Scientific Computing: HW10 Solution

June 6, 2021

Professor Lai

Xu Shengze 3190102721

Problem 1

(1)Strictly diagonal dominance:
Assume that A is irreversible, then $det(A) = 0$, so $AX = 0$ has a non-zero solution, suppose that $X = (x_1, x_2, \cdots, x_n)^T$, $|x_k| = max|x_i|$.
According to the condition, we have $\sum_{j=1}^{n} a_{kj}x_j = 0$, so $|a_{kk}||x_k| = |\sum_{j\neq k} a_{kj}x_j|$.
Meanwhile, A is Strict Diagonal Dominance Matrix, so $|a_{kk}||x_k| \geq |x_k|\sum_{j\neq k}|a_{kj}| > \sum_{j\neq k}|a_{kj}||x_j| \geq |\sum_{j\neq k} a_{kj}x_j|$. It contradicts, so A is reversible.
(2)Irreducible diagonal advantage:
Assume that A is irreversible, then $det(A) = 0$, so $AX = 0$ has a non-zero solution, suppose that $X = (x_1, x_2, \cdots, x_n)^T$, $K = \{k|\forall i, s.t.|x_k| \geq |x_i|, \exists j, s.t.|x_k| > |x_j|\}$.
If $K = \emptyset$, so $\forall i$, $|x_k| \geq |x_i|$ and $|x_k| \leq |x_i|$, then $|x_k| = |x_i|$, Therefore, we have $|x_1| = |x_2| = \cdots = |x_n|$.
For $\sum_{j=1}^{n} a_{kj}x_j = 0$, so $|a_{kk}x_k| = |\sum_{j\neq k} a_{kj}x_j|$, then $|a_{kk}||x_k| \leq \sum_{j\neq k}|a_{kj}x_j|$, but it contradicts with the fact that A is diagonally dominant. Therefore, $K \neq \emptyset$.
For $|a_{kk}||x_k| \leq \sum_{j\neq k}|a_{kj}x_j|$, we have $|a_{kk}| \leq \sum_{j\neq k}|a_{kj}|\frac{x_j}{x_k}$. Meanwhile we have $|a_{kk}| \geq \sum_{j\neq k}|a_{kj}|$, so $\sum_{j\neq k}|a_{kj}| \leq \sum_{j\neq k}|a_{kj}|\frac{x_j}{x_k}$. If $|x_j| < |x_k|$, we have $a_{kj} = 0$, but according to this condition, we can easily know that A is reducible. This contradicts the conditions of the title. So the assumption is wrong, A is reversible.

Problem 2

According to $A = D - L - L^T$, we have $B_{G-S} = (D - L)^{-1}L^T$.
Suppose λ is one of the eigenvalues of B_{G-S}, x is the corresponding feature vector, then we have $(D - L)^{-1}L^T x = \lambda x$, so $L^T x = \lambda(D - L)x$, so $x^T L^T x = \lambda x^T(D - L)x$.
A is positive definite matrix, so $p = x^T Dx > 0$, let $x^T L^T x = a$, then $x^T Ax = x^T(D - L - L^T)x = p - a - a = p - 2a > 0$.
Then we have $\lambda = \frac{x^T L^T x}{x^T(D-L)x} = \frac{a}{p-a}$, so $\lambda^2 = \frac{a^2}{(p-a)^2} = \frac{a^2}{p(p-2a)+a^2} < 1$.
Therefore, spectral radius $\rho(B_{G-S}) < 1$, we know that Gauss–Seidel method must converge.

Problem 3

Write the code of Jacobi iteration method.

```
function [X,Result]=Jacobi(A,b,X0,Norm,epsilon,Max)
        a=[];x=[];[N N]=size(A);X=X0;
        [L,D,U]=LUD(A);
        B=eye(N)-inv(D)*A;
        d=inv(D)*b;
        X1=A\b;
        Result=Ifconverge(B);
        for i=1:Max
                X=B*X+d;
                err=norm(X-X1,Norm);
                a(i)=err;
                x=i;
                if err<epsilon
                        return
                end
        end
```

```
end
```

Write the code of Gauss-Seidel iterative method.

```
function [X,Result]=Gauss_Seidel(A,b,X0,Norm,epsilon,Max)
        a=[];x=[];[N N]=size(A);X=X0;
        [L,D,U]=LUD(A);
        B=-inv(D+L)*U;
        d=inv(D+L)*b;
        X1=A\b;
        Result=Ifconverge(B);
        for i=1:Max
                X=B*X+d;
                err=norm(X-X1,Norm);
                a(i)=err;
                x=i;
                if err<epsilon
                        return
                end
        end
end
```

Write the function of matrix factorization.

```
function [L U D]=LUD(A)
        [n m]=size(A);
        L=zeros(size(A));
        U=zeros(size(A));
        D=zeros(size(A));
        for i=1:n-1
                L(i+1:end,i)=A(i+1:end,i);
                U(i,i)=A(i,i);
                D(i,i+1:end)=A(i,i+1:end);
        end
        U(n,n)=A(n,n);
end
```

Write a function to determine whether to converge.

```
function Result=Ifconverge(B)
        syms k;
        l=length(B);
        L=zeros(size(B));
        for i=1:l
                L(i)=limit(B(i)^k,k,inf);
        end
        if L==0
                Result=1;
        else
                Result=0;
```

```
        end
end
```

Substituting the question data, we get the corresponding results and list them in the following table.

	Jacobi		Gauss-Seidel	
Equations 1	[-0.9999999997087035, -3.999999999594850, -2.999999999534157]	Converge	[-0.9999999998404607, -3.999999999875405, -2.999999999928966]	Converge
Equations 2	[-0.666666666666666, -0.6666666666666666, 1.333333333333333]	Doesn't converge	[0.7499999999808467, 0.7500000005506776, 1.250000000265762]	Converge

Problem 4

First, write the corresponding function program according to the power method.

```
function [c,y]= Power(A,x0,eps,N)
        k=1;
        z=0;
        y=x0./max(abs(x0));
        x=A*y;
        xmax=max(x);
        if abs(z-xmax)<eps
                c=max(x);
                return;
        end
        while abs(z-xmax)>eps && k<N
                k=k+1;
                z=xmax;
                y=x./max(abs(x));
                x=A*y;
                xmax=max(x);
        end
        [m,i]=max(abs(x));
        c=x(i);
end
```

Next, use the function to find the unknown quantity required by the problem.

```
A=[1 -1 0
   -2 4 -2
   0 -1 1];
B=[2 -1 0
   -1 0 2
   1 1 3];
x0=[1;0;0];
eps=1e-9;
N=10000;
```

By running the program, we get the evaluated value as follow.
(1) $\lambda = 5$, $v = [0.25, -1, 0.25]^T$.
(2) $\lambda = 3.000299940011996$, $v = [-0.999400119976008, 0.9997000599880039, 1]^T$.

Problem 5

First, write the corresponding function program according to the inverse power method.

```
function [c,y]=Inversepower(A,x0,eps,N)
        k=1; r=0;
        y=x0./max(abs(x0));
        [L,U]=lu(A);
        z=L\y;
        x=U\z;
        xmax=max(x);
        c=1/xmax;
        if abs(xmax-r)<eps
                return
        end
        while abs(xmax-r)>eps && k<N
                k=k+1;
                r=xmax;
                y=x./max(abs(x));
                z=L\y;
                x=U\z;
                xmax=max(x);
        end
        [m,i]=max(abs(x));
        c=1/x(i);
end
```

Next, use the function to find the unknown quantity required by the problem.

```
format long e
A=[-4 14 0
   -5 13 0
   -1 0 2];
x0=[1;1;1];
eps=1e-9;
N=10000;
[eigenvalue,eigenvector]=Inversepower(A,x0,eps,N);
disp(eigenvalue);
disp(eigenvector);
```

By running the program, we get the evaluated value as follow.
$\lambda = 2.000000005576279$, $v = [-8.364418839284320 \times 10^{-9}, -4.182209419642158 \times 10^{-9}, 1]^T$.