function [x, phi] = convection\_diffusion\_fem(NP, u, k, L, alpha, Pe\_values)

% Inputs:

% NP - Number of points

% u - Velocity (will be adjusted according to Pe)

% k - Diffusion coefficient (base value, will be adjusted according to Pe)

% L - Total length of the domain

% alpha - Upwinding factor (Petrov-Galerkin weighting)

% Pe\_values - Array of Peclet numbers to simulate

% Initialize figure for plotting

figure;

hold on;

for Pe = Pe\_values

% Adjust the velocity u or diffusion coefficient k to get the desired Peclet number

if Pe == 0

current\_u = 0; % Velocity is set to zero for Pe = 0

else

current\_u = u; % Maintain the specified velocity for other Pe numbers

k = current\_u \* L / Pe; % Adjust diffusion coefficient based on Pe

end

% Preprocessing

NE = NP - 1; % Number of elements

h = L / NE; % Length of each element

x = linspace(0, L, NP); % Mesh points

% Initialize global stiffness matrix and force vector

K\_global = zeros(NP, NP);

F\_global = zeros(NP, 1);

% Connectivity matrix

intma = zeros(NE, 2);

for i = 1:NE

intma(i, :) = [i, i+1];

end

% Assembly of element matrices

for i = 1:NE

% Local-to-global mapping

node1 = intma(i, 1);

node2 = intma(i, 2);

% Element convection and diffusion matrix

emat\_convection = (current\_u/2) \* [-1 1; -1 1];

emat\_diffusion = (k/h) \* [1 -1; -1 1];

emat\_pg = (alpha \* current\_u/2) \* [-1 1; -1 1]; % Petrov-Galerkin part

% Total element matrix

emat = emat\_convection + emat\_diffusion + emat\_pg;

% Assemble into global matrix

K\_global([node1, node2], [node1, node2]) = ...

K\_global([node1, node2], [node1, node2]) + emat;

end

% Apply boundary conditions

K\_global(1, :) = 0; K\_global(1, 1) = 1; F\_global(1) = 1; % Dirichlet BC at x=0

K\_global(NP, :) = 0; K\_global(NP, NP) = 1; F\_global(NP) = 0; % Dirichlet BC at x=L

% Solve the system of equations

phi = K\_global \ F\_global;

% Plotting

plot(x, phi, 'DisplayName', ['Pe = ' num2str(Pe)]);

end

% Finalize the plot

title('Solution of Convection-Diffusion Equation for Different Peclet Numbers');

xlabel('Domain (x)');

ylabel('Solution (\phi)');

legend show;

hold off;

end

% Specify the Peclet numbers to simulate

Pe\_values = [0, 10, 100, 1000, 10000];

% Call the function with desired parameters and Peclet numbers

[x, phi] = convection\_diffusion\_fem(10, 1, 0.1, 1, 0.5, Pe\_values);