Assignment 2

Zou Yuan
Student No. 21960216

October 7, 2019

1 Jacobi Method

The problems to be solved in this section are below:

Use the Jacobi Method to solve the sparse system within three correct decimal places (forward error in the infinity norm) for $n = 100$. The correct solution is $[1, -1, 1, -1, \ldots, 1, -1]$. Report the number of steps needed and the backward error. The system is

$$\begin{bmatrix} 2 & 1 & & & \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \\ & & & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ \\ \vdots \\ \\ x_n \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}$$

The Matlab code is as follows:

```
n=100;
[a,b]=sparsesetup(n);
[x,cout]=jacobi(a,b)

function [a,b] = sparsesetup(n)
e = ones(n,1);
n2=n/2;
a = spdiags([e 2*e e],-1:1,n,n);
b=zeros(n,1);
```

```
b(1)=1;b(n)=-1;
end


function [x2,cout]=jacobi(B,c)

D=diag(diag(B));
U=D-triu(B);
L=D-(triu(B))';
x1=(rand(1,size(B,1)))';
cout=0;
while 1
cout=cout+1;
x2=inv(D)*(U+L)*x1+inv(D)*c;
t=ones(1,size(B,1));
t(2:2:size(B,1))=-1;
t=t';
t1=x2-x1;
t2=x2-t;
if (norm(t1,inf)<10e-3)&&(norm(t2,inf)<10e-3)
break
end
x1=x2;
end

end
```

The output is:

$$cout = 10896$$
$$back\_error = 0.0137$$

As shown above, the number of iterations is 10896 which varies slightly with the initial value, and the backward error is 0.0137.

2 Gauss-Seidel Method

This chapter uses the Gauss - Seidel method method and the SOR method to solve the problems in the previous chapter.

The Matlab code are below:

(a)

```
n=100;
[a,b]=sparsesetup(n);
x1=(rand(1,n))';
[x,cout]=seidel(a,b)
back_error=norm(b-a*x,inf)

function [a,b] = sparsesetup(n)
e = ones(n,1);
n2=n/2;
a = spdiags([e 2*e e],-1:1,n,n);
b=zeros(n,1);                                   r.h
    .s. b
b(1)=1;b(n)=-1;
end
function [x2,cout]=seidel(A,b)

D=diag(diag(A));
L = -tril(A,-1);%求A的下三角矩阵
U = -triu(A,1);%求A的上三角矩阵
G = (D-L)\U;
f = (D-L)\b;
x0=(rand(1,size(A,1)))';
x = G*x0+f;
cout=0;  %迭代次数
t=ones(1,size(A,1));
t(2:2:size(A,1))=-1;
t=t';
```

```
while 1
x0 = x;
x = G*x0+f;
cout = cout+1;
if (norm(x-x0,inf)<10e-3)&&(norm(x-t,inf)
    <10e-3)
break
end
x2=x;
end

end

n=100;
[a,b]=sparsesetup(n);
x1=(rand(1,n))';
[x,cout]=seidel(a,b)
back_error=norm(b-a*x,inf)

function [a,b] = sparsesetup(n)
e = ones(n,1);
n2=n/2;
a = spdiags([e 2*e e],-1:1,n,n);
b=zeros(n,1);
b(1)=1;b(n)=-1;
end
function [x2,cout]=seidel(A,b)

D=diag(diag(A));
L = -tril(A,-1);%求A的下三角矩阵
U = -triu(A,1);%求A的上三角矩阵
w=1.5;
G = (D-w*L)\((1-w)*D+w*U);
f = (D-w*L)\(w*b);
```

```
x0=(rand(1,size(A,1)))';
x = G*x0+f;
cout=0;  %迭代次数
t=ones(1,size(A,1));
t(2:2:size(A,1))=-1;
t=t';
while 1
x0 = x;
x = G*x0+f;
cout = cout+1;
if (norm(x-x0,inf)<10e-3)&&(norm(x-t,inf)
    <10e-3)
break
end
x2=x;
end

end
```

The Seidel method output is:

$$cout = 5031$$

$$back\_error = 9.6863e - 06$$

The SOR output is:

$$cout = 1659$$

$$back\_error = 9.7578e - 06$$

As shown above, These methods can converge faster and get Higher precision than Jacobi method.