```

% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    nA = z(1);
    nB = z(2);
    nY = z(3);
    CA = nA/V;
    CB = nB/V;
    CY = nY/V;
    CZ = nZ0/V;
    r = k*CA*CB*(1 - CY*CZ/K/CA/CB);
    dzdt = [
        -V*r
        -V*r
        V*r
    ];
end % of internal function odeqns

```

Listing 10. Code that evaluates the ODEs.

The initial values were then entered, and the ODEs were solved, Listing 11. (The final value was passed in as an argument and therefore did not need to be entered.)

```

% Initial and final values of the dependent variables (values for the
% independent variables are passed in as arguments)
z0 = [
    nA0
    nB0
    nY0
];

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

% Extract the final values of the independent and dependent variables
last_value = length(t);
t_f = t(last_value); % final value of the independent variable
z = zz(last_value,:); % final values of the dependent variables

```

Listing 11. Setting the initial and final conditions.

The final modification simply involves extracting the results so they will be properly returned, Listing 12.

```

% Set the values of the return variables (t is already set)
ts2 = t;
nAs2 = zz(:,1);
nBs2 = zz(:,2);
nYs2 = zz(:,3);
nZs2 = nZ0*ones(length(t),1);

end % of Example_32_2_semi_b.m

```

Listing 12. Setting the return values for the first stage of semi-batch processing.

Coordinating the calculations. The MATLAB function that coordinates the calculations, Example_32_2.m, is straightforward and will not be listed here (a copy accompanies this solution). Briefly, it begins with the entry of the known constants and a total processing time for the requested plot. It then calls Example_32_2_equil to solve the equilibrium equations, and reports the results. It next calls Example_32_2_semi_a to evaluate the first stage of semi-batch processing, printing out the duration of stage 1 and the corresponding conversion. Example_32_2_semi_b is then called to evaluate the second stage of semi-batch processing. The results from the two stages are concatenated, the conversion is calculated and the plot is made. Executing the function generates the plot (Figure 4) and the output shown in Listing 13.

```

>> Example_32_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 =

    3.9258
    0.0167

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

f =

    1.0e-05 *
   -0.2987
   -0.0000

Equilibrated Batch Reactor

pct_conversion =

    1.8542

final_pressure_atm =

    20.8304

Semi-Batch Reactor

first_stage_duration_min =

    0.6696

first_stage_pct_conversion =

    0.1335

```

Listing 13. Output from Example_32_2.