Packed Bed Catalysis Model Solution Exercise

This exercise uses the finite difference grid and weight Matlab function developed in the previous Matlab exercises to solve a mathematical model for a packed bed catalytic reactor developed in class.

Task 1: Scale the equations to dimensionless form

From the lectures, we have the following system of equations to describe the mole fraction distributions of reactant A and product B in a packed bed:

$$-cD_L\frac{\partial^2 x_A}{\partial z^2} + cV_{z,A}\frac{\partial x_A}{\partial z} = -\frac{1-\varepsilon}{\varepsilon}\frac{3}{R_p}\left[\frac{2cD_{AB}}{\delta}\ln\left(\frac{1}{1-\frac{1}{2}x_A}\right)\right]$$

$$-cD_L\frac{\partial^2 x_B}{\partial z^2} + cV_{z,A}\frac{\partial x_B}{\partial z} = +\frac{1-\varepsilon}{\varepsilon}\frac{3}{R_p}\left[\frac{cD_{AB}}{\delta}\ln\left(\frac{1}{1-\frac{1}{2}x_A}\right)\right]$$

$$\text{at } z=0, \quad x_A = 1,\ x_B = 0$$

$$\text{at } z=L, \quad \frac{\partial x_A}{\partial z} = 0,\ \frac{\partial x_B}{\partial z} = 0$$

Task 1a: Define a characteristic velocity and length scale for the system.

Characteristic length, can select L and R of the PBR, or the radius of the particle R_p

Characteristic velocity, can select multiple, f.ex. $V_{Z,A}$, diffusion velocity, reaction rate (instant) and more

As the system curves varies much with the z-direction / length of the PBR
We select L as the characteristic length

Assuming the system is convection dominated system, then a good choice for characteristic velocity is $V_{Z,A}$

Task 1b: Define which dimensionless variables should be used.

We can easily create a dimensionless variable in the length direction:

$$\hat{z} = \frac{z}{L} \quad \Rightarrow \quad dz = L \cdot d\hat{z}$$

Task 1c: Write the system equations and boundary conditions in dimensionless form

Inserting the result from 1b, the equations become:

$$-\frac{cD_L}{L^2}\frac{\partial x_A^2}{\partial \hat{z}^2}+\frac{cV_{z,A}}{L}\cdot\frac{\partial x_A}{\partial \hat{z}}=-\frac{1-\varepsilon}{\varepsilon}\cdot\frac{3}{R_p}\left[\frac{2cD_{AB}}{\delta}\cdot \ln\left(\frac{1}{1-\frac{1}{2}x_A}\right)\right]$$

$$-\frac{cD_L}{L^2}\frac{\partial x_B^2}{\partial \hat{z}^2}+\frac{cV_{z,A}}{L}\cdot\frac{\partial x_B}{\partial \hat{z}}=-\frac{1-\varepsilon}{\varepsilon}\cdot\frac{3}{R_p}\left[\frac{cD_{AB}}{\delta}\cdot \ln\left(\frac{1}{1-\frac{1}{2}x_A}\right)\right]$$

We want the highest order derivative to not have a constant in front => Divide by $-\frac{c\cdot D_L}{L^2}$

$$\boxed{\begin{aligned}\frac{\partial x_A^2}{\partial \hat{z}^2}-\frac{L\cdot V_{z,A}}{D_L}\cdot\frac{\partial x_A}{\partial \hat{z}}&=\frac{6L^2D_{AB}}{R_p\delta\cdot D_L}\cdot\frac{1-\varepsilon}{\varepsilon}\cdot\ln\left(\frac{1}{1-\frac{1}{2}x_A}\right)\\ \frac{\partial x_B^2}{\partial \hat{z}^2}-\frac{L\cdot V_{z,A}}{D_L}\cdot\frac{\partial x_B}{\partial \hat{z}}&=\frac{3L^2D_{AB}}{R_p\delta\cdot D_L}\cdot\frac{1-\varepsilon}{\varepsilon}\cdot\ln\left(\frac{1}{1-\frac{1}{2}x_A}\right)\end{aligned}}$$

The boundary conditions become:

at $z=0$, $\hat{z}=\frac{z}{L}=\frac{0}{L}=0 \quad \Rightarrow x_A=1,\ x_B=0$

at $z=L$, $\hat{z}=\frac{z}{L}=\frac{L}{L}=1 \Rightarrow \frac{dx_A}{dz}=\frac{1}{L}\cdot\frac{dx_A}{d\hat{z}}=0 \Rightarrow \frac{dx_A}{d\hat{z}}=0$

Similarily, $\frac{dx_B}{dz}=0 \Rightarrow \frac{dx_B}{d\hat{z}}=0$

Then:

at $\hat{z}=0$, $x_A=1$, $x_B=0$

at $\hat{z}=1$, $\frac{dx_A}{d\hat{z}}=\frac{dx_B}{d\hat{z}}=0$

Task 1d: Identify the principle dimensionless coefficients in the dimensionless model. How many are there? What do they mean? Based on the system of equations defined above, do you need to solve both equations?

The coefficients are: $\alpha = \frac{L \cdot V_{z,A}}{D_L}$, $\beta = \frac{6L^2 D_{AB}}{R_p \delta \cdot D_L} \cdot \frac{1-\varepsilon}{\varepsilon}$

$\tau_F \propto \frac{R_p \delta}{D_{AB}}$

- There are 2 dimensionless coefficients
- Meaning:

α: $\frac{D_L}{L}$ can be viewed as a diffusion velocity in the length direction, and $V_{z,A}$ as the "average" velocity of the bulk liquid.

then α can be interpreted as the ratio between diffusion in the z-direction and convection, meaning that for large α, the mass flux through the reactor is convection controlled and for small α, it's diffusion controlled.

β: $\frac{R_p \delta}{D_{AB}} \propto \tau_F$, the recidence time of the film

$\frac{L^2}{D_L} \propto \tau_L$, the recidence time of the reactor

$\frac{1-\varepsilon}{\varepsilon} = \frac{\text{"area" avalible for convection}}{\text{"area" occupied by catalyst particles}} \approx$ Packing parameter

$\beta \propto \frac{\tau_L}{\tau_F}$ ratio of recidence time in the reactor vs. film

=> Large values of β implies that the recidence time of the reactor is large compared to the recidence time in the film. This means that the reaction is limited by the rate of mass transfer in the bulk through the reactor (diffusion through film "appears to be" instant), for low values of β, the reaction is limited by the diffusion through the stagnant films surrounding the catalyst particles.

- As we have a binary system => $X_B = 1 - X_A$, and no source terms inside the reactor except for the reaction, we only need to solve one of them. As the equation for X_A is independent of X_B, we choose that one

Task 2: Write a Matlab script to solve the model

Write a Matlab script that uses the Finite Difference grid and weight function you wrote in the last exercise to provide the discretized coordinate grid and weights as a basis to transform the non-linear differential equations into a set of non-linear algebraic equations. Then use the Matlab function "fsolve" to solve the system. Once you have finished Task 1, ask Brian or the teaching assistants for numerical values for your dimensionless coefficients. These should be the only numerical inputs needed for the model beside the number, N, of grid points which you should set yourself.

The final result should be a plot of x_A and x_B as a function of z.

let $\alpha = \frac{L \cdot V_{z,A}}{D_L}$, $\beta = \frac{6L^2 D_{AB}}{R_p \delta \cdot D_L} \cdot \frac{1-\varepsilon}{\varepsilon}$

and $f(X_A) = \ln\left(\frac{1}{1-\frac{1}{2}X_A}\right)$

Then we can rewrite the equation for X_A:

$$\frac{\partial^2 X_A}{\partial \hat{z}} - \alpha \frac{\partial X_A}{\partial \hat{z}} = f(X_A)$$

This must be discretized to solve numerically:

$$A \cdot \vec{X_A} - \beta \cdot B \cdot \vec{X_A} = \alpha \cdot f(\vec{X_A})$$

Where A is the matrix containing the discretization rule for 2nd order derivative, and B is the matix containing the discretization rule for the 1st order derivative in exercise 5.

fsolve needs the answer to the equation to be zero:

$$A \cdot \vec{X_A} - \beta \cdot B \cdot \vec{X_A} - \alpha \cdot f(\vec{X_A}) = 0$$

However, we need to take care of the boundaries, Inserting boundary conditions:

$$\hat{z} = 0,\ X_A = 1 \Rightarrow \vec{X_A}(1) = 1 \Rightarrow \vec{X_A}(1) - 1 = 0$$

$$\hat{z} = 1,\ \frac{\partial X_A}{\partial \hat{z}} = 0 \Rightarrow (B \cdot \vec{X_A})(N) = 0$$

The equations using $\alpha = 10$ and $\beta = 1000$, which was the values provided during exercise hour.

Code bits:

Functions utilized:

```
%% Functions used in the problem

% Creating a function for the f(xA) defined on my submission
function toReturn = f(x)
   toReturn = log(1/(1-0.5*x));
end

% Creating the equation system to be solved
function wantToBeZero = sysEquations(xA, dxA_matrix, ...
    d2xA2_matrix, param1, param2, numberOfNodes)

    %Creating empty array to optimize performance
    wantToBeZero = zeros(numberOfNodes, 1);

    % Computing the differentials values beforehand using the grid
    d2xA_dz2 = d2xA2_matrix*xA;
    dxA_dz = dxA_matrix*xA;

    %% Boundary conditions
    % At z=0
    wantToBeZero(1) = xA(1)-1;

    %% General solution for the middle nodes:
    for i = 2:(numberOfNodes-1)
        wantToBeZero(i) = d2xA_dz2(i) - param1*dxA_dz(i) - param2*f(xA(i));
    end
    % At z=1
    wantToBeZero(numberOfNodes) = dxA_dz(numberOfNodes);
end
```

Finitedifferencegrid from ex5:

```
function [x, w, A, B] = FiniteDifferenceGrid (N)

% The distance between nodes will be used multiple times.
dx = 1/(N-1);

% a) Creating equally spaced vector
x = linspace(0,1,N);

%b) Creating the weight array
w = dx * ones(1, N);
% Changing the first and last elements:
w(1) = dx/2;
w(end) = dx/2;

% c) Creating the matrix for the 1st derivative
A = zeros(N,N);
A(1:N+1:end) = -1;
A(N+1:N+1:end) = 1;
% Fixing the end node:
A(end, end) = 1;
A(end, end-1) = -1;
% Multiplying by 1/dx:
A = 1/dx * A;


% d) Creating the matrix for the 2nd derivative
B = zeros(N,N);
B(1:N+1:end) = -2;
B(2:N+1:end) = 1;
B(N+1:N+1:end) = 1;
% Fix the boundaries
B(1,1) = 1;
B(1,2) = -2;
B(1,3) = 1;
B(end, end) = 1;
B(end, end-1) = -2;
B(end, end-2) = 1;
% Multiply with 1/dx^2
B = 1/dx^2 * B;
```

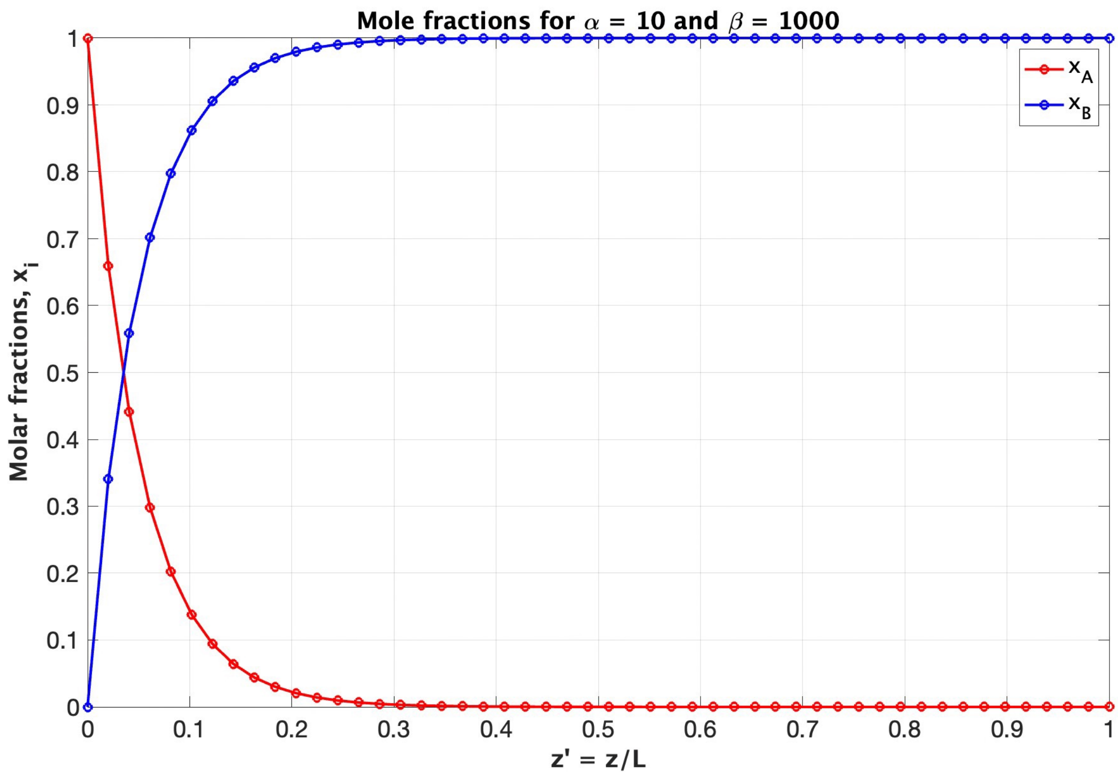
Solving the equations:

```
% Task 2

% Number of nodes in the grid
N = 50;


% Collecting the grid used in th previous exercise (the nodes will be along
% the z-axis)
[z_hat, w, A, B] = FiniteDifferenceGrid (N);

% The dimensionless parameters
alpha = 10;
beta = 1000;

% Need initial guesses for fsolve, choosing values in the middle of range:
xA0 = 0.5*ones(N,1);

% Calling fsolve to find xA:
xA = fsolve(@(xA) sysEquations(xA, A, B, alpha, beta, N), xA0);

% Calculating xB from the fact that we have a binary mixture
xB = 1-xA;
```

Plotting the result:

```
%% Plotting the result

% Creating a title-string:
titletext = sprintf('Mole fractions for \alpha = %d and \beta = %d\n', alpha, beta);
% Properties of the plot
figProps = struct('Color', [1 1 1], 'OuterPosition', [170, 170, 1000, 700]);
fontProps = struct('FontName', 'Calibri', 'FontSize', 18, 'FontWeight', 'bold');
% Initializing the plot
fig = figure(1);
ax = axes;
% Plotting the results
plot(z_hat, xA, 'Color', 'r', 'LineWidth', 2, 'Marker', 'o', 'MarkerEdgeColor', 'r', 'MarkerFaceColor', 'none', 'DisplayName', 'x_A');
hold on;
plot(z_hat, xB, 'Color', 'b', 'LineWidth', 2, 'Marker', 'o', 'MarkerEdgeColor', 'b', 'MarkerFaceColor', 'none', 'DisplayName', 'x_B');
% Tidying up
set(fig, figProps);
set(ax, 'FontSize', fontProps.FontSize);
set(xlabel("z' = z/L", 'interpreter', 'tex'), fontProps);
set(ylabel('Molar fractions, x_i'), fontProps);
set(title(['Mole fractions for \alpha = ', num2str(alpha), ' and \beta = ', num2str(beta)],'interpreter', 'tex'), fontProps);
xlim([0 1]);
box("on");
grid("on");
set(legend(), 'FontName', fontProps.FontName)
%Saving the figure
saveas(fig, 'Ex6_2_plot', 'jpg')
```

The resulting plot:



Decreasing N seems to yield close to the same results, however N must be at least 40 for the curve to look smooth. However, in task 3, when playing with values, I discovered that N must be a lot higher when increasing α. F.ex ≈ 300 for $\alpha = 500$, if not X_A will initially increase to be larger than 1.

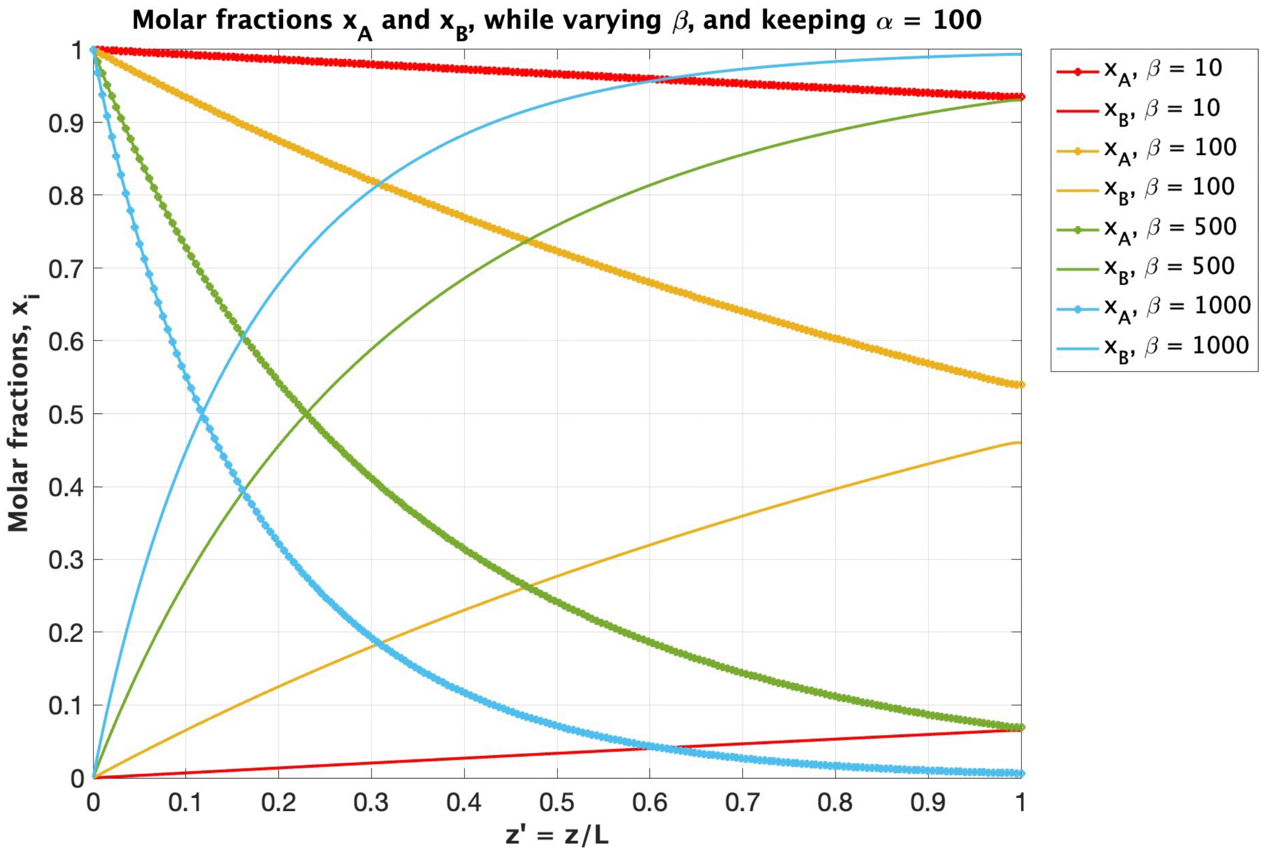
The resulting plot makes sence, as at the inlet, where $z' = 0$, $X_A = 1$, and at the outlet, $z' = 1$, X_A has it's lowest value (~ 0)

Task 3: Evaluate the parametric behavior of the model

Choose 4 potential values for each of your dimensionless equation coefficients of your model. For each dimensionless equation coefficient, plot the axial (z-direction) profiles of x_A and x_B on the same plot (this means there will be 8 lines on the same plot). Use different line types, markers, and/or colors to distinguish the different curves. You should produce 1 plot for each dimensionless coefficient you have identified in Task 1.

Analyze your plots, try to relate the behavior you see in the profiles to the mathematical form of the equations and the parameters that constitute the dimensionless equation coefficients.

Keeping α constant, while varying β:
All parameters are visible on the plot



As explained in problem 1d, the β-parameter is analogous to the ratio between the recidence time of the bulk in the reactor, to the recidence time of the film.

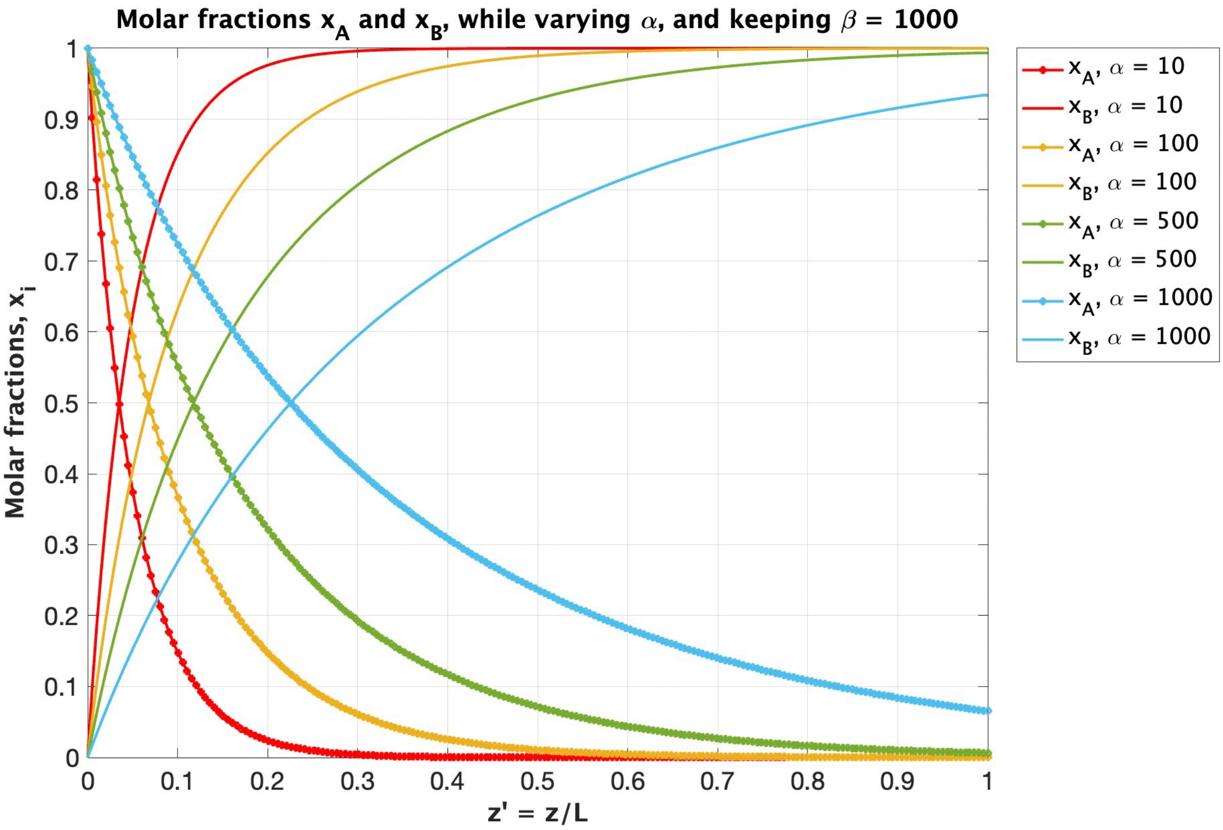
This "definition" of β agrees with the plot of variying β.

For small values of β, the concentration profiles appears to be linear, and they do not look like they will reach a steady state either as $z' \to 1$

Increasing β increases the conversion of A, and the system appears to reach a "steady state" as z increases.

This is because for small β, the reaction is limited by the diffusion as it takes a long time for A to diffuse from the bulk to the catalyst particle. The conversion of A is therefore slow, and a lot of A will not react while passing through the reactor

Increasing β gives the effect of a system that becomes gradually more convection controlled, meaning that the diffusion through the stagnant particle film appears to be almost instant compared to the convection for large enough values of β.
This, in turn increases the conversion of A at any point of the reactor, which is why the slopes for the curves of X_A and X_B increases with increasing β.



As stated in problem 1d), α can be intepreted as the ratio between the bulk velocity and the diffusion velocity through the reactor

For small values of α, the mass transfer of A and B is diffusion controlled. Meaning that the flow rate is small. This gives a large recidence time in the reactor, allowing for more for A to diffuse to the catalyst surface, which increases the conversion. This is why the x_A curve quickly reaches 0 for small values of α.

Increasing α will in turn decrease the recidence time of the reactor, decreasing the reaction rate. As shown in the plot, this allows for A to penetrate further into the reactor, while also decreasing the total conversion through the reactor (except for $\alpha=10 \rightarrow \alpha=100$, where $x_A=0$ at $z'=1$ in both cases)