

# A First Course on Kinetics and Reaction Engineering

## Example 19.3

### Problem Purpose

This problem illustrates the analysis of a batch reactor where a perfectly mixed jacket containing flowing coolant is used to remove heat.

### Problem Statement

The rate expression for reaction (1) is given in equation (2). The heat of reaction (1) may be taken to be constant and equal to  $-22,200 \text{ cal mol}^{-1}$ . A jacketed, perfectly mixed batch reactor is charged with 4 L of a 2 M solution of A and heated to  $60 \text{ }^\circ\text{C}$ , at which time 18% of the A has been converted to B and the jacket is filled with water at  $60 \text{ }^\circ\text{C}$ . At that point,  $20 \text{ }^\circ\text{C}$  cooling water flow to the perfectly mixed jacket is initiated. The pressure in the reactor headspace is held constant during processing. The reaction is allowed to continue until the concentration of A reaches 0.2 M. The heat capacity of the solution is approximately constant and equal to  $440 \text{ cal L}^{-1} \text{ K}^{-1}$ , and its density is constant. The jacket volume is 0.5 L; the heat transfer area is  $0.6 \text{ ft}^2$  with a heat transfer coefficient of  $1.13 \times 10^4 \text{ cal ft}^{-2} \text{ h}^{-1} \text{ K}^{-1}$ . The cooling water may be taken to have a constant density of  $1 \text{ g cm}^{-3}$  and a constant heat capacity of  $1 \text{ cal g}^{-1} \text{ K}^{-1}$ . Plot the concentration of A versus time and the temperature versus time for water flow rates of  $2 \text{ kg min}^{-1}$ ,  $1 \text{ kg min}^{-1}$  and  $0.1 \text{ kg min}^{-1}$ .



$$r_1 = (2.59 \times 10^9 \text{ min}^{-1}) \exp\left(\frac{-16500 \text{ cal mol}^{-1}}{RT}\right) C_A \quad (2)$$

### Problem Analysis

The reactor is a batch reactor; the reactions taking place and their corresponding rate expressions are provided, so this is a reaction engineering problem. Mole balance design equations will be written for each reactant and product, and an energy balance will be written for the reactor. The coolant temperature may also vary over time, so an energy balance will also need to be written on the coolant. Solving the resulting set of equations will provide the answers to the questions posed in the problem statement.

### Problem Solution

The general mole balance for a batch reactor is given in equation (3). In this problem, there is only one reaction taking place, so the sum consists of a single term. Writing this equation for A and for B leads to equations (4) and (5).

$$\frac{dn_i}{dt} = V \sum_{\substack{j=\text{all} \\ \text{reactions}}} \nu_{i,j} r_j \quad (3)$$

$$\frac{dn_A}{dt} = -Vr_1 \quad (4)$$

$$\frac{dn_B}{dt} = Vr_1 \quad (5)$$

The energy balance on the reacting fluid in a batch reactor is given in equation (6). In this problem, the reacting fluid is a liquid, and it will be assumed to be an ideal solution. Since the headspace pressure is constant, the derivative of the pressure with respect to time is negligible. The fluid density is specified as being constant, so the solution volume will also be constant and the derivative of the volume with respect to time will equal zero. It is further assumed that any agitator being used does negligible work on the system. In this problem, the individual molar heat capacities are not provided; instead, the mass-specific heat capacity of the solution as a whole is given, and the problem notes that it is constant. In order to use the data provided in the problem statement, the sensible heat term must be replaced as was done in Example 19.1. The heat transfer between the coolant in the jacket and the reacting fluid is described using an overall heat transfer coefficient, as given in equation (7). With these assumptions, substitution of the heat transfer expression into the energy balance and expansion of the summation leads to the energy balance given in equation (8)

$$\dot{Q} - \dot{W} = \frac{dT}{dt} \sum_{\substack{i=\text{all} \\ \text{species}}} (n_i \hat{C}_{p,i}) + V \sum_{\substack{j=\text{all} \\ \text{reactions}}} (r_j \Delta H_j) - V \frac{dP}{dt} - P \frac{dV}{dt} \quad (6)$$

$$\dot{Q} = UA(T_e - T) \quad (7)$$

$$UA(T_e - T) = V\tilde{C}_{p,\text{soln}} \frac{dT}{dt} + V\Delta H_1 r_1 \quad (8)$$

The reactor jacket is assumed to be perfectly mixed with cooling water entering at  $T_e^0 = 20^\circ\text{C}$ . Within the jacket and at its outlet, the coolant temperature,  $T_e$ , is the same everywhere, due to the perfect mixing in the jacket. Since the density and heat capacity of the coolant are constant, an energy balance on the coolant is given by equation (9).

$$\dot{m}\tilde{C}_{p,e}(T_e^0 - T_e) - UA(T_e - T) = \rho_e V_e \tilde{C}_{p,e} \frac{dT_e}{dt} \quad (9)$$

The mole and energy balances can be rearranged so that the derivatives appear on the left side of the equals sign, as shown in equations (10) through (13). These equations represent a complete model for the batch reactor processing step described in the problem statement. They are a set of ordinary differential equations (ODEs) where the independent variable is  $t$ , and the dependent variables are  $n_A$ ,  $n_B$ ,  $T$  and  $T_e$ .

$$\frac{dn_A}{dt} = -Vr_1 \quad (10)$$

$$\frac{dn_B}{dt} = Vr_1 \quad (11)$$

$$\frac{dT}{dt} = \frac{UA(T_e - T) - V\Delta H_1 r_1}{V\hat{C}_{p,\text{soln}}} \quad (12)$$

$$\frac{dT_e}{dt} = \frac{\dot{m}\tilde{C}_{p,e}(T_e^0 - T_e) - UA(T_e - T)}{\rho_e V_e \tilde{C}_{p,e}} \quad (13)$$

The problem specification provides enough information to calculate the initial value of the dependent variables, so the design equations can be solved numerically using software for the solution of initial-value ODEs. Supplemental Unit S5 provides a brief overview of how such software works. There are many software packages you can use in order to do this; you should pick the one you are most comfortable using. No matter what software you elect to use, you will need to provide three things as input to that software:

- the initial values of the independent and dependent variables
- the final value of either  $t$  or one of the dependent variables
- code that evaluates each of the derivatives (i. e. the right side of equations (10) through (13)) given a value for  $t$  and values for each of the dependent variables along with the additional information provided in the problem specification

The processing step being analyzed in this problem is the second step in the operational protocol for the batch reactor. The time in the design equations is the elapsed time since the start of this second step in the protocol, so at the start of this step, the elapsed time is zero ( $t = 0$ ). The problem notes that at the start of the protocol, the reactor was charged with 4 L of a 2 M solution of A. In that first step, the solution was heated to 60 °C, and during the time it was being heated, 18% of the A was converted to B. Therefore at the start of the current processing step, the number of moles of A is 82 % of the original charge, and, according to the reaction stoichiometry, the number of moles of B at the start of the current processing step must equal 18% of the original A charged. The amount of A charged can be found from the concentration of A in the original solution ( $(C_A^0)_{\text{step 1}} = 2 \text{ M}$ ) and the reaction volume, ( $V = 4 \text{ L}$ ). Thus, at the start of the current processing step, the moles of A and B are given by equations (14) and (15). The initial temperature of both the reacting solution and the coolant are given as 60 °C in the problem statement.

$$n_A(t = 0) = n_A^0 = 0.82(C_A^0)_{\text{step 1}} V \quad (14)$$

$$n_B(t = 0) = n_B^0 = 0.18(C_A^0)_{\text{step 1}} V \quad (15)$$

The second thing that must be provided in order to solve the design equations is either the final value of  $t$  or the final value of one of the dependent variables. In this problem, the step ends when the concentration of A reaches 0.2 M. Knowing the reaction volume, this permits the calculation of the final moles of A according to equation (16).

$$n_A(t = t_f) = C_{A,f}V \quad (16)$$

The final thing that must be provided in order to solve the design equations numerically is code that evaluates each of the derivatives (i. e. the right hand sides of equations (10) through (13)) given values for the independent ( $t$ ) and dependent ( $n_A$ ,  $n_B$ ,  $T$  and  $T_e$ ) variables and the other information provided in the problem statement or found in handbooks and other reference sources. In order to evaluate the right hand side of equations (10) and (11), the reaction volume and the rate are needed. The former is given in the problem statement and the latter can be calculated using equation (2). To calculate the rate using equation (2), the gas constant, temperature and concentration of A are needed. The gas constant is known and the temperature is a dependent variable, so its value will be given. The concentration can be calculated using equation (17); the reaction volume is given in the problem statement and the moles of A is a dependent variable, so its value will be given.

$$C_A = \frac{n_A}{V} \quad (17)$$

In order to evaluate the right hand side of equation (12), the additional quantities we need are  $U$ ,  $A$ ,  $\widehat{C}_{p,\text{soln}}$  and  $\Delta H_1$  which are known from the problem statement and  $T_e$  which is a dependent variable, so its value will be given. In order to evaluate the right hand side of equation (13) we additionally need  $\dot{m}$ ,  $\tilde{C}_{p,e}$ ,  $\rho_e$ , and  $V_e$ . Once again, all of these quantities are given in the problem statement.

We now have everything we need in order to solve the design equations numerically. Doing so for each of the three specified coolant flow rates yields Figures 1 through 3. Looking at Figure 1, we see that it takes over 300 minutes to reach the desired final concentration of A. A closer examination reveals that at the beginning of the step, the rate is reasonable, but it decreases quickly. Too much heat is being removed through the reactor jacket, causing the reactor temperature to fall until it is essentially equal to the coolant temperature. At that low temperature, the rate is low and consequently it takes a very long time to reach the desired final concentration.

When the coolant flow rate is reduced to  $1 \text{ kg min}^{-1}$ , Figure 2, the desired final concentration of A is reached in just over 20 minutes. Looking at Figure 2b we see that initially the coolant is removing slightly less heat than is being generated by the reaction. As a consequence, the reactor temperature rises for the first 15 minutes. As the reactant is consumed, the rate decreases and consequently less heat is released by the reaction. Eventually, at ca. 15 minutes, the heat being removed by the coolant becomes greater than the heat being released, and from that point on the reactor temperature decreases. However, it never falls as low as in Figure 1 because the specified conversion is reached soon after the temperature starts decreasing.

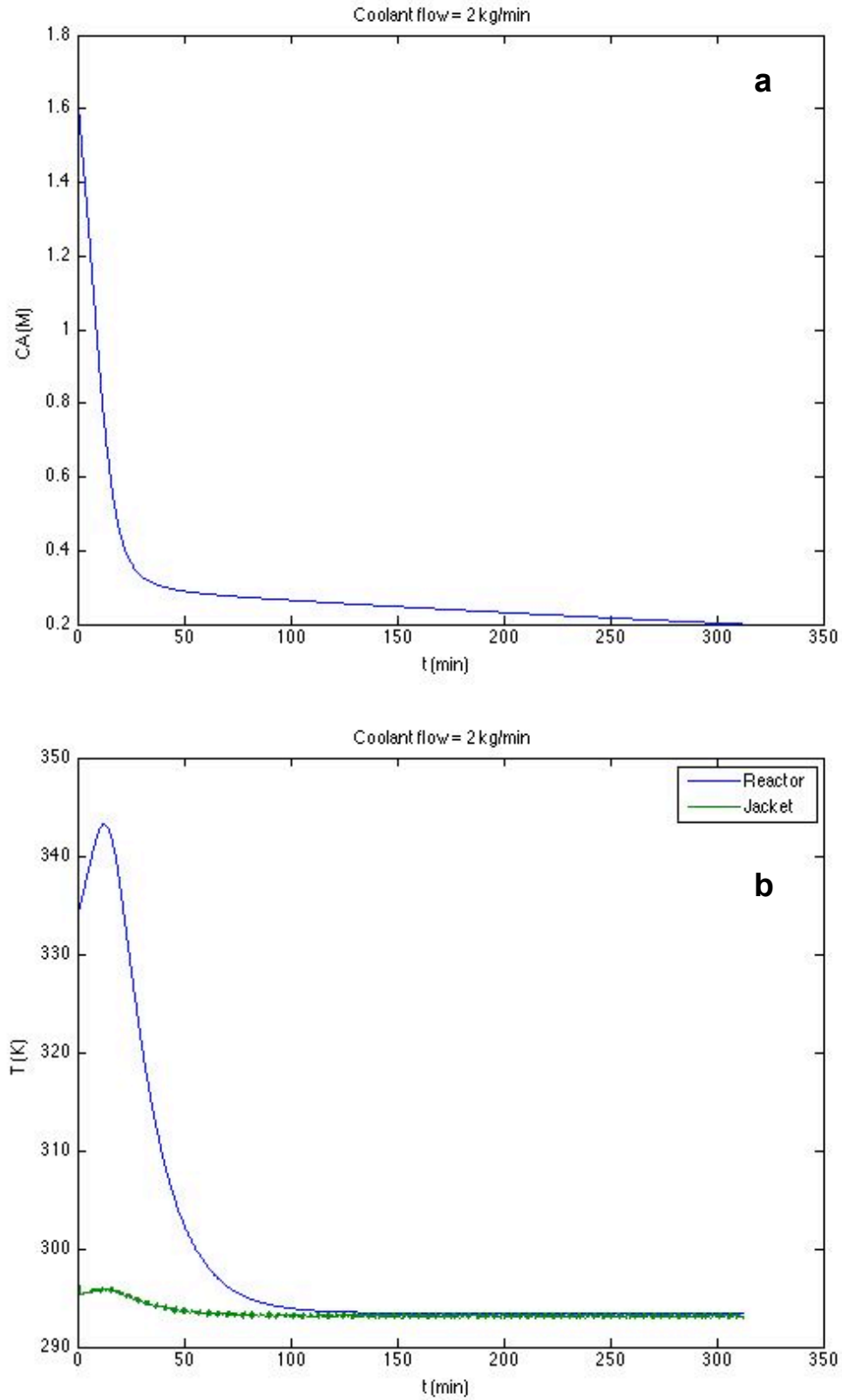


Figure 1. Batch reactor performance with a coolant flow of 2 kg/min. (a) concentration of A versus time and (b) reactor and coolant temperatures versus time.

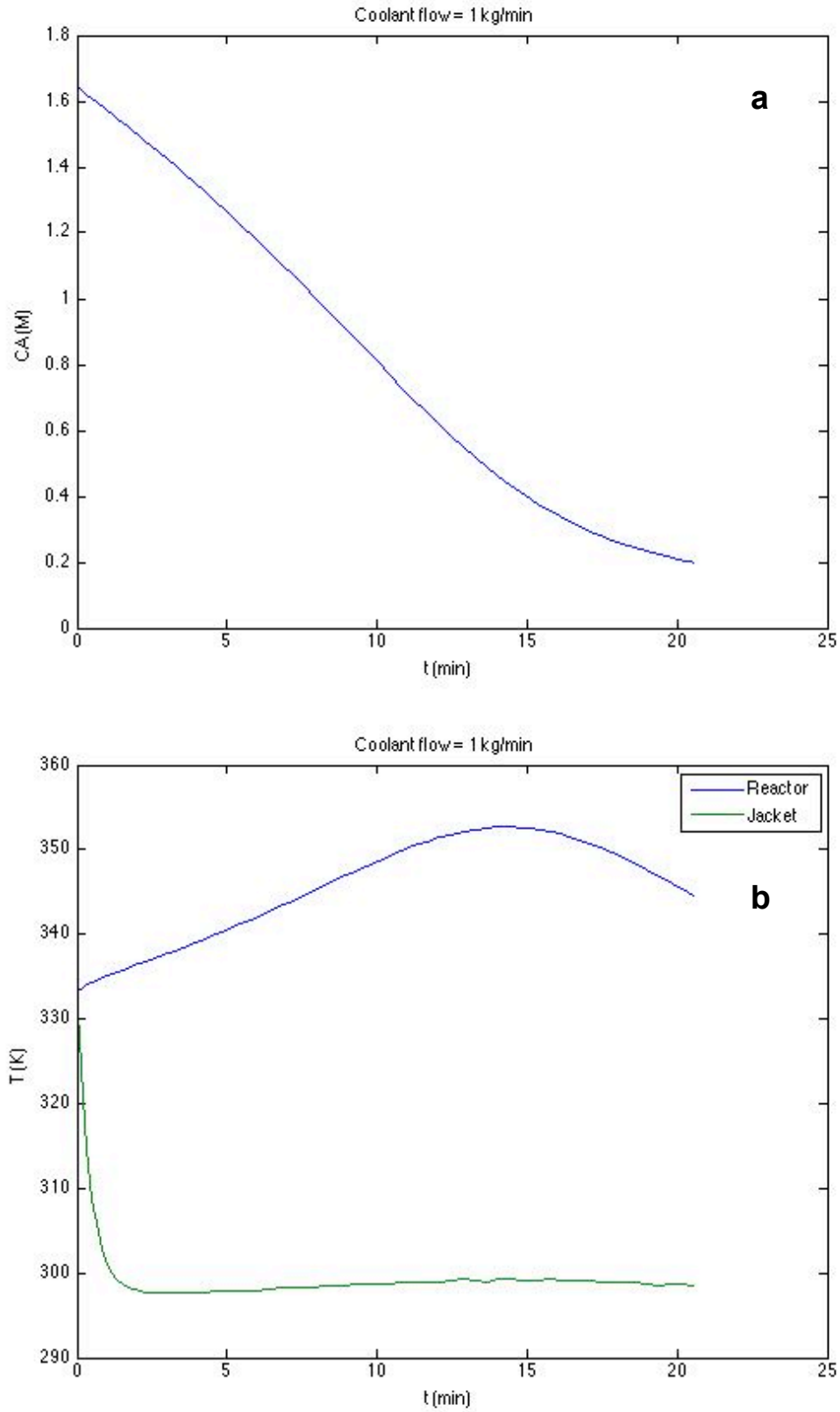


Figure 2. Batch reactor performance with a coolant flow of 1 kg/min. (a) concentration of A versus time and (b) reactor and coolant temperatures versus time.

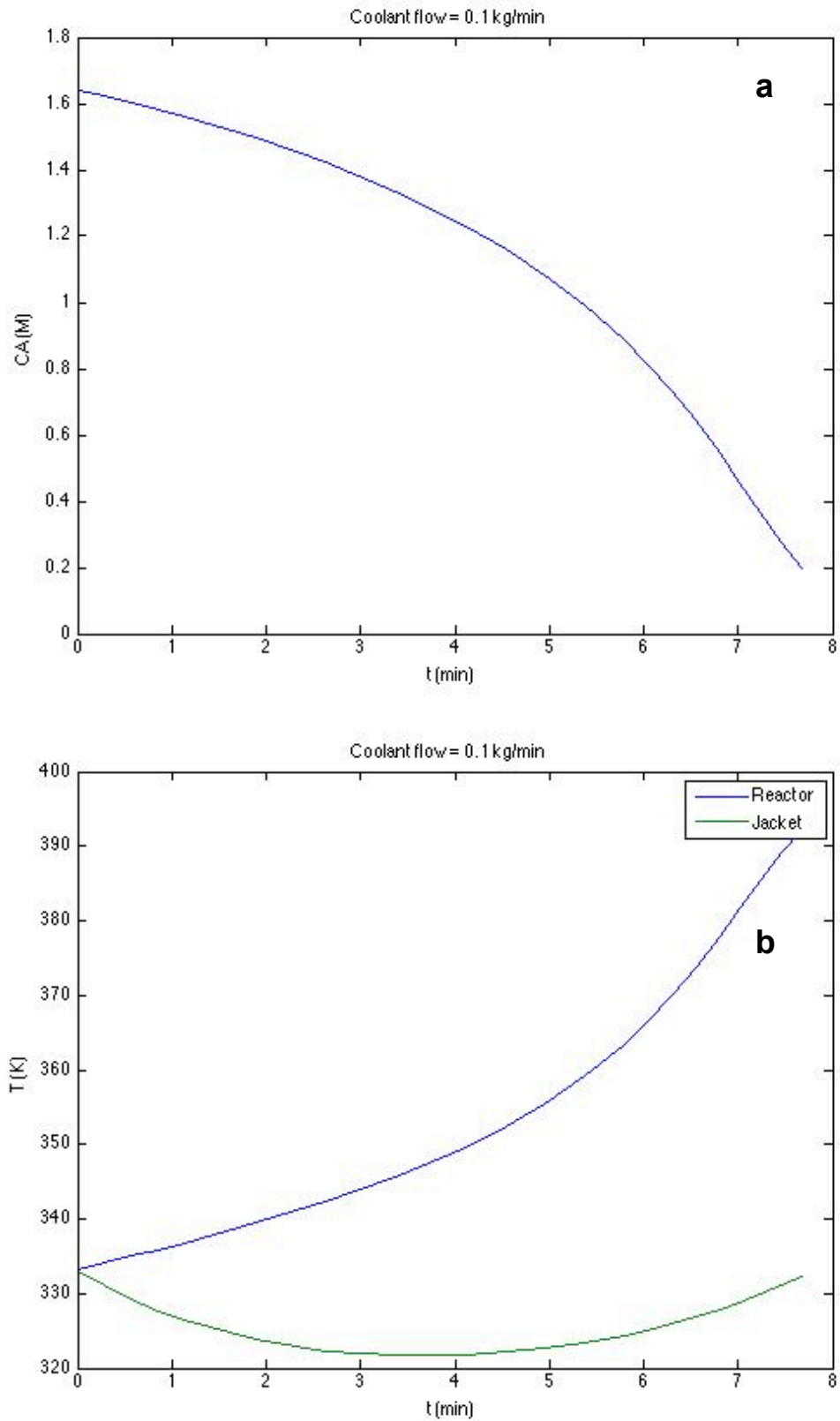


Figure 3. Batch reactor performance with a coolant flow of 0.1 kg/min. (a) concentration of A versus time and (b) reactor and coolant temperatures versus time.

If the coolant flow rate is reduced further to  $0.1 \text{ kg min}^{-1}$ , Figure 3, the specified final concentration is reached in less than 8 minutes. However, an examination of the reactor temperature shows that it continually rises, so that by the time the specified final concentration of A is reached it has increased by nearly  $60 \text{ }^\circ\text{C}$ . It does not appear that the reactor is running away, which would be indicated by near-vertical temperature versus time behavior. (In fact, if the processing step is allowed to run until essentially all the A is consumed, the temperature never reaches  $400 \text{ K}$ .) If the reactants and products remained stable at the final temperature and no additional reactions occurred at the higher temperature near the end of the processing step, this might be acceptable. However, if either the reactant or the product degraded at the higher temperatures, these processing conditions would not be acceptable.

### Calculation Details Using MATLAB

If you elect to use MATLAB to solve the design equations, Supplemental Unit S5 provides template files that can be used. In this problem, the equations are initial value ODEs and 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_19_3.m`; a copy of that file accompanies this solution. Since the function name must match the filename, I changed the name of the function to `Example_19_3`. At the same time, knowing that I won't need to use the results from these calculations in subsequent calculations, I changed the function so that it does not return any values. However I know I will want to run it using different coolant flow rates, so I additionally changed the function statement so that the coolant flow rate is passed in as an argument named `m`. 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. As a result of making them, the beginning of the file looks as shown in Listing 1.

```
% Modified version of the MATLAB template file SolvIVDifD.m used in the
% solution of Example 19.3 of "A First Course on Kinetics and Reaction
% Engineering."
%
function Example_19_3(m)
```

*Listing 1. Non-required modifications made at the beginning of the template file.*

The first *required* modification involves entering all the known quantities from the problem statement along with constants that will be needed (from handbooks or other reference sources). As these are entered, they should be converted to a consistent set of units. For this problem, I decided to use units of cal, mol, K, min,  $\text{ft}^2$ , kg and L. Knowing that I would use equation (2) to calculate the rate, I decided to enter the pre-exponential factor and the activation energy in that equation as constants. Doing so keeps all the constants at the top of the file and avoids equations that have long numbers in them. This is just a matter of programming style; it isn't required. These modifications are made immediately following the code in Listing 1, and they are shown in Listing 2.



```
% Known quantities and constants (cal, mol, K, min, ft2, kg, L)
k0 = 2.59e9; % min-1
E = 16500; % cal/mol
R = 1.987; % cal/mol/K
dH = -22200; % cal/mol
V = 4; % L
CA0_step1 = 2; % mol/l
nA0 = 0.82*CA0_step1*V; % mol
nB0 = 0.18*CA0_step1*V; % mol
Te0 = 20 + 273.15; %K
cp = 440; % cal/L/K
Ve = 0.5; %L
A = 0.6; % ft2
U = 1.13e4/60; % cal/ft2/min/K
rho = 1; % kg/L
cpe = 1000; % cal/kg/K
T0 = 60 + 273.15; %K
CAf = 0.2; % mol/L
% m in kg/min is passed in as an argument
```

*Listing 2. Results of the first required modification.*

The second *required* modification involves entering the code to evaluate the right hand side of the ODEs when they are written in the form shown in equation (18). In fact, the design equations (10) through (13) are written in this form, with the derivatives on the left side of the equals sign. Notice that the equations are provided as a vector quantity. Thus, it is necessary to map the dependent variables used in the problem statement ( $n_A$ ,  $n_B$ ,  $T$  and  $T_e$ ) to a vector  $z$ , and the corresponding derivatives are mapped to a vector  $dz/dt$ . I find it useful at the start of the internal function that will evaluate the derivatives, to define local variables with the names used in the problem statement. This modification is not required, but in my opinion, it makes the code more readable and easier to debug. In addition, the list of variables here serves as a reminder of the mapping of the problem statement variables to the vector  $z$ . The rate was calculated as described in the problem solution, after which the derivatives were evaluated using equations (10) through (13), saving the results in the vector  $dz/dt$ . The resulting code is shown in Listing 3.

$$\frac{dz}{dt} = f(z, t) \quad (18)$$

The third *required* modification involves providing the initial values of the independent and dependent variables. The initial values of the dependent variables are entered as a vector named  $z_0$ , and they must use the same mapping of the problem variables to the vector  $z_0$  as was used previously for  $z$ . The results of performing this modification are shown in Listing 4.

```

% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    nA = z(1);
    nB = z(2);
    T = z(3);
    Te = z(4);
    r1 = k0*exp(-E/R/T)*nA/V;
    dzdt = [
        -r1*V
        r1*V
        (U*A*(Te-T)-V*dH*r1)/V/cp
        (m*cpe*(Te0-Te)-U*A*(Te-T))/rho/Ve/cpe
    ];
end % of internal function odeqns

```

*Listing 3. Results of the second required modification.*

```

% Initial values
t0 = 0.0;
z0 = [
    nA0
    nB0
    T0
    Te0
];

```

*Listing 4. Results of the third required modification.*

The fourth and fifth required modifications involve providing the final value of one dependent variable, in this problem  $n_A$ . The first step (fourth required modification) is to select a value for  $t_f$  that is much greater than the actual final time. Of course, you don't know what the final time is at this point, so all you can do is set  $t_f$  to a large number. It is important to check the answer to make sure that the final time that is reported is smaller than this value; otherwise, you need to make this value larger and repeat the calculations. For this problem, I arbitrarily set  $t_f$  to 5000 min, so I'll need to check the final time that is calculated and make sure it is less than 5000 min.

The second step (fifth required modification) is to provide the final condition. Basically, this involves adding the code to calculate the value of a variable named `stop_when`. The variable, `stop_when`, should equal zero when the final condition is reached. Here the final condition we want to reach is that the concentration of A should equal 0.2 M. When I entered the data from the problem statement, I used `CAf` to represent this concentration. Hence, noting that `z(1)` is the moles of A, the final condition is given by equation (19). I can rearrange that equation so that there is a zero on the left-hand side as shown in equation (20). Clearly, the final condition will be reached when the right hand side of equation (20) equals zero. Therefore, for the fifth required modification, I set the variable `stop_when` equal to the right hand side of equation (20). The results of the fourth and fifth modifications are shown in Listing 5.

$$\frac{z(1)}{V} = C_{A,f} \quad (19)$$

$$0 = C_{A,f} - \frac{z(1)}{V} \quad (20)$$

```
tf = 5000.0;
options = odeset('Events',@stop);
[t, zz, te, ze, ie] = ode45(@odeqns,[t0, tf],z0,options);

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

    stop_when = CAf - z(1)/V;
end % of internal function stop
```

*Listing 5. Results of the fourth and fifth required modifications.*

The sixth and final required modification is to use the results from solving the ODEs to calculate whatever the problem requested. This problem specified that two plots should be made, and therefore code was added to generate those plots. The results of these modifications are shown in Listing 6.

```
% Plot CA vs. t
CA = zz(:,1)/V;
figure
plot(t,CA)
title(['Coolant flow = ',num2str(m),' kg/min'])
xlabel('t (min)')
ylabel('CA (M)')
% Plot Te and T vs. t
figure
plot(t,zz(:,3),t,zz(:,4))
title(['Coolant flow = ',num2str(m),' kg/min'])
xlabel('t (min)')
ylabel('T (K)')
legend('Reactor','Jacket')
```

*Listing 6. Results of the sixth required modification.*

Once the file containing all the modifications had been saved, it was executed by typing Example\_19\_3(2) at the MATLAB command prompt. This generated the plots shown previously as Figure 1 (a) and (b). The other figures were generated by typing Example\_19\_3(1) and Example\_19\_3(0.1) at the MATLAB command prompt.