HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

CHAPTER 2: SYSTEMS OF LINEAR EQUATIONS

Vu Van Thieu, Dinh Viet Sang, Nguyen Khanh Phuong

SCIENTIFIC COMPUTING

Introduction

System of Linear Equations

Definition

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$
$$\ldots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m$$

Denote
$A = (a_{ij})$ where $i = 1, \cdots, m$ and $j = 1, \cdots, n$ is the coefficient matrix A.
$b = (b_1, b_2, \cdots, b_m)^T$ is the right-hand side vector.
$x = (x_1, x_2, \cdots, x_n)^T$ is the variable vector.
We can rewrite the system of linear equations in matrix form

$$Ax = b$$

System of Linear Equations

Example 1 :

Consider a system of linear equations with

- Coefficient Matrix $A = \begin{pmatrix} 3 & 2 \\ 1 & -1 \end{pmatrix}$
- The right-hand side vector is $b = \begin{pmatrix} -first \\ first \end{pmatrix}$

then the system has a unique solution $x = \begin{pmatrix} 0.2 \\ -0.8 \end{pmatrix}$

System of Linear Equations

Example 2 :

Consider a system of linear equations with

- Coefficient Matrix $A = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & -5 \end{pmatrix}$
- The right-hand side vector is $b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

then the system has infinitely many solutions $x = \begin{pmatrix} 1-3t \\ 2+5t \\ t \end{pmatrix}$ for every $t \in \mathbb{R}$.

System of Linear Equations

Example 3 :

Consider a system of linear equations with

- Coefficient Matrix $A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 3 & 4 \end{pmatrix}$
- The right-hand side vector is $b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

then the system has no solution.

System of Linear Equations

For a system of possible linear equations

- $m = n$: square system (number of equations equals the unknowns, often with a unique solution)
- $m < n$: missing system (the number of equations is less than the number of unknowns, the system is usually infinitely many solutions)
- $m > n$: residual system (number of pequations more than unknowns, the system usually has no solution)

Solve a system of linear equations

System of square equations

$$Ax = b$$

where $A \in \mathbb{R}^{n \times n}$ and x and b are vectors $\in \mathbb{R}^n$

Solve the system of square equations

- If the matrix A is not singular, the only solution to the equation is

$$x = A^{-1}b$$

Matlab

» **x=inv(A)*b**

Solve a system of linear equations

Example 4: :

Solve the system of equations $A = (7)$ and $b = (21)$ or equation

$$7x = 21$$

- Method 1: Solve the division directly $x = 21/7 = 3$
- Method 2: Invert 7^{-1} and multiply by 21 will result in

$$x = 7^{-1} \times 21 = 0.142857 \times 21 = 2.99997$$

Obviously, method 1 is better than method 2, in addition, method 2 has a larger computational load when determining the inverse 7^{-1}.

Solve a system of linear equations

Comment

The use of the inverse matrix gives a less precise solution.
When solving a system of linear equations, we often find the solution directly and only use the inverse matrix A^{-1} in a few situations.

Some methods:

- LU Factorization (LU Factorization)
- Cholesky Analysis (Cholesky Factorization)
- QR Decomposition

Solve a system of linear equations

Matrix division operator in Matlab

If A is any matrix and B is a matrix with the same number of rows as A, then the solution of the system of equations

$$AX = B$$

then we use *left division* $X = A \backslash B$.
And the solution of the system of equations

$$XA = B$$

then we use *right division* $X = B/A$.

Solve a system of linear equations

Example 5: Split left

» A=[3 2;1 -1];b=[-1;1];
» x = A\b;
x = 0.2000
-0.8000

Example 6: Right split

» AA=A';bb=b';
» xx = bb/AA;
xx = 0.2000 -0.8000

Solve a system of linear equations

Basic quantities when solving systems of square equations

- **Determinant** is an important numerical feature of square matrices that allows the determination of the number of solutions of HPT (zero or infinitely many solutions, or unique solutions).
- **Trace** is the sum of the main diagonal elements.
- **Rank** is the maximum number of linearly independent rows or columns of the matrix.

Matlab

» **D=det(A)**
» **T=trace(A)**
» **R=rank(A)**

Solve a system of linear equations

Kronecker-Capelli theorem

The system of linear equations $Ax = b$ has a solution if and only if

$$rank(A) = rank(Ab)$$

Example 7 : same rank

```
» A=[1 2 3; 4 5 6; 8 10 12];
» b=[5;6;12];
» rA=rank(A);
» rAb=rank([Ab])
rA = 2
rAb = 2
```

Solve a system of square linear equations

Instances when solving $Ax = b$

- The system of equations has a unique solution if $det(A) \neq 0$.
- When $det(A) = 0$ the system of equations can have infinitely many solutions or no solutions (We can apply the Kronecker-Capelli theorem to determine whether it has no solution or infinitely many solutions).
- When $det(A) \neq 0$, there exists an inverse matrix of A and A which is called a non-degenerate matrix.
- When $det(A) = 0$ then the inverse matrix A^{-1} does not exist and A is called degenerate matrix.

Solve a system of square linear equations

Example 8 :

```
» A1=[-1 1; -2 2];b1=[1 ; 0];
% system has no solution
» x1 = A1\b1, D1 = det(A1)
Warning : Matrix is singular to working precision.
x1 = Inf
     Inf
D1=0
```

Solve a system of square linear equations

Example 9 :

```
» A2=[-1 1; -2 2];b2=[1 ; 2];
% system has infinitely many solutions
» x2 = A2\b2
Warning : Matrix is singular to working precision.
x2 = -1
      0
```

Solve a system of square linear equations

Solve a system of square linear equations

3-dimensional example

Given the following system of 3rd degree equations:

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \\ 6 \end{pmatrix}$$

We rewrite it as a system of linear equations

$$\begin{aligned} 10x_1 - 7x_2 &= 7 \\ -3x_1 + 2x_2 + 6x_3 &= 4 \\ 5x_1 - x_2 + 5x_3 &= 6 \end{aligned}$$

Solve a system of square linear equations

3-dimensional example (continued)

System of linear equations

$$\begin{aligned} 10x_1 - 7x_2 &= 7 && (1) \\ -3x_1 + 2x_2 + 6x_3 &= 4 && (2) \\ 5x_1 - x_2 + 5x_3 &= 6 && (3) \end{aligned}$$

we proceed to solve

- Remove $x_1 \Rightarrow (2) - (1) \times (-0.3)$ and $(3) - (1) \times 0.5$

The factor of 10 of the hidden x_1 in (1) is called **pivot element**, the coefficients -0.3 and 0.5 are called **factor**.

Solve a system of square linear equations

3-dimensional example (continued)

System of linear equations after eliminating x_1

$$\begin{aligned} 10x_1 - 7x_2 \qquad &= 7 && (4) \\ -0.1x_2 + 6x_3 &= 6.1 && (5) \\ 2.5x_2 + 5x_3 &= 2.5 && (6) \end{aligned}$$

we continue to solve

- Remove $x_2 \Rightarrow$ because the pivot element of x_2 in (5) is -0.1 has a small absolute value, we proceed to swap the two equations (5) and (6) and then proceed to eliminate x_2.

This is called **rotation**

Solve a system of square linear equations

3-dimensional example (continued)

System of linear equations after performing the rotation

$$
\begin{aligned}
10x_1 - 7x_2 \qquad &= 7 \qquad (7)\\
2.5x_2 + 5x_3 &= 2.5 \qquad (8)\\
-0.1x_2 + 6x_3 &= 6.1 \qquad (9)
\end{aligned}
$$

we continue to solve

- Remove $x_2 \Rightarrow (9) - (8) \times (-0.04)$

Solve a system of square linear equations

3-dimensional example (continued)

System of linear equations after performing the elimination x_2

$$
\begin{aligned}
10x_1 - 7x_2 \quad\quad &= 7 && (10)\\
2.5x_2 + 5x_3 &= 2.5 && (11)\\
6.2x_3 &= 6.2 && (12)
\end{aligned}
$$

we continue to solve

- From equation (12) $\Rightarrow x_3 = 1$.
- replace x_3 in (11) then $2.5x_2 + 5 \times (1) = 2.5 \Rightarrow x_2 = -1$.
- replace x_2 in (10) then $10x_1 - 7 \times (-1) = 7 \Rightarrow x_1 = 0$.

Solve a system of square linear equations

Matrixes L,U,P

The entire solution just presented can be encapsulated in the following matrices

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & -0.04 & 1 \end{pmatrix}, U = \begin{pmatrix} 10 & -7 & 0 \\ 0 & 2.5 & 5 \\ 0 & 0 & 6.2 \end{pmatrix}, P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

With

- L is the matrix containing the factors
- U is the final coefficient matrix
- P is the permutation matrix describing the rotation

then we have

$$LU = PA$$

Solve a system of square linear equations

Solve a system of square linear equations

Permutation matrix

The permutation matrix is obtained from the unit matrix I by permuting its rows.

- With permutation matrix : $P^{-1} = P^T$
- Multiplication PX to permute rows of matrix X
- Multiplication XP to permute matrix columns X

Example 10

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Solve a system of square linear equations

Example 10 (continued)

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Short form in Matlab

$$p = [2 \;\; 4 \;\; 3 \;\; 1]$$

Solve a system of square linear equations

Matrix triangle

The matrix $X \in \mathbb{R}^{n\times n}$ is **upper triangular matrix** if $x_{ij} = 0 \in X \quad \forall i < j$ means of the form

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ 0 & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_{nn} \end{pmatrix}$$

- When this matrix has major diagonal elements $x_{ii} = 1 \;\; \forall i = 1, \cdots, n$ is called **triangular matrix per unit**.
- The determinant of an upper triangular matrix is non-zero if and only if all the elements lie on the main diagonal are different from zero.

$\Rightarrow$ Similar definition we have **lower triangle matrix** and **subunit triangle matrix**.

Solve a system of square linear equations

Matrix triangle

The system of equations with the matrix of triangular coefficients can be solved easily. Start solving the last row equation to find the last unknown; then alternately substitute the above equations to find the remaining unknowns.

Example 11 :

Solve a system of triangle equations on $Ux = b$

```
»x = zeros(n,1);
for k = n:-1:1
    x(k) = b(k)/U(k,k);
    i=(1:k-1)';
    b(i) = b(i) - x(k) * U(i,k);
end
```

Solve a system of square linear equations

Example 12 :

Solving a system of linear equations

$$\begin{aligned} 3x_1 + 4x_2 + 5x_3 &= 7 \\ 2x_2 - 3x_3 &= 8 \\ 5x_3 &= 11 \end{aligned}$$

```
% Matlab Programs
» U=[3,4,5;0.2,-3;0,0,5]; b = [7;8;11];n=3;x=zeros(n,1);
» for k=n:-1:1
»       x(k) = b(k)/U(k,k);
»       i=(1:k-1)';
»       b(i) = b(i) - x(k) * U(i,k);
» end;
```

Solve a system of square linear equations

Analyze LU

The common algorithm used to solve a system of square linear equations has two stages

- Forward elimination is the transformation of the square matrix into the above triangular form used to eliminate each unknown, with compatible factors and pivot elements combined with rotation.
  - consists of $n-1$ steps
  - at step $k = 1, \cdots, n-1$ multiply the equation k by the factor and then subtract the remaining equations to de-anonymize the number x_k.
  - If the coefficients of x_k are small then we should swap the equations.
- Backward subtitution solves the last row equation to find the last hidden, and then reverses the upper rows to find the remaining unknowns. (see example 12)

Solve a system of square linear equations

LU analysis (continued)

- Let P_k be the permutation matrices at steps $k = 1, \cdots, n-1$
- Let M_k be the subunit triangular matrices obtained by inserting the "addition" factors used in the k step below the diagonal position of the k column of the unit matrix .
- Let U be the final upper triangular matrix obtained at the end of the forward elimination phase.

The reduction process is rewritten in matrix form as follows

$$U = M_{n-1}P_{n-1}\cdots M_1P_1A$$

Solve a system of square linear equations

LU analysis (continued)

The equation can be rewritten as:

$$L_1 L_2 \cdots L_{n-1} U = P_{n-1} \cdots P_1 A$$

where L_k is obtained from M_k by permutation and sign of the factors below the diagonal. So if we put

$$L = L_1 L_2 \cdots L_{n-1}$$
$$P = P_{n-1} \cdots P_2 P_1$$

then we get the final formula

$$LU = PA$$

Solve a system of square linear equations

Example 12 :

Going back to the first 3-dimensional example, we have

$A = \begin{pmatrix} 10 & -7 & 0 \\ -3 & 2 & 6 \\ 5 & -1 & 5 \end{pmatrix}$ then the matrices determined in the forward elimination process are

$$P_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, M_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0.3 & 1 & 0 \\ -0.5 & 0 & 1 \end{pmatrix}$$

$$P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, M_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.04 & 1 \end{pmatrix}$$

Solve a system of square linear equations

Example 12 (continued):

The matrices L_1,L_2 are respectively

$$L_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ -0.3 & 0 & 1 \end{pmatrix}, L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.04 & 1 \end{pmatrix}$$

Note :

When calculating the reduction phase, we will calculate directly on the rows of the matrix, not perform the matrix multiplication as above.

Analyze LU

- The relation $LU = PA$ just described is called *LU analysis* or *triangular decomposition* of the matrix A.

Solve a system of square linear equations

Solve the system of equations by LU analysis

With the system of equations

$$Ax = b$$

where the matrix A is non-degenerate and $PA = LU$ is the LU analysis of A, the system of equations can be solved in two steps.

- **Forward elimination** Solve the system

  $$Ly = Pb$$

  to find y, since L is a lower unit matrix, y can be found with a forward elimination (from top to bottom).
- **Backward substitution** Solve the system

  $$Ux = y$$

  by the backward substitution method to get x.

Role of the pivot element

pivot element

- The elements that lie on the main diagonal of the matrix U.
- The k-th pivot element is the coefficient of the hidden x_k in the k-th equation at the k step of the reduction phase.
- Both the forward elimination and backward substitution steps need to be divided by the pivot element, so they cannot be zero.

Intuition :

The system of equations solves badly if the pivot element is close to zero.

Role of the pivot element

Example 13 :

Slight change in the second row of the above examples

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & 2.099 & 6 \\ 5 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 3,901 \\ 6 \end{pmatrix}$$

Thus, suppose all calculations are performed on a hypothetical computer equipped with arithmetic operations with 5-digit floating point real numbers.

- Coefficient x_2 in row two changed from 2,000 to 2,099
- Also the corresponding right side changed from 4,000 to 3,901

the goal is to keep the solution $(0, -1, 1)^T$ of the system of equations.

Role of the pivot element

Example 13 (continued) :

The first step of the elimination phase

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.001 & 6 \\ 0 & 2.5 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.001 \\ 2.5 \end{pmatrix}$$

continue to perform elimination even though the pivot element -0.001 is small compared to other coefficients of the matrix without performing the rotation. So we

- Multiply the second row equation by 2.5×10^3 and then add the third row.
- On the right hand side of this equation, multiplying $6,001$ by 2.5×10^3 results in $1,50025 \times 10^4$ rounded to $1,5002 \times 10^4$

Role of the pivot element

Example 13 (continued) :

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.001 & 6 \\ 0 & 0 & 1,5005 \times 10^4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.001 \\ 1,5004 \times 10^4 \end{pmatrix}$$

continue ...

- The result on the right hand side of the second equation is rounded $1,5002 \times 10^4$ is added to the 2.5 which is the right hand side of the third equation and is rounded again.

So equation three becomes $1,5005 \times 10^4 x_3 = 1,5004 \times 10^4$ solving we have

$$x_3 = \frac{1.5004 \times 10^4}{1.5005 \times 10^4} = 0.99993$$

Obviously, with the exact value of the unknown $x_3 = 1$, the value solved by this equation is not worrisome.

Role of the pivot element

Example 13 (continued) :

$$\begin{pmatrix} 10 & -7 & 0 \\ 0 & -0.001 & 6 \\ 0 & 0 & 1,5005 \times 10^4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.001 \\ 1,5004 \times 10^4 \end{pmatrix}$$

continue ...

- for unknown x_2

$$-0.001x_2 + 6 \times (0.99993) = 6.001$$

Candlestick $x_2 = \frac{1.5 \times 10^{-3}}{-1.0 \times 10^{-3}} = -1.5$

- Finally substitute the first equation to find the hidden x_1

$$10x_1 - 7 \times (-1.5) = 7$$

deduce $x_1 = -3.5$

Role of the pivot element

Example 13 (continued) :

Thus, when not performing rotation, select the pivot element

$$\begin{pmatrix} 10 & -7 & 0 \\ -3 & -0.001 & 6 \\ 5 & 2.5 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6.001 \\ 2.5 \end{pmatrix}$$

instead of the solution $(0, -1, 1)^T$ we get the solution $(-0.35, -1.5, 0.99993)^T$.

Why is this problem?

The error is because we choose **the pivot element is too small**. So we should choose the pivot element with the largest absolute value at each k-th elimination step.

Effect of rounding error

How to measure difference

Normally, when we get a solution x^* that is different from the correct solution x, we often use two ways of measuring the difference:

- Error : $e = x - x^*$
- Offset : $r = b - Ax^*$

Theoretically, if A is not degenerate, then these two quantities are equal to zero, but when calculating in the computer these two quantities are not in sync.

Effect of rounding error

Example 14 :

Consider the system of equations

$$
\begin{aligned}
0.780x_1 + 0.563x_2 &= 0.217 \\
0.913x_1 + 0.659x_2 &= 0.254
\end{aligned}
$$

Gaussian elimination like the previous example, apply the rule of choosing the largest pivot element, but all calculations are accurate to 3 digits after the decimal point.

Effect of rounding error

Example 14 (continued):

continue ....

- Perform the rotation, so that 0.913 becomes the pivot element.

$$0.913x_1 + 0.659x_2 = 0.254$$
$$0.780x_1 + 0.563x_2 = 0.217$$

Calculate the coefficient $0.780/0.913 = 0.854$

- Multiply the factor 0.854 by the first equation and then subtract the second equation. We have:

$$0.913x_1 + 0.659x_2 = 0.254$$
$$0.001x_2 = 0.001$$

Effect of rounding error

Example 14 (continued):

$$\begin{aligned} 0.913x_1 + 0.659x_2 &= 0.254 \\ 0.001x_2 &= 0.001 \end{aligned}$$

continue ....

- Hide $x_2 = 0.001/0.001 =$1,000 (exactly)
- Substitute for the above pt, $x_1 = (0.254 - 0.659x_2)/0.913 = -0.443$

Finally we get the solution $x^* = (-0.443, 1, 000)^T$

Effect of rounding error

Example 14 (continued):

Measuring the difference, it is clear that the true solution of the system $x = (1, -1)^T$

- Error : $e = x - x^* = (1,433, -2)^T$
- Deviation :

$$r = b - Ax^* = \begin{pmatrix} 0.217 - (0.780(-0.443) + 0.563(1,000)) \\ 0.254 - (0.913(-0.443) + 0.659(1,000)) \end{pmatrix}$$

$$= \begin{pmatrix} -0.000460 \\ -0.000541 \end{pmatrix}$$

Obviously, while the deviation is acceptable when we round to three decimal places, the error is even larger than the solution.

Effect of rounding error

Questions for rounding errors

- Why is the offset so small?
- Why is the error so large?
- The determinant of the system $0.780 \times 0.659 - 0.913 \times 0.563 = 10^{-6}$ is not near the cause of this phenomenon?

Effect of rounding error

Example 14 (continued):

Replacing the assumption of rounding with 3 decimals to rounding with 6 decimals after the decimal point, we get a system of equations after eliminating Gauss

$$\begin{pmatrix} 0.913000 & 0.659000 \\ 0.000000 & 0.000001 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.254000 \\ -0.0000001 \end{pmatrix}$$

Notice, the value of the right hand side of the second equation has changed. In fact **the approximate solution is also the exact solution** of the system

$$x_1 = \frac{-0.00001}{0.00001} = -1.000000$$
$$x_2 = \frac{0.254 - 0.659x_2}{0.913} = 1.000000$$

Effect of rounding error

Explain why the deviation is small

- The small deviation of the two equations due to the near degenerate matrix $det(A) = 10^{-6}$ leads to the two equations being almost linearly dependent.
- So the hidden pair (x_1, x_2) satisfying the first equation also satisfies the second equation

$\Rightarrow$ If we know for sure that the determinant is zero, we don't need to worry about the second equation because every solution of the first system of equations satisfies the second system of equations.

Important conclusion :

When we perform Gaussian elimination with the maximum pivot element on the column **make sure** the deviation $r = b - Ax^*$ is small.

Effect of rounding error

```
function [L, U ,p]=lutx(A)
[n,n]=size(A);
p=(1:n)';
for k=1:n-1
[r,m]=max(abs(A(k:n,k)));
m=m+k-1;
if(A(m,k)~=0)
if(m~=k)
A([k m],:)=A([m k],:);
p([k m])=p([m k]);
end
i=k+1:n;
A(i,k)=A(i,k)/A(k,k);
j=k+1:n;
A(i,j)=A(i,j)-A(i,k)*A(k,j);
end
end
L=tril(A,-1)+eye(n,n);
U=triu(A);
end
```

Exercise

Write a function bslashtx that implements MatLab's (simplified) left division to solve a system of linear equations.

```
function x=bslashtx(A,b)
n=size(A,1);
%Su use lutx(A);
...
% Downside
...
%The source
...
```

Bad determinism and matrix conditions

Coefficients are rarely known exactly

because of

- For systems of equations that appear in the application, the coefficients that are normally attributed to the empirical value should be associated with the observation error.
- Many other systems of equations have coefficients calculated by formulas and therefore they are known exactly to rounding errors when calculated according to the given formula.
- Even for systems of equations that are stored exactly in the computer, errors are inevitable (Review the representations of integers, signed integers, and floating-point real numbers in the textbook). h General Informatics)

So the question is:

If there is an error in the representation of coefficients of a system of linear equations, how does that affect the solution? Or when solving $Ax = b$ how can the sensitivity of x be measured when there is a change in A, b ?

Bad determinism and matrix conditions

There are a few comments after

- If A is degenerate then for some b x either has no solution or infinitely many solutions. In the case where A has a small determinant, a small change in A and b can lead to a large change in the solution.
- Think about the size of the pivot elements and the concept of near degeneracy. Because if the arithmetic operations are performed exactly, all the pivot elements are non-zero if and only if the matrix is non-degenerate. From that, the following statement is drawn: 'If the pivot elements are small, the matrix is near degenerate' the opposite is not true, or in other words, there is a near degenerate matrix where the pivot elements are not small.

Bad determinism and matrix conditions

Vector Normals

Definition: The function $v : \mathbb{R}^n \mapsto R$ is said to be a vector norm over $\mathbb{R}^n$ if and only if

1. $v(x) \geq 0 \ \ \forall x \in \mathbb{R}^n$ and $v(x) = 0$ if and only if $x = 0$
2. $v(\alpha x) = |\alpha| v(x) \ \ \forall x \in \mathbb{R}^n, \forall \alpha \in \mathbb{R}$
3. $v(x + y) \leq v(x) + v(y) \ \ \forall x, y \in \mathbb{R}^n$ this is the triangle inequality.

Normally $v(x)$ is denoted by $||x||$

Bad determinism and matrix conditions

Vector Normalization (continued)

Some commonly used standards

- $||x||_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$ is (l_2) or Euclidean
- $||x||_1 = \sum_{i=1}^{n} |x_i|$ is (l_1)
- $||x||_\infty = \max_{1 \le i \le n} |x_i|$ is (l_∞)
- $||x||_p = (\sum_{i=1}^{n} |x_i|^p)^{1/p}$ is (l_p)

Matlab

norm(x,p) for l_p
and for $p = 2$ the function is simpler than **norm(x)**

Bad determinism and matrix conditions

Matrix normal

Definition: Function $||.|| : \mathbb{R}^{n\times n} \mapsto \mathbb{R}$ is said to be matrix normal if

$$||A|| = \max_{||x||=1, x\in\mathbb{R}^n} ||Ax|| = \max_{||x||\neq 0, x\in\mathbb{R}^n} \frac{||Ax||}{||x||}$$

where $||Ax||$ is the norm of the vector Ax. Of course, we have the inequality $||Ax|| \leq ||A||\ ||x||$

Bad determinism and matrix conditions

Matrix normal (continued)

The properties of the matrix norm

1. $||A|| \geq 0; ||A|| = 0$ if and only if $A = 0$.
2. $||\alpha A|| = ||\alpha|| ||A||, \alpha \in \mathbb{R}$
3. $||A + B|| \leq ||A|| + ||B||$
4. $||AB|| \leq ||A|| \times ||B||$
5. $||Ax|| \leq ||A|| ||x||$

Bad determinism and matrix conditions

Matrix normal (continued)

The vector normals generate the corresponding matrix normals

- Euclidean normal : $||A||_2 = \max_{||x||_2=1} ||Ax||_2$
- max "total lines" :
  $||A||_\infty = \max_{||x||_\infty=1} ||Ax||_\infty = \max_{1\leq i\ leqn} \sum_{j=1}^{n} |a_{ij}|$
- max "total column" :
  $||A||_1 = \max_{||x||_1=1} ||Ax||_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}|$
- Frobenius Standard : $||A||_F = \left(Tr(A^T A)^{1/2}\right) = \left(\sum_{i,j=1}^{n} a_{ij}^2\right)^{1/2}$

In Matlab

norm(A,p) where $p = 1, 2, inf$

Bad determinism and matrix conditions

Number of matrix conditions

Definition : The condition number **cond(A)**, usually denoted by $\kappa_p(A)$, of a square matrix A computed for a given matrix p standard is a number

$$cond(A) = ||A|| \cdot ||A^{-1}||$$

where, we convention $cond(A) = \infty$ when A is degenerate. Because of,

$$||A|| \cdot ||A^{-1}|| = \frac{\max_{x \neq 0} \frac{||Ax||}{||x||}}{\min_{x \neq 0} \frac{||Ax||}{||x||}}$$

so the number of conditions that measure the ratio of the maximum expansion to the maximum contraction that the matrix can act on for a vector is non-zero.

Bad determinism and matrix conditions

Number of conditions of the matrix (continued)

The number of conditions indicates how close the matrix is to degeneracy: the larger the matrix, the closer **near degenerate** (the corresponding system of equations is poorly defined), whereas the closer the matrix to 1 the number of conditions. far from **near degenerate**.

Note :

The matrix determinant is not a good characterization for approximation. Although when $det(A) = 0$, the matrix is degenerate, but the magnitude or small of the determinant is not contains information about whether the matrix is near degenerate or not.

For example, for the matrix $det(\alpha\mathbb{I}_n) = \alpha^n$ may be a very small number when $|\alpha| < 1$ but the matrix $\alpha\mathbb{I}_n$ has good conditions with $cond(\alpha\mathbb{I}_n) = 1$. Where $\mathbb{I}_n$ is the n dimensional unit matrix.

Bad determinism and matrix conditions

Some conditional numerical properties of matrices

1. For all matrices A : $cond(A) \geq 1$
2. For all unit matrices $\mathbb{I}$: $cond(\mathbb{I}) = 1$
3. For any permutation matrix P : $cond(P) = 1$
4. For all matrices A and non-zero reals α : $cond(\alpha A) = cond(A)$
5. For any diagonal matrix $D = diag(d_i)$: $cond(D) = \frac{\max\{|d_i|\}}{\min\{|d_i|\}}$
6. The number of conditions is important in evaluating the accuracy of the solution of a system of linear equations.

Bad determinism and matrix conditions

Matlab with number of conditions

cond(A,p) to calculate $\kappa_p(A)$ with $p = 1, 2, inf$.
cond(A) or **cond(A,2)** calculates $\kappa_2(A)$. Use **svd(A)**. Should be used with small matrix.
cond(A,1) calculates $\kappa_1(A)$. Use the function **inv(A)**. Requires less computation time than **cond(A,2)**.
cond(A,inf) calculates $\kappa_\infty(A)$. Use the function **inv(A)**. Requires less computation time than **cond(A,1)**.
condest(A) to evaluate $\kappa_1(A)$. Using the function **lu(A)** and the Higham-Tisseur algorithm. Recommended for large matrices.
rcond(A) to evaluate $1/\kappa_1(A)$

Bad determinism and matrix conditions

Evaluate the error when knowing the number of conditions of the matrix

Let x be the exact solution of $Ax = b$, and x^* be the solution of the system $Ax^* = b + \Delta b$ (note we only consider b to be additive noise.). Put $\Delta x = x^* - x$, we have $b + \Delta b = Ax^* = A(x + \Delta x) = Ax + A\Delta x$ since $Ax = b$ substitute in, then $\Delta x = A^{-1}\Delta b$.

$$b = Ax \Rightarrow ||b|| \leq ||A||||x|| \tag{13}$$

$$\Delta x = A^{-1}\Delta b \Rightarrow ||\Delta x|| \leq ||A^{-1}||||\Delta b|| \tag{14}$$

Multiply the two inequalities (13) (14) and use the definition $cond(A) = ||A||||A^{-1}||$ we have evaluation

$$\frac{||\Delta x||}{||x||} \leq cond(A)\frac{||\Delta b||}{||b||}$$

Bad determinism and matrix conditions

Evaluate error when knowing the number of conditions of the matrix (continued)

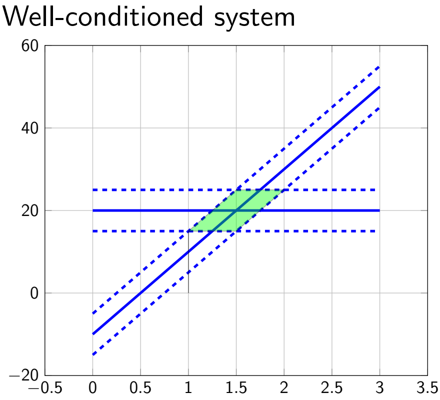
continue...

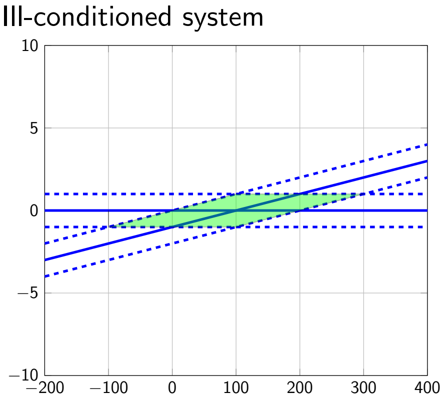
$$\frac{||\Delta x||}{||x||} \le cond(A)\frac{||\Delta b||}{||b||}$$

So the number of conditions allows us to determine the relative error variation in the solution $\frac{||\Delta x||}{||x||}$ given the relative change in the right-hand side. $\frac{||\Delta b||}{||b||}$

- When $cond(A)$ is large or the system is near degenerate, the relative transformation of the right side will 'force' the corresponding error change in the solution.
- Conversely, when $cond(A)$ approaches 1 or the system is well-conditioned, the equivalent transformation of the right hand side and the solution are the same.

Bad determinism and matrix conditions





Bad determinism and matrix conditions

Evaluate the error when knowing the number of conditions of the matrix (Conclusion)

If the input data is represented approximately to computer accuracy, then the relative error estimate of the calculated solution is given by the formula:

$$\frac{||x^* - x||}{||x||} \approx cond(A)\epsilon_M$$

The calculated solution will lose an interval $\log_{10}(cond(A))$ in decimal places in the relative error of the data precision.

Conclusion

The system of linear equations $Ax = b$ is ill-conditioned if $cond(A)$ is large, then a small change in the data can lead to a large change in the solution.

Bad determinism and matrix conditions

Example 15 :

Consider the system of equations

$$0.789x_1 + 0.563x_2 = 0.127$$
$$0.913x_1 + 0.659x_2 = 0.254$$

Results when using Matlab

» A=[0.789 0.563;0.913 0.659];

» fprintf('cond(A)=%d ; det(A)=%d ',cond(A),det(A))

» cond(A) = 2.193219e+006 ; det(A)=1.000000e-006

Bad determinism and matrix conditions

Example 16 :

Consider the system of equations

$$\begin{pmatrix} 4.1 & 2.8 \\ 9.7 & 6.6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4.1 \\ 9.7 \end{pmatrix}$$

This is an ill-conditioned system because $cond(A, 1) = 2494.4$ and the exact solution of the system is $x = (1, 0)^T$. If we substitute the right hand side $b + \Delta b = (4.11, 9.70)^T$ then the solution of the system will be $x^* = (0.34, 0.97)^T$.

In Matlab we have

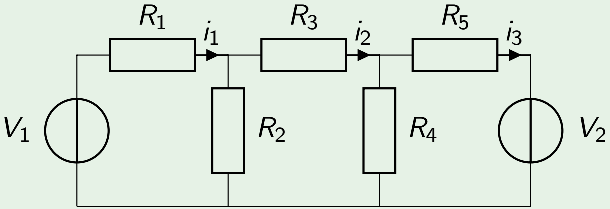
```
» A = [4.1 2.8; 9.7 6.6]; b = [4.1 ; 9.7]; b1=[4.11 ; 9.7];
» x = (A \ b)', x1 = (A \ b1)'
x = 1 0
x1 = 0.3400 0.9700
```

Bad determinism and matrix conditions

Solve a system of linear equations using matrix analysis

Electrical network analysis

Solving a system of linear equations has important applications in power network analysis. Example for the following electrical network



Solving a system of linear equations by matrix analysis (continued)

Electrical network analysis (continued)

According to Kirchoff's law, the voltage across the loops must be zero. We have the following system of linear equations

$$\begin{aligned} -V_1 + R_1 i_1 + R_2(i_1 - i_2) &= 0 \\ R_2(i_2 - i_1) + R_3 i_2 + R_4(i_2 - i_3) &= 0 \\ R_4(i_3 - i_2) + R_5 i_3 + V_2 &= 0 \end{aligned}$$

converted into

$$\begin{pmatrix} R_1 + R_2 & -R_2 & 0 \\ -R_2 & R_2 + R_3 + R_4 & -R_4 \\ 0 & -R_4 & R_4 + R_5 \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} = \begin{pmatrix} V_1 \\ 0 \\ -V_2 \end{pmatrix}$$

System of Linear Equations

Summary

- Features of a system of linear equations (linear algebra)
- Recalling permutation matrices and triangular matrices
- LU analysis is used to solve system of linear equations
- The role of cylinders in LU analysis can cause errors in the results
- Rounding error effect
- Determines the condition of a matrix in a linear equation that causes the resulting error
- An example of applying a system of linear equations

Solve a system of linear equations using matrix analysis

More homework

We can use many methods besides LU analysis to solve the system of equations

- Cholesky Analysis
  - The concept of a positive semi-deterministic matrix
  - If A is a positive definite matrix then there exists a positive lower triangular matrix L such that $A = LL^T$
  - Eliminate forward $Ly = b$, backward substitution $L^T x = y$
- Decay QR
  - Orthogonal matrix concept
  - Decay QR : if $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and has rank n then there exists an orthogonal matrix $Q \in \mathbb{R}^{m \times n}$ and a triangular matrix on $R \in \mathbb{R}^{n \times n}$ with positive elements on the diagonal such that $A = QR$
  - Solve least squares problem

$$\min\{||Ax - b||^2 | x \in \mathbb{R}\}$$

so the solution x is the stopping point when minimizing the above problem.

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

soict.hust.edu.vn/ fb.com/groups/soict