

# A First Course on Kinetics and Reaction Engineering

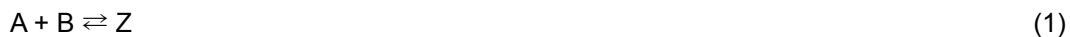
## Example 22.1

### Problem Purpose

This problem illustrates the quantitative analysis of a steady state CSTR.

### Problem Statement

The rate of liquid-phase reaction (1) is adequately described by the rate expression given in equation (2). Reactant A is fed to a steady state CSTR at a rate of  $0.01 \text{ lbmol min}^{-1}$ , and reactant B is fed at a rate of  $0.25 \text{ lbmol min}^{-1}$ . This corresponds to an inlet volumetric flow rate of  $0.08 \text{ ft}^3 \text{ min}^{-1}$ . The CSTR has a fluid volume of  $18 \text{ ft}^3$ , and it operates adiabatically. The heat of reaction may be taken to be constant and equal to  $-1.7 \times 10^4 \text{ BTU lbmol}^{-1}$ . The heat capacities of A, B and Z are equal to 1000, 180 and  $1200 \text{ BTU lbmol}^{-1} \text{ }^\circ\text{R}^{-1}$ , respectively, and they may be considered to be independent of temperature. What will the conversion and outlet temperatures equal if the combined feed enters at 600, 650 or  $700 \text{ }^\circ\text{R}$ .



$$r_1 = \left(1.2 \times 10^{14} \text{ ft}^3 \text{ lbmol}^{-1} \text{ min}^{-1}\right) \exp\left\{\frac{-23000 \text{ }^\circ\text{R}}{T}\right\} C_A C_B$$
$$\times \left[ 1 - \frac{C_Z}{\left(6.5 \times 10^{-13} \text{ ft}^3 \text{ lbmol}^{-1}\right) \exp\left\{\frac{20000 \text{ }^\circ\text{R}}{T}\right\} C_A C_B} \right] \quad (2)$$

### Problem Analysis

In this problem, a reaction is taking place in a CSTR and the kinetics of the reaction are known. Therefore it is a CSTR reaction engineering problem. The problem specifies that the reactor is operating at steady state. To model the system, a mole balance will be written for each reactant and product and an energy balance will be written for the reaction volume. Since the reactor is adiabatic, there is no heat transfer fluid, and consequently an energy balance on the heat transfer fluid is not needed. Thus the complete set of design equations for modeling the system comprise the mole balances and the energy balance on the reaction volume. Solving these equations will allow the calculation of the conversion and the outlet temperature for any set of operating parameters.

### Problem Solution

Figure 1 shows a schematic representation of the reactor. The quantities provided in the problem statement are given in a consistent set of units, and they have been entered in the diagram. The reacting fluid is stated to be a liquid, and it will be assumed that the liquid density is constant. With this

assumption, the outlet volumetric flow rate will equal the inlet volumetric flow rate as indicated in the schematic diagram.

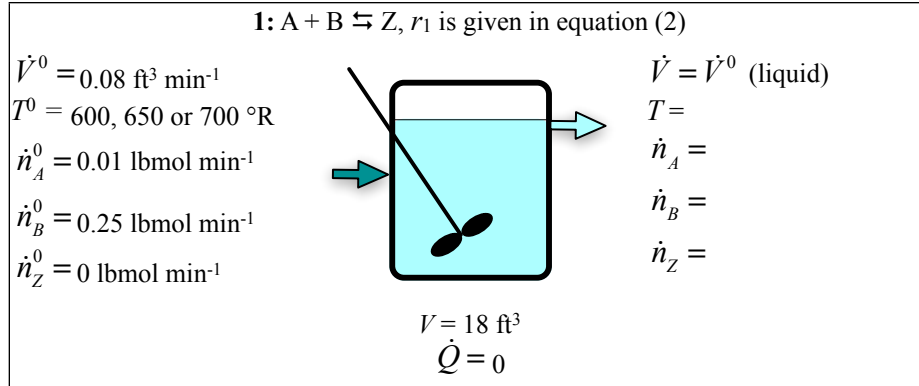


Figure 1. Schematic representation of the CSTR.

Mole balances can be written for each species present in the system. The generalized, steady state mole balance equation is given by equation (4). There is only one reaction taking place, so the summation reduces to a single term. Writing equation (4) once for each reactant and product in the system leads to equations (5) through (7).

$$0 = \dot{n}_i^0 - \dot{n}_i + V \sum_{\substack{j=\text{all} \\ \text{reactions}}} \nu_{i,j} r_j \quad (4)$$

$$0 = \dot{n}_A^0 - \dot{n}_A + V \nu_{A,1} r_1 = \dot{n}_A^0 - \dot{n}_A - V r_1 \quad (5)$$

$$0 = \dot{n}_B^0 - \dot{n}_B + V \nu_{B,1} r_1 = \dot{n}_B^0 - \dot{n}_B - V r_1 \quad (6)$$

$$0 = \dot{n}_Z^0 - \dot{n}_Z + V \nu_{Z,1} r_1 = \dot{n}_Z^0 - \dot{n}_Z + V r_1 \quad (7)$$

The general, steady state energy balance is given by equation (8). The heat term equals zero because the reactor is adiabatic and the work term can be assumed to be negligible because there are no shafts or moving boundaries other than, possibly, an agitator. The heat capacities are stated to be constant allowing evaluation of the integrals. Applying these simplifications results in the energy balance given in equation (9).

$$0 = \sum_{\substack{i=\text{all} \\ \text{species}}} \left( \dot{n}_i^0 \int_{T^0}^T \hat{C}_{p,i} dT \right) + V \sum_{\substack{j=\text{all} \\ \text{reactions}}} r_j \Delta H_j(T) - \dot{Q} + \dot{W} \quad (8)$$

$$0 = \left( \dot{n}_A^0 \hat{C}_{p,A} + \dot{n}_B^0 \hat{C}_{p,B} + \dot{n}_Z^0 \hat{C}_{p,Z} \right) (T - T^0) + V r_1 \Delta H_1(T) \quad (9)$$

In order for the design equations (equations (5) through (7) and equation (9)) to be solved, the number of unknowns must equal the number of design equations, in this case four. Figure 1 indicates that

the three outlet molar flow rates and the outlet temperature are unknown, so they will be chosen as the unknown variables. In that case the three mole balances and the energy balance can be written as shown in equations (10) through (13).

$$0 = f_1(\dot{n}_A, \dot{n}_B, \dot{n}_Z, T) = \dot{n}_A^0 - \dot{n}_A - Vr_1 \quad (10)$$

$$0 = f_2(\dot{n}_A, \dot{n}_B, \dot{n}_Z, T) = \dot{n}_B^0 - \dot{n}_B - Vr_1 \quad (11)$$

$$0 = f_3(\dot{n}_A, \dot{n}_B, \dot{n}_Z, T) = \dot{n}_Z^0 - \dot{n}_Z + Vr_1 \quad (12)$$

$$0 = f_4(\dot{n}_A, \dot{n}_B, \dot{n}_Z, T) = (\dot{n}_A^0 \hat{C}_{p,A} + \dot{n}_B^0 \hat{C}_{p,B} + \dot{n}_Z^0 \hat{C}_{p,Z})(T - T^0) + Vr_1 \Delta H_1(T) \quad (13)$$

Equations (10) through (13) do not contain integrals or derivatives. As such, one could attempt to solve them manually, using symbolic algebra software or using numerical methods software. A variety of software packages are available for doing so, and you should use the one you feel most comfortable with. Supplemental Unit S2 presents a brief introduction to the numerical solution of sets of non-differential equations like these using numerical methods software. No matter what software package you use, you typically will need to provide two things as input in order to solve the equations:

- Code that evaluates the functions,  $f_1$  through  $f_4$  in equations (10) through (13), given values of the unknown variables,  $\dot{n}_A$ ,  $\dot{n}_B$ ,  $\dot{n}_Z$  and  $T$
- A guess for the solution, that is, a guess for the values of  $\dot{n}_A$ ,  $\dot{n}_B$ ,  $\dot{n}_Z$  and  $T$  that cause all of the functions,  $f_1$  through  $f_4$ , to equal zero

The functions to be evaluated contain quantities other than the four unknown variables. In order to evaluate the functions, values will be needed for each of those other quantities. Here the inlet molar flow rates ( $\dot{n}_A^0$ ,  $\dot{n}_B^0$ ,  $\dot{n}_Z^0$ ), the reaction volume ( $V$ ), the heat capacities ( $\hat{C}_{p,A}$ ,  $\hat{C}_{p,B}$  and  $\hat{C}_{p,Z}$ ), the inlet temperature ( $T^0$ ) and the heat of reaction ( $\Delta H_1(T)$ ) are all known constants whose values are given in the problem statement.

The value of the rate ( $r_1$ ) will also be needed in order to evaluate the functions  $f_1$  through  $f_4$  in equations (10) through (13). The rate can be computed using equation (2), which introduces the concentrations of A, B and Z. Those concentrations can be calculated by expressing them in terms of the molar flow rates and the volumetric flow rate as in equations (14) through (16). The code that evaluates the functions will be given  $\dot{n}_A$ ,  $\dot{n}_B$  and  $\dot{n}_Z$ , and the volumetric flow rate is another known constant.

$$C_A = \frac{\dot{n}_A}{\dot{V}} \quad (14)$$

$$C_B = \frac{\dot{n}_B}{\dot{V}} \quad (15)$$

$$C_Z = \frac{\dot{n}_Z}{\dot{V}} \quad (16)$$

That is everything that is needed in order to evaluate the functions,  $f_1$  through  $f_4$ . The only other thing that will be needed in order to solve the equations numerically is a guess for the values of the unknowns. Since all that is needed is a guess, one possibility is to use the inlet molar flow rates and temperature as the guess. With that, equations (10) through (13) can be solved numerically to find  $\dot{n}_A$ ,  $\dot{n}_B$ ,  $\dot{n}_Z$  and  $T$ . The problem asked for the conversion and the outlet temperature. The latter is found directly by solving the design equations. The fractional conversion can be computed using equation (17) once the design equations have been solved to obtain  $\dot{n}_A$  ( $\dot{n}_A^0$  is a known constant).

$$f_A = \frac{\dot{n}_A^0 - \dot{n}_A}{\dot{n}_A^0} \quad (17)$$

Performing these calculations one finds for a feed temperature of 600 °R, the outlet temperature is 602 °R and the conversion is 68%; for a feed temperature of 650 °R, the outlet temperature is 653 °R and the conversion is 95% and for a feed temperature of 700 °R, the outlet temperature is 703 °R and the conversion is 82%. That is, the conversion passes through a maximum as the feed temperature increases. This is expected when a reversible exothermic reaction takes place adiabatically. Initially as the temperature is increased, the rate increases and this leads to increased conversion. Eventually the conversion approaches the equilibrium conversion. Then, as the temperature increases, the conversion decreases because the equilibrium constant, and hence the equilibrium conversion, decreases. Thus, one would expect that there is an inlet temperature that will maximize the conversion in this situation.

### Calculation Details Using MATLAB

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 used Example\_22\_1.m. Since the function name must match the filename, I changed the name of the function to Example\_22\_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. However I know I will want to run it using different feed temperatures, so I additionally changed the function statement so that the feed temperature is passed in as an argument named T0. 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 SolvNonDif.m used in the
% solution of Example 22.1 of "A First Course on Kinetics and Reaction
% Engineering."
%
function Example_22_1(T0)
```

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

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. Listing 2 shows the next part of Example\_22\_1.m where these modifications were made.

```
% Known quantities and constants (in consistent units)
VFR = 0.08; % ft3/min
nA0 = 0.01; % lbmol/min
nB0 = 0.25; % lbmol/min
nZ0 = 0; % lbmol/min
V = 18; % ft3
CpA = 1000; % BTU/lbmol/degR
CpB = 180; % BTU/lbmol/degR
CpZ = 1200; % BTU/lbmol/degR
dH = -1.7e4; % BTU/lbmol
k0 = 1.2e14; % ft3/lbmol/min
EoverR = 23000; % degR
K0 = 6.5e-13; % ft3/lbmol
HoverR = 20000; % degR
```

*Listing 2. Portion of the modified template file SolvNonDif.m where problem-specific and universal constants were entered in consistent units.*

The second required modification involves entering the code to evaluate functions  $f_1$  through  $f_4$ , equations (10) through (13). In the code, this occurs within an internal function named evalEqns; within evalEqns, both the unknowns and the equations are provided as vector quantities named z and f, respectively. Thus, it is necessary to map the variables used in the problem solution to represent the unknowns ( $\dot{n}_A$ ,  $\dot{n}_B$ ,  $\dot{n}_Z$  and  $T$ ) 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 statement variables to the vector z.

Recall from the solution that the functions contained variable quantities ( $r_1$ ,  $C_A$ ,  $C_B$  and  $C_Z$ ) that depend upon the unknowns. In the code being written here to evaluate the functions, we are given values of the unknowns as just described. Therefore, these variable quantities can be evaluated here using the equations given in the problem statement, specifically equations (4), (15), (16) and (2). The constants from the problem statement have already been entered and are available at this point in the MATLAB file, so the functions  $f_1$  through  $f_4$  can next be evaluated. The code containing all these modifications is shown in Listing 3.

```

% Function that evaluates the equations
function f = evalEqns(z)
    % mapping of unknowns into vector z
    nA = z(1);
    nB = z(2);
    nZ = z(3);
    T = z(4);
    % calculate variables quantities
    CA = nA/VFR;
    CB = nB/VFR;
    CZ = nZ/VFR;
    r1 = k0*exp(-EoverR/T)*CA*CB*(1-CZ/(K0*exp(HoverR/T)*CA*CB));
    f = [
        nA0 - nA - V*r1
        nB0 - nB - V*r1
        nZ0 - nZ + V*r1
        (nA0*CpA + nB0*CpB + nZ0*CpZ)*(T - T0) + V*r1*dH
    ];
end % of internal function evalEqns

```

*Listing 3. Portion of the modified template file SolvNonDif.m showing the entry of the code to evaluate the functions being solved.*

The third required modification is where guesses for the unknowns are provided. The guesses are entered in the array named `z_guess`. They must be entered using the same mapping of the unknowns to the vector `z` as was used above. 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. As noted in the problem solution, one possibility for a guess is to simply use the inlet values of  $\dot{n}_A$ ,  $\dot{n}_B$ ,  $\dot{n}_Z$  and  $T$ , and that is what is done in Listing 4.

```

% guesses for the solution
z_guess = [
    nA0
    nB0
    nZ0
    T0
];

```

*Listing 4. Portion of the modified template file SolvNonDif.m where guesses are provided for the unknowns.*

The final required modification only applies if you need to use the results from solving the set of equations to calculate other quantities. In this case, the problem asked for the conversion of A, which can be calculated using equation (17). Listing 5 shows the code that was used to do this; it also prints out the value of the temperature separately from the other unknowns, since the problem also asked for its value.

```

% Report the results
T = z(4)
pct_conv = 100*(nA0-z(1))/nA0

```

*Listing 5. Portion of the modified template file SolvNonDif.m where the results of solving the set of non-differential equations are used to calculate additional quantities.*

At this point, the modified template file can be used to solve the equations. To do so, the function name is typed at the MATLAB command prompt, with a feed temperature (in °R) as an argument within parentheses following the function name. Numerical methods for solving non-differential equations don't always converge to an acceptably accurate final answer (see Supplemental Unit S2 for more reasons why and what to do if this happens). Therefore, you should always examine the output from MATLAB closely; it should indicate that the equations were solved. If it doesn't, then you need to determine why a solution was not obtained. In addition, you should check the values of the functions that are calculated using the final solution and printed. These should all be close to zero if the solution is accurate.

In the present problem, accurate solutions are found for each of the feed temperatures specified. Listing 6 shows the output from MATLAB for a feed temperature of 700 °R.

```
>> Example_22_1(700)

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 solver found the following values for the unknowns:

z =
    0.0018
    0.2418
    0.0082
   702.5288

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

f =
  1.0e-10 *
    0.0000
    0.0000
   -0.0000
    0.4957

T =
   702.5288

pct_conv =
   81.8139
```

*Listing 6. Output from the execution of Example\_22\_1.m for a feed temperature of 700 °R.*