

A First Course on Kinetics and Reaction Engineering

Example 19.1

Problem Purpose

This problem illustrates the analysis of a single step in the operational protocol for a perfectly mixed batch reactor.

Problem Statement

Irreversible, liquid phase reactions (1) and (2) take place in an adiabatic batch reactor where the headspace pressure is held constant. The rate expressions are given in equations (3) and (4). The rate coefficients exhibit Arrhenius type dependence on temperature with pre-exponential factors of $k_{0,1} = 8,110 \text{ L mol}^{-1} \text{ s}^{-1}$ and $k_{0,2} = 9,730 \text{ L mol}^{-1} \text{ s}^{-1}$ and activation energies of $E_1 = 75 \text{ kJ mol}^{-1}$ and $E_2 = 66 \text{ kJ mol}^{-1}$. The heats of reaction may be taken to be constant: $\Delta H_1 = -41 \text{ kJ mol}^{-1}$ and $\Delta H_2 = -36 \text{ kJ mol}^{-1}$. The solution may be treated as ideal with a constant heat capacity of $1.23 \text{ cal g}^{-1} \text{ K}^{-1}$ and a constant density of 1.02 g cm^{-3} .

The reactor is charged with 1900 L of solution at 180 °C and containing only A and B at concentrations of $C_A^0 = 2.9 \text{ mol L}^{-1}$ and $C_B^0 = 3.2 \text{ mol L}^{-1}$. An appropriate amount of catalyst is then added to start the reaction. How long will it take for the concentration of S to reach 1.0 mol L^{-1} ? What will the concentrations of B, R and Z equal at that time, and what will be the temperature?



$$r_1 = k_{0,1} \exp\left(\frac{-E_1}{RT}\right) C_A C_B \quad (3)$$

$$r_2 = k_{0,2} \exp\left(\frac{-E_2}{RT}\right) C_R C_B \quad (4)$$

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. Solving those equations will provide the answers to the questions posed in the problem statement.

Problem Solution

The mole balance design equation for a batch reactor is given in equations (5). In this problem, there are two reactions taking place, so the summation will include two terms. Writing equation (5) for the reactant, A, gives equation (6). Writing equation (5) for each of the other reactants and products, and

similarly expanding the summations and substituting values for the stoichiometric coefficients gives equations (7) through (11).

$$\frac{dn_i}{dt} = V \sum_{\substack{j=all \\ reactions}} \nu_{i,j} r_j \quad (5)$$

$$\frac{dn_A}{dt} = V(\nu_{A,1}r_1 + \nu_{A,2}r_2) = V((-1)r_1 + (0)r_2) = -Vr_1 \quad (6)$$

$$\frac{dn_B}{dt} = V(-r_1 - r_2) \quad (7)$$

$$\frac{dn_R}{dt} = V(r_1 - r_2) \quad (8)$$

$$\frac{dn_S}{dt} = V(r_1) \quad (9)$$

$$\frac{dn_Y}{dt} = V(r_2) \quad (10)$$

$$\frac{dn_Z}{dt} = V(r_2) \quad (11)$$

The energy balance design equation for a batch reactor is given in equation (12), which applies to reacting fluids that are ideal gas or liquid solutions. According to the problem statement, the reactor operates adiabatically, so the heat term is equal to zero, equation (13). The problem does not specify that the reactor has any moving boundaries; there is likely a rotating shaft attached to an impeller that is used to stir the reactor, but the work input from such a shaft is negligible, so the work term is also equal to zero, equation (14). 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 shown in equation (15). The density of the reacting solution is specified to be constant, so the solution volume will be constant and its derivative with respect to time will be equal to zero, equation (16). The pressure in the headspace is held constant, so the solution pressure will also be constant and the derivative of the pressure with respect to time will equal to zero, equation (17). After substitution of equations (13) through (17), expansion of the remaining summation and rearrangement, the energy balance takes the form given in equation (18).

$$\dot{Q} - \dot{W} = \frac{dT}{dt} \sum_{\substack{i=all \\ species}} (n_i \hat{C}_{p,i}) + V \sum_{\substack{j=all \\ reactions}} (r_j \Delta H_j) - V \frac{dP}{dt} - P \frac{dV}{dt} \quad (12)$$

$$\dot{Q} = 0 \quad (13)$$

$$\dot{W} = 0 \quad (14)$$

$$\frac{dT}{dt} \sum_{\substack{i=all \\ species}} (n_i \hat{C}_{p,i}) \Rightarrow \rho_{\text{soln}} V \tilde{C}_{p,\text{soln}} \frac{dT}{dt} \quad (15)$$

$$\frac{dV}{dt} = 0 \quad (16)$$

$$\frac{dP}{dt} = 0 \quad (17)$$

$$\frac{dT}{dt} = \frac{-V(r_1 \Delta H_1 + r_2 \Delta H_2)}{\rho_{\text{soln}} V \tilde{C}_{p,\text{soln}}} \quad (18)$$

The mole balances, equations (6) through (11), and the energy balance, equation (18) constitute a complete model for the batch reactor. We can therefore use those design equations to analyze the processing step described in the problem statement. To do so, we will need to solve the design equations; they are a set ordinary differential equations (ODEs). The independent variable is t , and the dependent variables are n_A , n_B , n_R , n_S , n_Y , n_Z and T . 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 given a value for t and values for each of the dependent variables along with the additional information provided in the problem specification, that is, when the equations are written with the derivatives on the left hand side, as in equations (6) through (11) and (18), code that evaluates the right hand side of the equations

First let's consider the initial values. At the start of the processing step described in the problem, the elapsed time for that step is zero ($t = 0$). The problem also notes that at that time, the only reagents present in the reactor are A and B. Therefore n_R , n_S , n_Y and n_Z are each equal to zero at $t = 0$. The initial moles of A and B are not specified, but their concentrations are, and that information can be used to calculate the initial moles of each, equations (19) and (20), since the reaction volume is specified. The initial temperature is specified to equal 180 °C in the problem statement, providing the last initial value.

$$n_A(t = 0) = n_A^0 = VC_A^0 \quad (19)$$

$$n_B(t = 0) = n_B^0 = VC_B^0 \quad (20)$$

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, we are asked to calculate the final time, so that clearly is not given. Instead, we are asked to calculate the time required for the concentration of S to reach 1 M. This can be used to calculate the final moles of S, equation (21), since, again, the reaction volume is specified.

$$n_S(t = t_f) = VC_{S,f} \quad (21)$$

The final thing that must be provided in order to solve the design equations numerically is code that evaluates each of the derivatives given values for the independent and dependent variables and the other information provided in the problem statement. Looking at the mole balances, the only quantities that appear on the right hand sides of those equations are the reaction volume, which is given and the rates of the two reactions. The rates can be calculated using equations (3) and (4). The necessary pre-exponential factors, activation energies and ideal gas constant are given in the problem statement or known. The temperature is one of the dependent variables, so it will be given. Thus, additional quantities are the concentrations of A, B and R. It is trivial to calculate these using equations (22) through (24) since the moles of the species are dependent variables and will be given and the reaction volume is a known constant specified in the problem statement.

$$C_A = \frac{n_A}{V} \quad (22)$$

$$C_B = \frac{n_B}{V} \quad (23)$$

$$C_R = \frac{n_R}{V} \quad (24)$$

Looking at the energy balance, the only additional quantities it contains are the heats of the reactions, the fluid density and the fluid heat capacity. These are all constants whose values are given in the problem statement. Thus, we can write code to evaluate the derivatives.

At this point, all the input that is needed to solve the design equations numerically is available. Doing so will yield the final value of the remaining independent and dependent variables, in this case t , n_A , n_B , n_R , n_Y , n_Z and T . The resulting values can be used to calculate the final concentrations of B, R and Z using equations (23), (24) and (25) (the final time and final temperature are found directly and do not need to be calculated).

$$C_Z = \frac{n_Z}{V} \quad (25)$$

The results are as follows:

$$t_f = 130 \text{ min}$$

$$T_f = 193.5 \text{ }^\circ\text{C}$$

$$C_B = 1.366 \text{ M}$$

$$C_R = 0.166 \text{ M}$$

$$C_Z = 0.834 \text{ M}$$

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. 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_19_1.m; a copy of that file accompanies this solution. Since the function name must match the filename, I changed the name of the function to Example_19_1. At the same time, knowing that I won't need to use the results from these calculations in subsequent calculations, I changed the function so that it does not return any values. The template file begins with a long set of comments describing what it does and how to use it; I replaced these comments with a brief comment stating the purpose of the modified version. None of these modifications were required. 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 19.1 of "A First Course on Kinetics and Reaction
% Engineering."
%
function Example_19_1
```

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 L, mol, s, K, J and g. These modifications are made immediately following the code in Listing 1, and they are shown in Listing 2.

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 (26). 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 , etc.) 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 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

two rates according to equations (3) and (4) (with equations (22) through (24) substituted). It then evaluates the derivatives using equations (6) through (11) and (18), saving the results in the vector $dzdt$. The resulting code is shown in Listing 3.

$$\frac{dz}{dt} = \underline{f}(\underline{z}, t) \quad (26)$$

```
% Known quantities and constants (in L, mol, s, K, J, g)
k01 = 8110;
k02 = 9730;
E1 = 75000;
E2 = 66000;
dH1 = -41000;
dH2 = -36000;
cp = 1.23*4.184;
rho = 1020;
V = 1900;
T0 = 180 + 273.15;
CA0 = 2.9;
CB0 = 3.2;
CSf = 1.0;
R = 8.314;
```

Listing 2. Results of the first required modification.

```
% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    nA = z(1);
    nB = z(2);
    nR = z(3);
    nS = z(4);
    nY = z(5);
    nZ = z(6);
    T = z(7);
    r1 = k01*exp(-E1/R/T)*(nA/V)*(nB/V);
    r2 = k02*exp(-E2/R/T)*(nR/V)*(nB/V);
    dzdt = [
        -V*r1
        V*(-r1 - r2)
        V*(r1 - r2)
        V*r1
        V*r2
        V*r2
        -V*(r1*dH1 + r2*dH2)/(rho*V*cp)
    ];
end % of internal function odeqns
```

Listing 3. Results of the second required modification.

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 $z0$, and they must use the same mapping of the problem variables to the vector $z0$ as was used previously for z .

The results of performing this modification are shown in Listing 4 where equations (19) and (20) were used to calculate the initial moles of A and B.

```
% Initial values
t0 = 0;
z0 = [
      CA0*V
      CB0*V
      0
      0
      0
      0
      T0
    ];
```

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_S . The first step (fourth 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 re-execute the function, Example_19_1. For this problem, I arbitrarily set t_f to 100,000 s, so I'll need to check the final time that is calculated and make sure it is less than 100,000 s.

The second step (fifth modification) is to provide the final condition. Basically, this involves adding the code to calculate the value of a variable named `stop_when`. The variable, `stop_when`, should equal zero when the final condition is reached. Here the final condition we want to reach is that the concentration of S should equal 1.0 M. When I entered the data from the problem statement, I used `CSf` to represent this concentration. Hence, noting that $z(4)$ is the moles of S, the final condition is given by equation (27). I can rearrange that equation so that there is a zero on the left-hand side as shown in equation (28). Clearly, the final condition will be reached when the right hand side of equation (28) equals zero. Therefore, for the fifth required modification, I set the variable `stop_when` equal to the right hand side of equation (28). The results of these two modifications are shown in Listing 5.

$$\frac{z(4)}{V} = C_{S,f} \quad (27)$$

$$0 = C_{S,f} - \frac{z(4)}{V} \quad (28)$$

The sixth and final *required* modification is to use the results from solving the ODEs to calculate whatever the problem requested. The values of t_f and T_f are calculated directly as `t_f` and `z(7)`. However, I decided that the final time would be more useful if the units were minutes, not seconds, and similarly I decided to calculate the final temperature in °C instead of K since the initial temperature was

given in °C. The problem also asked for the final concentrations of B, R and Z. These were calculated using equations (23), (24) and (25). The results of these modifications are shown in Listing 6.

```
tf = 100000;
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 = CSf - z(4)/V;
end % of internal function stop
```

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

```
t_f_min = t_f/60
CS = z(4)/V
CB = z(2)/V
CR = z(3)/V
CZ = z(6)/V
T_C = z(7)-273.15
```

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_1 at the MATLAB command prompt. The resulting output is shown in Listing 7.

```
>> Example_19_1

t_f_min =

    129.8046

CS =

     1

CB =

    1.3660

CR =

    0.1660

CZ =

    0.8340

T_C =

    193.5305
```

Listing 7. Results of executing the modified template file.