

A First Course on Kinetics and Reaction Engineering

Example 23.2

Problem Purpose

This problem illustrates the analysis of the start-up of a CSTR.

Problem Statement

A 500 cm³ CSTR needs to be started up. It will operate adiabatically with a feed of 1.0 cm³ s⁻¹ containing equal amounts of A and B (0.015 mol cm⁻³) at 50 °C. The heat capacity of the fluid is essentially equal to that of the solvent, 0.35 cal g⁻¹ K⁻¹ and can be considered to be constant. The (constant) density of the fluid is 0.93 gm cm⁻³. The reaction of interest is given in equation (1). The standard heat of this reaction is constant and equal to -20 kJ mol⁻¹. The rate expression is given in equation (2). For start-up purposes, the reactor will first be filled with pure solvent at 190 °C, and then a valve will be opened, admitting the feed. How long will it take for the reactor to reach 260 °C, and what will the conversion of A equal?



$$r_1 = 3.24 \times 10^{12} \left(\frac{\text{cm}^3}{\text{mol s}} \right) \exp \left\{ \frac{-25 \text{ kcal mol}^{-1}}{RT} \right\} C_A C_B \quad (2)$$

Problem Analysis

This problem involves a CSTR, and the reaction kinetics are known, so it is a reaction engineering problem. The CSTR is being started-up, therefore it is a transient CSTR problem. Appropriate transient mole and energy balances will be written and solved, and the results will be used to answer the questions posed.

Problem Solution

This problem involves the transient operation of a CSTR. The schematic in Figure 1 represents the system just after the inlet flow has been initiated. Some of the quantities provided in the problem statement will need to be converted to different units so that all quantities are in a consistent set of units. One way to do this is to convert the inlet temperature from °C to K and to convert the fluid heat capacity and the activation energy from cal or kcal to J. By doing so all volumes will use cm³, all times will use s, all molar quantities will use mol and all energies will use J. The problem states that the reacting fluid has a constant density, so the outlet and inlet volumetric flow rates will be equal and constant. The reactor is full at the start of the transient, and so, the reaction volume will also be constant.

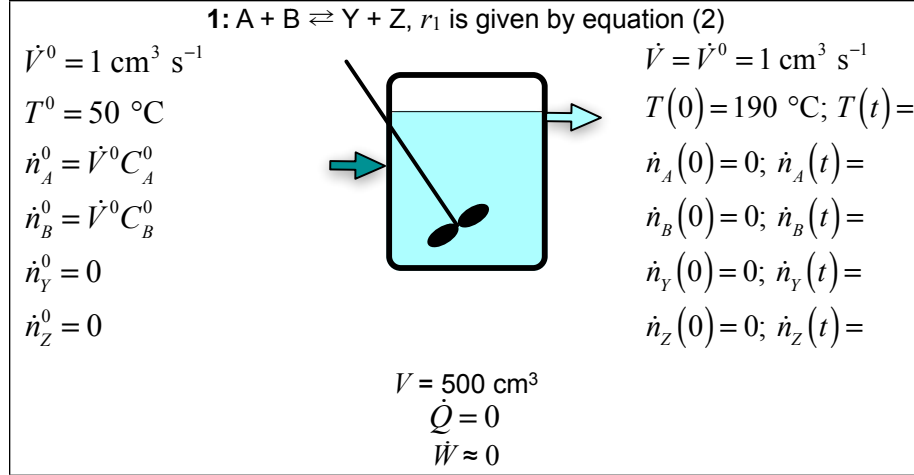


Figure 1. Schematic representation of the CSTR.

Mole balances can be written for each species present in the system. The generalized mole balance equation is given in equation (3). In this case, there is only one reaction taking place, so the summation reduces to a single term with $j = 1$. The fluid volume is constant (the reactor is full initially), as is the volumetric flow rate, so their derivatives with respect to time will equal zero. Applying these simplifications leads to equation (4).

$$\frac{V}{\dot{V}} \frac{d\dot{n}_i}{dt} + \frac{\dot{n}_i}{\dot{V}} \frac{dV}{dt} - \frac{\dot{n}_i V}{\dot{V}^2} \frac{d\dot{V}}{dt} = \dot{n}_i^0 - \dot{n}_i + V \sum_{\substack{j=\text{all} \\ \text{reactions}}} v_{i,j} r_j \quad (3)$$

$$\frac{d\dot{n}_i}{dt} = \frac{\dot{V}}{V} (\dot{n}_i^0 - \dot{n}_i + V v_{i,1} r_1) \quad (4)$$

Writing this mole balance for each reactant and product then leads to equations (5) through (8).

$$\frac{d\dot{n}_A}{dt} = \frac{\dot{V}}{V} (\dot{n}_A^0 - \dot{n}_A - V r_1) \quad (5)$$

$$\frac{d\dot{n}_B}{dt} = \frac{\dot{V}}{V} (\dot{n}_B^0 - \dot{n}_B - V r_1) \quad (6)$$

$$\frac{d\dot{n}_Y}{dt} = \frac{\dot{V}}{V} (\dot{n}_Y^0 - \dot{n}_Y + V r_1) \quad (7)$$

$$\frac{d\dot{n}_Z}{dt} = \frac{\dot{V}}{V} (\dot{n}_Z^0 - \dot{n}_Z + V r_1) \quad (8)$$

The general energy balance equation is given in equation (9). The heat term equals zero because the reactor is adiabatic and the work term is negligible. Instead of summing the individual heat capacities of the reagents, we can use the heat capacity of the fluid as a whole in this problem. However, that is provided as a mass heat capacity, so it will need to be multiplied by the total mass of the fluid, which, in

turn, is equal to the fluid volume times the fluid density. Also, the heat capacity is a constant allowing evaluation of the integral. The system involves a constant density liquid, and since the reactor is full at the start of the transient, the reaction volume will not change with time. Similarly, the pressure will be constant. This means that the derivatives of the reaction volume and the pressure with respect to time equal zero. There is only one reaction, so that summation consists of a single term. This results in the simplified energy balance given in equation (10), which was re-arranged so that only the derivative appears on the left side of the equals sign. There is no heat transfer fluid, so an energy balance on the heat transfer fluid is not used.

$$V \left(\sum_{\substack{i=\text{all} \\ \text{species}}} \frac{\dot{n}_i \hat{C}_{p-i}}{\dot{V}} \right) \frac{dT}{dt} - P \frac{dV}{dt} - V \frac{dP}{dt} = \dot{Q} - \dot{W} - \sum_{\substack{i=\text{all} \\ \text{species}}} \left(\dot{n}_i^0 \int_{T^0}^T \hat{C}_{p-i} dT \right) - V \sum_{\substack{j=\text{all} \\ \text{reaction}}} r_j \Delta H_j(T) \quad (9)$$

$$\frac{dT}{dt} = \frac{-\dot{V} \rho_{\text{fluid}} \tilde{C}_{p,\text{fluid}} (T - T^0) - V r_1 \Delta H_1(T)}{V \rho_{\text{fluid}} \tilde{C}_{p,\text{fluid}}} \quad (10)$$

The design equations (5) through (8) and (10) constitute a complete model for the transient reactor. We can therefore use those design equations to analyze the start-up procedure 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 \dot{n}_A , \dot{n}_B , \dot{n}_Y , \dot{n}_Z and T . The problem specification provides enough information to calculate the initial value of these 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

First let's consider the initial values. At the beginning of the startup procedure described in the problem, the elapsed time is zero ($t = 0$), and at that time, the reactor contains only solvent. Therefore \dot{n}_A , \dot{n}_B , \dot{n}_Y and \dot{n}_Z are each equal to zero at $t = 0$. (If the reactor contains only solvent, then there can't be any A, B, Y or Z in the effluent.) The initial temperature is specified to equal 190 °C in the problem statement, providing the last initial value.

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 time required to reach 260 °C, and the corresponding conversion of A. Thus the final condition for this analysis is that the temperature equals 260 °C.

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 and the energy balance, the only quantities other than the dependent variables (\dot{n}_A , \dot{n}_B , \dot{n}_Y , \dot{n}_Z and T) that appear on the right hand sides of those equations are the volumetric flow rate (\dot{V}), the reaction volume (V), the inlet molar flow rates (\dot{n}_A^0 , \dot{n}_B^0 , \dot{n}_Y^0 and \dot{n}_Z^0), the reaction rate (r_1), the fluid density (ρ_{fluid}), the fluid mass-specific heat capacity ($\tilde{C}_{p,fluid}$), the inlet temperature (T^0) and the heat of reaction ($\Delta H_1(T)$). All of these quantities, except the reaction rate, are known constants.

The reaction rate can be calculated using equation (2), which introduces the concentrations of A and B. It is trivial to calculate these using equations (11) and (12) since the moles of the species are dependent variables and will be given and the volumetric flow rate is a known constant specified in the problem statement.

$$C_A = \frac{\dot{n}_A}{\dot{V}} \quad (11)$$

$$C_B = \frac{\dot{n}_B}{\dot{V}} \quad (12)$$

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 , \dot{n}_A , \dot{n}_B , \dot{n}_Y and \dot{n}_Z . With that information, the final conversion can be calculated using equation (13). Doing so reveals that the specified final temperature is reached in 1450 s and the corresponding conversion of A is 97.3 %.

$$f_A = \frac{\dot{n}_A^0 - \dot{n}_A}{\dot{n}_A^0} \quad (13)$$

If you plot the outlet concentration of A versus time and the outlet temperature versus time, Figures 2 and 3, you can see that the system appears to be approaching a steady state. Before it does, however, it displays some interesting transient behavior. This behavior is easy to understand. Initially as the reactant flows into the reactor, it mixes with the solvent and causes the temperature to drop since the feed is colder than the solvent. Then, as the concentration of A builds, the reaction rate increases, leading to the release of energy (since the reaction is exothermic) and causing the temperature to increase (since the reactor is adiabatic). As it does so, the amount of A within the reactor decreases due to the larger reaction rate.

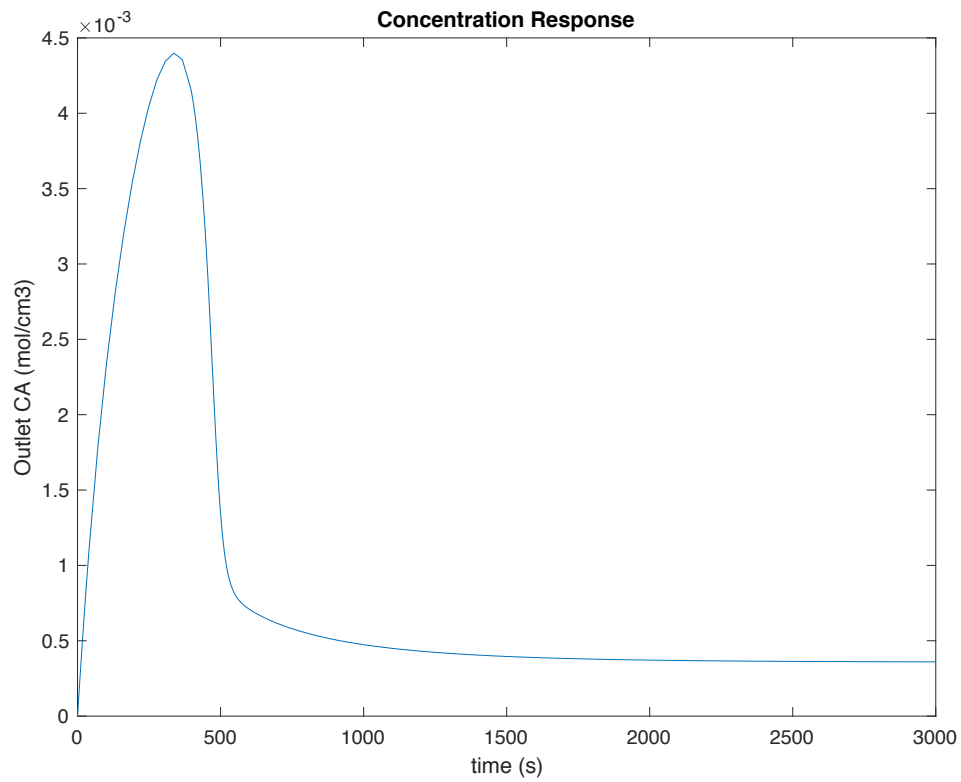
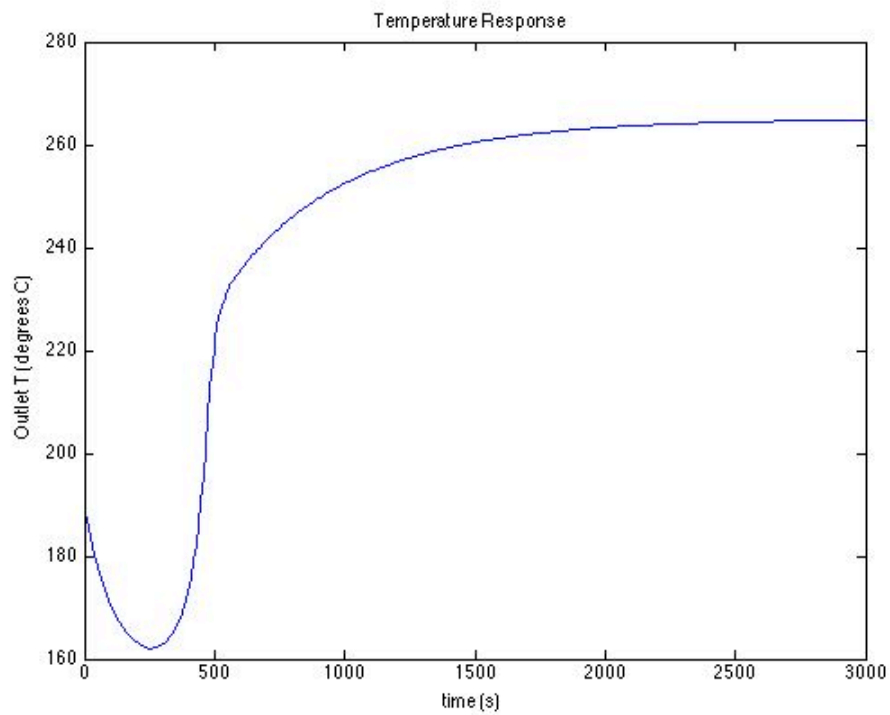


Figure 2. Outlet concentration of A vs. time.



*Figure 3. Outlet temperature versus time.***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_23_2.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_23_2. 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 SolvIVDifI.m used in the
% solution of Example 23.2 of "A First Course on Kinetics and Reaction
% Engineering."
%
function Example_23_2
```

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 cm^3 , mol, s, K, J and g. At the same time, I calculated the constant inlet molar flow rates of A and B. Note, to avoid confusing inlet quantities with initial values, I used an added "in" for the inlet quantities instead of an added "0". 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 (14). 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}_B , \dot{n}_Y , \dot{n}_Z and T) 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 next calculates the outlet concentrations of A and B and the rate according to equations (11), (12) and (2). It

then evaluates the derivatives using equations (5) through (8) and (10), saving the results in the vector $dzdt$. The resulting code is shown in Listing 3.

$$\frac{dz}{dt} = \underline{f}(\underline{z}, t) \quad (14)$$

```
% Known quantities and constants (in consistent units)
V = 500; % cm3
VFR = 1; % cm3/s
CAin = 0.015; % mol/cm3
CBin = CAin;
Tin = 50 + 273.15; % K
cp = 0.35*4.1840; % J/g/K
rho = 0.93; % g/cm3
dH = -20000; % J/mol
k0 = 3.24e12; % cm3/mol/s
E = 25000*4.1840; % J/mol
R = 8.31446; % J/mol/K
Tout0 = 190 + 273.15; % K
Toutf = 260 + 273.15; % K
nYin = 0; % mol/s
nZin = 0; % mol/s
% Other calculated constants
nAin = CAin*VFR; % mol/s
nBin = CBin*VFR; % mol/s
```

Listing 2. Results of the first required modification.

```
% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    nA = z(1);
    nB = z(2);
    nY = z(3);
    nZ = z(4);
    T = z(5);
    % Other variable quantities
    CA = nA/VFR;
    CB = nB/VFR;
    r = k0*exp(-E/(R*T))*CA*CB;
    % Evaluate the ODEs
    dzdt = [
        VFR/V*(nAin - nA - V*r)
        VFR/V*(nBin - nB - V*r)
        VFR/V*(nYin - nY + V*r)
        VFR/V*(nZin - nZ + V*r)
        (-VFR*rho*cp*(T-Tin) - V*r*dH)/(V*rho*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 .

Recall that since the reactor initially contains only solvent at 190 °C, the outlet flow rate of A, B, Y and Z are initially equal to zero since they are not present in the reactor. The results of performing this modification are shown in Listing 4.

```
% Initial values
t0 = 0;
z0 = [
    0
    0
    0
    0
    Tout0;
];
```

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 T . 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_23_2. For this problem, I arbitrarily set t_f to 3000 s, so I'll need to check the final time that is calculated and make sure it is less than 3000 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 outlet temperature should equal 260 °C. When I entered the data from the problem statement, I used `Tout_f` to represent this temperature. Hence, noting that $z(5)$ is the outlet temperature, the final condition is given by equation (15). I can rearrange that equation so that there is a zero on the left-hand side as shown in equation (16). Clearly, the final condition will be reached when the right hand side of equation (16) equals zero. Therefore, for the fifth required modification, I set the variable `stop_when` equal to the right hand side of equation (16). The results of these two modifications are shown in Listing 5.

$$z(5) = T_{out,f} \quad (15)$$

$$0 = z(5) - T_{out,f} \quad (16)$$

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 is calculated directly as t_f . The conversion can be calculated using equation (13), noting that $z(1)$ is the final moles of A. The results of these modifications are shown in Listing 6.

Once the file containing all the modifications had been saved, it was executed by typing Example_23_2 at the MATLAB command prompt. The resulting output is shown in Listing 7. An additional MATLAB file named Example_23_2_plot.m was also created to produce the plots shown as Figures 2 and 3. The code in that file will not be discussed here; the primary difference is that a final value was chosen for the time, so the code is based on the template file SolvDiflVI.m, and it generates the plots as output.

```
tf = 3000; % Don't want to reach this value
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;
    % final condition
    stop_when = z(5) - Toutf;
end % of internal function stop
```

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

```
% Report the required time and the conversion
required_time = t_f
conversion = (nAin - z(1))/nAin
```

Listing 6. Results of the sixth required modification.

```
>> Example_23_2

required_time =
    1.4507e+03

conversion =
    0.9734
```

Listing 7. Results of executing the modified template file.