A First Course on Kinetics and Reaction Engineering

Example 38.1

Problem Purpose

This example illustrates the calculation of the effectiveness factor and demonstrates its use in the ideal PFR design equations for a first-order reaction with spherical catalyst particles.

Problem Statement

Consider the gas phase, heterogeneous catalytic reaction A → B taking place in a packed bed PFR operating isothermally and with negligible pressure drop. The reaction rate per unit catalyst particle volume is first order in A, and at the operating temperature, the rate coefficient is equal to 2 x 10^{-2} s^{-1}. The void fraction within the packed bed is equal to 0.4. Pure A, flowing at 1.0 cm^3 s^{-1} is fed to the reactor, this corresponds to a feed concentration of 1.16 mol L^{-1}. There are no temperature gradients anywhere in the reactor or in the catalyst particles, and there are no concentration gradients external to the spherical catalyst particles (diameter = 3 mm). The effective diffusion coefficient of A within the porous catalyst particles is equal to 2.7 x 10^{-7} cm^2 s^{-1}. If the reactor diameter is 2.5 cm and the conversion of A in the reactor is to equal 85%, what reactor length is required?

Problem Analysis

This is a reaction engineering problem involving a packed bed reactor. If concentration gradients are present within the catalyst particles, then the ideal PFR design equations must be modified or replaced so that the gradients are accounted for properly. Since the reaction is first order and there are no temperature effects, the effectiveness factor is expected to be constant throughout the reactor. Therefore, the ideal PFR design equations can be used as long as the reaction rate is multiplied by the effectiveness factor.

Problem Solution

The following quantities are specified in the problem statement: k = 2 x 10^{-2} s^{-1}, ε = 0.4, $\dot{V}$ = 1.0 cm^3 s^{-1} (since the total moles do not change upon reaction and the reactor is isothermal, the volumetric flow rate will be constant), $C_{A,in}$ = 1.16 mol L^{-1}, D_p = 3 mm, D_{eA} = 2.7 x 10^{-7} cm^2 s^{-1}, D_{PFR} = 2.5 cm and f_A = 0.85. We are asked to determine the reactor length, L.

As noted in the problem analysis, the ideal PFR design equation can be used as long as the rate is multiplied by the effectiveness factor, η. Thus, a mole balance on A is given by equation (1), where the first order rate expression has been substituted for the rate. The (1 − ε) term in equation (1) is needed to convert the rate per volume of catalyst (which is what the given rate coefficient yields) to the rate per volume of reactor (which is used in the ideal PFR design equation). Specifically ε is the void volume per reactor volume, so (1 − ε) is the catalyst volume per reactor volume, and multiplying it by the rate per catalyst volume gives the rate per reactor volume. The initial condition needed to solve equation (1) is that the molar flow rate of A at the inlet is equal to the volumetric flow rate times the feed concentration of A, as given in equation (2). The final value of the molar flow rate of A is known from the specified conversion, as given in equation (3).

$$\frac{d\dot{n}_A}{dz} = \frac{-\pi D_{PFR}^2}{4}\left(1-\varepsilon\right)\eta\left(kC_A\right) \tag{1}$$

$$\dot{n}_A\left(z=0\right) = \dot{V}C_{A,in} \tag{2}$$

$$\dot{n}_A\left(z=L\right) = \dot{V}C_{A,in}\left(1-f_A\right) \tag{3}$$

Equation (1) can be solved numerically; to do so, in addition to the initial and final conditions, code must be provided that evaluates the right-hand side of equation (1) given values for z and $\dot{n}_A$. Looking at the right-hand side of equation (1), the values of D_{PFR}, ε and k are known. The concentration of A can be calculated using equation (4), where $\dot{V}$ is also known. That leaves η as the only quantity that is needed in order to evaluate the right-hand side of equation (1).

$$C_A = \frac{\dot{n}_A}{\dot{V}} \tag{4}$$

The problem states that the catalyst particles are spherical and the reaction is first-order. In that case, the Thiele modulus can be calculated using equation (5). You may notice that in equation (38.14) in the informational reading for Unit 38, the rate coefficient is multiplied by the apparent catalyst density in the numerator of the expression for the Thiele modulus. That is because equation (38.14) assumes the rate coefficient to be normalized per unit catalyst mass. Thus, multiplying by the apparent density re-normalizes the rate coefficient per unit catalyst particle volume. Here, in this problem, the rate coefficient is already normalized per unit catalyst particle volume, and therefore it should not be multiplied by the apparent catalyst density.

$$\phi = R_{part}\sqrt{\frac{k}{D_{eA}}} \tag{5}$$

The rate coefficient and the effective diffusivity are known. The particle diameter is also known; dividing it in half gives the particle radius and allows the Thiele modulus to be calculated. A value of 40.8 is found upon doing so. Knowing the Thiele modulus, the effectiveness factor for an isothermal first order reaction in a spherical catalyst particle can be computed using equation (6). The resulting value is found to be η = 0.072.

$$\eta = \frac{3}{\phi}\left(\frac{1}{\tanh\phi} - \frac{1}{\phi}\right) \tag{6}$$

With that, the code needed to solve equation (1) numerically can be written, and equation (1) can be solved to find that the required reactor length is L = 450 cm. The effectiveness factor in this case is very small. Had one made the mistake of using the ideal PFR design equation to calculate the length without correcting for the catalyst concentration gradients, the length would have incorrectly been found to equal 32.2 cm. If a 32.2 cm long reactor was built and the reaction was run using it, the conversion would only be 12.7% instead of the specified 85%. Clearly, it is very important to properly account for concentration (and temperature) gradients when they exist!

Calculation Details Using MATLAB

The equation to be solved here, equation (1), is an initial value ODE. The final value of the dependent variable is known, equation (3), so the MATLAB template file SolvIVDifD.m was modified and used to solve that ODE. To begin, I saved a copy of the template file as Example_38_1.m and changed the function declaration statement to match the filename without the .m extension. I deleted the long comment at the top of the file and added a short comment on the first line of the function to describe its purpose. I then entered the values of all quantities given in the problem statement in consistent units. In order to solve equation (1) I will need a value for the effectiveness factor, and since it is a constant in this problem, I entered code to calculate it next. To do so, the Thiele modulus was first calculated using equation (5) and then the effectiveness factor was calculated using equation (6). The results of these modifications are shown in Listing 1.

```
function Example_38_1
%Example_38_1 Code used in the solution of Example 38.1 of AFCoKaRE

    % Known quantities and constants
    k = 2.0E-2; % /s
    eps = 0.4;
    VFR = 1.0; % cm^3/s
    CA0 = 1.16E-3; % mol/cm^3
    Rp = 0.3/2; % cm
    DA = 2.7E-7; % cm^2/s
    fA = 0.85;
    D = 2.5; % cm

    % Calculate the effectiveness factor
    Thiele = Rp*sqrt(k/DA) % solution, equation 5
    eff_factor = 3/Thiele*(1/tanh(Thiele) - 1/Thiele) % solution, equation 6
```

Listing 1. Function declaration, initial comment, entry of given quantities and calculation of the effectiveness factor.

The template file requires that the internal function named odeqns be modified so that it returns the right hand side of equation (1). In order to do that, the concentration of A must be calculated according to equation (4) and substituted into equation (1). I chose to simply substitute equation (4) into equation (1) and then evaluate the resulting expression directly. This is shown in Listing 2.

```
% Function that evaluates the ODEs
function dzdt = odeqns(t,z)
    dzdt = [
        -pi()*D^2/4*k*eff_factor*(1-eps)*z(1)/VFR;
    ]; % solution, equations 1 and 4
end % of internal function odeqns
```

Listing 2. Required modification of the internal function odeqns.

The template file was next modified to provide the initial and final conditions. In this problem, the initial value of z is z = 0 and the initial value of $\dot{n}_A$ is calculated according to equation (2). The final condition is specified by making two modifications. First, a large final value is specified for the *independent* variable (`tf` in the MATLAB code) so that the known final value of the *dependent* variable is reached first. Second, the internal function named `stop` is modified so that the variable `stop_when` will equal zero when the final condition is reached. Here, the final condition is given by equation (3). Subtracting the molar flow rate of A at the reactor outlet from both sides of equation (3) yields equation (7). Clearly, the right-hand side of equation (7) is equal to zero when the final condition is satisfied, so the variable, `stop_when`, is set equal to the right-hand side of equation (7). These modifications are shown in Listing 3.

$$0 = \dot{V}C_{A,in}\left(1 - f_A\right) - \dot{n}_A\left(z = L\right) \qquad (7)$$

```
% Initial values
t0 = 0;
z0 = [
    CA0*VFR % solution, equation 2
];

tf = 460.;
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 = (1-fA)*CA0*VFR - z(1); % solution, equation 7
end % of internal function stop
```

Listing 3. Modifications required to specify the initial and final conditions.

With those modifications, the template file is set up to solve equation (1) with the initial and final conditions specified in equations (2) and (3). This actually occurs when the call is made to the built-in MATLAB function ode45 in Listing 3. The template file was modified so that after the solution has been obtained, the conversion is calculated (as a check to make sure it equals 0.85 and the length of the reactor (the final value of the independent variable) is listed. Those modifications can be seen in Listing 4.

```
    % 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

    % Warn if the independent variable reached tf above
    if (t_f >= tf)
        display('WARNING: the final value of the independent variable was')
        display('    reached, instead of the final value of the dependent')
        display('    variable. Set tf to a larger value and re-execute');
        display('    the function')
    end % if (t_f >= tf)

    check = (CA0*VFR - z(1))/CA0/VFR
    L = t_f

end % of Example_38_1
```

Listing 4. Remainder of Example_38_1.m after modifications are complete.

The modified file, Example_38_1.m accompanies this solution. It can be executed by placing it in the MATLAB path and typing “Example_38_1” at the MATLAB command prompt. Doing so produces the output shown in Listing 5.

```
>> Example_38_1

Thiele =
   40.8248

eff_factor =
    0.0717

check =
    0.8500

L =
  449.2802
```

Listing 5. Output generated upon execution of Example_38_1.m.