function gaussian\_elimination\_diffusion(nx, ny)

% This function solves the 2D steady-state diffusion equation using

% Gaussian elimination method for a given grid size (nx, ny).

% Domain size

Fx = 1; Fy = 1;

dx = Fx / (nx-1);

dy = Fy / (ny-1);

% Grid points

x = linspace(0, Fx, nx);

y = linspace(0, Fy, ny);

% Preallocate the coefficient matrix and right-hand side vector

A = zeros(nx\*ny, nx\*ny);

b = zeros(nx\*ny, 1);

% Fill the coefficient matrix A and the RHS vector b

for j = 1:ny

for i = 1:nx

idx = (j-1)\*nx + i; % Flattened index

if i == 1

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

A(idx, idx) = 1;

b(idx) = phi\_bc;

elseif i == nx

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

A(idx, idx) = 1;

b(idx) = phi\_bc;

elseif j == 1

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

A(idx, idx) = 1;

b(idx) = phi\_bc;

elseif j == ny

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

A(idx, idx) = 1;

b(idx) = phi\_bc;

else

A(idx, idx) = -2/dx^2 - 2/dy^2;

A(idx, idx-1) = 1/dx^2;

A(idx, idx+1) = 1/dx^2;

A(idx, idx-nx) = 1/dy^2;

A(idx, idx+nx) = 1/dy^2;

S\_phi = 50000 \* exp(-50 \* ((1 - x(i))^2 + y(j)^2)) ...

\* (100 \* ((1 - x(i))^2 + y(j)^2) - 2);

b(idx) = S\_phi;

end

end

end

% Solve the linear system using Gaussian elimination

tic;

phi = A\b;

cpu\_time = toc;

% Reshape phi into 2D grid

phi = reshape(phi, [nx, ny]);

% Plot the contour of the computed phi field

figure;

contourf(x, y, phi', 50, 'LineColor', 'none');

colorbar;

title(['\phi Field (Gaussian Elimination), Grid: ' num2str(nx) 'x' num2str(ny)]);

xlabel('x');

ylabel('y');

% Display CPU run time

fprintf('CPU run time for grid %dx%d: %f seconds\n', nx, ny, cpu\_time);

end

% Run the function for 21x21, 41x41, and 81x81 grids and record CPU times

grids = [21, 41, 81];

cpu\_times = zeros(size(grids));

total\_points = zeros(size(grids));

for k = 1:length(grids)

nx = grids(k);

ny = grids(k);

gaussian\_elimination\_diffusion(nx, ny);

cpu\_times(k) = toc;

total\_points(k) = nx \* ny;

end

% Plot CPU run time vs total number of grid points

figure;

plot(total\_points, cpu\_times, '-o');

xlabel('Total Number of Grid Points');

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

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

grid on;