Numerical Analysis

Programming Assignment # 2

Zaid Abid (18100129)

1. Code:

```
%Question Number 1

A = []; %input A matrix
b = []'; %input b vector here
n = length(b);

if all((2*abs(diag(A))) >= sum(abs(A),2))
    disp('A is diagonally dominant')
else
    disp('A is not diagonally dominant')
end

D = diag(diag(A));
L = -tril(A,-1);
U = -triu(A,1);
tol = 10^-5; %input tolerance here

%Jacobi Method

x0 = zeros(n,1); %input initial guess here
x1 = x0;
x2 = ones(n,1);
iter_j = 0;

while abs(max(x2-x1)) > tol
    x2 = inv(D)*(L+U)*x0 + inv(D)*b;
    x1 = x0;
    x0 = x2;
    iter_j = iter_j + 1;
end

disp('Number of Iterations')
disp(iter_j)
disp('Answer by Jacobi')
disp(x2')

%Gauss Siedel Method

x0 = zeros(n,1); %input initial guess here
x1 = x0;
x2 = ones(n,1);
iter_g = 0;

while abs(max(x2-x1)) > tol
    x2 = inv(D-L)*U*x0 + inv(D-L)*b;
    x1 = x0;
    x0 = x2;
    iter_g = iter_g + 1;
end

disp('Number of Iterations')
disp(iter_g)
disp('Answer by Gauss Siedel')
disp(x2')
```

2. Code:

```
%Question Number 2

A = [1 2 -2; 1 1 1; 2 2 1]; %input A matrix
b = [7 2 5]'; %input b vector here
n = length(b);

if all((2*abs(diag(A))) >= sum(abs(A),2))
    disp('A is diagonally dominant')
else
    disp('A is not diagonally dominant')
end

D = diag(diag(A));
L = -tril(A,-1);
U = -triu(A,1);
tol = 10^-5; %input tolerance here

%part a

T_j = inv(D)*(L+U);
e_j = eig(T_j);
spectral_radius_T_j = max(abs(e_j))

T_g = inv(D-L)*U;
e_g = eig(T_g);
spectral_radius_T_g = max(abs(e_g))

%part b

x0 = zeros(n,1); %input initial guess here
x1 = x0;
x2 = ones(n,1);
iter_j = 0;

while abs(max(x2-x1)) > tol
    x2 = inv(D)*(L+U)*x0 + inv(D)*b;
    x1 = x0;
    x0 = x2;
    iter_j = iter_j + 1;
end

disp('Number of Iterations')
disp(iter_j)
disp('Answer by Jacobi')
disp(x2')

%part c

x0 = zeros(n,1); %input initial guess here
x1 = x0;
x2 = ones(n,1);
iter_g = 0;
max_iter = 25;

while abs(max(x2-x1)) > tol && iter_g < max_iter
    x2 = inv(D-L)*U*x0 + inv(D-L)*b;
    x1 = x0;
    x0 = x2;
    iter_g = iter_g + 1;
end

disp('Number of Iterations')
disp(iter_g)
disp('Answer by Gauss Siedel')
disp(x2')
```

ANSWER Q2 Part A)

spectral_radius_T_j = 1.0809e-05;
This is approximately equal to 0

spectral_radius_T_g = 2

ANSWER Q2 Part B)

Number of Iterations
4

Answer by Jacobi
[1 2 -1]

ANSWER Q2 Part C)

Number of Iterations
25

Answer by Gauss Siedel
1.0e+09 * [1.3086 -1.3254 0.0336];
This is nowhere close to the solution

3. Code:

```
%Question Number 3

A = [4 1 -1 1; 1 4 -1 -1; -1 -1 5 1; 1 -1 1 3]; %input A matrix
b = [-2 -1 0 1]'; %input b vector here
n = length(b);

D = diag(diag(A));
C = D^(1/2); %pre-condition
A = inv(C)*A*inv(C');
b = inv(C)*b;
tol = 10^-5; %input tolerance here

x0 = zeros(n,1); %input initial guess here
r0 = b - A*(C'*x0);
v1 = r0;
x1 = x0;
x2 = ones(n,1);
iter = 0;

while abs(max(x2-x1)) > tol
    t1 = (r0'*r0)/(v1'*(A*v1));
    x2 = inv(C')*((C'*x0) + t1*v1);
    r1 = r0 - t1*A*v1;
    s1 = (r1'*r1)/(r0'*r0);
    v2 = r1 + s1*v1;
    r0 = r1;
    x1 = x0;
    x0 = x2;
    v1 = v2;
    iter = iter+1;
end

disp('Number of Iterations')
disp(iter)
disp('Answer by CGM')
disp(x2')
```

ANSWER Q3)

Number of Iterations
5

Answer by CGM
[-0.7534 0.0411 -0.2808 0.6918]