```
% Shiqi Su
% MATH0033 Numerical Methods
% Theoretical sheet 2
```

Setup

```
close all, clear all, clc
fs=14;
set(0,'defaulttextfontsize',fs);
set(0,'defaultaxesfontsize',fs);
```

Exercise 1(a)

For $n \geq 5$ using strictly diagonal dominant condition, we can compute $|a_{ii}| > \sum_{i=1, j \neq i}^{n} |a_{ij}|$. So $1 > \epsilon + \epsilon + \epsilon^2 + \epsilon^2$ with $\epsilon > 0$ and therefore it's easy to compute $0 < \epsilon < \frac{\sqrt{3}-1}{2}$.

Exercise 1(b)

```
n=5;
epsi=0.3;
[A,b] = matrix(n,epsi);
x0=zeros(n,1);
nmax=1000;
tol=1e-10;
% Jacobi method
[x_J, niter_J, relresiter_J, xiter_J] = itermeth(A,b,x0,nmax,tol,'J');
```

```
itermeth converged in 50 iterations.
```

```
% Gauss-Seidel method
[x_G, niter_G, relresiter_G, xiter_G] = itermeth(A,b,x0,nmax,tol,'G');
```

```
itermeth converged in 14 iterations.
```

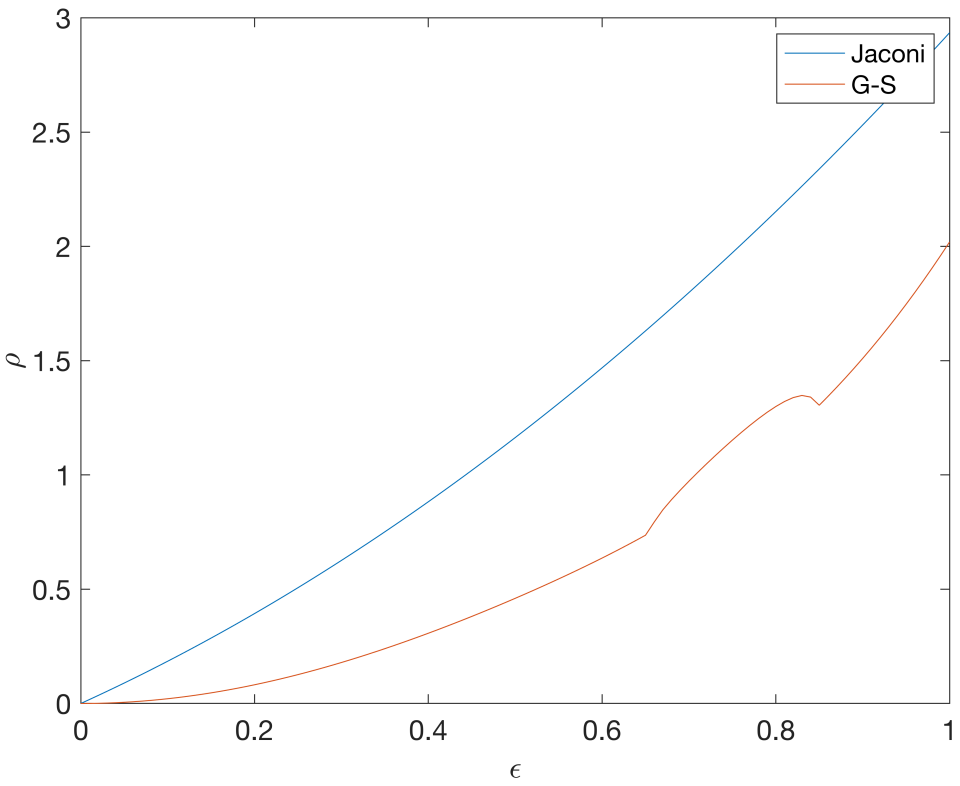
Jacobi method converged in 50 iterations.

Gauss-Seidel converged in 14 iterations.

Exercise 1(c)

```
n=5;
epsi_c=[0:0.01:1];
x0=zeros(n,1);
nmax=1000;
tol=1e-10;
D_epsi=eye(n);

eig_J=zeros(101,1);
eig_G=zeros(101,1);
```

```
for i = 1:101
    % calculate 101 A and b matrices
    [A_epsi,b_epsi] = matrix(n,epsi_c(i));

    % decomponent matrix A,b
    E_epsi=tril(-A_epsi,-1);
    F_epsi=triu(-A_epsi,1);
    B_J=D_epsi\(E_epsi+F_epsi);
    B_G=(D_epsi-E_epsi)\F_epsi;
    % calculate spectural radius for both method
    eig_J(i)= max(abs(eig(B_J)));
    eig_G(i)= max(abs(eig(B_G)));
    if eig_J(i)<1
        max_J = epsi_c(i);
    end
    if eig_G(i)<1
        max_GS = epsi_c(i);
    end

end

% plot
figure(1)
plot(0:0.01:1,eig_J)
hold on
plot(0:0.01:1,eig_G)
hold off
xlabel('\epsilon')
ylabel('\rho')
legend('Jaconi','G-S')
```




```
fprintf('Jacobi converges when epsi smaller than %d\n',max_J)
```

```
Jacobi converges when epsi smaller than 4.400000e-01
```

```
fprintf('GS converges when epsi smaller than %d\n',max_GS)
```

```
GS converges when epsi smaller than 7.000000e-01
```

Two method converge as long as $\epsilon \in [0, 0.44)$,take intersection of [0,0.44) and [0,0.7) approximately. The difference is that, the range looked through graph is larger than answer in (a).

I expect Gauss-Seidel method converges faster because $\rho(B_{GS}) < 1$ always hold for $\epsilon \in [0, 1]$.

```
n=5;
epsi_check = 0.5;
D_epsi=eye(n);
% calculate 101 A and b matrices
[A_epsi,b_epsi] = matrix(n,epsi_check);

% decomponent matrix A,b
E_epsi=tril(-A_epsi,-1);
F_epsi=triu(-A_epsi,1);
B_J=D_epsi\(E_epsi+F_epsi);
B_G=(D_epsi-E_epsi)\F_epsi;
% calculate spectural radius for both method
eig_J_check= max(abs(eig(B_J)));
eig_G_check= max(abs(eig(B_G)));
```

```
fprintf('spactrual radiu %d of Jacobi method when n=5\n',eig_J_check)
```

```
spactrual radiu 1.163568e+00 of Jacobi method when n=5
```

```
fprintf('spactrual radiu %d of G-S method when n=5\n',eig_G_check)
```

```
spactrual radiu 4.601861e-01 of G-S method when n=5
```

Since the spactural radius of G-S method is much smaller than Jacobi method which implies faster convergence. And it support my recommendation.

Exercise 2(a)

```
clear all
N = [5,10,20,40,80];
Nvec = [5,10,20,40,80];
rhoBJ =zeros(1,5);
rhoBGS =zeros(1,5);
for i = 1:length(N)
    % calculate A and b matrices

    h = 1/N(i);
    A = (2/h^2)*diag(ones(N(i)-1,1)) - (1/h^2)*diag(ones(N(i)-2,1),1)...
     - (1/h^2)*diag(ones(N(i)-2,1),-1);
    b = sin(pi*h*(1:N(i)-1)');
    D = diag(diag(A));
    % decomponent matrix A,b
    E=tril(-A,-1);
    F=triu(-A,1);
    B_J=D\(E+F);
    B_G=(D-E)\F;
    % calculate spectural radius for Jacobi and G-S
    rhoBJ(i)= max(abs(eig(B_J)));
    rhoBGS(i)= max(abs(eig(B_G)));
end
A_1 = [rhoBJ;Nvec];
A_2 = [rhoBGS;Nvec];
fprintf('%8.3f is the spectral radii of Jacobi method when N=%d\n',A_1)
```

```
   0.809 is the spectral radii of Jacobi method when N=5
   0.951 is the spectral radii of Jacobi method when N=10
   0.988 is the spectral radii of Jacobi method when N=20
   0.997 is the spectral radii of Jacobi method when N=40
   0.999 is the spectral radii of Jacobi method when N=80
```

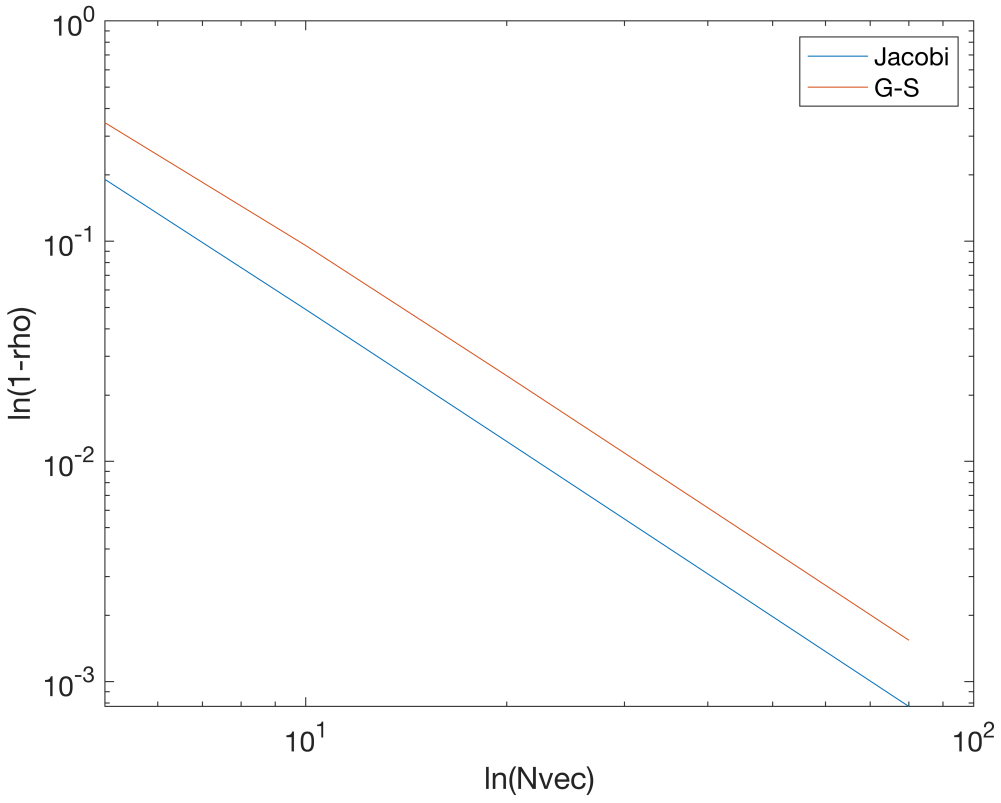
```
fprintf('%8.3f is the spectral radii of G-S method when N=%d\n',A_2)
```

```
   0.655 is the spectral radii of G-S method when N=5
   0.905 is the spectral radii of G-S method when N=10
   0.976 is the spectral radii of G-S method when N=20
   0.994 is the spectral radii of G-S method when N=40
   0.998 is the spectral radii of G-S method when N=80
```

1.Both method converge because all spectural radius value are smaller than 1 based on different N. And I suppose G-S method converge faster since smaller convergence rate.

2. As N increases, $\rho(B_J)$ and $\rho(B_{GS})$ converge to 1. so the 2 methods will have more number of iterations to approximate true value because covergence speed slow down.

```
% calculate slope of loglog plot,i.e alpha
alpha_J = (log(1-rhoBJ(end))-log(1-rhoBJ(1)))/(log(Nvec(end))-log(Nvec(1)));
alpha_GS = (log(1-rhoBGS(end))-log(1-rhoBGS(1)))/(log(Nvec(end))-log(Nvec(1)));
% calculate offset of loglog plot, i.e log(C)
C_J=exp(1-rhoBJ(1));
C_GS=exp(1-rhoBGS(1));

figure(2)
loglog(Nvec,1-rhoBJ)
hold on
loglog(Nvec,1-rhoBGS)
hold off
xlabel('ln(Nvec)')
ylabel('ln(1-rho)')
legend('Jacobi','G-S')
```




```
fprintf('%d is the slop of Jacobi method\n',alpha_J)
```

```
-1.988141e+00 is the slop of Jacobi method
```

```
fprintf('%d is the C of Jacobi method\n',C_J)
```

```
1.210439e+00 is the C of Jacobi method
```

```
fprintf('%d is the slop of G-S method\n',alpha_GS)
```

```
-1.952082e+00 is the slop of G-S method
```

```
fprintf('%d is the C of G-S method',C_GS)
```

```
1.412684e+00 is the C of G-S method
```

Through loglog plot and write linear equation $ln(1-\rho) = \alpha ln(N) + ln(C)$,where α is slope and $ln(C)$is constant offset. Rearranging the equation can see relation clearly, i.e $\rho = 1 - CN^{\alpha}$.

To find the slop, select 2 points on linear equation $(ln(N_1), ln(1-\rho_1)), (ln(N_2), ln(1-\rho_2))$. And use $\alpha = \frac{ln(1-\rho_2) - ln(1-\rho_1)}{ln(N_2) - ln(N_1)}$.

Considering offset C, I just select the first element in $1-\rho$ and take exponential.

And for iteration k_{tol}, when N increases,because ρ tends to 1, we should let k increases inoder to achivev a fixed solution accuracy.

Excercise 2 (b)

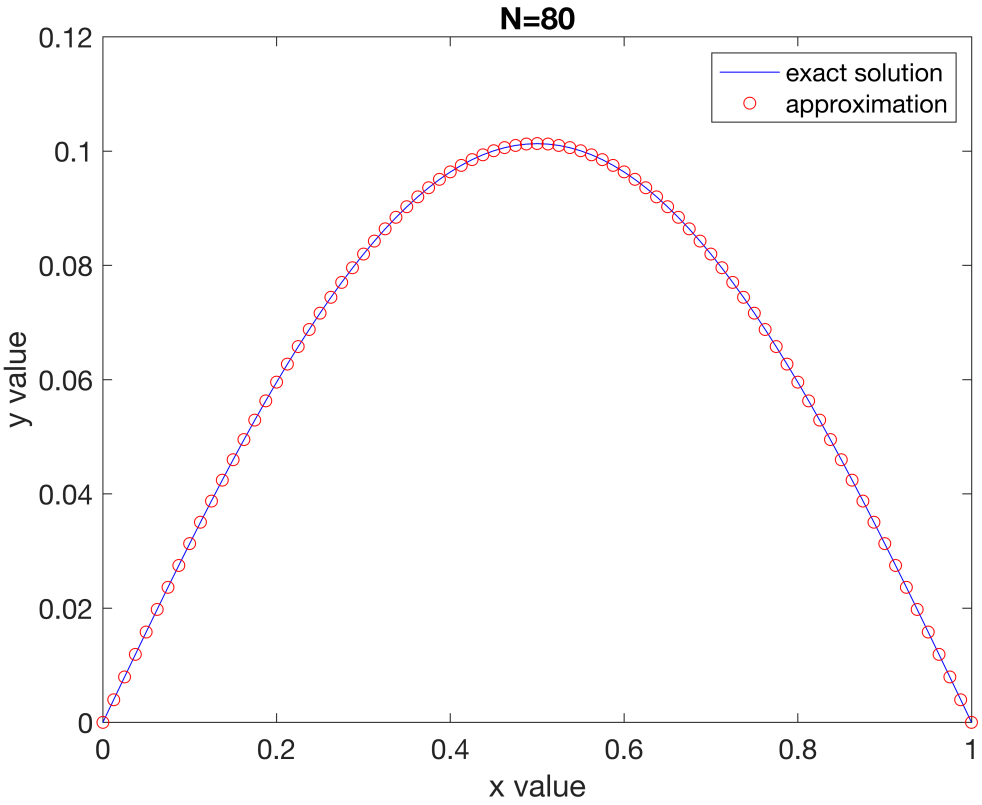
```
clear all
nmax = 1e+5;
tol = 1e-10;
N = [5,10,20,40,80];

error_vect = zeros(5,1);
for i = 1:length(N)
    h = 1/N(i);
    u_init = zeros(N(i)-1,1);

    A = (2/h^2)*diag(ones(N(i)-1,1)) - (1/h^2)*diag(ones(N(i)-2,1),1)...
     - (1/h^2)*diag(ones(N(i)-2,1),-1);
    b = sin(pi*h*(1:N(i)-1)');

   % Gauss-Seidel method
   [u_G, niter_G, relresiter_G, uiter_G] = itermeth(A,b,u_init,nmax,tol,'G');
   % compute error
   x = linspace(0,1,N(i)+1);
   y = pi^(-2)*sin(pi*x);
   u_G = [0,u_G',0];

   error_vect(i) = max(abs(u_G-y));

end
```

```
itermeth converged in 56 iterations.
itermeth converged in 231 iterations.
itermeth converged in 931 iterations.
```

```
itermeth converged in 3731 iterations.
itermeth converged in 14929 iterations.
```

The G-S method converges as N increase.

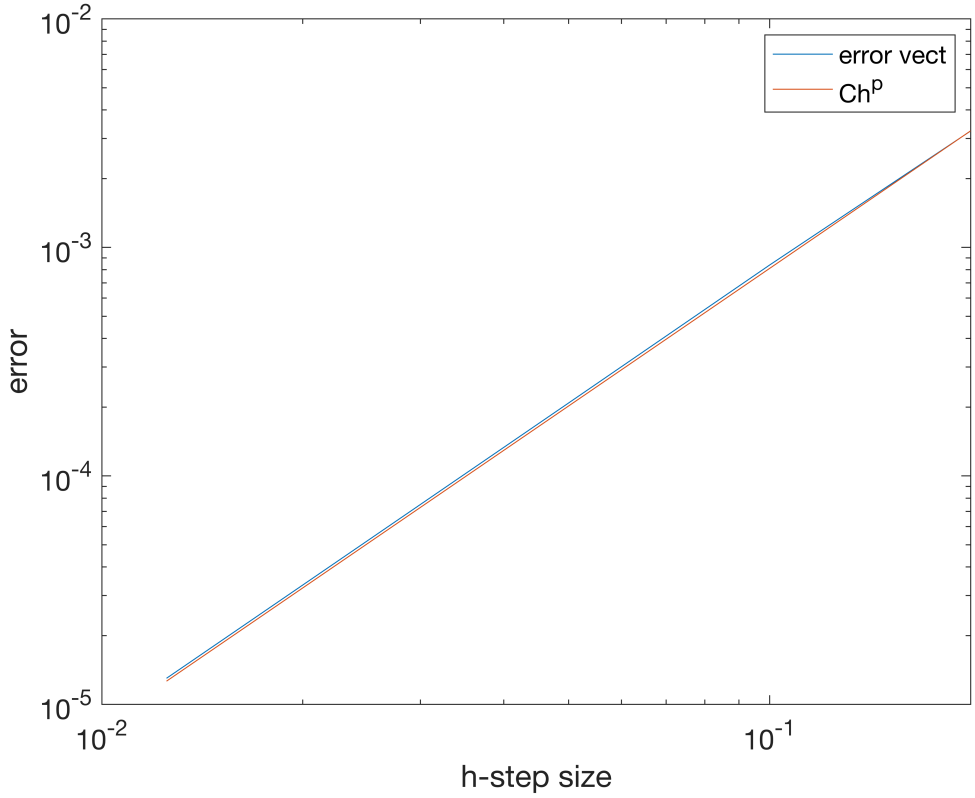
```
% Plot the result at N=80
x = linspace(0,1,81);
y = pi^(-2)*sin(pi*x);

figure(3)
plot(x,y,'b')
hold on
plot(x,u_G,'ro')
hold off
xlabel('x value')
ylabel('y value')
title('N=80')
legend('exact solution','approximation')
```



plot loglog step size h against error

```
hvect = 1./N;
[para_BGS]=regress(log(error_vect),[ones(5,1),log(hvect')]);
C_GS = exp(para_BGS(1));
alpha_GS = para_BGS(2);
p=2;
```

```
% plot step size against global error
figure(4)
loglog(hvect,error_vect)
hold on
%loglog(hvect,C*hvect.^logC_GS)
loglog(hvect,C_GS*hvect.^p)
hold off
xlabel('h-step size')
ylabel('error')
legend('error vect','Ch^p')
```



My plot supports theoretical extimate.The approximation is quite close to the true y, a quratic equation and the finite method is 2nd order accuracy. It can be seen from graph,error vector is below truncation error which demostrates $E(N) \leq Ch^2$ as N tends to infinite.

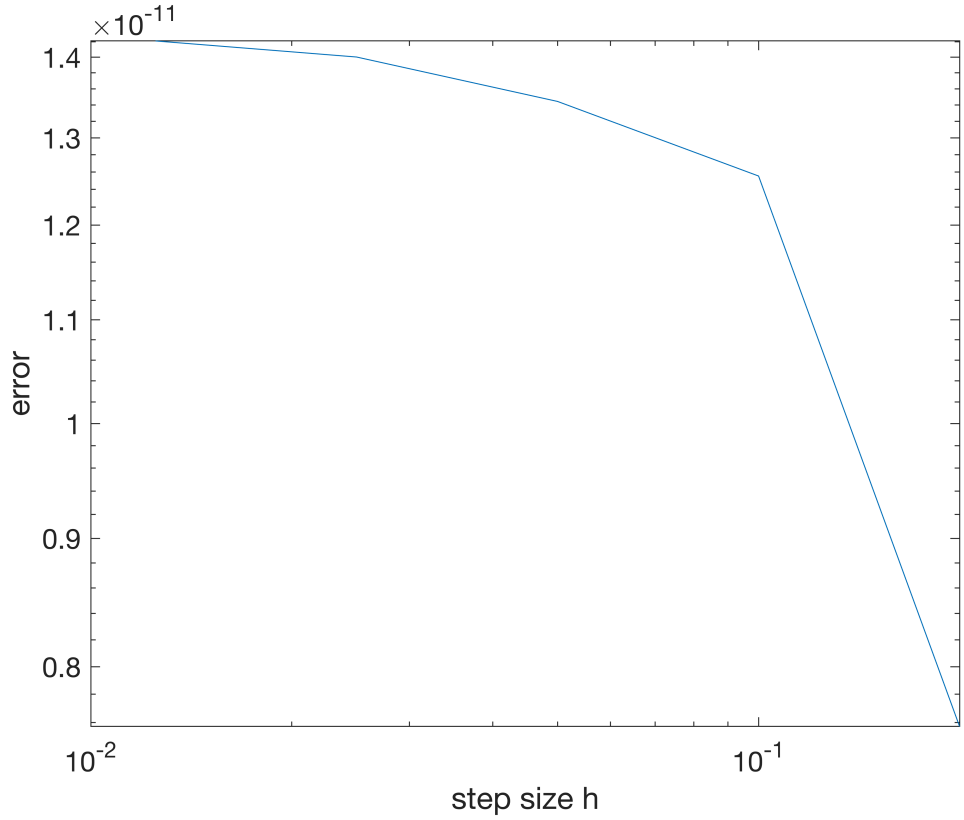
Excercise 2(c)

```
clear all
nmax = 1e+5;
tol = 1e-10;
N = [5,10,20,40,80];
hvect = 1./N;
error_vect = zeros(1,5);

for i = 1:length(N)
    h = 1/N(i);
    u_init = zeros(N(i)-1,1);
```

```
    A = (2/h^2)*diag(ones(N(i)-1,1)) - (1/h^2)*diag(ones(N(i)-2,1),1)...
     - (1/h^2)*diag(ones(N(i)-2,1),-1);
    b = ones(N(i)-1,1);
    %b = h*(1:N(i)-1);


   % Gauss-Seidel method
   [u_GS, niter_GS, relresiter_GS, uiter_GS] = itermeth(A,b,u_init,nmax,tol,'G');
   % compute error
   x = linspace(0,1,N(i)+1);

   y_exact = 1/2*x.*(1-x) ;
   u_GS = [0,u_GS',0];

   error_vect(i) = max(abs(u_GS-y_exact));

end
```

```
itermeth converged in 56 iterations.
itermeth converged in 230 iterations.
itermeth converged in 928 iterations.
itermeth converged in 3716 iterations.
itermeth converged in 14865 iterations.
```

```
loglog(hvect,error_vect)
xlabel('step size h')
ylabel('error')
```



The relation between log(hvect) and log(error_vect) is not linear. As the step size h increase (N decreases), the rate of error decreases faster.