Project 2

Due on Friday, May 5th, 2017 (Four weeks)

Consider the tridiagonal matrix $\mathbf{A} = (a_{i,j})_{1\leq i,j\leq n}$ given by

$$a_{i,j} = \begin{cases} -\frac{1}{h^2} & |i-j| = 1 \\ \frac{2}{h^2} & i = j \\ 0 & Otherwise \end{cases} \tag{1}$$

obtained when the following ODE

$$\begin{aligned} -u''(x) &= f(x), \quad x \in [0,1] \\ u(0) &= u(1) = 0, \end{aligned} \tag{2}$$

is discretized using second order centered differences:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f(x_i) \quad i = 1, 2, \ldots, n, \tag{3}$$

where $h = 1/(n+1)$.

1. For each $k = 1, 2, \ldots, n$, show that the vector $\mathbf{u}^{(k)}$ given by

   $$u_i^{(k)} = \sin\left(\frac{\pi k i}{n+1}\right), \quad i = 1, 2, \ldots, n \tag{4}$$

   is an eigenvector of the matrix A, and determine the corresponding eigenvalue λ_k.

2. Set up the Jacobi iteration for system (3), and show that the vectors (4) are also eigenvectors of the Jacobi iteration matrix, $\mathbf{T_J}$.

3. Determine the spectral radius of $\mathbf{T_J}$, $\rho(\mathbf{T_J})$.

4. The Jacobi iteration can be written as

   $$\mathbf{x}^{(k+1)} = \mathbf{T_J}\mathbf{x}^{(k)} + \mathbf{c} \tag{5}$$

   From the previous steps, we know that $\mathbf{T_J}$ is symmetric and diagonalizable. Use this fact to show that if $\mathbf{x}^*$ is the (unique) fixed point of (5), then

   $$\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 \leq \rho(\mathbf{T_J})^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2 \tag{6}$$

5. Use formula (6) and the spectral radius obtained earlier to estimate the number of iterations necessary for the error to be less than a given ϵ as a function of the number of grid points used, n. You should end up with a formula of the form $Iter = O(n^\alpha)$ for some α.

6. Fix $\epsilon = 10^{-4}$. Consider the vector $\mathbf{u}$ such that

$$u_i = \sin\left(\frac{\pi i}{n+1}\right) \tag{7}$$

and construct the right hand side $\mathbf{f} = \mathbf{Au}$. Solve the system of equations

$$\mathbf{Ax} = \mathbf{f} \tag{8}$$

using Jacobi's method. Use the values $n = 10, 20, 40, 80, 160, 320$. Do a log-log plot of the number of Jacobi iterations necessary for the error to satisfy

$$\|\mathbf{x}^{(k)} - \mathbf{u}\|_2 \leq \epsilon \|\mathbf{u}\|. \tag{9}$$

Explain theoretically why this is the expected number of iterations.

7. Repeat the previous part with Gauss-Seidel's method. How much faster is it?

Solution.

1. Consider the tridiagonal matrix $\mathbf{A} = (a_{i,j})_{1 \leq i,j \leq n}$ given by

$$a_{i,j} = \begin{cases} -\frac{1}{h^2} & |i-j| = 1 \\ \frac{2}{h^2} & i = j \\ 0 & \textit{Otherwise} \end{cases} \tag{10}$$

i.e.,

$$\mathbf{A} = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & -1 \\ 0 & 0 & \cdots & -1 & 2 \end{pmatrix}_{n \times n}$$

According to the definition of eigenvalue and eigenvector of the matrix, it is easy to check the vector $\mathbf{u}^{(k)}$ given by

$$u_i^{(k)} = \sin\left(\frac{\pi k i}{n+1}\right), \quad i = 1, 2, \ldots, n$$

is an eigenvector of the matrix $\mathbf{A}$, for each $k = 1, 2, \ldots, n$. When $i \neq 1, n$, we can take the i-th component of $\mathbf{Au}^{(k)}$, i.e., we need to show that

$$\begin{aligned}(\mathbf{Au}^{(k)})_i &= \frac{1}{h^2}\left(-\sin\frac{\pi k(i-1)}{n+1} + 2\sin\frac{\pi k i}{n+1} - \sin\frac{\pi k(i+1)}{n+1}\right)\\ &\overset{(?)}{=} \lambda_k \mathbf{u}_i^{(k)} = \lambda_k \sin\left(\frac{\pi k i}{n+1}\right), \end{aligned} \tag{11}$$

note that

$$\sin(A \pm B) = \sin A \cos B \pm \cos A \sin B,$$
$$\sin(A+B) + \sin(A-B) = 2\sin A \cos B.$$

Then,

$$\begin{aligned}(\mathbf{Au}^{(k)})_i &= \frac{1}{h^2}\left(2\sin\frac{\pi k i}{n+1} - 2\sin\frac{\pi k i}{n+1}\cdot\cos\frac{\pi k}{n+1}\right)\\ &= \frac{1}{h^2}2\sin\frac{\pi k i}{n+1}\left(1-\cos\frac{\pi k}{n+1}\right), \end{aligned} \tag{12}$$

consequently,

$$\lambda_k = \frac{2}{h^2}\left(1-\cos\frac{\pi k}{n+1}\right), \quad k = 1, \ldots, n.$$

Let us check the first and the last component, for $i = 1$, we need to show

$$\begin{aligned}(\mathbf{Au}^{(k)})_1 &= \frac{1}{h^2}\left(2\sin\frac{\pi k}{n+1} - \sin\frac{\pi 2k}{n+1}\right)\\ &= \frac{1}{h^2}\left(2\sin\frac{\pi k}{n+1} - 2\sin\frac{\pi k}{n+1}\cos\frac{\pi k}{n+1}\right)\\ &= \frac{2}{h^2}\left(1-\cos\frac{\pi k}{n+1}\right)\sin\frac{\pi k}{n+1}\\ &= \lambda_k \sin\frac{\pi k}{n+1}, \end{aligned} \tag{13}$$

for $i = n$, we need to show

$$
\begin{aligned}
(\mathbf{A}\mathbf{u}^{(k)})_n &= \frac{1}{h^2}\left(-\sin\frac{\pi k(n-1)}{n+1} + 2\sin\frac{\pi kn}{n+1}\right) \\
&= \frac{1}{h^2}\left(2\sin\frac{\pi kn}{n+1} - \sin\frac{\pi kn}{n+1}\cos\frac{\pi k}{n+1} + \cos\frac{\pi kn}{n+1}\sin\frac{\pi k}{n+1}\right) \\
&\overset{(*)}{=} \frac{2}{h^2}\left(1 - \cos\frac{\pi k}{n+1}\right)\sin\frac{\pi kn}{n+1} \\
&= \lambda_k \sin\frac{\pi kn}{n+1}, \qquad (14)
\end{aligned}
$$

where the identity $(*)$ is true, if

$$
-\sin\frac{\pi kn}{n+1}\cos\frac{\pi k}{n+1} + \cos\frac{\pi kn}{n+1}\sin\frac{\pi k}{n+1} = -2\cos\frac{\pi k}{n+1}\sin\frac{\pi kn}{n+1}
$$

i.e.,

$$
\begin{aligned}
\cos\frac{\pi k}{n+1}\sin\frac{\pi kn}{n+1} + \cos\frac{\pi kn}{n+1}\sin\frac{\pi k}{n+1} &= \sin\left(\frac{\pi kn}{n+1} + \frac{\pi k}{n+1}\right) \\
&= \sin\left(\frac{\pi k(n+1)}{n+1}\right) \\
&= 0. \qquad (15)
\end{aligned}
$$

Therefore, the vectors $\mathbf{u}^{(k)}$ with components

$$
u_i^{(k)} = \sin\left(\frac{\pi ki}{n+1}\right), \quad i = 1, 2, \ldots, n,
$$

and the corresponding eigenvalue

$$
\lambda_k = \frac{2}{h^2}\left(1 - \cos\frac{\pi k}{n+1}\right), \quad k = 1, \ldots, n.
$$

2. Consider the Jacobi iteration is $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$, where $\mathbf{U} = \mathbf{L}^T$,

$$
\mathbf{L} = \frac{1}{h^2}\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}_{n\times n}
$$

$$\mathbf{D} = \frac{1}{h^2}\begin{pmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 2 \end{pmatrix}_{n\times n}$$

Set up the Jacobi iteration for system (3)

$$\mathbf{A}\mathbf{x} = \mathbf{f} \tag{16}$$

i.e.,

$$(\mathbf{D} - \mathbf{L} - \mathbf{U})\mathbf{x} = \mathbf{f}.$$

Then

$$\mathbf{D}\mathbf{x}^{(k+1)} = (\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} + \mathbf{f}.$$

We can obtain the Jacobi iteration as the following that

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{f}.$$

Introducing the notation

$$\mathbf{x}^{(k+1)} = \mathbf{T_J}\mathbf{x}^{(k)} + \mathbf{c}$$

where $\mathbf{T_J} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$, and $\mathbf{c} = \mathbf{D}^{-1}\mathbf{f}$. We rewrite the scheme in component,

$$\begin{cases} \mathbf{x}_1^{(k+1)} & = \frac{1}{2}\mathbf{x}_2^{(k)} + \frac{h^2}{2}f_1 \qquad i = 1, \\ \\ \mathbf{x}_i^{(k+1)} & = \frac{1}{2}\left(\mathbf{x}_{i-1}^{(k)} + \mathbf{x}_{i+1}^{(k)}\right) + \frac{h^2}{2}f_i \qquad i = 2 : n-1, \\ \\ \mathbf{x}_n^{(k+1)} & = \frac{1}{2}\mathbf{x}_{n-1}^{(k)} + \frac{h^2}{2}f_n \qquad i = n. \end{cases}$$

the vectors (4) are also eigenvectors of the Jacobi iteration matrix, $\mathbf{T_J}$. Indeed,

$$\begin{aligned} \mathbf{T_J} &= \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}) \\ &= \mathbf{D}^{-1}(\mathbf{D} - \mathbf{A}) \\ &= -\mathbf{D}^{-1}\mathbf{A} + \mathbf{I} \\ &= -\frac{h^2}{2}\mathbf{A} + \mathbf{I}, \end{aligned}$$

the final equality is due to $\mathbf{D} = \frac{2}{h^2}\mathbf{I}$. Then,

$$\begin{aligned}
\mathbf{T_J}\mathbf{u}^{(k)} &= \left(-\frac{h^2}{2}\mathbf{A} + \mathbf{I}\right)\mathbf{u}^{(k)} \\
&= -\frac{h^2}{2}\lambda_k \mathbf{u}^{(k)} + \mathbf{u}^{(k)} \\
&= \left(-\frac{h^2}{2}\lambda_k + 1\right)\mathbf{u}^{(k)}.
\end{aligned}$$

Note that

$$\lambda_k = \frac{2}{h^2}\left(1 - \cos\frac{\pi k}{n+1}\right), \quad k = 1, \ldots, n.$$

we get that the eigenvalue of $\mathbf{T_J}$ are

$$\begin{aligned}
\mu_k &= -\frac{h^2}{2}\lambda_k + 1 \\
&= -\frac{h^2}{2}\cdot\frac{2}{h^2}\left((1 - \cos\frac{\pi k}{n+1}\right) + 1 \\
&= \cos\frac{\pi k}{n+1},
\end{aligned}$$

so $\sigma(\mathbf{T_J}) = \{\mu_k\}_{k=1}^{n}$.

3. Consider the spectral radius of $\mathbf{T_J}$, $\rho(\mathbf{T_J})$.

$$\begin{aligned}
\rho(\mathbf{T_J}) &= \max_k |\mu_k| \\
&= \max_k |\cos\frac{\pi k}{n+1}| \\
&= \cos\frac{\pi}{n+1} \\
&\approx 1 - \frac{1}{2}\left(\frac{\pi}{n+1}\right)^2 \\
&< 1.
\end{aligned}$$

The approximation is due to $\sin x \sim x$, when $x \to 0$, from the last inequality, we know that the convergence of Jacobi iteration.

4. Consider The Jacobi iteration can be written as

$$\mathbf{x}^{(k+1)} = \mathbf{T_J}\mathbf{x}^{(k)} + \mathbf{c},$$

where $\mathbf{T_J} = \mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})$, and $\mathbf{c} = \mathbf{D}^{-1}\mathbf{f}$. Then,

$$\mathbf{T_J}' = (\mathbf{L}+\mathbf{U})'(\mathbf{D}^{-1})' = (\mathbf{L}+\mathbf{U})\mathbf{D}^{-1} = \mathbf{D}^{-1}(\mathbf{L}+\mathbf{U}) = \mathbf{T_J},$$

it is easy to obtain $\mathbf{T_J}$ is symmetric and diagonalizable. If $\mathbf{x}^*$ is the (unique) fixed point of (5), then

$$\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 \leq \rho(\mathbf{T_J})^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2$$

Indeed, for a symmetric matrix $\mathbf{T_J}$, it can be shown that $\|\mathbf{T_J}\|_2 = \sqrt{\lambda_{\max}(\mathbf{T_J}'\mathbf{T_J})} = \rho(\mathbf{T_J})$, so we need to show

$$\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 \leq \|\mathbf{T_J}\|_2^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2.$$

Observe that $\mathbf{x}^* = \mathbf{T_J}\mathbf{x}^* + \mathbf{c}$ and $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|$, then

$$\begin{aligned}
\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 &= \|\mathbf{T_J}\mathbf{x}^{(k-1)} + \mathbf{c} - (\mathbf{T_J}\mathbf{x}^* + \mathbf{c})\|_2 \\
&= \|\mathbf{T_J}(\mathbf{x}^{(k-1)} - \mathbf{x}^*)\|_2 \\
&\leq \|\mathbf{T_J}\|_2 \|\mathbf{x}^{(k-1)} - \mathbf{x}^*\|_2 \\
&\leq \|\mathbf{T_J}\|_2^2 \|\mathbf{x}^{(k-2)} - \mathbf{x}^*\|_2 \\
&\leq \cdots \\
&\leq \|\mathbf{T_J}\|_2^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2,
\end{aligned}$$

i.e.,

$$\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2 \leq \rho(\mathbf{T_J})^k \|\mathbf{x}^{(0)} - \mathbf{x}^*\|_2.$$

5. Consider the formula as above and the spectral radius $\rho(\mathbf{T_J})$, we take $\mathbf{x}^{(0)} = 0$. Then, the relative error is

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2} \leq \rho(\mathbf{T_J})^k.$$

We want $\rho^k \leq \epsilon$, we take logarithm on both side, it follows that $k \log \rho \leq \log \epsilon$, we note that $\log \rho < 0$ due to $\rho < 1$, and also $\log \epsilon < 0$, therefore,

$$k|\log \rho| \geq |\log \epsilon|.$$

Then

$$k \geq \frac{|\log \epsilon|}{|\log \rho|},$$

we can set $k = \frac{|\log \epsilon|}{|\log \rho|}$, we need to show $k = Cn^{\alpha}$, i.e.,

$$\log k = \alpha \log n + \log C.$$

So, setting $\epsilon = 10^{-r}$, where r is a constant. We consider the relationship between

$$\log k = \log \frac{|\log \epsilon|}{|\log \rho|} = \log |r| - \log |\log \cos \frac{\pi}{n+1}|.$$

and $\log n$, we need to do a work of linear regression using LSM. We list as the following that

n	$\log n$	$\log \lvert \log \cos \frac{\pi}{n+1} \rvert$
10	2.30258509	-3.1857
20	2.99573227	-4.4890
40	3.688879454	-5.8299
80	4.38202663	-7.1923
160	5.075173815	-8.5664
320	5.76832099579	-9.9466

We can draw the graph as follows

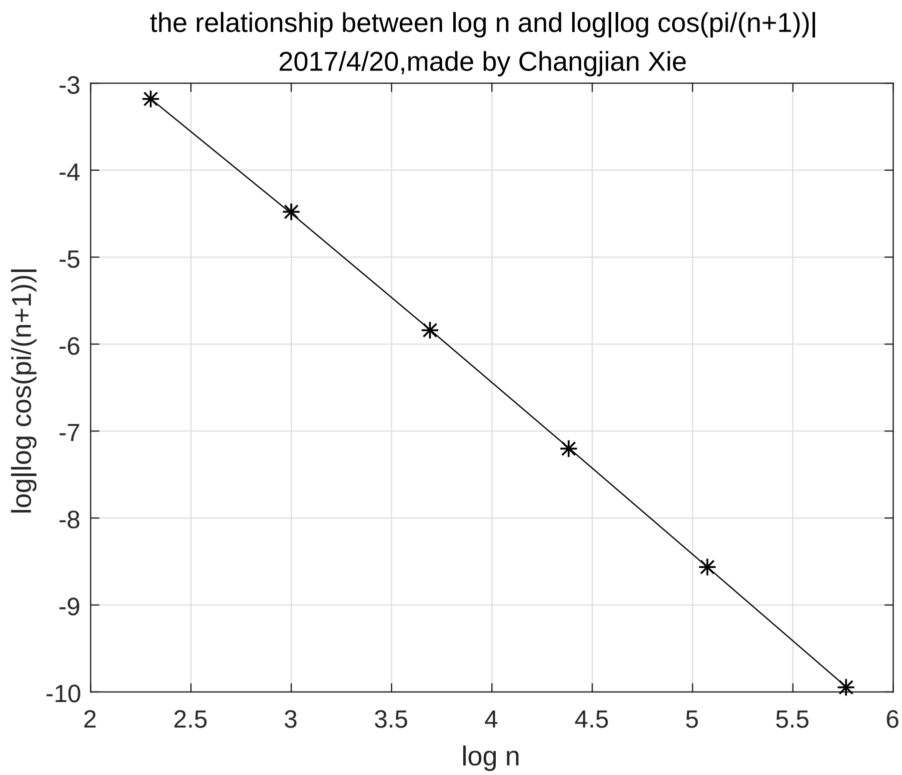

Figure 1: Using the values n = 10,20,40,80,160,320. Do a plot of $\log n$ and $\log|\log\cos\frac{\pi}{n+1}|$

The first problem transfers into

$$\log|\log\cos\frac{\pi}{n+1}| = \log C_1 + \alpha_1 \log n,$$

we have implied the method in Project 1, now, I omit the detail and only give the result as follows

α_1
-1.9537886863

Thus, we obtain that $\alpha = -\alpha_1 = 1.9537886863$, $Iterk = O(n^{\alpha})$ for some α. In fact, we can also do the Taylor extension of the function, then, we deserve the same result.

6. Consider Fix $\epsilon = 10^{-3}$ and note that $\mathbf{f} = \mathbf{A}\mathbf{u}$. Then, the vector $\mathbf{u}$ is the exact solution. In the following, we solve the system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{f}$$

using Jacobi's method with $\mathbf{x}^{(0)} = 0$.

Jacobi iterative algorithm
To solve $\mathbf{A}\mathbf{x} = \mathbf{f}$ given an initial approximation $\mathbf{x}^{(0)}$,

INPUT. The number of equations and unknowns n, the entries a_{ij}, $1 \le i, j \le n$ of the matrix $\mathbf{A}$, the entries f_i of $\mathbf{f}$, the entries XO_i, $1 \le i \le n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$, tolerance TOL; maximum number of iterations N.

OUTPUT. The approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Step 1. Set $k = 1$.

Step 2. While $k \le N$ do Steps $3 - 6$.

Step 3. For $i = 1, \ldots, n$, set

$$x_i = \frac{-\sum_{\substack{j=1 \\ j \neq i}}^{n} (a_{ij} XO_j) + f_i}{a_{ii}}$$

Step 4. If $\|\mathbf{x} - \mathbf{XO}\| < TOL$, then OUTPUT $(x_1, \ldots, x_n)$; (The procedure was successful). STOP.

Step 5. Set $k = k + 1$.

Step 6. For $i = 1, \ldots, n$, set $XO_i = x_i$.

Step 7. OUTPUT (Maximum number of iterations exceeded); (The procedure was successful). STOP.

Using the values $n = 10, 20, 40, 80, 160, 320$. Do a log-log plot of the number of Jacobi iterations necessary for the error to satisfy

$$\|\mathbf{x}^{(k)} - \mathbf{u}\|_2 \le \epsilon \|\mathbf{u}\|.$$

We obtain that

n	$iter\ k$
10	223
20	821
40	3135
80	12243
160	48377
320	192314

The matlab code of Jacobi function file as following that

```
function [x, iter]=myjacobi(a,xexact,x0,tol)

    b=a*xexact;

    n=length(x0);
    u0=x0;
    x=x0;

    iter = 0;
    error=1;

    while error > tol
        for i=1:n
            x(i) = a(i,1:i-1)*x0(1:i-1)+a(i,i+1:n)*x0(i+1:n);
            x(i)=(b(i)-x(i))/a(i,i);
        end
        error=norm(x-xexact)/norm(u0-xexact);
        x0=x;
        iter = iter + 1;
     end
end
```

The matlab code of Jacobi iteration as following that

```
%
% Project 2
% Jacobi method
%
```

```
clear
results=[];
for k=0:5,

    n=10*2^k;
    h=1./(n+1);
    %
    % Define Grid
    %
    x=zeros(n,1);
    for i=1:n,
        x(i) = i*h;
    end
    %
    % Initialization
    %
    sol = sin(pi*x);
    e=ones(n,1)/h^2;
    a=spdiags([-e 2*e -e], -1:1, n, n);
    f = a *sol;

    %
    % Iteration starts here
    %
    tol = 1.e-4;

    u0 = zeros(n,1);
    u1 = zeros(n,1);
    error=norm(u0-sol)/norm(sol);

    iter = 0;
    while error > tol
        u1(1) = (h^2*f(1)+u0(2))/2.;
        for i=2:n-1
            u1(i) = (h^2*f(i)+u0(i+1)+u0(i-1))/2.;
        end
        u1(n) = (h^2*f(n)+u0(n-1))/2.;
```

```
        u0 = u1;

        error=norm(u0-sol)/norm(sol);
        iter = iter + 1;
     %   fprintf('Iteration %d, Error: %16.10g\n', iter, norm(u1-sol));
    end

    results=[results; n error iter];

    fprintf('Iterations for n=%d: %d\n', n, iter);
end
%%
loglog(results(:,1),results(:,3),'o-'),hold on,grid on
title({['Log-Log plot of the number of Jacobis
iterations'];['Due on 2017/4/20, Changjian Xie']})
xlabel('n');
ylabel('#the number of Jacobi Iterations')
```

Do a log-log plot of the number of Jacobi iterations as follows

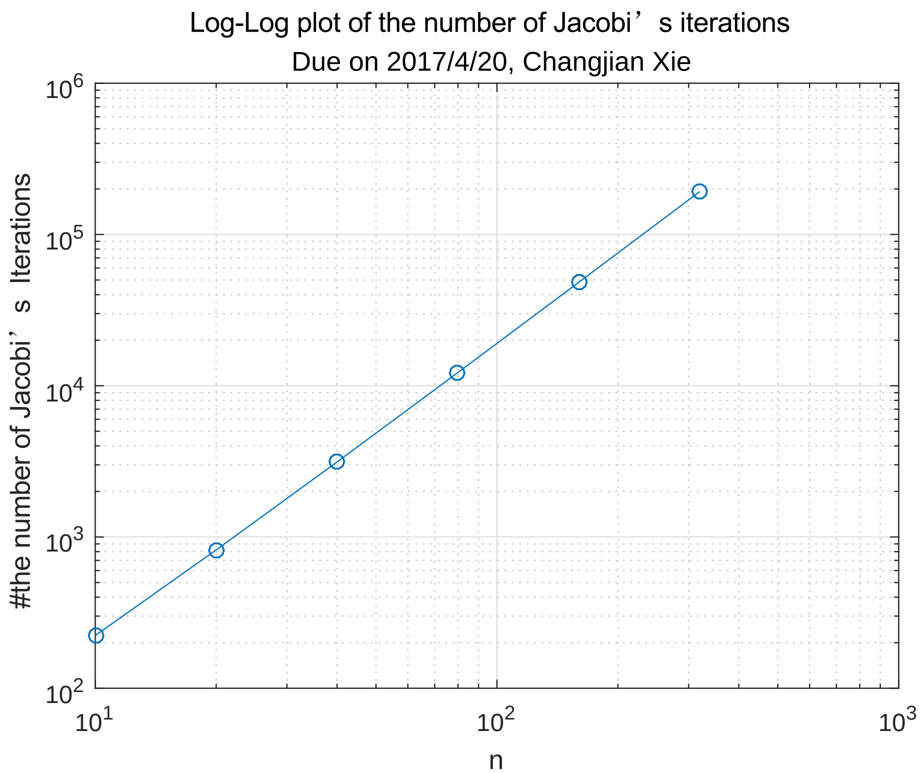

Figure 2: Using the values n = 10,20,40,80,160,320. Do a log-log plot of the number of Jacobi iterations necessary for the error to satisfy $\|\mathbf{x}^{(k)} - \mathbf{u}\|_2 \leq \epsilon\|\mathbf{u}\|$.

Theoretically this is the expected number of iterations. Indeed, we know the result is linear of $\log k$ and $\log n$. Then, Obviously, $k = O(n^\alpha)$, from task (5), of course, it's true.

7. Repeat the previous part with Gauss-Seidel's method.

G-S iterative algorithm
To solve $\mathbf{Ax} = \mathbf{f}$ given an initial approximation $\mathbf{x}^{(0)}$,

INPUT. The number of equations and unknowns n, the entries a_{ij}, $1 \leq i, j \leq n$ of the matrix $\mathbf{A}$, the entries f_i of $\mathbf{f}$, the entries XO_i, $1 \leq i \leq n$ of $\mathbf{XO} = \mathbf{x}^{(0)}$, tolerance TOL; maximum number of iterations N.

OUTPUT. The approximate solution $x_1, \ldots, x_n$ or a message that the number of iterations was exceeded.

Step 1. Set $k = 1$.

Step 2. While $k \leq N$ do Steps $3 - 6$.

Step 3. For $i = 1, \ldots, n$, set

$$x_i = \frac{-\sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^{n} a_{ij} XO_j + f_i}{a_{ii}}$$

Step 4. If $\|\mathbf{x} - \mathbf{XO}\| < TOL$, then OUTPUT $(x_1, \ldots, x_n)$; (The procedure was successful). STOP.

Step 5. Set $k = k + 1$.

Step 6. For $i = 1, \ldots, n$, set $XO_i = x_i$.

Step 7. OUTPUT (Maximum number of iterations exceeded); (The procedure was successful). STOP. The matlab code of G-S function file as following that

```
function [x, iter]=mygs(a,xexact,x0,tol)

    b=a*xexact;

    n=length(x0);
    x=x0;

    iter = 0;
    error=1;

    while error > tol
        for i=1:n
            x(i) = a(i,1:i-1)*x(1:i-1)+a(i,i+1:n)*x(i+1:n);
            x(i)=(b(i)-x(i))/a(i,i);
        end
        error=norm(x-xexact)/norm(x0-xexact);
        iter = iter + 1;
```

```
    end
end
```

The matlab code of G-S iteration as following that

```
%
% Project 2
% Gauss-Seidel method
%
clear
results=[];
for k=0:5,

    n=10*2^k;
    h=1./(n+1);
    %
    % Define Grid
    %
    x=zeros(n,1);
    for i=1:n,
        x(i) = i*h;
    end
    %
    % Initialization
    %
    sol = sin(pi*x);
    e=ones(n,1)/h^2;
    a=spdiags([-e 2*e -e], -1:1, n, n);
    f = a *sol;

    %
    % Iteration starts here
    %
    tol = 1.e-4;

    u = zeros(n,1);
    error=norm(u-sol)/norm(sol);
```

```
    iter = 0;
    while error > tol
        u(1) = (h^2*f(1)+u(2))/2.;
        for i=2:n-1
            u(i) = (h^2*f(i)+u(i+1)+u(i-1))/2.;
        end
        u(n) = (h^2*f(n)+u(n-1))/2.;


        error=norm(u-sol)/norm(sol);
        iter = iter + 1;
     %   fprintf('Iteration %d, Error: %16.10g\n', iter, norm(u1-sol));
    end

    results=[results; n error iter];

    fprintf('Iterations for n=%d: %d\n', n, iter);
end
%%
loglog(results(:,1),results(:,3),'o-'),hold on,grid on
title({['Log-Log plot of the number of Gauss-Seidels
iterations'];['Due on 2017/4/20, Changjian Xie']})
xlabel('n');
ylabel('#the number of Gauss-Seidels Iterations')


%% The same graph
clf
loglog(results(:,1),results(:,3),'o-'),hold on,grid on
xlabel('n');
ylabel('#the number of Iterations'),
title({['Log-Log plot of the number of
iterations'];['Due on 2017/4/20, Changjian Xie']}),
hold on
%% after the fisrt one
% note that the output result of J and G-s  is different
loglog(results(:,1),results(:,3),'s-.'),hold on,
legend('Jacobis  Iterations','Gauss-Seidels Iterations')
```

We obtain that

n	$iter\ k$
10	112
20	411
40	1568
80	6122
160	24189
320	96157

Do a log-log plot of the number of G-S iterations as follows

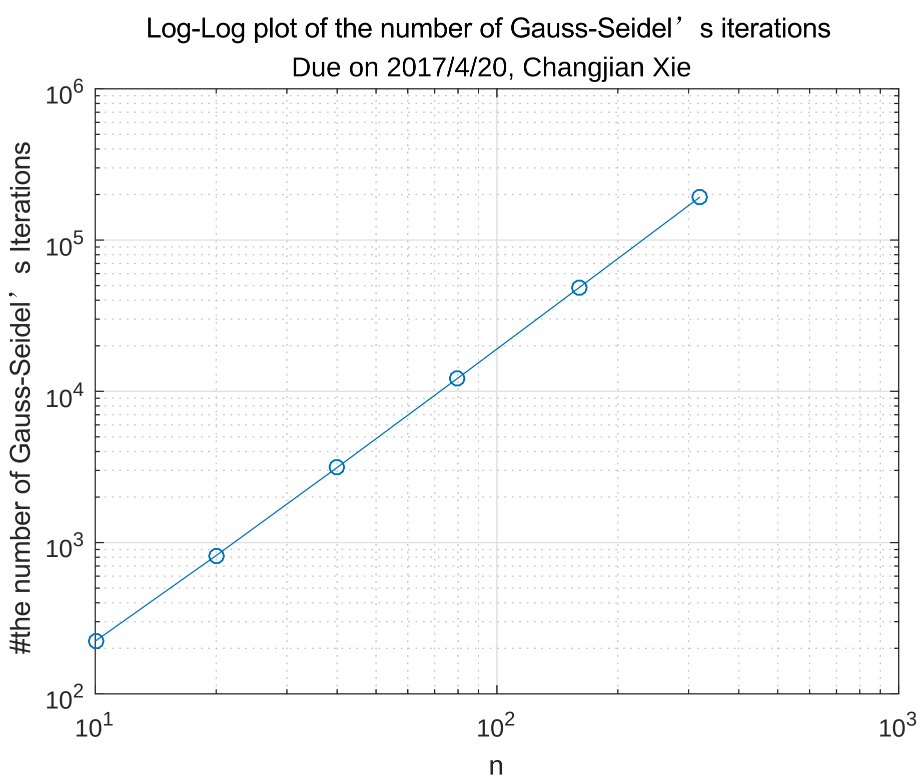

Figure 3: Using the values n = 10,20,40,80,160,320. Do a log-log plot of the number of G-S iterations necessary for the error to satisfy $\|\mathbf{x}^{(k)} - \mathbf{u}\|_2 \leq \epsilon\|\mathbf{u}\|$.

We can draw together as follows.

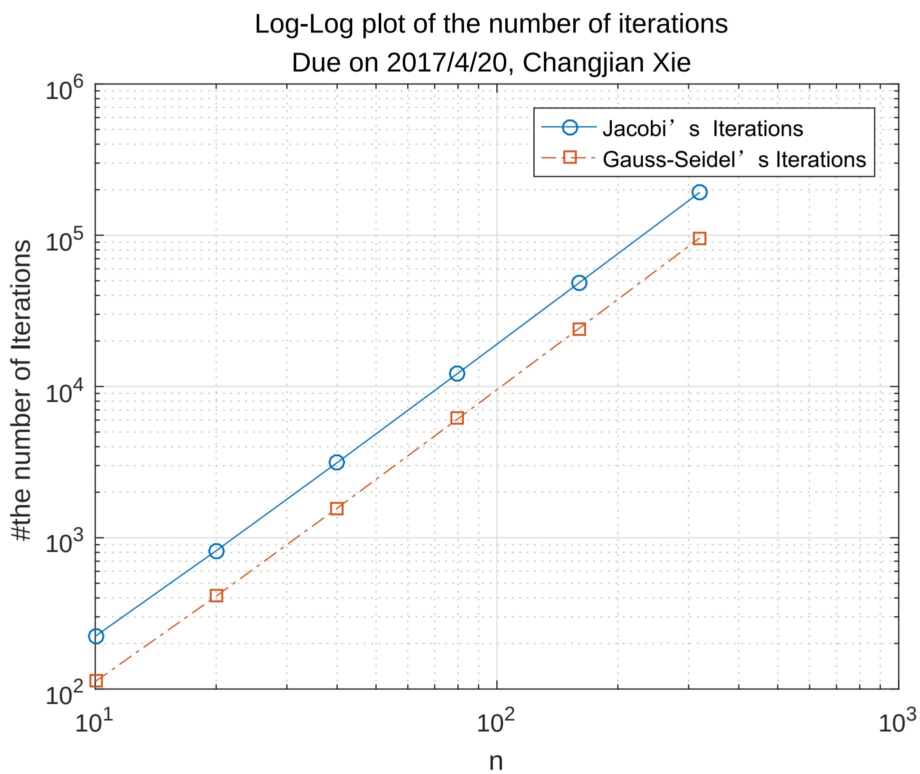

Figure 4: Using the values n = 10,20,40,80,160,320. Do a log-log plot of the number of Jacobi and G-S iterations necessary for the error to satisfy $\|\mathbf{x}^{(k)} - \mathbf{u}\|_2 \leq \epsilon\|\mathbf{u}\|$.

We can get the result from above graph that G-S iterations is a bit better than Jacobi iterations.