# A First Course on Kinetics and Reaction Engineering

**Class 35 on Unit 33**

# Where We're Going

- Part I - Chemical Reactions

- Part II - Chemical Reaction Kinetics

- Part III - Chemical Reaction Engineering

  ‣ A. Ideal Reactors

  ‣ B. Perfectly Mixed Batch Reactors

  ‣ C. Continuous Flow Stirred Tank Reactors

  ‣ D. Plug Flow Reactors

  ‣ E. Matching Reactors to Reactions

- Part IV - Non-Ideal Reactions and Reactors

  ‣ A. Alternatives to the Ideal Reactor Models

    - 33. Axial Dispersion Model

    - 34. 2-D and 3-D Tubular Reactor Models

    - 35. Zoned Reactor Models

    - 36. Segregated Flow Model

    - 37. Overview of Multi-Phase Reactors

  ‣ B. Coupled Chemical and Physical Kinetics

# Axial Dispersion Model

- The axial dispersion model is a modified form of the plug flow reactor model
  - ‣ it adds a diffusion-like term that accounts for mixing in the axial direction

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

  - ‣ the quantity $D_{ax}$ is not a diffusion coefficient
    - it is called the dispersion coefficient
    - it has one value that is used for all species being modeled
    - it can be found
      - from correlations of the axial Peclet number (or its inverse, the dispersion number) as a function of the Reynolds number

$$Pe_{ax} = \frac{ul}{D_{ax}}$$

      - experimentally, for example using stimulus and response measurements involving a tracer (as is done when measuring the age function

- In commercial scale packed bed reactors, including axial dispersion doesn't affect conversion as long as the reactor length is greater than ca. 50 catalyst particle diameters
  - ‣ The axial dispersion model can still be used to empirically model a reactor
  - ‣ When $D_{ax} = 0$, it is a PFR; as $D_{ax}$ becomes very large, it approaches a CSTR

# Boundary Conditions and Numerical Solution

- The axial dispersion model is a second order differential equation
  - There are several ways the boundary conditions could be formulated
  - Danckwerts boundary conditions are very common

$$\text{at } z = 0; \quad uC_A(0) - D_{ax}\left.\frac{dC_A}{dz}\right|_{z=0} = u^0 C_{A,feed}$$

$$\text{at } z = L; \quad \left.\frac{dC_A}{dz}\right|_{z=L} = 0$$

- First Danckwerts boundary condition recognizes that as soon as the feed enters the reactor, it will be diluted by axial mixing
  - Much like what happens to feed entering a CSTR
  - It requires that the flow of A upstream of the reactor entrance (where no mixing has occurred) must be equal to the net flow of A (due to convection and mixing) at the inlet to the reactor
- Second Danckwerts boundary condition simply requires that the concentration stops changing at the point where the flow leaves the reactor.
- The axial dispersion model involves mixed boundary value ordinary differential equations
  - Numerical solution requires different methods than are used for initial value problems
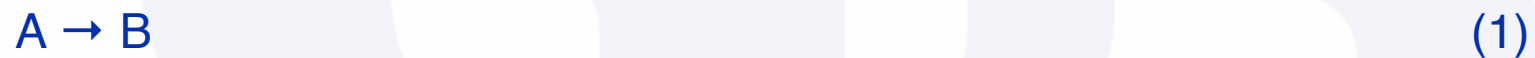  - Discussed in Supplemental Unit S6

Questions?

# Activity 33.1

Suppose liquid phase reaction (1) is second order in the concentration of A. The reaction proceeds isothermally with a rate coefficient equal to 0.05 $dm^3$ $mol^{-1}$ $min^{-1}$. The liquid density is constant. The feed to the process is taken from a tank containing A at a concentration of 1 M and enters the reactor with a superficial velocity of 6.0 $dm$ $min^{-1}$. Plot the concentration of A versus axial position within a tubular reactor for axial dispersion coefficients of 0.1, 1, 10, 100 and 1000 $dm^2$ $min^{-1}$.

$$A \rightarrow B \qquad\qquad\qquad (1)$$

Read through the problem statement. Each time you encounter a quantity, write it down and equate it to the appropriate variable. When you have completed doing so, if there are any additional constant quantities that you know will be needed and that can be calculated from the values you found, write the equations needed for doing so.

Suppose liquid phase reaction (1) is second order in the concentration of A. The reaction proceeds isothermally with a rate coefficient equal to 0.05 dm$^3$ mol$^{-1}$ min$^{-1}$. The liquid density is constant. The feed to the process is taken from a tank containing A at a concentration of 1 M and enters the reactor with a superficial velocity of 6.0 dm min$^{-1}$. Plot the concentration of A versus axial position within a tubular reactor for axial dispersion coefficients of 0.1, 1, 10, 100 and 1000 dm$^2$ min$^{-1}$.

$$A \rightarrow B \hspace{6cm} (1)$$

Read through the problem statement. Each time you encounter a quantity, write it down and equate it to the appropriate variable. When you have completed doing so, if there are any additional constant quantities that you know will be needed and that can be calculated from the values you found, write the equations needed for doing so.

$k = 0.05$ dm$^3$ mol$^{-1}$ min$^{-1}$

$C_{A,feed} = 1$ mol L$^{-1}$

$u_s = 6.0$ dm min$^{-1}$

$D_{ax} = 0.1, 1, 10, 100$ or $1000$ dm$^2$ min$^{-1}$

Suppose liquid phase reaction (1) is second order in the concentration of A. The reaction proceeds isothermally with a rate coefficient equal to 0.05 $dm^3$ $mol^{-1}$ $min^{-1}$. The liquid density is constant. The feed to the process is taken from a tank containing A at a concentration of 1 M and enters the reactor with a superficial velocity of 6.0 dm $min^{-1}$. Plot the concentration of A versus axial position within a tubular reactor for axial dispersion coefficients of 0.1, 1, 10, 100 and 1000 $dm^2$ $min^{-1}$.

$$A \rightarrow B \tag{1}$$

Generate the mole balance design equations needed to model the axially-dispersed tubular reactor by simplification of the general design equations found in Unit 33 or on the AFCoKaRE Exam Handout.

Suppose liquid phase reaction (1) is second order in the concentration of A. The reaction proceeds isothermally with a rate coefficient equal to 0.05 dm$^3$ mol$^{-1}$ min$^{-1}$. The liquid density is constant. The feed to the process is taken from a tank containing A at a concentration of 1 M and enters the reactor with a superficial velocity of 6.0 dm min$^{-1}$. Plot the concentration of A versus axial position within a tubular reactor for axial dispersion coefficients of 0.1, 1, 10, 100 and 1000 dm$^2$ min$^{-1}$.

$$A \rightarrow B \qquad\qquad\qquad (1)$$

Generate the mole balance design equations needed to model the axially-dispersed tubular reactor by simplification of the general design equations found in Unit 33 or on the AFCoKaRE Exam Handout.

$$-D_{ax}\frac{d^2C_i}{dz^2}+\frac{d}{dz}(u_sC_i)=\sum_{\substack{j=all \\ reactions}}\nu_{i,j}r_j \qquad\qquad -D_{ax}\frac{d^2C_A}{dz^2}+u_s\frac{dC_A}{dz}=-r$$

$$\text{at } z=0;\ u_sC_i(z=0)-D_{ax}\left.\frac{dC_i}{dz}\right|_{z=0}=u_sC_{i,feed} \qquad u_sC_A(0)-D_{ax}\left.\frac{dC_A}{dz}\right|_{z=0}-u_sC_{A,feed}=0$$

$$\text{at } z=L;\ \left.\frac{dC_i}{dz}\right|_{z=L}=0 \qquad\qquad \left.\frac{dC_i}{dz}\right|_{z=L}=0$$

$$-D_{ax}\frac{d^2 C_A}{dz^2} + u_s\frac{dC_A}{dz} = -r \qquad u_s C_A(0) - D_{ax}\frac{dC_A}{dz}\bigg|_{z=0} - u_s C_{A,feed} = 0 \qquad \frac{dC_i}{dz}\bigg|_{z=L} = 0$$

- Identify the independent and dependent variables, if appropriate, and the unknown quantities to be found by solving the equations

$$-D_{ax}\frac{d^2C_A}{dz^2}+u_s\frac{dC_A}{dz}=-r \qquad u_sC_A(0)-D_{ax}\frac{dC_A}{dz}\bigg|_{z=0}-u_sC_{A,feed}=0 \qquad \frac{dC_i}{dz}\bigg|_{z=L}=0$$

- Identify the independent and dependent variables, if appropriate, and the unknown quantities to be found by solving the equations
  - ‣ Independent variable: $z$
  - ‣ Dependent variable: $C_A$
  - ‣ Solution will give $C_A(z)$ between $z = 0$ and $z = L$
- Assuming that the equations will be solved numerically, specify the information that must be provided and show how to calculate any unknown values

$$-D_{ax}\frac{d^2C_A}{dz^2}+u_s\frac{dC_A}{dz}=-r \qquad u_sC_A(0)-D_{ax}\frac{dC_A}{dz}\bigg|_{z=0}-u_sC_{A,feed}=0 \qquad \frac{dC_i}{dz}\bigg|_{z=L}=0$$

- Identify the independent and dependent variables, if appropriate, and the unknown quantities to be found by solving the equations
  - ‣ Independent variable: $z$
  - ‣ Dependent variable: $C_A$
  - ‣ Solution will give $C_A(z)$ between $z = 0$ and $z = L$
- Assuming that the equations will be solved numerically, specify the information that must be provided and show how to calculate any unknown values
  - ‣ Must supply code that is given $C_A$ and its derivatives as a function of $z$ and uses them to evaluate the ODEs
    - $D_{ax}$ and $u_s$ are known constants
    - $r = k(C_A)^2$
      - $k$ is a known constant
  - ‣ Must supply code that evaluates the boundary conditions given values of $C_A$, $z$ and the derivatives at $z = 0$ and $z = L$
    - $D_{ax}$, $C_{A,feed}$, $u_s$ and $L$ are known constants
- Identify what variables will become known upon solving the design equations and show how those variables can be used to answer the questions that were asked in the problem

$$-D_{ax}\frac{d^2C_A}{dz^2}+u_s\frac{dC_A}{dz}=-r \qquad u_sC_A(0)-D_{ax}\frac{dC_A}{dz}\bigg|_{z=0}-u_sC_{A,feed}=0 \qquad \frac{dC_i}{dz}\bigg|_{z=L}=0$$

- Identify the independent and dependent variables, if appropriate, and the unknown quantities to be found by solving the equations
  - ‣ Independent variable: $z$
  - ‣ Dependent variable: $C_A$
  - ‣ Solution will give $C_A(z)$ between $z = 0$ and $z = L$
- Assuming that the equations will be solved numerically, specify the information that must be provided and show how to calculate any unknown values
  - ‣ Must supply code that is given $C_A$ and $z$ and uses them to evaluate all other quantities in the ODE except derivatives
    - $D_{ax}$ and $u_s$ are known constants
    - $r = k(C_A)^2$
      - $k$ is a known constant
  - ‣ Must supply code that evaluates the boundary conditions given values of $C_A$, $z$ and the derivatives at $z = 0$ and $z = L$
    - $D_{ax}$, $C_{A,feed}$, and $u_s$ are known constants
- Identify what variables will become known upon solving the design equations and show how those variables can be used to answer the questions that were asked in the problem
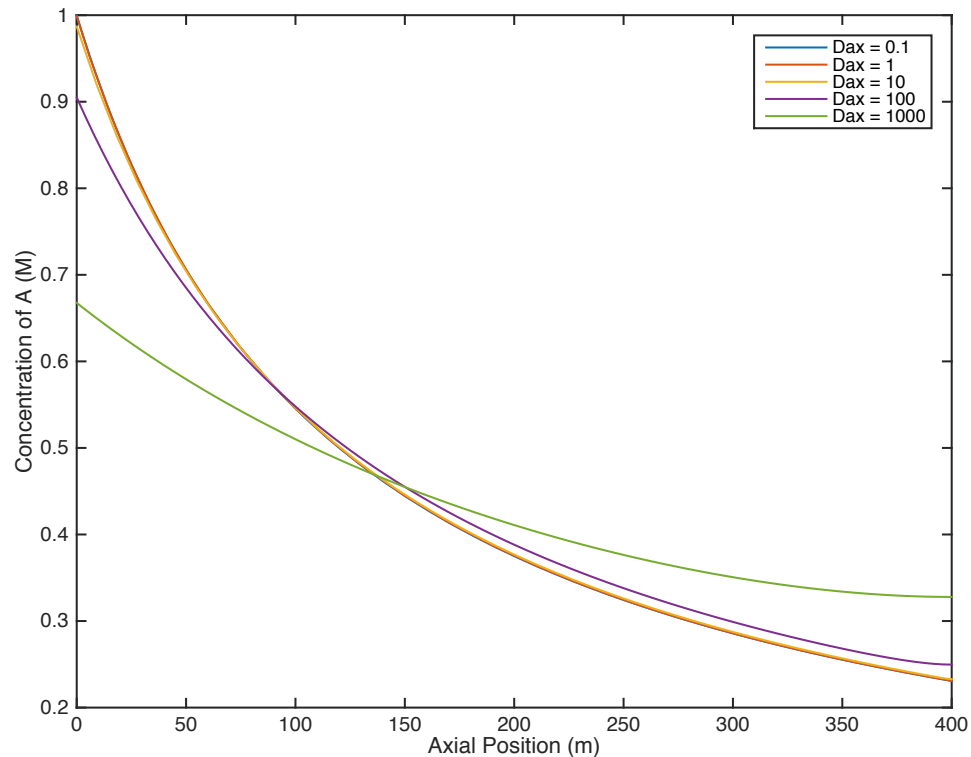  - ‣ $C_A(z)$ between $z = 0$ and $z = L$ will be known, can use it to make the requested plot

Plot legend:
- Dax = 0.1
- Dax = 1
- Dax = 10
- Dax = 100
- Dax = 1000

y-axis: Concentration of A (M)
x-axis: Axial Position (m)

- The results are shown in this plot
- If $D_{ax} = 0$, the reactor is a PFR
  - What would the plot look like?
- If $D_{ax} = \infty$, the reactor is a CSTR
  - What would the plot look like?
- Do the results at intermediate $D_{ax}$ make sense physically?

# Numerical Solution of Boundary Value ODEs

- Identification of boundary value ordinary differential equations (ODEs)
  - No partial derivatives
  - All ordinary derivatives with respect to the same independent variable
  - Independent variable ranges from some known lower limit to a different upper limit
    - some boundary conditions apply at the lower limit and other boundary conditions apply at the upper limit

- Requirements imposed by the MATLAB solver, bvp4c
  - Higher order ODEs reduced to sets of first order ODEs
  - Each ODE written in the form $\dfrac{dy_i}{dx} = \sum_{\text{all } j} S_{ij} \dfrac{y_j}{x} + f_i\left(x, \underline{y}\right)$
    - $S_{ij}$ equal zero or constant (here, $S_{ij}$ is equal to zero)

- To convert any second order ODE into a set of two first order ODEs
  - Define a new variable that is equal to the first derivative of the original dependent variable with respect to the independent variable
    - This definition becomes one of the two first order ODEs in the set
  - In the original second order ODE
    - Replace any occurrence of the second derivative of the original dependent variable with the first derivative of the new variable
    - Replace any occurrence of the first derivative of the original dependent variable with the new variable
  - In the boundary conditions, replace the first derivative of the original dependent variable with the new variable

# Continuing the Present Problem

$$-D_{ax}\frac{d^2C_A}{dz^2}+u_s\frac{dC_A}{dz}=-r$$

- To solve numerically we need to convert the second order ODE to two first order ODEs

- Let $Y=\dfrac{dC_A}{dz}$

  ‣ This becomes one of the two first order ODEs

  ‣ Note that $\dfrac{dY}{dz}=\dfrac{d^2C_A}{dz^2}$

- Substitute $Y$ and $\dfrac{dY}{dz}$ for $\dfrac{d^2C_A}{dz^2}$ and $\dfrac{dC_A}{dz}$ in the original ODE

  ‣ $-D_{ax}\dfrac{dY}{dz}+u_sY=-r$

  ‣ This becomes the second of the two first order ODEs

- Also re-write the boundary conditions replacing $\dfrac{dC_A}{dz}$ with $Y$

$$u_sC_A(0)-D_{ax}\frac{dC_A}{dz}\bigg|_{z=0}-u_sC_{A,feed}=0 \quad \Rightarrow \quad u_sC_A(0)-D_{ax}Y(0)-u_sC_{A,feed}=0$$

$$\frac{dC_i}{dz}\bigg|_{z=L}=0 \quad \Rightarrow \quad Y(L)=0$$

# Choosing the MATLAB Template File

- The equations are boundary value ODEs; they do not contain a singularity at x = 0
    - The MATLAB template file SolvBVDif.m can be used to solve them
    - Follow the step by step instructions provided with this Supplemental Unit S6
- Save a copy of SolvBVDif.m as S6_Example_1.m
    - Change the initial comment to reflect the purpose of the modified file
    - Change the function declaration to match the filename, but without the ".m"

        -

```
% Modified version of the MATLAB template file SolvBVDif.m that was used to
% solve the in-class problem in Class 35 of CE 329, Fall 2014.
%
function result = S6_Example_1
    % Known quantities and constants (in consistent units)
    Dax = 0.1;
    us = 0.6.0;
    k = 0.05;
    CA0 = 1.0;
    L = 400; % May need to change this; just a guess
```

# Second Required Modification

```matlab
    % Function that evaluates the derivatives
    function dydx = bvodes(x,y)
% EDIT HERE (Required modification 2 of 6):
        dydx = [
            % Evaluate dy1/dx here
            % Evaluate dy2/dx here
            % and so on, one dyi/dx per line
        ];
    end % of internal function bvodes
```

$$\frac{dC_A}{dz} = Y$$

$$\frac{dY}{dz} = \frac{1}{D_{ax}}(r - u_s Y)$$

```matlab
    % Function that evaluates the derivatives
    function dydx = bvodes(x,y)
        dydx = [
            y(2)
            (1/Dax)*(k*y(1)^2 - us*y(2))
        ];
    end % of internal function bvodes
```

# Third Required Modification

```
    % Function that calculates the errors at the boundaries
    function res = bvs(y_at_start,y_at_end)
        % y_at_start is a column vector containing y values at the starting
        % x boundary and y_at_end is a column vector containing y values at
        % the ending x boundary.
% EDIT HERE (Required modification 3 of 6):
        res = [
            % Calculate the error in the first boundary condition here
            % Calculate the error in second boundary condition here
            % and so on, one boundary condition per line
        ];
```

$$u_s C_A(0) - D_{ax} Y(0) - u_s C_{A,feed} = 0$$

$$Y(L) = 0$$

```
    % Function that calculates the errors at the boundaries
    function res = bvs(y_at_start,y_at_end)
        % y_at_start is a column vector containing y values at the starting
        % x boundary and y_at_end is a column vector containing y values at
        % the ending x boundary.
        res = [
            us*y_at_start(1)-Dax*y_at_start(2)-us*CA0
            y_at_end(2)
        ];
    end % of internal function bvs
```

# Fourth Required Modification

```
% EDIT HERE (Required modification 4 of 6):
    % Set up the initial mesh
    x_range_low = % insert starting value of the valid range of x here
    x_range_high = % insert ending value of the valid range of x here
    n_mesh_points = % insert the number of mesh points here
```

$$0 \leq x \leq L$$

```
    % Set up the initial mesh
    x_range_low = 0.0;
    x_range_high = L;
    n_mesh_points = 40;
```

# Fifth Required Modification

```
    % Guesses
% EDIT HERE (Required modification 5 of 6):
    yinit = [
        % Enter a guess for y1 (one value for the whole x range) here
        % Enter a guess for y2 (one value for the whole x range) here
        % and so on, one guess per line
    ];
```

```
    % Guesses
    yinit = [
        CA0
        -CA0/L
    ];
```

# Sixth and Final Required Modification

```
% EDIT HERE (Required modification 6 of 6):
    % Calculate any other desired quantities from the results, noting
    % result.x(i) contains the final x value of mesh point i
    % result.y(i,j) contains the final value of yj at mesh point i
```

- Here, just use results to plot $C_A$ vs z

# Where We're Going

- Part I - Chemical Reactions

- Part II - Chemical Reaction Kinetics

- **Part III - Chemical Reaction Engineering**

  ‣ A. Ideal Reactors

  ‣ B. Perfectly Mixed Batch Reactors

  ‣ C. Continuous Flow Stirred Tank Reactors

  ‣ D. Plug Flow Reactors

  ‣ E. Matching Reactors to Reactions

- **Part IV - Non-Ideal Reactions and Reactors**

  ‣ **A. Alternatives to the Ideal Reactor Models**

    - 33. Axial Dispersion Model

    - 34. 2-D and 3-D Tubular Reactor Models

    - 35. Zoned Reactor Models

    - 36. Segregated Flow Model

    - 37. Overview of Multi-Phase Reactors

  ‣ **B. Coupled Chemical and Physical Kinetics**