Tsinghua University

Numberical Analysis

Fall 2024

Homework 6

Lin Zejin 2024 年 10 月 14 日

- **Collaborators:** I finish this homework by myself.

Problem 1.

$$
\begin{aligned}
\hat{r} - r &= Ax - A\hat{x} \\
&= A(A^TA)^{-1}A^Tb - A(A^TA+F)^{-1}A^Tb \\
&= A(A^TA+F)^{-1}\left((A^TA+F) - (A^TA)\right)(A^TA)^{-1}A^Tb \\
&= A(A^TA+F)^{-1}Fx
\end{aligned}
$$

We know from the lecture that

$$
\mathcal{K}_2(A)^2 = \mathrm{cond}_2(A^TA) = \frac{|\rho(A^TA)|}{|\sigma_n(A)|^2} = \frac{||A||_2^2}{|\sigma_n(A)|^2}
$$

Noticed that $||(A^TA)^{-1}||_2 = ||A^\dagger||_2^2 = \dfrac{1}{|\sigma_n(A)|^2} \le \dfrac{1}{2||F||_2}$, hence

$$
||(A^TA+F)^{-1}||_2 \le \frac{||(A^TA)^{-1}||_2}{1 - ||F||_2 \cdot ||(A^TA)^{-1}||_2} \le \frac{||A^\dagger||_2^2}{\frac{1}{2}}
$$

Therefore

$$
\begin{aligned}
||\hat{r} - \hat{r}||_2 &\le ||A||_2 \cdot ||(A^TA+F)^{-1}||_2 \cdot ||F||_2 \cdot ||x||_2 \\
&\le 2||A||_2 \cdot ||A^\dagger||_2^2 \cdot ||F||_2 \cdot ||x||_2 \\
&= 2\mathcal{K}_2(A)^2 \frac{||F||_2}{||A||_2}||x||_2
\end{aligned}
$$

Problem 2.

$$\begin{aligned}||x-\hat{x}||_2 &= ||(A^TA)^{-1}f||_2\\ &\le ||(A^TA)^{-1}||\cdot||f||_2\\ &\le ||A^\dagger||_2^2\cdot\epsilon||A||_2||b||_2\\ &= \epsilon\mathcal{K}_2(A)^2\frac{||A||_2||b||_2}{||A||_2^2}\\ &= \epsilon\mathcal{K}_2(A)^2\frac{||A||_2||b||_2}{||A^TA||_2}\end{aligned}$$

Then

$$\frac{||x-\hat{x}||_2}{||x||_2}\le\epsilon\mathcal{K}_2(A)^2\frac{||A||_2||b||_2}{||A^TA||_2||x||_2}\le\epsilon\mathcal{K}_2(A)^2\frac{||A||_2||b||_2}{||A^TAx||_2}=\epsilon\mathcal{K}_2(A)^2\frac{||A||_2||b||_2}{||A^Tb||_2}$$

Problem 3. We already have method tp compute A^TA, A^Tb, and we do Gauss elimination to it to get an equation:

$$U'x = b'$$

where U is upper triangular matrix, with

$$U' = \begin{bmatrix} U & B \\ 0 & 0 \end{bmatrix}, b' = \begin{bmatrix} \hat{b'} \\ 0 \end{bmatrix}$$

U is the upper triangular matrix with all diagonal elements 1.

Then solutions of $A^TAx = A^Tb$ are

$$x = \begin{bmatrix} U^{-1}(\hat{b'} - Bx') \\ x' \end{bmatrix} = \begin{bmatrix} U^{-1}\hat{b'} \\ 0 \end{bmatrix} - \begin{bmatrix} U^{-1}B \\ -I \end{bmatrix} x'$$

Therefore the solution of least norm is exactly the least square solution of

$$\begin{bmatrix} U^{-1}B \\ -I \end{bmatrix} x' = \begin{bmatrix} U^{-1}\hat{b'} \\ 0 \end{bmatrix}$$

which can be solved by the least square method.

Then we can find x such that the 2-norm is least among all least square solution of $Ax = b$.

Problem 4. For each step of the Givens transformation, we make one element in one row be 0.

The step between x, y should compute

$$t = \frac{y}{x},\ s = \mathrm{sgn}(x)(1+t^2)^{\frac{1}{2}}, c = st$$

So there is about 3 times of computation.

And in total, if we want to get an upper triangular matrix, it should be $3(n-1)$.

So the Complexity is $o(n)$.

Problem 5. (a)

$$A = \begin{bmatrix} \frac{\sqrt{2}}{2}\alpha + \frac{1}{2} & -\frac{\sqrt{2}}{2}\alpha & \frac{\sqrt{2}}{2}\alpha - \frac{1}{2} \\ \\ -\frac{\sqrt{6}}{6}\alpha + \frac{\sqrt{3}}{2} & \frac{\sqrt{6}}{6}\alpha & -\frac{-\sqrt{6}}{6}\alpha - \frac{\sqrt{3}}{2} \end{bmatrix} \tag{5.1}$$

$$A^{\dagger} = \begin{bmatrix} \frac{\alpha+\sqrt{2}}{4\alpha} & -\frac{\sqrt{2}}{4\alpha} & -\frac{\alpha-\sqrt{2}}{4\alpha} \\ \frac{3\sqrt{3}\alpha-\sqrt{6}}{12\alpha} & \frac{6}{12\alpha} & -\frac{3\sqrt{3}\alpha+\sqrt{6}}{12\alpha} \end{bmatrix} \tag{5.2}$$

Then the solution is

$$x = A^{\dagger}b = \begin{bmatrix} 1 \\ \sqrt{3} \end{bmatrix}$$

(b)

$$\mathcal{K}(A)^2 = \mathrm{cond}_2(A^TA) = \frac{|\lambda_1|}{|\lambda_n|} = \frac{\max(2, 2x^2)}{\min(2, 2x^2)}$$

since the eigen value of A^TA is $2x^2$ and 2

(3) The obtained data is in the appendix **A** and main code is in the

appendix **B**.

So easy to see that The Cholesky method is more efficient but have no accuracy. And also it causes wrong message when $x = 10^9$.

However, the QR method using Givens transformation is more accurate, actually, it is very accurate even if $x = 10^9$. And it also costs more time.

A Obtained Data

(a) $x = 10^5$

Cholesky method

ans =

1.4142e+05

d1 =

1.0e+05 *

2.0000 0.0000

cost 0.004769s

Gauss method

ans =

1.4142e+05

d1 =

1.0e+05 *

2.0000 0.0000

cost 0.001538s

QR Givens method

ans =

0.3660

d2 =

1.0000 1.7321

cost 0.002946 s

(b) $x = 10^7$

Cholesky method

ans =

1.4149e+07

d1 =

1.0e+07 *

2.0010 0.0000

cost 0.002820s

Gauss method

ans =

1.4149e+07

d1 =

1.0e+07 *

2.0010 0.0000

cost 0.000503 s

QR Givens method

ans =

0.3660

d2 =

1.0000 1.7321

cost 0.001068 s

(c) $x = 10^9$

Cholesky method

There has to be some problem

ans =

2.3094e+09

d1 =

1.0e+09 *

3.2660 0.0000

cost 0.003693 s

Gauss method

There has to be some problem

ans =

2.3094e+09

d1 =

1.0e+09 *

3.2660 0.0000

cost 0.000513 s

QR givens method

ans =

0.3660

d2 =

1.0000 1.7321

cost 0.001796 s

B Source Code

Listing 1: QR_Givens.m

```
function X=QR_Givens(A,b);
[row,col]=size(A);

now=[A,b];
for i=1:col
    for j=i+1:row
        if now(j,i)==0;
            continue
        end
        m=sqrt(now(i,i)^2+now(j,i)^2);
        c=now(i,i)/m;
        s=now(j,i)/m;
        tmp=now(i,:);
        now(i,:)=now(i,:)*c+s*now(j,:);
        now(j,:)=-tmp*s+c*now(j,:);
```

```
    end
    now(i,:)=now(i,:)/now(i,i);
end
X=solution(now);
end
```

Listing 2: Cholesky.m

```
%Cholesky method
function X=Cholesky(A,b);
now=[A,b];
[row,col]=size(A);
for j=1:row
    L(j,j)=A(j,j);
    for k=1:j-1
        L(j,j)=L(j,j)-L(j,k)*L(j,k);
    end
    L(j,j)=L(j,j)^0.5;
    for i=j+1:col
        L(i,j)=A(i,j);
        for k=1:j-1
            L(i,j)=L(i,j)-L(i,k)*L(j,k);
        end
        L(i,j)=L(i,j)/L(j,j);
    end
end
x=Gauss(L,b);
X=solution([L',x]);
end
```

Listing 3: Gauss.m

```
%Gauss elimination method
function[X]=Gauss(A,b);
    now=[A,b];
    [row,col]=size(A);
    for i=1:row
        if now(i,i)~=0
            for j=i+1:row
                t=now(j,i)/now(i,i);
                now(j,:)=now(j,:)-t*now(i,:);
            end
            now(i,:)=now(i,:)/now(i,i);

        else
        fprintf("There has to be some problem");
        end
    end
    [X]=solution(now)';
end
```

Listing 4: solution.m

```
%solve upper triangular matrix
function [X]=solution(T);
    [row,col]=size(T);
    for i=col-1:-1:1
        X(i)=T(i,col);
        for j=i+1:col-1
            X(i)=X(i)-X(j)*T(i,j);
        end
    end
```

```
end
```

The code to compute the least square solution.

Listing 5: least_square

```
%test file
q3=sqrt(3);
q2=sqrt(2);

A=[(q2*x)/2 + 1/2,   3^(1/2)/2 - (3^(1/2)*q2*x)/6
;      -(q2*x)/2,                    (3^(1/2)*q2*x)/6;
(q2*x)/2 - 1/2, - 3^(1/2)/2 - (3^(1/2)*q2*x)/6];



Ans=[1;sqrt(3)]


tic;

S=[         (3*x^2)/2 + 1/2, -(3^(1/2)*(x^2 - 1))/2;
-(3^(1/2)*(x^2 - 1))/2,              x^2/2 + 3/2]; %
    S=A^T*A
h=A'*b;
d1=Cholesky(S,h);
norm(d1-Ans)/norm(Ans)
d1
toc;

tic;
```

```

d1=Cholesky(S,A'*b);
norm(d1-Ans)/norm(Ans)
d1
toc;


tic;

d2=QR_Givens(A,b);
norm(d2-Ans)/norm(Ans)
d2
toc;
```

The method to compute $(a),(b)$

Listing 6: compute.m

```
q3=sqrt(3);
q2=sqrt(2);
syms x;

A=[(q2*x)/2 + 1/2,   3^(1/2)/2 - (3^(1/2)*q2*x)/6
;     -(q2*x)/2,                  (3^(1/2)*q2*x)/6;
(q2*x)/2 - 1/2, - 3^(1/2)/2 - (3^(1/2)*q2*x)/6];

S=[        (3*x^2)/2 + 1/2, -(3^(1/2)*(x^2 - 1))/2;
-(3^(1/2)*(x^2 - 1))/2,             x^2/2 + 3/2]; %
    S=A^T*A


```

```

h=A'*b;
Ans=simplify(S^(-1)*transpose(A))
Ans=Ans*b
```