function [x, y, phi, source] = initialize\_grid(n)

Lx = 1.0;

Ly = 1.0;

x = linspace(0, Lx, n);

y = linspace(0, Ly, n);

phi = zeros(n, n);

source = zeros(n, n);

grid\_sizes = [41, 81, 161];

tol = 1e-6; % Convergence tolerance

max\_iter = 10000; % Maximum iterations

% Calculate source term

for i = 1:n

for j = 1:n

dist\_squared = (1 - x(i))^2 + y(j)^2;

source(i, j) = 50000 \* exp(-50 \* dist\_squared) \* (100 \* dist\_squared - 2);

end

end

end

% Function to apply boundary conditions

function phi = apply\_boundary\_conditions(phi, x, y, n)

% Bottom boundary: phi(x, 0)

for i = 1:n

phi(i, 1) = 100 \* x(i) + 500 \* exp(-50 \* (1 - x(i))^2);

end

% Top boundary: phi(x, 1)

for i = 1:n

phi(i, n) = 500 \* exp(-50 \* ((1 - x(i))^2 + 1));

end

% Left boundary: phi(0, y)

for j = 1:n

phi(1, j) = 500 \* exp(-50 \* (1 + y(j)^2));

end

% Right boundary: phi(1, y)

for j = 1:n

phi(n, j) = 100 \* (1 - y(j)) + 500 \* exp(-50 \* y(j)^2);

end

end

% Iterate through each grid size

for g = 1:length(grid\_sizes)

n = grid\_sizes(g); % Number of grid points in each direction

[x, y, phi, source] = initialize\_grid(n);

dx = x(2) - x(1);

dy = y(2) - y(1);

% Apply boundary conditions

phi = apply\_boundary\_conditions(phi, x, y, n);

% Gauss-Seidel Iteration

residuals = zeros(max\_iter, 1);

for iter = 1:max\_iter

phi\_old = phi;

for i = 2:n-1

for j = 2:n-1

% Update phi using Gauss-Seidel formula

phi(i,j) = 0.25 \* (phi(i+1,j) + phi(i-1,j) + phi(i,j+1) + phi(i,j-1) ...

- dx^2 \* source(i,j));

end

end

% Calculate residual

residual = max(max(abs(phi - phi\_old)));

residuals(iter) = residual;

% Check for convergence

if residual < tol

residuals = residuals(1:iter); % Trim the residuals array

break;

end

end

% Stop the timer and record the time

time\_all(g) = toc;

residuals\_all{g} = residuals;

fprintf('Grid size: %dx%d, Iterations: %d, Residual: %e\n', n, n, iter, residual);

end

% Plot residuals vs. iterations for each grid size

figure;

hold on;

for g = 1:length(grid\_sizes)

plot(1:length(residuals\_all{g}), residuals\_all{g}, 'LineWidth', 2);

end

xlabel('Iteration');

ylabel('Residual');

legend('41x41', '81x81', '161x161');

title('Residual vs Iterations for Different Grid Sizes');

grid on;

hold off;

% Contour plot for the finest grid

figure;

contourf(x, y, phi', 50, 'linecolor', 'none'); % Transpose phi for correct orientation

xlabel('x');

ylabel('y');

title('\phi Contour Plot for 161 x 161 grid (Gauss-Seidel)')

colorbar;

% Plot CPU run time vs. number of grid points

figure;

total\_grid\_points = grid\_sizes.^2;

plot(total\_grid\_points, time\_all, '-o', 'LineWidth', 2);

xlabel('Total Number of Grid Points');

ylabel('CPU Run Time (seconds)');

title('CPU Run Time vs Total Number of Grid Points');

grid on;