clear;

clc;

%% Lab 3

%% Part 1)Generate sparse matrix

D=diag([-4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4 -4],0);

D1=diag([1 1 1 1 1 1 1 1 1 1 1 1],-4);

D2=diag([1 1 1 1 1 1 1 1 1 1 1 1],4);

D3=diag([1 1 1 0 1 1 1 0 1 1 1 0 1 1 1],1);

D4=diag([1 1 1 0 1 1 1 0 1 1 1 0 1 1 1],-1);

D4=D+D1+D2+D3+D4;

%% Part 2) Matlab Backslash

X = [0 0 0 -10 0 0 0 -10 0 0 0 -10 0 0 0 -10]';

tstart = tic;

for n = 1:1000

Y = D4\X;

end

elaspse\_time = toc(tstart); %time is 0.0227/1000 s

disp(elaspse\_time);

matrix\_x = zeros(6,6);

count = 1;

for i = 2 : 5

for j = 2:5

matrix\_x(i,j) = Y(count);

count = count + 1;

end

end

for i = 2:5

matrix\_x(i,6) = 10;

end

% Plot

figure(1);

surf(matrix\_x);

title('Potential 3D plot');

xlabel('x');

ylabel('y');

zlabel('Potential');

%% Part 2

matrix\_y = zeros(6,6);

for i = 2:5

matrix\_y(i,6) = 10;

end

tstart = tic;

iteration = 0;

true\_error = 100; % initialize to be 100%

while(true\_error > 1)

max\_change = 0; % initialize

xy\_cod = [1,1]; % initialize

for i = 2:5

for j = 2:5

old\_val = matrix\_y(i,j);

matrix\_y(i,j) = (matrix\_y(i+1,j) + matrix\_y(i-1,j) + matrix\_y(i,j+1)+ matrix\_y(i,j-1))/4;

cur\_change = abs(matrix\_y(i,j) - old\_val);

if cur\_change > max\_change

max\_change = cur\_change; % updated maximum error

xy\_cod = [i,j]; %updated the coordinate

true\_error = abs(matrix\_y(i,j) - matrix\_x(i,j))/matrix\_x(i,j) \* 100; % in percent

end

end

end

figure(2);

surf(matrix\_y)

iteration = iteration + 1;

% disp(matrix\_y);

% disp(true\_error);

% disp(iteration);

% pause;

end

elapse\_time = toc(tstart);

disp(elapse\_time);

%% Relaxation method

sta\_val = 0.8;

end\_val = 1.2;

[row column] = size(sta\_val:0.01:end\_val);

total\_iteration = zeros(1,column);

count = 1;

for relax\_coef = sta\_val:0.01:end\_val

iteration = 0;

true\_error = 100; % initialize to be 100%

matrix\_y = zeros(6,6);

for i = 2:5

matrix\_y(i,6) = 10;

end

while(true\_error > 1)

max\_change = 0; % initialize

xy\_cod = [1,1]; % initialize

for i = 2:5

for j = 2:5

old\_val = matrix\_y(i,j);

matrix\_y(i,j) = (1-relax\_coef)\*old\_val + relax\_coef\*(matrix\_y(i+1,j) + matrix\_y(i-1,j) + matrix\_y(i,j+1)+ matrix\_y(i,j-1))/4;

cur\_change = abs(matrix\_y(i,j) - old\_val);

if cur\_change > max\_change

max\_change = cur\_change; % updated maximum error

xy\_cod = [i,j]; %updated the coordinate

true\_error = abs(matrix\_y(i,j) - matrix\_x(i,j))/matrix\_x(i,j) \* 100; % in percent

end

end

end

iteration = iteration + 1;

% disp(matrix\_y);

% disp(iteration);

end

total\_iteration(count) = iteration;

count = count + 1;

end

relax\_coef = sta\_val:0.01:end\_val;

figure(3);

plot(total\_iteration, relax\_coef);

title('The plot of total iterations versus the relaxation coefficient given 1% true error');

xlabel('The number of iterations');

ylabel('The relaxation coefficients');

grid on;

%%Matrix Inverse

D\_inv = inv(D4);

% D\_max = max(max(D\_inv)); % absolute max

% D\_min = min(min(D\_inv)); % absolute min

D\_max = max(max(abs(D\_inv))); % absolute max

D\_min = min(min(abs(D\_inv))); % absolute min

D\_cond = cond(D4); % conditional number to indicate the relationship to singular matrix

tstart = tic;

for n = 1:1000

D\_inv=inv(D4);

Y = D\_inv\*X;

end

elaspse\_time = toc(tstart);

disp(elaspse\_time);

%%

for m = 1:16

for n = 1:16

if(D\_inv(m,n) == -D\_max)

x\_max = m;

y\_max = n;

% break;

end

end

end

for m = 1:16

for n = 1:16

if(D\_inv(m,n) == -D\_min)

x\_min = m;

y\_min = n;

% break;

end

end

end

fprintf('The maximum value is %d\n',D\_inv(x\_max,y\_max));

fprintf('The minimum value is %d\n',D\_inv(x\_min,y\_min));