Xiaowen Lin
xlin11
CS450

- 2.7 p97

(a) What is the determinant of A?

$det(A) = |A| = 1 \times 1 - (1+\epsilon) \times (1-\epsilon) = \epsilon^2$

(b) In floating-point arithmetic, for waht range of values of ϵ will the computed value of the determinant be zero?

For $\epsilon^2$to be computed as zero, there are two possibilities: one is that $1+\epsilon$ and $1-\epsilon$ are rounded to one; the other is that ϵ^2 is smaller than UFL.

SP:$\epsilon_{mach} = \frac{1}{2} \times 2^{1-24} = 2^{-24}$, $UFL = 2^{-126}$. So $\epsilon < 2^{-24}$ or $\epsilon < 2^{-63}$. So $\epsilon < 2^{-24}$.

DP:$\epsilon_{mach} = \frac{1}{2} \times 2^{1-53} = 2^{-53}$, $UFL = 2^{-1022}$. Similarly, $\epsilon < 2^{-53}$.

In both cases, $\epsilon < \epsilon_{mach}$ leads determinant to be zero.

(c) What is the LU factorization of A?

$L = \begin{bmatrix} 1 & 0 \\ \epsilon - 1 & 1 \end{bmatrix}$, $U = \begin{bmatrix} 1 & 1+\epsilon \\ 0 & \epsilon^2 \end{bmatrix}$

(d) In floating-point arithmetic, for what range of values of ϵ will the computed value of U be singular?

Whether U is singular or not depends on the $U(2,2)$ value, which is calculated by $(\epsilon - 1) \times (\epsilon + 1) + 1 = \epsilon^2$. It is the same as it in (b). $\epsilon < \epsilon_{mach}$ leads $U(2,2)$ to be zero, making U singular

- 2.26 p98

(a)If A is nonsigular, A^{-1}must exist.

$(I-uv^t)^{-1}(I-uv^t) = (I+u(1-v^tu)^{-1}v^t)(I-uv^t) = I + \frac{1}{1-v^tu}uv^t - uv^t - \frac{v^tu}{1-v^tu}uv^t = I$

So for any u, v that $v^tu \neq 1$, A^{-1} exist and A is nonsigular.

(b)According to (a), A^{-1} is equal to $I + \frac{1}{1-v^tu}uv^t$. So $\sigma = \frac{1}{v^tu-1}$.

(c)$M_k = 1 - me_k^t$,$e_k^t m = 0$, which is not equal to 1. So $M_k$is nonsingular.

$u = m, v = e_k, \sigma = \frac{1}{v^tu-1} = -1$

3.17 p150

$\alpha = \pm\|a\|_2 = \pm 2$

$$v = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} \alpha \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \alpha = -2$$

3.22 p151

(a) Show that $R^T R = LL^T$

$A^T A = \left[Q \begin{bmatrix} R \\ O \end{bmatrix} \right]^T Q \begin{bmatrix} R \\ O \end{bmatrix} = \begin{bmatrix} R^T & O \end{bmatrix} Q^T Q \begin{bmatrix} R \\ O \end{bmatrix} = \begin{bmatrix} R^T & O \end{bmatrix} \begin{bmatrix} R \\ O \end{bmatrix} = R^T R$

$A^T A = LL^T$

So $R^T R = LL^T$

(b) Can one conclude that $R = L^T$?

No. L has to be positive entidiagonal entries, while R doesn't.

For example, $R = -L^T$.

$A^T A = \begin{bmatrix} -L & O \end{bmatrix} Q^T Q \begin{bmatrix} -L^T \\ O \end{bmatrix} = LL^T$

The requirements still hold. But $R = -L^T$.

3.23 p151

$A = QR$

For this sparse matrix, use Givens QR factorization

$$\begin{bmatrix} c_1 & s_1 & 0 \\ -s_1 & c_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ \epsilon \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ 0 \\ 0 \end{bmatrix}$$

$\alpha_1 = \sqrt{1^2 + \epsilon^2} = 1$

$c_1 = \frac{1}{\sqrt{1^2+\epsilon^2}} = 1$

$s_1 = \frac{\epsilon}{\sqrt{1^2+\epsilon^2}} = \epsilon$

$$H_1 A = \begin{bmatrix} 1 & \epsilon & 0 \\ -\epsilon & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ \epsilon & 0 \\ 0 & \epsilon \end{bmatrix} = \begin{bmatrix} 1+\epsilon^2 & 1 \\ 0 & -\epsilon \\ 0 & \epsilon \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & -\epsilon \\ 0 & \epsilon \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_2 & s_2 \\ 0 & -s_2 & c_2 \end{bmatrix} \times \begin{bmatrix} 1 \\ -\epsilon \\ \epsilon \end{bmatrix} = \begin{bmatrix} 1 \\ \alpha_2 \\ 0 \end{bmatrix}$$

$\alpha_2 = \sqrt{\epsilon^2 + \epsilon^2} = \sqrt{2}\epsilon = 1.414\epsilon$

$c_2 = \frac{-\epsilon}{\sqrt{\epsilon^2+\epsilon^2}} = \frac{-1}{\sqrt{2}} = -0.707$

$s_2 = \frac{\epsilon}{\sqrt{\epsilon^2+\epsilon^2}} = \frac{1}{\sqrt{2}} = 0.707$

$$H_2 H_1 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.707 & 0.707 \\ 0 & -0.707 & -0.707 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & -\epsilon \\ 0 & \epsilon \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1.414\epsilon \\ 0 & 0 \end{bmatrix}$$

So R is not singular.

2.1 p100

(a) show the matrix is singular with exact arithmetic (approximating computer calculation is in the next page).

$$Ux = y, U = \begin{bmatrix} 0.7 & 0.8 & 0.9 \\ 0 & \frac{0.6}{7} & \frac{1.2}{7} \\ 0 & 0 & 0 \end{bmatrix}, y = \begin{bmatrix} \frac{1}{2} & \frac{1}{35} & 0 \end{bmatrix}^T$$

$$\begin{bmatrix} \frac{7}{10} & \frac{4}{5} & \frac{9}{10} \\ 0 & \frac{3}{35} & \frac{6}{35} \\ 0 & 0 & 0 \end{bmatrix} x = \begin{bmatrix} \frac{1}{2} & \frac{1}{35} & 0 \end{bmatrix}^T$$

(b) with exact arithmetic, at what point would this process fail?

After it replace A with $\begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0 & -0.3 & -0.6 \\ 0 & -0.6 & -1.2 \end{bmatrix}$ and try to eliminate row 3 with row 2. It becomes $\begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0 & -0.3 & -0.6 \\ 0 & 0 & 0 \end{bmatrix}$.The rank is lower than 3. It fails when it begine to do back-substitution for upper triangular system because the pivot (3,3) is zero.

(c) Next page.

```
>> cp02_01old
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND
=  2.590520e-17.
> In cp02_01old at 10
(c)So the solution from computer is
    x
    0.1615
    0.6771
   -0.1719

Give the set of solution from (a) the value of x from (c).
The left side is
5.000000000e-01
2.857142857e-01
0.000000000e+00
The right side is
5.000000000e+00
2.857142857e-01
0.000000000e+00
The solution fits into conclusion from (a) very well.
cond(A) is 2.111896834e+16
Computed solution has 16-log10(cond(A)) decimal digits of accuracy.
The result is -0.32.
It means we expect result to be no digits to trust.
>>
```

```
function cp02_01
A=[0.1 0.2 0.3; 0.4 0.5 0.6; 0.7 0.8 0.9];
B=[0.1 0.3 0.5]';

U1=[7/10 4/5 9/10; 0 6/7 12/7; 0 0 0];
y1=[5 2/7 0];

[L,U,P] = lu(A);
y=linsolve(L, P*B);
x=linsolve(U, y);
disp('(c)So the solution from computer is');
disp('    x')
disp(x);
disp('Give the set of solution from (a) the value of x from (c).')
disp('The left side is ')
fprintf('%13.9e \n', U1*x);
disp('The right side is ')
fprintf('%13.9e \n', y1);
disp('The solution fits into conclusion from (a) very well. ')
fprintf('cond(A) is %13.9e \n', cond(A));
fprintf('Computed solution has 16-log10(cond(A)) decimal digits of accuracy. \n')
fprintf('The result is %3.2f. \n', 16-log10(cond(A)));
fprintf('It means we expect result to be no digits to trust. \n')
```

```
>> cp02_09
(a)
    x0
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000
    0.9999    1.0000
    0.9992    1.0000
    2.2204    1.0000
         0    1.0000
         0    1.0000

It can been seen from the data that as k increases, x is more and more far from [1 1]T.
(b)
    x1
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000
    1.0000    1.0000

After one iteration of iterative refinement, the accuracy is more proved.
>>
```

```
function cp02_09

M=zeros(2,10);
M1=zeros(2,10);
for k=1:10,
    X=[1 1]';
    A=[10^(-2*k) 1; 1 1];
    b=[1+10^(-2*k) 2]'; B=b;

    [n,n]=size(A);
    L=eye(n); U=A;
    for i=1:n
        for j=i+1:n
            if U(j,j)==0
                continue;
            else
                L(j,i)=U(j,i)/U(i,i);
                U(j,:)=U(j,:)-L(j,i)*U(i,:);
            end
        end
    end

    %Ly=b
    y=zeros(n,1);
    for j=1:n
        if (L(j,j)==0)
            fprintf('During forward substitution for lower triangular systems, at the row
of %d, the process fails. \n',j);
            error('Matrix is singular!');
            disp(j);
        end;
        y(j)=b(j)/L(j,j);
        b(j+1:n)=b(j+1:n)-L(j+1:n,j)*y(j);
    end

    x=zeros(n,1);
    for l=n:-1:1
        if (U(l,l)==0)
            fprintf('During back substitution for upper triangular systems, at the row of
%d, the process fails. \n',j);
            error('Matrix is singular!');
        end;
        x(l)=y(l)/U(l,l);
        y(1:l-1)=y(1:l-1)-U(1:l-1,l)*x(l);
    end


    r0=B-A*x;

    %Ly=b
    y0=zeros(n,1);
```

```
    for m=1:n
        if (L(m,m)==0)
            fprintf('During forward substitution for lower triangular systems, at the row
of %d, the process fails. \n',j);
            error('Matrix is singular!');
            disp(m);
        end;
        y0(m)=r0(m)/L(m,m);
        r0(m+1:n)=r0(m+1:n)-L(m+1:n,m)*y0(m);
    end

    s0=zeros(n,1);
    for h=n:-1:1
        if (U(h,h)==0)
            fprintf('During back substitution for upper triangular systems, at the row of
%d, the process fails. \n',j);
            error('Matrix is singular!');
        end;
        s0(h)=y0(h)/U(h,h);
        y0(1:h-1)=y0(1:h-1)-U(1:h-1,h)*s0(h);
    end

    x1=x+s0;
    M(:,k)=x;
    M1(:,k)=x1;
end
disp('(a)');
disp('    x0');
disp(M');
disp('It can been seen from the data that as k increases, x is more and more far from [1
1]T. ');
disp('(b)')
disp('    x1');
disp(M1');
disp('After one iteration of iterative refinement, the accuracy is more proved. ');
```

```
>> cp03_04
(a)Solution of the system is
    1.0000
    1.0000

(b)Solution of the system is
    7.0089
   -8.3957

(c)According to the textbook, norm(dX,2)/norm(x,2)<cond(A)/cos(theta)*norm(dB,2)/norm(B,2).
Assign Boundary=cond(A)/cos(theta)*norm(dB,2)/norm(B,2).

For (a), dB is about transpose(e e e), where e is the rounding error of B in binary computer.
The norm(dB,2) about  1.922963e-16
And the resulting Boundary is about  6.334147e-14

For (b), dB is norm(b1-b,2).
The norm(b1-b,2) is  3.741657e-02
And the resulting Boundary is about 12.325.

cond(A) is unchanged. cos(theta) of (a) is  1.000000e+00 and cos(theta) of (b) is 9.999791e-01 with a perturbation rate of -2.094311e-05.
So the major change is introduced by a slight perturbance of B.
>>
```

```
function cp03_04
A=[0.16 0.10; 0.17 0.11; 2.02 1.29];
%(a)
b=[0.26 0.28 3.31]';
x=linsolve(A, b);
disp('(a)Solution of the system is');
disp(x);


%(b)
b1=[0.27 0.25 3.33]';
x1=linsolve(A, b1);
disp('(b)Solution of the system is');
disp(x1);


%(c)
cos=norm(A*x,2)/norm(b,2);
cos1=norm(A*x1,2)/norm(b1,2);
(cos1-cos)/cos;
dNorm=norm([2^(-53) 2^(-53) 2^(-53)]', 2);
Norm=norm(b,2);

Ratio_b=norm(b1-b,2)/norm(b,2);
disp('(c)According to the textbook, norm(dX,2)/norm(x,2)<cond(A)/cos(theta)*norm(dB,2)/norm(B,2).');
disp('Assign Boundary=cond(A)/cos(theta)*norm(dB,2)/norm(B,2).');
fprintf('\n');
%fprintf('cond(A) is %13.6e. \n', cond(A));
%fprintf('cos(theta) is %13.6e for (a). And cos(theta) is %13.6e for (b) with a perturbation rate of %13.6e.\n', cos, cos1, (cos1-cos)/cos);
fprintf('For (a), dB is about transpose(e e e), where e is the rounding error of B in binary computer. \n');
fprintf('The norm(dB,2) about %13.6e \n', dNorm);
fprintf('And the resulting Boundary is about %13.6e \n', cond(A)/cos*dNorm/Norm);
fprintf('\n');
fprintf('For (b), dB is norm(b1-b,2). \n');
fprintf('The norm(b1-b,2) is %13.6e \n', norm(b1-b,2));
fprintf('And the resulting Boundary is about %5.3f. \n', cond(A)/cos1*Ratio_b);
fprintf('\n');
fprintf('cond(A) is unchanged. cos(theta) of (a) is %13.6e and cos(theta) of (b) is %13.6e with a perturbation rate of %13.6e. \n', cos, cos1, (cos1-cos)/cos);
disp('So the major change is introduced by a slight perturbance of B.')
```