Computational Problem 2

Table of Contents

Setup

```
close all, clear all, clc
format long, format compact
fs = 16;
set(0,'defaulttextfontsize',fs);
set(0,'defaultaxesfontsize',fs);
```

EXERCISE 1

(a)

Since matrix A_ϵ is a pentadiagonal matrix, so only main diagonal, and the first two upper and two lower diagonals exists nonzero elements.

Thus, for size n, each row has a maximum of 5 nonzero elements. Because if A_ϵ is strictly diagonal dominant, it must satisfy $|a_{ii}| > \sum_{j=1,\ j\neq i}^{n} |a_{ij}|$, for $i = 1, \ldots, n$.

So, matrix A_ϵ is strictly diagonal dominant if $1 > 2 \times (\epsilon^2 + \epsilon)$. And we can get $\epsilon < 0.366$.

Therefore, $\epsilon \in [0, 0.366)$.

```
syms e
cond1 = e >= 0;
cond2 = e <= 1;
cond3 = 2*(e^2 + e) < 1;
conds = [cond1 cond2 cond3];

sol = solve(conds, e, 'ReturnConditions', true);
vpa(sol.conditions)
```

```
ans =
0.0 <= x & x < 0.366025403784438646763723170752936
```

(b)

```
tol = 1e-10;
```

```
nmax = 1000;
x0 = [0; 0; 0; 0; 0];
[A, b] = matrix(5, 0.3);

% Jacobi method
[x, niter, relresiter, xiter] = itermeth(A, b, x0, nmax, tol, 'J');
niter

% Gauss-Seidel method
[x, niter, relresiter, xiter] = itermeth(A,b,x0,nmax,tol,'G');
niter
```

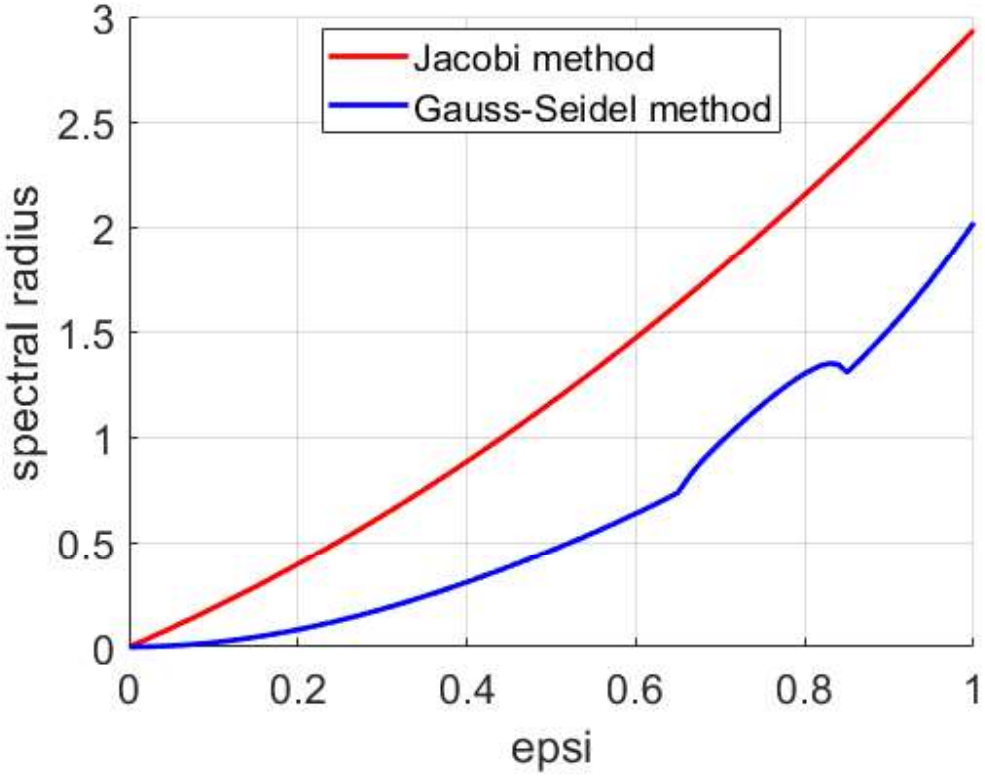
```
itermeth converged in 50 iterations.
niter =
    50
itermeth converged in 14 iterations.
niter =
    14
```

For Jacobi method, it needs 50 iterations to convergence; and for Gauss-Seidel method, it requires 14 iterations to convergence.

(c)

Plot the spectral radius against the value of ϵ for both methods

```
x=linspace(0, 1, 101);
figure
grid on
xlabel('epsi')
ylabel('spectral radius')
hold on

radius_BJ = zeros(size(101));
radius_BGS = zeros(size(101));

for e = 0:0.01:1
    [A, b] = matrix(5, e);
    D = diag(diag(A));  % diagonal part of A_epsi
    L = tril(A,-1);  % lower triangular part of A_epsi
    U = triu(A,1);  % lower triangular part of A_epsi

    B_J = -(D^(-1)) * (L + U);
    B_GS = -(D + L)^(-1) * U;

    radius_BJ(round(e*100) + 1) = max(abs(eig(B_J)));  % spectral radius is
 the maximum modulus of eigenvalues
    radius_BGS(round(e*100) + 1) = max(abs(eig(B_GS)));
end

plot(x, radius_BJ, 'r','LineWidth', 2);
plot(x, radius_BGS, 'blue','LineWidth', 2);
legend({'Jacobi method', 'Gauss-Seidel method'}, 'Location', 'Best');
```



Since the Jacobi method and the Gauss-Seidel method converge if the spectral radius of both B_J and B_{GS} less than 1. Therefore, according to the graph, when $x \in [0, 0.44]$, both B_J and B_{GS} less than 1, and this result is 0.074 larger than the answer of 0.366 from question (a).

According to the graph, if both methods converge, the Gauss-Seidel method is faster than the Jacobi method, because its spectral radius is smaller than the Jacobi method.

When n = 5 and ϵ = 0.5, I recommend using the Gauss-Seidel method.

```
[A, b] = matrix(5, 0.5);
D = diag(diag(A));  % diagonal part of A_epsi
L = tril(A,-1);  % lower triangular part of A_epsi
U = triu(A,1);  % lower triangular part of A_epsi

B_J = -(D^(-1)) * (L + U);
B_GS = -(D + L)^(-1) * U;

radius_BJ = max(abs(eig(B_J)))  % spectral radius is the maximum modulus of
 eigenvalues
radius_BGS = max(abs(eig(B_GS)))

tol = 1e-10;
nmax = 1000;
x0 = [0; 0; 0; 0; 0];
[A, b] = matrix(5, 0.5);
```

```
% Jacobi method
[x, niter, relresiter, xiter] = itermeth(A, b, x0, nmax, tol, 'J');

% Gauss-Seidel method
[x, niter, relresiter, xiter] = itermeth(A,b,x0,nmax,tol,'G');
```

```
radius_BJ =
   1.163568373244976
radius_BGS =
   0.460186129658958
itermeth reached the maximum iteration number without converging.
itermeth converged in 27 iterations.
```

Since spectral radius of B_J larger than 1, the Jacobi method will not convergence. Therefore, it should use the Gauss-Seidel method When n = 5 and ϵ = 0.5.

Published with MATLAB® R2022b