

A First Course on Kinetics and Reaction Engineering

Example 33.2

Problem Purpose

This problem will help you determine whether you have mastered the learning objectives for this unit. Specifically, it illustrates the use of the axial dispersion model and shows how it will predict results intermediate between a CSTR and a PFR for an appropriate value of the dispersion coefficient.

Problem Statement

The reaction between A and B produces the desired product, D according to reaction (1), but B also decomposes according to reaction (2), generating the undesired product U. When the reactions take place in a constant density solution, the heat of reaction is so small that the reactions are effectively isothermal. Reaction (1) is first order in each of the two reactants, second order overall. Reaction (2) is second order in B. At the process temperature the rate coefficient for reaction (1) is $8 \times 10^{-2} \text{ L mol}^{-1} \text{ min}^{-1}$ and that for reaction (2) is $4 \times 10^{-2} \text{ L mol}^{-1} \text{ min}^{-1}$. When a CSTR is used, the selectivity is 6.32 mol D per mol U at 90% conversion of B, and when a PFR is used at the same conversion, the selectivity is 3.03 mol D per mol U. Compare these selectivities to the selectivity predicted by the axial dispersion model when the axial dispersion coefficient is equal to $4 \text{ dm}^2 \text{ s}^{-1}$. The feed to the 35 dm long tubular reactor will be taken from a tank that contains a solution of A and B with equal concentrations of 0.75 M and flow with a superficial velocity of 0.1 dm s^{-1} . (The flow rate and reactor length were chosen to give a conversion of ca. 90% to permit an equitable comparison to the PFR and CSTR selectivities.)



Problem Analysis

The problem statement instructs us to use the axial dispersion model to compute the selectivity. It also states that the reactions occur isothermally. To solve the problem, axial dispersion model mole balances will need to be written for at least two of the reagents that participate in the reactions. Those equations can be solved to find the concentrations of D and U at the reactor outlet, allowing the calculation of the selectivity.

Problem Solution

The following quantities are given in the problem statement: $k_1 = 8 \times 10^{-2} \text{ L mol}^{-1} \text{ min}^{-1}$, $k_2 = 4 \times 10^{-2} \text{ L mol}^{-1} \text{ min}^{-1}$, $D_{ax} = 4 \text{ dm}^2 \text{ s}^{-1}$, $L = 35 \text{ dm}$, $C_{A,feed} = 0.75 \text{ mol L}^{-1}$, $C_{B,feed} = 0.75 \text{ mol L}^{-1}$, $u_s = 0.1 \text{ dm s}^{-1}$. The problem states that the density of the solution is constant, so the volumetric flow rate and the superficial velocity are also constant.

The general form of the axial dispersion model mole balance is given in equation (1). In the present problem, there are two reactions taking place, so the summation expands to two term. In addition, the superficial velocity is a constant and can be taken outside of the derivative. Mole balances of the form

given in equation (1) can be written for each of the reagents, A, B, D and U, leading to equations (2) through (5).

$$-D_{ax} \frac{d^2 C_i}{dz^2} + \frac{d}{dz}(u_s C_i) = \sum_{\substack{j=all \\ reactions}} v_{i,j} r_j \quad (1)$$

$$-D_{ax} \frac{d^2 C_A}{dz^2} + u_s \frac{dC_A}{dz} = -r_1 \quad (2)$$

$$-D_{ax} \frac{d^2 C_B}{dz^2} + u_s \frac{dC_B}{dz} = -r_1 - 2r_2 \quad (3)$$

$$-D_{ax} \frac{d^2 C_D}{dz^2} + u_s \frac{dC_D}{dz} = r_1 \quad (4)$$

$$-D_{ax} \frac{d^2 C_U}{dz^2} + u_s \frac{dC_U}{dz} = r_2 \quad (5)$$

We will use the Danckwerts boundary conditions to solve the mole balances. The general form of the Danckwerts boundary conditions are given in equations (6) and (7). Those equations can be used to write the boundary conditions for A, B, D and U, as given in equations (8) through (15), where the equations were also re-arranged so there is a zero on the right of the equals sign.

$$\text{at } z = 0; \quad u_s C_i(z=0) - D_{ax} \left. \frac{dC_i}{dz} \right|_{z=0} = u_s C_{i,feed} \quad (6)$$

$$\text{at } z = L; \quad \left. \frac{dC_i}{dz} \right|_{z=L} = 0 \quad (7)$$

$$u_s C_A(0) - D_{ax} \left. \frac{dC_A}{dz} \right|_{z=0} - u_s C_{A,feed} = 0 \quad (8)$$

$$\left. \frac{dC_A}{dz} \right|_{z=L} = 0 \quad (9)$$

$$u_s C_B(0) - D_{ax} \left. \frac{dC_B}{dz} \right|_{z=0} - u_s C_{B,feed} = 0 \quad (10)$$

$$\left. \frac{dC_B}{dz} \right|_{z=L} = 0 \quad (11)$$

$$u_s C_D(0) - D_{ax} \left. \frac{dC_D}{dz} \right|_{z=0} - u_s C_{D,feed} = 0 \quad (12)$$

$$\left. \frac{dC_D}{dz} \right|_{z=L} = 0 \quad (13)$$

$$u_s C_U(0) - D_{ax} \left. \frac{dC_U}{dz} \right|_{z=0} - u_s C_{U,feed} = 0 \quad (14)$$

$$\left. \frac{dC_U}{dz} \right|_{z=L} = 0 \quad (15)$$

Some of the boundary conditions are specified at $z = 0$ while other are specified at $z = L$, making the set of ordinary differential equation (ODEs) to be solved mixed boundary value ODEs. We will solve these equations numerically (see Supplemental Unit S6). No matter what software is used, solving equations (2) through (5) numerically will require us to provide code that is given values for the dependent variable and its derivatives as a function of the independent variable and that uses those values to evaluate the ODEs and the boundary conditions.

In the present case, the four quantities appearing in the ODEs other than the derivatives, dependent variables and independent variable are the axial dispersion coefficient, the superficial velocity and the rates of the two reactions. The first two are constants that were given in the problem statement, as listed above. The problem specifies the reaction orders, and the rate coefficients are given (they are constant since the reaction occurs isothermally). Thus, the rates can be calculated using equations (16) and (17). All of the quantities appearing in the boundary conditions are known constants whose values are given in the problem statement.

$$r_1 = k_1 C_A C_B \quad (16)$$

$$r_2 = k_2 C_B^2 \quad (17)$$

Thus, at this point the code needed to solve the the axial dispersion model mole balances numerically can be written and the equations can be solved. Doing so yields the concentrations of A, B, D and U as a function of axial position within the reactor. The resulting values of the concentrations of D and U at the reactor outlet can be used to compute the selectivity according to equation (18). Doing so, one finds the selectivity to equal 3.4 mol D per mol U. Recall that at the same conversion the selectivity in a PFR is 3.03 and when a CSTR is used it equals 6.32. As expected, the axial dispersion model yields a selectivity intermediate between a PFR and a CSTR because the amount of back-mixing in the axial dispersion model is intermediate between the zero back-mixing of the PFR and the complete back-mixing of the CSTR.

$$S_{D/U} = \frac{C_D(L)}{C_U(L)} \quad (18)$$

Calculation Details Using MATLAB

The axial dispersion mole balance design equation is a mixed boundary ODE. The solution of mixed boundary ODEs is described in Supplemental Unit S6, which also provides template files for solving them. There are two template files provided. One is used if the ODE contains the reciprocal of the independent variable which causes a singularity at the point where that variable equals zero. Examining equations (2) through (5), it can be seen that the reciprocal of z is not present. Therefore the code presented here was generated by modifying the MATLAB template file SolvBVDif.m, a copy of which was saved as Example_33_2_ad.m and modified to solve this problem. The modified file accompanies this solution.

In order to use the template file, the second-order ODEs of equations (2) through (5) must be reduced to a set of first order ODEs, and the boundary conditions must be re-written in terms of the new dependent variables. This is a straightforward process and is explained in Supplemental Unit S6. We begin by defining C_A' as in equation (19). With that definition, the second derivative of C_A with respect to z will equal the first derivative of C_A' with respect to z , as shown in equation (20). Substitution of equations (19) and (20) into equation (2) leads to equation (21). Analogous treatment of the mole balances on B, D and U leads to equations (22) through (27).

$$\frac{dC_A}{dz} = C_A' \quad (19)$$

$$\frac{d^2C_A}{dz^2} = \frac{dC_A'}{dz} \quad (20)$$

$$-D_{ax} \frac{dC_A'}{dz} + u_s C_A' = -r_1 \quad \Rightarrow \quad \frac{dC_A'}{dz} = \frac{r_1 + u_s C_A'}{D_{ax}} \quad (21)$$

$$\frac{dC_B}{dz} = C_B' \quad (22)$$

$$\frac{dC_B'}{dz} = \frac{r_1 + 2r_2 + u_s C_B'}{D_{ax}} \quad (23)$$

$$\frac{dC_D}{dz} = C_D' \quad (24)$$

$$\frac{dC_D'}{dz} = \frac{-r_1 + u_s C_D'}{D_{ax}} \quad (25)$$

$$\frac{dC_U}{dz} = C_U' \quad (26)$$

$$\frac{dC_U'}{dz} = \frac{-r_2 + u_s C_U'}{D_{ax}} \quad (27)$$

Thus, the four second order ODEs, equations (2) through (5), are equivalent to the eight first order ODEs, equations (19) and (21) through (27). Similarly substituting equation (19), and its equivalents for B, D and U, into the original boundary conditions, equations (8) through (15), leads to the corresponding boundary conditions in terms of the new dependent variables, equations (28) and (through).

$$u_s C_A(0) - D_{ax} \left. \frac{dC_A}{dz} \right|_{z=0} - u_s C_{A,feed} = 0 \Rightarrow u_s C_A(0) - D_{ax} C'_A(0) - u_s C_{A,feed} = 0 \quad (28)$$

$$\left. \frac{dC_A}{dz} \right|_{z=L} = 0 \Rightarrow C'_A(L) = 0 \quad (29)$$

$$u_s C_B(0) - D_{ax} C'_B(0) - u_s C_{B,feed} = 0 \quad (30)$$

$$C'_B(L) = 0 \quad (31)$$

$$u_s C_D(0) - D_{ax} C'_D(0) - u_s C_{D,feed} = 0 \quad (32)$$

$$C'_D(L) = 0 \quad (33)$$

$$u_s C_U(0) - D_{ax} C'_U(0) - u_s C_{U,feed} = 0 \quad (34)$$

$$C'_U(L) = 0 \quad (35)$$

With the equations in this form, the modification of SolvBVDif.m can be undertaken. To begin, the initial comment was replaced, the function definition was changed so there are no return values and the known constants from the problem statement were entered. These modifications can be seen in Listing 1.

```
% Modified version of the MATLAB template file SolvBVDif.m used to model
% the tubular reactor with axial dispersion in Example 33.2 of "A First
% Course on Kinetics and Reaction Engineering."
%
function Example_33_2_ad
    % Known quantities and constants (in consistent units)
    k1 = 8e-2; % dm3/mol/min
    k2 = 4e-2; % dm3/mol/min
    CAfeed = 0.75; % mol/dm3
    CBfeed = 0.75; % mol/dm3
    CDfeed = 0.0; % mol/dm3
    CUfeed = 0.0; % mol/dm3
    us = 0.1 * 60; % dm/min
    L = 350.0; % dm
    Dax = 4. * 60; % dm2/min
```

Listing 1. Modified initial comment and function statement followed by entry of known constants.

The second modification occurs within an internal function named `bvodes`; this is where the ODEs are evaluated. This internal function is passed values of the independent and dependent variables (as x and y , respectively). I like to extract the dependent variables from the vector y , giving them more meaningful names. In this case I extracted them into variables named CA , CA_prime , etc. After doing that, the rates of the two reactions can be calculated using equations (16) and (17). If the ODEs have been written with one derivative on the left of the equals sign and everything else on the right side of the equals sign, as in equations (19) and (21) through (27), the right hand sides of those ODEs can then be evaluated and returned in the column vector named $dydx$. All this is shown in Listing 2.

```
% Function that evaluates the derivatives
function dydx = bvodes(x,y)
    CA = y(1);
    CA_prime = y(2);
    CB = y(3);
    CB_prime = y(4);
    CD = y(5);
    CD_prime = y(6);
    CU = y(7);
    CU_prime = y(8);
    r1 = k1*CA*CB;
    r2 = k2*CB^2;
    dydx = [
        CA_prime
        (us*CA_prime + r1)/Dax;
        CB_prime
        (us*CB_prime + r1 + 2*r2)/Dax;
        CD_prime
        (us*CD_prime - r1)/Dax;
        CU_prime
        (us*CU_prime - r2)/Dax;
    ];
end % of internal function bvodes
```

Listing 2. Modified version of the internal function bvodes.

The third modification occurs within an internal function named `bvs`; this is where the boundary conditions are evaluated. This internal function is passed the values of the dependent variables (C_A , C_A' , C_B , C_B' , C_D , C_D' , C_U and C_U') at $z = 0$ in a vector named `y_at_start` and it is passed the values of the same dependent variables at $z = L$ in a vector named `y_at_end`. I extracted the dependent variables from the two vectors, giving them more meaningful names. Notice that in equations (28) through (35) the boundary equations have been written with a zero on the right hand side of the equations. The modification of internal function `bvs` involves evaluating the left hand side of each of those equations and returning them in the form of a column vector named `res`. (The variable name “res” stands for “residual,”

because if the boundary condition is satisfied it will equal zero, otherwise it will be the residual, non-zero value.) The modifications to internal function bvs are shown in Listing 3.

```
% Function that calculates the errors at the boundaries
function res = bvs(y_at_start,y_at_end)
    CA_inlet = y_at_start(1);
    CA_prime_inlet = y_at_start(2);
    CB_inlet = y_at_start(3);
    CB_prime_inlet = y_at_start(4);
    CD_inlet = y_at_start(5);
    CD_prime_inlet = y_at_start(6);
    CU_inlet = y_at_start(7);
    CU_prime_inlet = y_at_start(8);
    CA_outlet = y_at_end(1);
    CA_prime_outlet = y_at_end(2);
    CB_outlet = y_at_end(3);
    CB_prime_outlet = y_at_end(4);
    CD_outlet = y_at_end(5);
    CD_prime_outlet = y_at_end(6);
    CU_outlet = y_at_end(7);
    CU_prime_outlet = y_at_end(8);
    res = [
        us*CA_inlet - Dax*CA_prime_inlet - us*CAfeed;
        CA_prime_outlet;
        us*CB_inlet - Dax*CB_prime_inlet - us*CBfeed;
        CB_prime_outlet;
        us*CD_inlet - Dax*CD_prime_inlet - us*CDfeed;
        CD_prime_outlet;
        us*CU_inlet - Dax*CU_prime_inlet - us*CUfeed;
        CU_prime_outlet;
    ];
end % of internal function bvs
```

Listing 3. Modified version of the internal function bvs.

The fourth modification involves creating an initial mesh of points within the reactor where the values of the dependent variables will be calculated. As can be seen in Listing 4, I chose to simply use 100 points between $z = 0$ and $z = L$. The fifth modification involves providing a guess for the value of the dependent variables at those mesh points. You only provide a single guess for each dependent variable, and that same guess is used at all of the mesh points. As can be seen in Listing 4, I used the feed concentration of A as my guess for the each of the concentrations, and I used the rate of the first reaction at the feed concentration divided by the superficial velocity as my guess for the the C_i' dependent variables. (Example 33.1 explains how I came up this this guess, but in the end it is simply a guess; you can use any guess that leads to a converged solution.)

```

% Set up the initial mesh
x_range_low = 0.0;
x_range_high = L;
n_mesh_points = 100.;

% The next line creates an array for the independent variable
x = linspace(x_range_low,x_range_high,n_mesh_points);

% Guesses
yinit = [
    CAfeed
    k1*CAfeed*CBfeed/us
    CAfeed
    k1*CAfeed*CBfeed/us
    CAfeed
    k1*CAfeed*CBfeed/us
    CAfeed
    k1*CAfeed*CBfeed/us
];

```

Listing 4. Modification to set up the initial mesh and provide guesses.

The final modification occurs after the equations have been solved. In this case it involves using the result to calculate the selectivity. I also calculated the conversion of B to make sure it was near 90% so that it could be equitably compared to the PFR and CSTR selectivities given in the problem statement. These modifications can be seen in Listing 5.

```

% Create a structure containing the mesh and guesses
solinit=bvpinit(x,yinit);

% Solve the odes
result = bvp4c(@bvodes,@bvs,solinit);

convB = 100*(CBfeed - result.y(3,end))/CBfeed
sel = result.y(5,end)/result.y(7,end)

end % of file Example_33_2_ad.m

```

Listing 5. Modification to assign values to the return variables.

After saving the modified template file, it can be executed by typing “Example_33_2_ad” at the MATLAB command prompt. Doing so generates the output in Listing 6.


```
>> Example_33_2_ad
```

```
convB =  
    89.8138
```

```
sel =  
    3.4215
```

Listing 6. Output from the execution of Example_33_2_ad.m