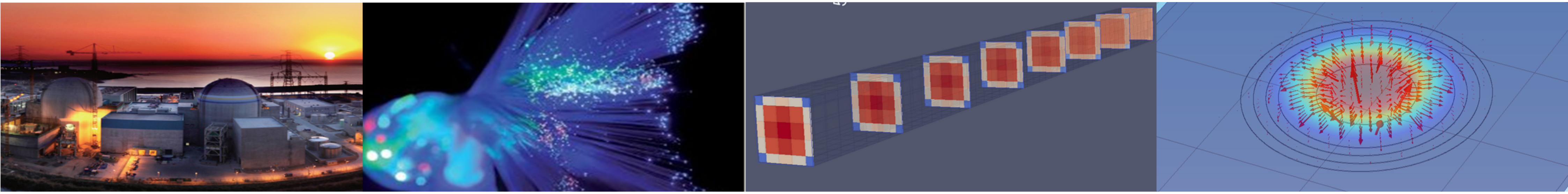


INTRODUCTION TO NUMERICAL ANALYSIS

Cho, Hyoung Kyu

Department of Nuclear Engineering

Seoul National University

4. A SYSTEM OF LINEAR EQUATIONS

NuTHEL

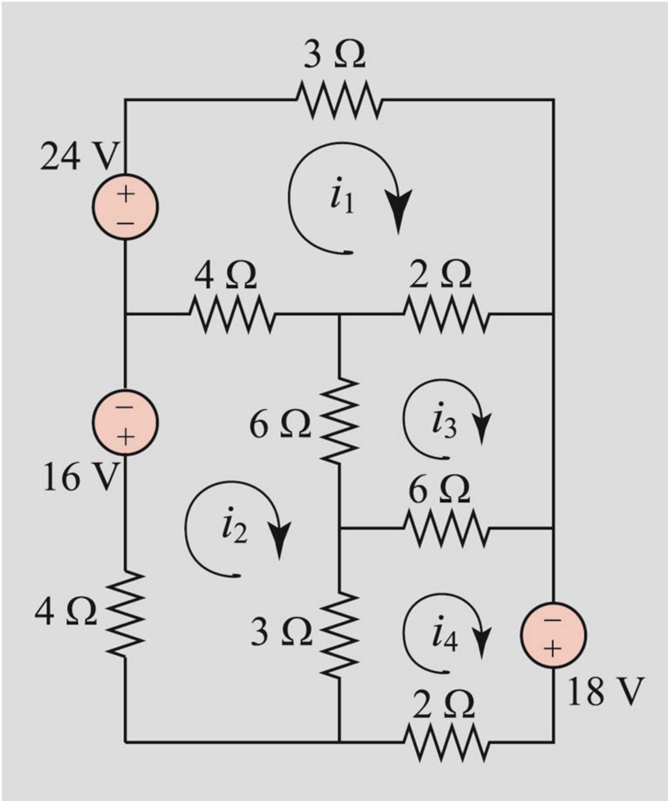
Nuclear Thermal Hydraulic Engineering Lab.

Systems of linear equations

- Occur frequently not only in engineering and science but in any disciplines

Example

- Electrical engineering
  - Kirchhoff's law



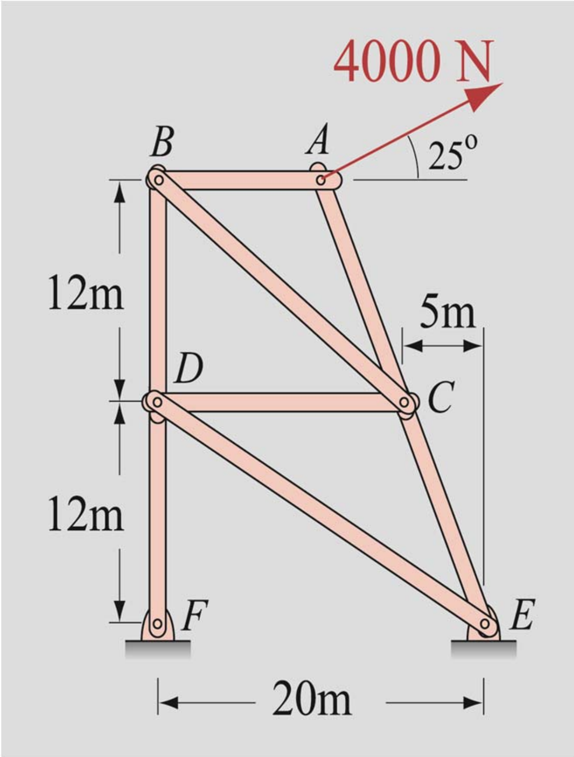
Systems of linear equations

- Occur frequently not only in engineering and science but in any disciplines

Example

- Force in members of a truss
  - Force balance

$$0.9231F_{AC} = 1690 \qquad -F_{AB} - 0.3846F_{AC} = 3625$$
$$F_{AB} - 0.7809F_{BC} = 0 \qquad 0.6247F_{BC} - F_{BD} = 0$$
$$F_{CD} + 0.8575F_{DE} = 0 \qquad F_{BD} - 0.5145F_{DE} - F_{DF} = 0$$
$$0.3846F_{CE} - 0.3846F_{AC} - 0.7809F_{BC} - F_{CD} = 0$$
$$0.9231F_{AC} + 0.6247F_{BC} - 0.9231F_{CE} = 0$$



Overview of numerical methods for solving a system of linear algebraic equations

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\
\vdots \qquad \vdots \qquad\quad & \quad \vdots \\
a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n
\end{aligned}
\quad \Rightarrow \quad
\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}
=
\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}
$$

- Direct methods vs. iterative methods

- Direct methods
  - The solution is calculated by performing arithmetic operations with the equations.
  - Three systems of equations that can be easily solved are the
    - Upper triangular
    - Lower triangular
    - Diagonal

❖ Overview of numerical methods for solving a system of linear algebraic equations

- Direct methods
  - Upper triangular form

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\
a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n &= b_2 \\
a_{33}x_3 + \ldots + a_{3n}x_n &= b_3 \\
\vdots \qquad\qquad \vdots \quad &\quad \vdots \\
a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n &= b_{n-1} \\
a_{nn}x_n &= b_n
\end{aligned}
$$

$$
\Rightarrow \quad
\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
=
\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}
$$

    – Ex) 4 equations

$$
x_4 = \frac{b_4}{a_{44}}, \quad x_3 = \frac{b_3 - a_{34}x_4}{a_{33}}, \quad x_2 = \frac{b_2 - (a_{23}x_3 + a_{24}x_4)}{a_{22}}, \quad x_1 = \frac{b_1 - (a_{12}x_2 + a_{13}x_3 + a_{14}x_4)}{a_{11}}
$$

    – In general

Overview of numerical methods for solving a system of linear algebraic equations

- Direct methods
  - Lower triangular form

$$\begin{aligned} a_{11}x_1 \qquad\qquad\qquad\qquad\qquad\qquad &= b_1 \\ a_{21}x_1 + a_{22}x_2 \qquad\qquad\qquad\qquad &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \qquad\qquad &= b_3 \\ \vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad &\;\;\vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n &= b_n \end{aligned}$$

$$\Rightarrow \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

    – Ex) 4 equations

$$x_1 = \frac{b_1}{a_{11}}, \quad x_2 = \frac{b_2 - a_{21}x_1}{a_{22}}, \quad x_3 = \frac{b_3 - (a_{31}x_1 + a_{32}x_2)}{a_{33}}, \quad x_4 = \frac{b_4 - (a_{41}x_1 + a_{42}x_2 + a_{43}x_3)}{a_{44}}$$

    – In general

Overview of numerical methods for solving a system of linear algebraic equations

- Direct methods
  - Diagonal form

$$
\begin{aligned}
a_{11}x_1 \quad\quad\quad\quad\quad\quad\quad\quad &= b_1 \\
a_{12}x_2 \quad\quad\quad\quad\quad\quad &= b_2 \\
a_{13}x_3 \quad\quad\quad\quad &= b_3 \\
\vdots \quad\quad\quad\quad & \quad \vdots \\
a_n x_n &= b_n
\end{aligned}
$$

$$
\begin{bmatrix} a_{11} & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_2 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_2 \\ b_4 \end{bmatrix}
$$

  - LU decomposition method
    - Lower and upper triangular form
  - Gauss-Jordan method
    - Diagonal form
  - Iterative methods
    - Jacobi
    - Gauss-Seidel

Gauss elimination method

- A general form is manipulated to be in upper triangular form
  - Back substitution

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3 \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= b_4 \end{aligned}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 &= b'_2 \\ a'_{33}x_3 + a'_{34}x_4 &= b'_3 \\ a'_{44}x_4 &= b'_4 \end{aligned}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a'_{33} & a'_{34} \\ 0 & 0 & 0 & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

Gauss elimination method

- A general form is manipulated to be in upper triangular form
  -
  - Step 1
    - Eliminate x_1 in all other equations except the first one.
    - First equation:
    - Coefficient a_{11} :

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

$$m_{21} = a_{21}/a_{11}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b_3 \\ b_4 \end{bmatrix}$$

$$\begin{array}{rl} & a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = b_2 \\ - & m_{21}(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4) = m_{21}b_1 \\ \hline & 0 + \underbrace{(a_{22} - m_{21}a_{12})}_{a'_{22}}x_2 + \underbrace{(a_{23} - m_{21}a_{13})}_{a'_{23}}x_3 + \underbrace{(a_{24} - m_{21}a_{14})}_{a'_{24}}x_4 = \underbrace{b_2 - m_{21}b_1}_{b'_2} \end{array}$$

Gauss elimination method

- A general form is manipulated to be in upper triangular form
  - Forward elimination
  - Step 1
    - Eliminate x_1 in all other equations except the first one.
    - First equation: **pivot equation**
    - Coefficient a_{11} : **pivot coefficient**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b_3 \\ b_4 \end{bmatrix}$$

$$m_{31} = a_{31}/a_{11}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b_4 \end{bmatrix}$$

$$\begin{array}{rr} & a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = b_3 \\ - & m_{31}(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4) = m_{31}b_1 \\ \hline \end{array}$$

$$0 + \underbrace{(a_{32} - m_{31}a_{12})}_{a'_{32}}x_2 + \underbrace{(a_{33} - m_{31}a_{13})}_{a'_{33}}x_3 + \underbrace{(a_{34} - m_{31}a_{14})}_{a'_{34}}x_4 = \underbrace{b_3 - m_{31}b_1}_{b'_3}$$

Gauss elimination method

- A general form is manipulated to be in upper triangular form
  - Forward elimination
  - Step 1
    - Eliminate x_1 in all other equations except the first one.
    - First equation: **pivot equation**
    - Coefficient a_{11} : **pivot coefficient**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b_4 \end{bmatrix}$$

$$m_{41} = a_{41}/a_{11}$$

$$\begin{array}{rl} & a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = b_4 \\ - & m_{41}(a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4) = m_{41}b_1 \\ \hline \end{array}$$

$$0 + \underbrace{(a_{42} - m_{41}a_{12})}_{a'_{42}}x_2 + \underbrace{(a_{43} - m_{41}a_{13})}_{a'_{43}}x_3 + \underbrace{(a_{44} - m_{41}a_{14})}_{a'_{44}}x_4 = \underbrace{b_4 - m_{41}b_1}_{b'_4}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

Gauss elimination method

- A general form is manipulated to be in upper triangular form
  - Forward elimination
  - Step 2
    - Eliminate x_2 in all other equations except the 1st and 2nd ones.
    - Second equation: **pivot equation**
    - Coefficient a'_{22} : **pivot coefficient**

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\
0 + a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 &= b'_2 \\
0 + a'_{32}x_2 + a'_{33}x_3 + a'_{34}x_4 &= b'_3 \\
0 + a'_{42}x_2 + a'_{43}x_3 + a'_{44}x_4 &= b'_4
\end{aligned}
$$

$$m_{32} = a'_{32}/a'_{22} \qquad m_{42} = a'_{42}/a'_{22}$$

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\
0 + a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 &= b'_2 \\
0 + \quad 0 \quad + a''_{33}x_3 + a''_{34}x_4 &= b''_3 \\
0 + \quad 0 \quad + a''_{43}x_3 + a''_{44}x_4 &= b''_4
\end{aligned}
$$

$$
\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & a''_{43} & a''_{44} \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
=
\begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b''_4 \end{bmatrix}
$$

Gauss elimination method

- A general form is manipulated to be in upper triangular form
  - Forward elimination
  - Step 3
    - Eliminate x_1 in 4th equations
    - Third equation: **pivot equation**
    - Coefficient a''_{33} : **pivot coefficient**

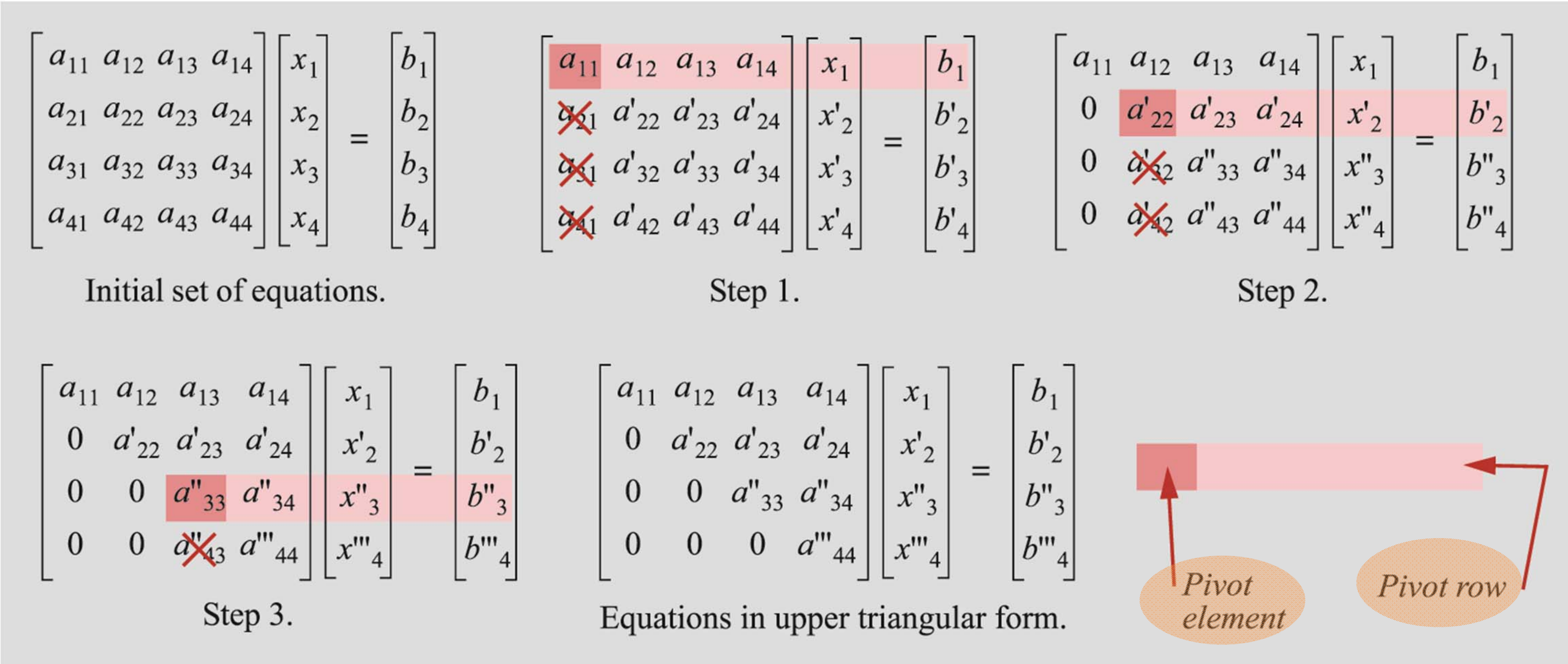
$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\
0 + a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 &= b'_2 \\
0 + \quad 0 \quad + a''_{33}x_3 + a''_{34}x_4 &= b''_3 \\
0 + \quad 0 \quad + a''_{43}x_3 + a''_{44}x_4 &= b''_4
\end{aligned}
$$

$$m_{43} = a''_{43} / a''_{33}$$

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\
0 + a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 &= b'_2 \\
0 + \quad 0 \quad + a''_{33}x_3 + a''_{34}x_4 &= b''_3 \\
0 + \quad 0 \quad + \quad 0 \quad + a'''_{44}x_4 &= b'''_4
\end{aligned}
$$

$$
\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a'''_{44} \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}
=
\begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b'''_4 \end{bmatrix}
$$

❖ Gauss elimination method



❖ Gauss elimination method

Example 4-1: Solving a set of four equations using Gauss elimination.

Solve the following system of four equations using the Gauss elimination method.

$$\begin{aligned} 4x_1 - 2x_2 - 3x_3 + 6x_4 &= 12 \\ -6x_1 + 7x_2 + 6.5x_3 - 6x_4 &= -6.5 \\ x_1 + 7.5x_2 + 6.25x_3 + 5.5x_4 &= 16 \\ -12x_1 + 22x_2 + 15.5x_3 - x_4 &= 17 \end{aligned}$$

- Answer

$$\begin{aligned} 4x_1 - 2x_2 - 3x_3 + 6x_4 &= 12 \\ 4x_2 + 2x_3 + 3x_4 &= 11.5 \\ 3x_3 - 2x_4 &= -10 \\ 4x_4 &= 2 \end{aligned}$$

$$\begin{aligned} x_4 &= 0.5 \\ x_3 &= -3 \\ x_2 &= 4 \\ x_1 &= 2 \end{aligned}$$

Gauss elimination method

Example 4-2: MATLAB user-defined function for solving a system of equations using Gauss elimination.

Write a user-defined MATLAB function for solving a system of linear equations, $[a][x]=[b]$, using the Gauss elimination method. For function name and arguments, use `x = Gauss(a,b)`, where `a` is the matrix of coefficients, `b` is the right-hand-side column vector of constants, and `x` is a column vector of the solution.

Use the user-defined function `Gauss` to

(*a*) Solve the system of equations of Example 4-1.

(b) Solve the system of Eqs. (4.1).

- Matrix ab

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ 0 & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ 0 & 0 & a_{33} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{nn} & b_n \end{bmatrix}$$

Back substitution !

4.2 Gauss Elimination Method

Gauss elimination method

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} & b_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} & b_n \end{bmatrix}$$

```
function x = Gauss(a,b)
% The function solve a system of linear equations [a][x]=[b] using the Gauss
% elimination method.
% Input variables:
% a  The matrix of coefficients.
% b  A column vector of constants.
% Output variable:
% x  A colum vector with the solution.

ab = [a,b];
[R, C] = size(ab);
for j = 1:R-1
    for i = j+1:R

    end
end
x = zeros(R,1);
x(R) = ab(R,C)/ab(R,R);
for i = R-1:-1:1

end
```

Append the column vector [b] to the matrix [a].

Forward elimination

Back substitution

Gauss elimination method

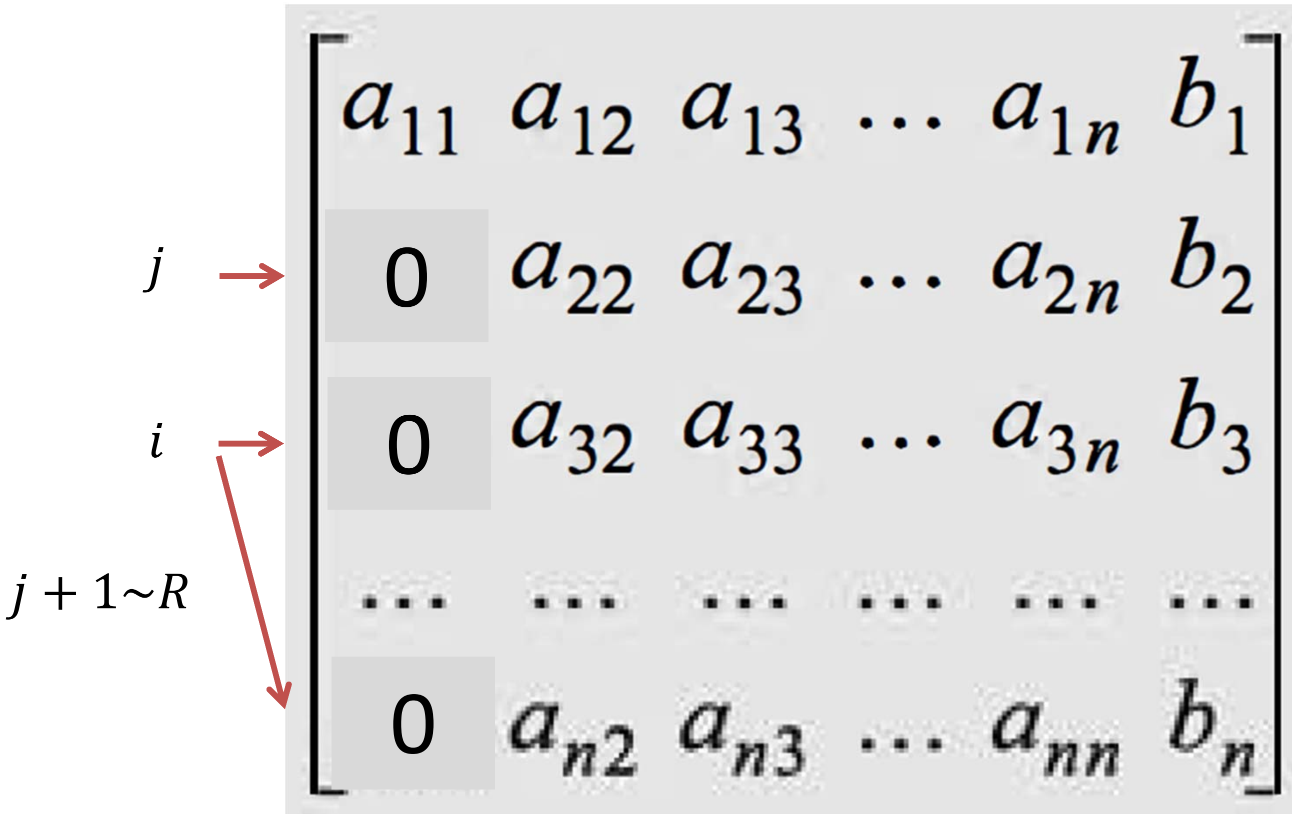
$j = 2$: pivot equation $(1 \sim R - 1)$
$i = 3$ $\quad\quad\quad\quad\quad\quad\quad\quad (j + 1 \sim R)$

$$\frac{a_{32}}{a_{22}} \Rightarrow \frac{a_{ij}}{a_{jj}}$$

$$\frac{a_{32}}{a_{22}} a_{23} \Rightarrow \frac{a_{ij}}{a_{jj}} a_{ji}$$

$$\frac{a_{32}}{a_{22}} a_{2c} \Rightarrow \frac{a_{ij}}{a_{jj}} a_{jc}$$

$$a_{3c} - \frac{a_{32}}{a_{22}} a_{2c} \Rightarrow a_{ic} - \frac{a_{ij}}{a_{jj}} a_{jc}$$



Gauss elimination method

$$x_n = \frac{b_n}{a_{nn}} \qquad x_i = \frac{b_i - \sum_{j=i+1}^{j=n} a_{ij} x_j}{a_{ii}}$$

C

i →

R →

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ 0 & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ 0 & 0 & a_{33} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{nn} & b_n \end{bmatrix}$$

Gauss elimination method

```
A=[4 -2 -3 6; -6 7 6.5 -6; 1 7.5 6.25 5.5; -12 22 15.5 -1];
B = [12; -6.5; 16; 17];
sola = Gauss(A,B)
```

```
sola =
    2.0000
    4.0000
   -3.0000
    0.5000
```

Potential difficulties when applying the Gauss elimination method

- The pivot element is zero.
  - Pivot row is divided by the pivot element.
  - If the value of the pivot element is equal to zero, a problem will arise.

Pivoting!

- The pivot element is small relative to the other terms in the pivot row.
  - Problem can be easily remedied by exchanging the order of the two equations.

- In general, a more accurate solution is obtained when the equations are arranged (and rearranged every time a new pivot equation is used) such that the pivot equation has the largest possible pivot element.

- Round-off errors can also be significant when solving large systems of equations even when all the coefficients in the pivot row are of the same order of magnitude.
- This can be caused by a large number of operations (multiplication, division, addition, and subtraction) associated with large systems.

❖ Potential difficulties when applying the Gauss elimination method

$$\begin{aligned} 0.0003x_1 + 12.34x_2 &= 12.343 \\ 0.4321x_1 + x_2 &= 5.321 \end{aligned}$$

$$x_1 = 10 \text{ and } x_2 = 1$$

$$m_{21} = 0.4321/0.0003 = 1440$$

$$(1440)(0.0003x_1 + 12.34x_2) = 1440 \cdot 12.34$$

$$0.4320x_1 + 17770x_2 = 17770$$

$$\begin{array}{rl} & 0.4321x_1 + x_2 = 5.321 \\ - & 0.4320x_1 + 17770x_2 = 17770 \\ \hline & 0.0001x_1 - 17770x_2 = -17760 \end{array}$$

$$\begin{aligned} 0.0003x_1 + 12.34x_2 &= 12.34 \\ 0.0001x_1 - 17770x_2 &= -17760 \end{aligned}$$

$$x_2 = \frac{-17760}{-17770} = 0.9994$$

$$x_1 = \frac{12.34 - (12.34 \cdot 0.9994)}{0.0003} = \frac{12.34 - 12.33}{0.0003} = \frac{0.01}{0.0003} = 33.33$$

❖ Potential difficulties when applying the Gauss elimination method

- Remedy

$$\begin{aligned} 0.4321x_1 + x_2 &= 5.321 \\ 0.0003x_1 + 12.34x_2 &= 12.343 \end{aligned}$$

$$m_{21} = 0.0003/0.4321 = 0.0006943$$

$$(0.0006943)(0.4321x_1 + x_2) = 0.0006943 \cdot 5.321$$

$$0.0003x_1 + 0.0006943x_2 = 0.003694$$

$$\begin{array}{rrl} & 0.0003x_1 + 12.34x_2 &= 12.34 \\ - & 0.0003x_1 + 0.0006943x_2 &= 0.003694 \\ \hline & 12.34x_2 &= 12.34 \end{array}$$

$$\begin{aligned} 0.4321x_1 + x_2 &= 5.321 \\ 0x_1 + 12.34x_2 &= 12.34 \end{aligned}$$

$$x_2 = \frac{12.34}{12.34} = 1$$

$$x_1 = \frac{5.321 - 1}{0.4321} = 10$$

Example

$$
\begin{aligned}
0x_1 + 2x_2 + 3x_3 &= 46 \\
4x_1 - 3x_2 + 2x_3 &= 16 \\
2x_1 + 4x_2 - 3x_3 &= 12
\end{aligned}
$$

- First pivot coefficient: 0 $\quad m_{21} = a_{21}/a_{11} \quad 4/0$

Pivoting

$$
\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & 0 & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & 0 & a'_{23} & a'_{24} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_3 \\ b'_2 \\ b'_4 \end{bmatrix}
$$

- Exchange of rows

Additional comments

- The numerical calculations are less prone to error and will have fewer round-off errors if the pivot element has a larger numerical absolute value compared to the other elements in the same row.
- Consequently, among all the equations that can be exchanged to be the pivot equation, it is better to select the equation whose pivot element has the largest absolute numerical value.
- Moreover, it is good to employ pivoting for the purpose of having a pivot equation with the pivot element that has a largest absolute numerical value at all times (even when pivoting is not necessary).

Partial pivoting

$$\begin{bmatrix} 1 & 4 & 1 & 8 & 3 & 2 & \cdots & 5 \\ 0 & 10^{-6} & 1 & 10 & 201 & 13 & & 4 \\ 0 & 9 & 4 & 6 & -8 & 2 & & 18 \\ 0 & 3 & 2 & -3 & 4 & 6003 & & 15 \\ 0 & 15 & 1 & 9 & 33 & -2 & & 1 \\ 0 & -155 & 23 & 4 & 25 & 73 & & 2 \\ & & & \vdots & & & & \\ 0 & 8 & 56 & 4 & -4 & 4 & \cdots & 88 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \\ \hat{r}_4 \\ \hat{r}_5 \\ \hat{r}_6 \\ \vdots \\ \hat{r}_n \end{pmatrix} \Rightarrow \begin{bmatrix} 1 & 4 & 1 & 8 & 3 & 2 & \cdots & 5 \\ 0 & -155 & 23 & 4 & 25 & 73 & & 2 \\ 0 & 9 & 4 & 6 & -8 & 2 & & 18 \\ 0 & 3 & 2 & -3 & 4 & 6003 & & 15 \\ 0 & 15 & 1 & 9 & 33 & -2 & & 1 \\ 0 & 10^{-6} & 1 & 10 & 201 & 13 & & 4 \\ & & & \vdots & & & & \\ 0 & 8 & 56 & 4 & -4 & 4 & \cdots & 88 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \hat{r}_1 \\ \hat{r}_6 \\ \hat{r}_3 \\ \hat{r}_4 \\ \hat{r}_5 \\ \hat{r}_2 \\ \vdots \\ \hat{r}_n \end{pmatrix}$$

❖ Additional comments

- The numerical calculations are less prone to error and will have fewer round-off errors if the pivot element has a larger numerical absolute value compared to the other elements in the same row.
- Consequently, among all the equations that can be exchanged to be the pivot equation, it is better to select the equation whose pivot element has the largest absolute numerical value.
- Moreover, it is good to employ pivoting for the purpose of having a pivot equation with the pivot element that has a largest absolute numerical value at all times (even when pivoting is not necessary).

Full pivoting

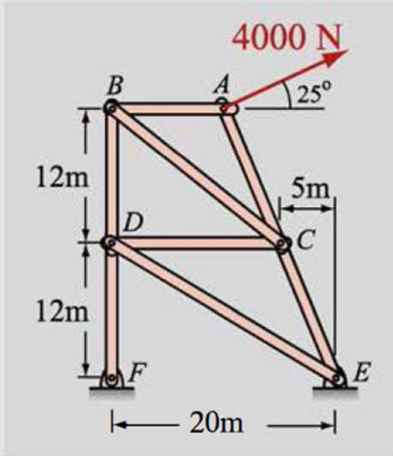
$$\begin{bmatrix} 1 & 4 & 1 & 8 & 3 & 2 & \cdots & 5 \\ 0 & 10^{-6} & 1 & 10 & 201 & 13 & & 4 \\ 0 & 9 & 4 & 6 & -8 & 2 & & 18 \\ 0 & 3 & 2 & -3 & 4 & 6003 & & 15 \\ 0 & 15 & 1 & 9 & 33 & -2 & & 1 \\ 0 & -155 & 23 & 4 & 25 & 73 & & 2 \\ & & & \vdots & & & & \\ 0 & 8 & 56 & 4 & -4 & 4 & \cdots & 88 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \hat{r}_1 \\ \hat{r}_2 \\ \hat{r}_3 \\ \hat{r}_4 \\ \hat{r}_5 \\ \hat{r}_6 \\ \vdots \\ \hat{r}_n \end{pmatrix} \Rightarrow \begin{bmatrix} 1 & 3 & 1 & 8 & 3 & 2 & \cdots & 5 \\ 0 & 201 & 1 & 10 & 10^{-6} & 13 & & 4 \\ 0 & -8 & 4 & 6 & 9 & 2 & & 18 \\ 0 & 4 & 2 & -3 & 3 & 6003 & & 15 \\ 0 & 33 & 1 & 9 & 15 & -2 & & 1 \\ 0 & 25 & 23 & 4 & -155 & 73 & & 2 \\ & & & \vdots & & & & \\ 0 & -4 & 56 & 4 & 8 & 4 & \cdots & 88 \end{bmatrix} \begin{pmatrix} x_1 \\ x_5 \\ x_3 \\ x_4 \\ x_2 \\ x_6 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \hat{b}_3 \\ \hat{b}_4 \\ \hat{b}_5 \\ \hat{b}_6 \\ \vdots \\ \hat{b}_n \end{pmatrix}$$

4.3 Gauss Elimination with Pivoting

❖ Example 4-3

Example 4-3: MATLAB user-defined function for solving a system of equations using Gauss elimination with pivoting.

Write a user-defined MATLAB function for solving a system of linear equations $[a][x] = [b]$ using the Gauss elimination method with pivoting. Name the function `x = GaussPivot(a,b)`, where `a` is the matrix of coefficients, `b` is the right-hand-side column vector of constants, and `x` is a column vector of the solution. Use the function to determine the forces in the loaded eight-member truss that is shown in the figure (same as in Fig. 4-2).



SOLUTION

The forces in the eight truss members are determined from the set of eight equations, Eqs. (4.2). The equations are derived by drawing free

$$\begin{bmatrix} 0 & 0.9231 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -0.3846 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0.8575 & 0 \\ 1 & 0 & -0.7809 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.3846 & -0.7809 & 0 & -1 & 0.3846 & 0 & 0 \\ 0 & 0.9231 & 0.6247 & 0 & 0 & -0.9231 & 0 & 0 \\ 0 & 0 & 0.6247 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -0.5145 & -1 \end{bmatrix} \begin{bmatrix} F_{AB} \\ F_{AC} \\ F_{BC} \\ F_{BD} \\ F_{CD} \\ F_{CE} \\ F_{DE} \\ F_{DF} \end{bmatrix} = \begin{bmatrix} 1690 \\ 3625 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Example 4-3

```
function x = GaussPivot(a,b)
% The function solve a system of linear equations ax=b using the Gauss
% elimination method with pivoting.
% Input variables:
% a  The matrix of coefficients.
% b  A column vector of constants.
% Output variable:
% x  A colum vector with the solution.

ab = [a,b]
[R, C] = size(ab);
for j = 1:R-1
% Pivoting section starts
    if ab(j,j)==0
        for k=j+1:R



            break
        end
    end
end
% Pivoting section ends
    for i = j+1:R
        ab(i,j:C) = ab(i,j:C)-ab(i,j)/ab(j,j)*ab(j,j:C);
    end
end
```

Check if the pivot element is zero.

If pivoting is required, search in the rows below for a row with nonzero pivot element. 1

Swap

Forward elimination

Example 4-3

```
x = zeros(R,1);
x(R) = ab(R,C)/ab(R,R);
for i = R-1:-1:1
    x(i)=(ab(i,C)-ab(i,i+1:R)*x(i+1:R))/ab(i,i);
end
```

Back substitution

- Result

```
% Example 4-3
a=[0 0.9231 0 0 0 0 0 0; -1 -0.3846 0 0 0 0 0 0; 0 0 0 0 1 0 0.8575 0; 1 0 -0.7809 0 0 0 0 0
   0 -0.3846 -0.7809 0 -1 0.3846 0 0; 0 0.9231 0.6247 0 0 -0.9231 0 0
   0 0 0.6247 -1 0 0 0 0; 0 0 0 1 0 0 -0.5145 -1];
b = [1690;3625;0;0;0;0;0;0];
Forces = GaussPivot(a,b)
```

```
Forces =
  -4.3291e+003
   1.8308e+003
  -5.5438e+003
  -3.4632e+003
   2.8862e+003
  -1.9209e+003
  -3.3659e+003
  -1.7315e+003
>>
```

$$\begin{bmatrix} F_{AB} \\ F_{AC} \\ F_{BC} \\ F_{BD} \\ F_{CD} \\ F_{CE} \\ F_{DE} \\ F_{DF} \end{bmatrix}$$

Gauss-Jordan Elimination Result

- In this procedure, a system of equations that is given in a general form is manipulated into an equivalent system of equations in **diagonal form with normalized elements** along the diagonal.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3 \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= b_4 \end{aligned}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

$$\begin{aligned} x_1 + 0 + 0 + 0 &= b'_1 \\ 0 + x_2 + 0 + 0 &= b'_2 \\ 0 + 0 + x_3 + 0 &= b'_3 \\ 0 + 0 + 0 + x_4 &= b'_4 \end{aligned}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

Procedure

- In this procedure, a system of equations that is given in a general form is manipulated into an equivalent system of equations in **diagonal form with normalized elements** along the diagonal.

- The pivot equation is normalized by dividing all the terms in the equation by the pivot coefficient. This makes the pivot coefficient equal to 1.

- The pivot equation is used to eliminate the off-diagonal terms in ALL the other equations.
- This means that the elimination process is applied to the equations (rows) that are above and below the pivot equation.
- In the Gaussian elimination method, only elements that are below the pivot element are eliminated.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a'''_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \\ b'''_4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \end{bmatrix}$$

4.4 Gauss-Jordan Elimination Method

❖ Procedure

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{bmatrix} \xrightarrow{\text{Gauss–Jordan procedure}} \begin{bmatrix} 1 & 0 & 0 & 0 & b'_1 \\ 0 & 1 & 0 & 0 & b'_2 \\ 0 & 0 & 1 & 0 & b'_3 \\ 0 & 0 & 0 & 1 & b'_4 \end{bmatrix}$$

- The Gauss-Jordan method can also be used for solving several systems of equations $[a][x] = [b]$ that have the same coefficients $[a]$ but different right-hand-side vectors $[b]$.
- This is done by augmenting the matrix $[a]$ to include all of the vectors $[b]$.
- The method is used in this way for calculating the inverse of a matrix.

Procedure

Example 4-4: Solving a set of four equations using Gauss–Jordan elimination.

Solve the following set of four equations using the Gauss–Jordan elimination method.

$$
\begin{aligned}
4x_1 - 2x_2 - 3x_3 + 6x_4 &= 12 \\
-6x_1 + 7x_2 + 6.5x_3 - 6x_4 &= -6.5 \\
x_1 + 7.5x_2 + 6.25x_3 + 5.5x_4 &= 16 \\
-12x_1 + 22x_2 + 15.5x_3 - x_4 &= 17
\end{aligned}
$$

$$
\begin{bmatrix} 4 & -2 & -3 & 6 \\ -6 & 7 & 6.5 & -6 \\ 1 & 7.5 & 6.25 & 5.5 \\ -12 & 22 & 15.5 & -1 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} =
\begin{bmatrix} 12 \\ -6.5 \\ 16 \\ 17 \end{bmatrix}
\Rightarrow
\begin{bmatrix} 4 & -2 & -3 & 6 & 12 \\ -6 & 7 & 6.5 & -6 & -6.5 \\ 1 & 7.5 & 6.25 & 5.5 & 16 \\ -12 & 22 & 15.5 & -1 & 17 \end{bmatrix}
$$

$$
\begin{bmatrix} \frac{4}{4} & \frac{-2}{4} & \frac{-3}{4} & \frac{6}{4} & \frac{12}{4} \\ -6 & 7 & 6.5 & -6 & -6.5 \\ 1 & 7.5 & 6.25 & 5.5 & 16 \\ -12 & 22 & 15.5 & -1 & 17 \end{bmatrix} =
\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ -6 & 7 & 6.5 & -6 & -6.5 \\ 1 & 7.5 & 6.25 & 5.5 & 16 \\ -12 & 22 & 15.5 & -1 & 17 \end{bmatrix}
$$

Pivot coefficient is normalized!

$$
\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ -6 & 7 & 6.5 & -6 & -6.5 \\ 1 & 7.5 & 6.25 & 5.5 & 16 \\ -12 & 22 & 15.5 & -1 & 17 \end{bmatrix}
\begin{matrix} \\ \leftarrow -(-6)[1\ -0.5\ -0.75\ 1.5\ 3] \\ \leftarrow -(1)[1\ -0.5\ -0.75\ 1.5\ 3] \\ \leftarrow -(-12)[1\ -0.5\ -0.75\ 1.5\ 3] \end{matrix}
=
\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ 0 & 4 & 2 & 3 & 11.5 \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6.5 & 17 & 53 \end{bmatrix}
$$

First elements in rows 2, 3, 4 are eliminated.

❖ Procedure

$$\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ 0 & \frac{4}{4} & \frac{2}{4} & \frac{3}{4} & \frac{11.5}{4} \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6.5 & 17 & 53 \end{bmatrix} = \begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6.5 & 17 & 53 \end{bmatrix}$$

The second pivot coefficient is normalized!

$$\begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 & 3 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 8 & 7 & 4 & 13 \\ 0 & 16 & 6.5 & 17 & 53 \end{bmatrix} \begin{matrix} \leftarrow -(-0.5)[0\ 1\ 0.5\ 0.75\ 2.875] \\ \\ \leftarrow -(8)[0\ 1\ 0.5\ 0.75\ 2.875] \\ \leftarrow -(16)[0\ 1\ 0.5\ 0.75\ 2.875] \end{matrix} = \begin{bmatrix} 1 & 0 & -0.5 & 1.875 & 4.4375 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 0 & 3 & -2 & -10 \\ 0 & 0 & -1.5 & 5 & 7 \end{bmatrix}$$

The second elements in rows 1, 3, 4 are eliminated.

$$\begin{bmatrix} 1 & 0 & -0.5 & 1.875 & 4.4375 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 0 & \frac{3}{3} & \frac{-2}{3} & \frac{-10}{3} \\ 0 & 0 & -1.5 & 5 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -0.5 & 1.875 & 4.4375 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & -1.5 & 5 & 7 \end{bmatrix}$$

The third pivot coefficient is normalized!

$$\begin{bmatrix} 1 & 0 & -0.5 & 1.875 & 4.4375 \\ 0 & 1 & 0.5 & 0.75 & 2.875 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & -1.5 & 5 & 7 \end{bmatrix} \begin{matrix} \leftarrow -(-0.5)[0\ 0\ 1\ -0.667\ -3.333] \\ \leftarrow -(0.5)[0\ 0\ 1\ -0.667\ -3.333] \\ \\ \leftarrow -(-1.5)[0\ 0\ 1\ -0.667\ -3.333] \end{matrix} = \begin{bmatrix} 1 & 0 & 0 & 1.5417 & 2.7708 \\ 0 & 1 & 0 & 1.0833 & 4.5417 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & 0 & 4 & 2 \end{bmatrix}$$

The third elements in rows 1, 2, 4 are eliminated.

Procedure

$$\begin{bmatrix} 1 & 0 & 0 & 1.5417 & 2.7708 \\ 0 & 1 & 0 & 1.0833 & 4.5417 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & 0 & \frac{4}{4} & \frac{2}{4} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1.5417 & 2.7708 \\ 0 & 1 & 0 & 1.0833 & 4.5417 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & 0 & 1 & 0.5 \end{bmatrix}$$

The fourth pivot coefficient is normalized!

$$\begin{bmatrix} 1 & 0 & 0 & 1.5417 & 2.7708 \\ 0 & 1 & 0 & 1.0833 & 4.5417 \\ 0 & 0 & 1 & -0.667 & -3.333 \\ 0 & 0 & 0 & 1 & 0.5 \end{bmatrix} \begin{matrix} \leftarrow -(1.5417)\begin{bmatrix}0 & 0 & 0 & 1 & 0.5\end{bmatrix} \\ \leftarrow -(1.0833)\begin{bmatrix}0 & 0 & 0 & 1 & 0.5\end{bmatrix} \\ \leftarrow -(-0.667)\begin{bmatrix}0 & 0 & 0 & 1 & 0.5\end{bmatrix} \\ \end{matrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 4 \\ 0 & 0 & 1 & 0 & -3 \\ 0 & 0 & 0 & 1 & 0.5 \end{bmatrix}$$

The fourth elements in rows 2, 3, 4 are eliminated.

- It is possible that the equations are written in such an order that during the elimination procedure a pivot equation has a pivot element that is equal to zero.
- Obviously, in this case it is impossible to normalize the pivot row (divide by the pivot element).
- As with the Gauss elimination method, the problem can be corrected by using **pivoting**.

4.4 Gauss-Jordan Elimination Method

Example code

```
function x = GaussJordan(a,b)
% The function solve a system of linear equations ax=b using the Gauss
% elimination method with pivoting. In each step the rows are switched
% such that pivot element has the largest absolute numerical value.
% Input variables:
% a The matrix of coefficients.
% b A column vector of constants.
% Output variable:
% x A column vector with the solution.

ab = [a,b];
[R, C] = size(ab);
for j = 1:R
    % Pivoting section starts
    pvtemp=ab(j,j);
    kpvt=j;
    % Looking for the row with the largest pivot element.
    for k=j+1:R
        if ab(k,j)~=0 && abs(ab(k,j)) > abs(pvtemp)
            pvtemp=ab(k,j);
            kpvt=k;
        end
    end
```

- **Example code**

```
% If a row with a larger pivot element exists, switch the rows.
    if kpvt~=j
        abTemp=ab(j,:);
        ab(j,:)=ab(kpvt,:);
        ab(kpvt,:)=abTemp;
    end
    % Pivoting section ends

    ab(j,:)= ab(j,:)/ab(j,j);
    for i = 1:R
        if i~=j
            ab(i,j:C) = ab(i,j:C)-ab(i,j)*ab(j,j:C);
        end
    end
end
x=ab(:,C);
```

Background

- The Gauss elimination method
  - Forward elimination procedure

    $$[a][x] = [b] \quad \rightarrow \quad [a'][x] = [b']$$

    – $[a']$: upper triangular.
  - Back substitution

- The elimination procedure requires many mathematical operations and significantly more computing time than the back substitution calculations.

- During the elimination procedure, the matrix of coefficients [a] and the vector [b] are both changed.

- This means that if there is a need to solve systems of equations that have the same left-hand-side terms (same coefficient matrix $[a]$) but different right-hand-side constants (different vectors $[b]$), the elimination procedure has to be carried out for each $[b]$ again.

Background

- Inverse matrix ?

$$[a][x] = [b] \Rightarrow [x] = [a]^{-1}[b]$$

  - Calculating the inverse of a matrix, however, requires many mathematical operations, and is computationally inefficient.
  - A more efficient method of solution for this case is the **LU decomposition method** !

- LU decomposition

$[a] = [L][U]$; $[L]$: lower triangular matrix ; $[U]$: upper triangular matrix

With this decomposition, the system of equations to be solved has the form:

; $\Leftarrow$

Gauss elimination method
Crout's method

LU decomposition using the Gauss elimination procedure

- Procedure
  - The elements of $[L]$ on the diagonal are all 1
  - The elements below the diagonal are the multipliers m_{ij}

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a'''_{44} \end{bmatrix}$$

  - Ex)

$$\begin{pmatrix} 2 & 3 & 1 & 5 \\ 6 & 13 & 5 & 19 \\ 2 & 19 & 10 & 23 \\ 4 & 10 & 11 & 31 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 2 & 1 & 7 & 1 \end{pmatrix} \times \begin{pmatrix} 2 & 3 & 1 & 5 \\ 0 & 4 & 2 & 4 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

LU decomposition using the Gauss elimination procedure

- Procedure
  - The elements of $[L]$ on the diagonal are all 1
  - The elements below the diagonal are the multipliers m_{ij}

$$\begin{pmatrix}2&3&1&5\\6&13&5&19\\2&19&10&23\\4&10&11&31\end{pmatrix}$$

$$\begin{pmatrix}1&0&0&0\\-3&1&0&0\\-1&0&1&0\\-2&0&0&1\end{pmatrix}\cdot\begin{pmatrix}2&3&1&5\\6&13&5&19\\2&19&10&23\\4&10&11&31\end{pmatrix}=\begin{pmatrix}2&3&1&5\\0&4&2&4\\0&16&9&18\\0&4&9&21\end{pmatrix}$$

$$\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&-4&1&0\\0&-1&0&1\end{pmatrix}\cdot\begin{pmatrix}2&3&1&5\\0&4&2&4\\0&16&9&18\\0&4&9&21\end{pmatrix}=\begin{pmatrix}2&3&1&5\\0&4&2&4\\0&0&1&2\\0&0&7&17\end{pmatrix}$$

$$\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&-7&1\end{pmatrix}\cdot\begin{pmatrix}2&3&1&5\\0&4&2&4\\0&0&1&2\\0&0&7&17\end{pmatrix}=\begin{pmatrix}2&3&1&5\\0&4&2&4\\0&0&1&2\\0&0&0&3\end{pmatrix}$$

LU decomposition using the Gauss elimination procedure

- Procedure

$$L_3 \cdot A_2 = L_3 \cdot L_2 \cdot L_1 \cdot A = A_3$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -3 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -2 & 0 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -4 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 4 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -7 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 7 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 4 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 7 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 2 & 1 & 7 & 1 \end{pmatrix}$$

4.5 LU Decomposition Method

LU decomposition using the Gauss elimination procedure

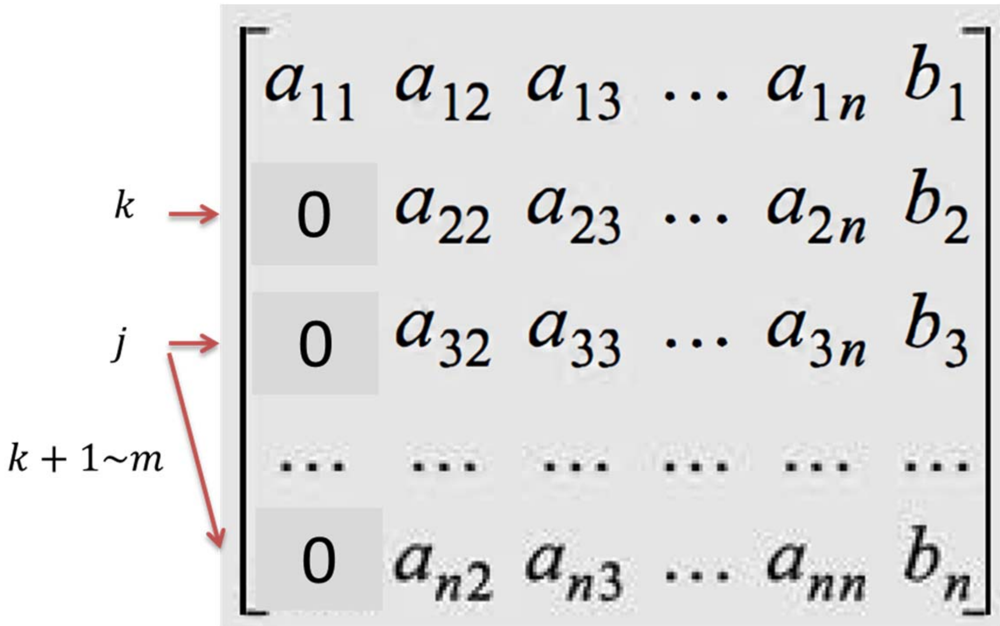
- Algorithm

Algorithm 20.1. Gaussian Elimination without Pivoting

$U = A, \quad L = I$

for $k = 1$ **to** $m - 1$

  for $j = k + 1$ **to** m



LU decomposition using Crout's method

$$[a][x] = [b]$$

- $[a] = [L][U]$; $[L]$: lower triangular matrix ; $[U]$: upper triangular matrix
  - The diagonal elements of the matrix $[U]$ are all 1 s.
- Illustration with 4x4 matrix

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix} \begin{bmatrix} 1 & U_{12} & U_{13} & U_{14} \\ 0 & 1 & U_{23} & U_{24} \\ 0 & 0 & 1 & U_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} L_{11} & (L_{11}U_{12}) & (L_{11}U_{13}) & (L_{11}U_{14}) \\ L_{21} & (L_{21}U_{12}+L_{22}) & (L_{21}U_{13}+L_{22}U_{23}) & (L_{21}U_{14}+L_{22}U_{24}) \\ L_{31} & (L_{31}U_{12}+L_{32}) & (L_{31}U_{13}+L_{32}U_{23}+L_{33}) & (L_{31}U_{14}+L_{32}U_{24}+L_{33}U_{34}) \\ L_{41} & (L_{41}U_{12}+L_{42}) & (L_{41}U_{13}+L_{42}U_{23}+L_{43}) & (L_{41}U_{14}+L_{42}U_{24}+L_{43}U_{34}+L_{44}) \end{bmatrix}$$

LU decomposition using Crout's method

- Illustration with 4x4 matrix

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} L_{11} & (L_{11}U_{12}) & (L_{11}U_{13}) & (L_{11}U_{14}) \\ L_{21} & (L_{21}U_{12}+L_{22}) & (L_{21}U_{13}+L_{22}U_{23}) & (L_{21}U_{14}+L_{22}U_{24}) \\ L_{31} & (L_{31}U_{12}+L_{32}) & (L_{31}U_{13}+L_{32}U_{23}+L_{33}) & (L_{31}U_{14}+L_{32}U_{24}+L_{33}U_{34}) \\ L_{41} & (L_{41}U_{12}+L_{42}) & (L_{41}U_{13}+L_{42}U_{23}+L_{43}) & (L_{41}U_{14}+L_{42}U_{24}+L_{43}U_{34}+L_{44}) \end{bmatrix}$$

$$L_{11} = a_{11} \qquad U_{12} = \frac{a_{12}}{L_{11}} \qquad U_{13} = \frac{a_{13}}{L_{11}} \qquad U_{14} = \frac{a_{14}}{L_{11}}$$

$$L_{21} = a_{21} \qquad L_{22} = a_{22} - L_{21}U_{12} \qquad U_{23} = \frac{a_{23} - L_{21}U_{13}}{L_{22}} \quad \text{and} \quad U_{24} = \frac{a_{24} - L_{21}U_{14}}{L_{22}}$$

$$L_{31} = a_{31}, \quad L_{32} = a_{32} - L_{31}U_{12}, \quad \text{and} \quad L_{33} = a_{33} - L_{31}U_{13} - L_{32}U_{23} \qquad U_{34} = \frac{a_{34} - L_{31}U_{14} - L_{32}U_{24}}{L_{33}}$$

$$L_{41} = a_{41}, \quad L_{42} = a_{42} - L_{41}U_{12}, \quad L_{43} = a_{43} - L_{41}U_{13} - L_{42}U_{23}, \quad L_{44} = a_{44} - L_{41}U_{14} - L_{42}U_{24} - L_{43}U_{34}$$

❖ LU decomposition using Crout's method

- For n×n matrix
  - Step 1: Calculating the first column of $[L]$:

    $for\ i = 1, 2, \ldots, n \qquad L_{i1} = a_{i1}$

  - Step 2: Substituting $1s$ in the diagonal of $[U]$: $U_{ii} = 1$
  - Step 3: calculating the elements in the first row of $[U]$ (except U_{11} which was already calculated):

    $for\ j = 2, 3, \ldots, n \qquad U_{1j} = \dfrac{a_{1j}}{L_{11}}$

  - Step 4: calculating the rest of the elements row after row. The elements of $[L]$ are calculated first because they are used for calculating the elements of $[U]$:

    $for\ i = 2, 3, \ldots, n$

    $for\ j = 2, 3, \ldots, i$

    $for\ j = (i + 1), (i + 2), \ldots, n$

LU decomposition using Crout's method

- Example

Example 4-5: Solving a set of four equations using *LU* decomposition with Crout's method.

Solve the following set of four equations (the same as in Example 4-1) using *LU* decomposition with Crout's method.

$$4x_1 - 2x_2 - 3x_3 + 6x_4 = 12$$
$$-6x_1 + 7x_2 + 6.5x_3 - 6x_4 = -6.5$$
$$x_1 + 7.5x_2 + 6.25x_3 + 5.5x_4 = 16$$
$$-12x_1 + 22x_2 + 15.5x_3 - x_4 = 17$$

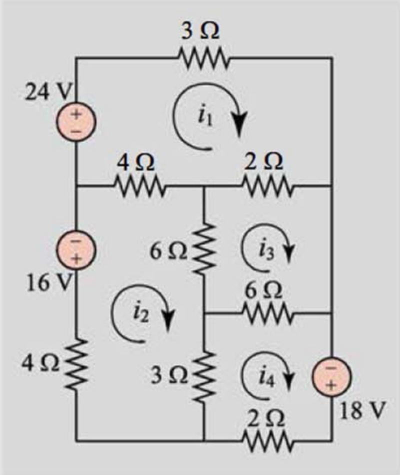
$$L = \begin{bmatrix} 4 & 0 & 0 & 0 \\ -6 & 4 & 0 & 0 \\ 1 & 8 & 3 & 0 \\ -12 & 16 & -1.5 & 4 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1 & -0.5 & -0.75 & 1.5 \\ 0 & 1 & 0.5 & 0.75 \\ 0 & 0 & 1 & -0.6667 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

❖ LU decomposition using Crout's method

- Example

Example 4-6: MATLAB user-defined function for solving a system of equations using *LU* decomposition with Crout's method.

Determine the currents i_1, i_2, i_3, and i_4 in the circuit shown in the figure (same as in Fig. 4-1). Write the system of equations that has to be solved in the form $[a][i] = [b]$. Solve the system by using the *LU* decomposition method, and use Crout's method for doing the decomposition.



SOLUTION

The currents are determined from the set of four equations, Eq. (4.1). The equations are derived by using Kirchhoff's law. In matrix form, $[a][i] = [b]$, the equations are:

$$\begin{bmatrix} 9 & -4 & -2 & 0 \\ -4 & 17 & -6 & -3 \\ -2 & -6 & 14 & -6 \\ 0 & -3 & -6 & 11 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix} = \begin{bmatrix} 24 \\ -16 \\ 0 \\ 18 \end{bmatrix} \tag{4.44}$$

LU decomposition using Crout's method

- Example

```
function [L, U] = LUdecompCrout(A)
% The function decomposes the matrix A into a lower triangular matrix L
% and an upper triangular matrix U, using Crout's method such that A=LU.
% Input variables:
% A  The matrix of coefficients.
% Output variable:
% L  Lower triangular matrix.
% U  Upper triangular matrix.

[R, C] = size(A);
for i = 1:R
    L(i,1) = A(i,1);
    U(i,i) = 1;
end
for j = 2:R
    U(1,j)= A(1,j)/L(1,1);
end
for i = 2:R
    for j = 2:i

    end
    for j = i+1:R

    end
end
```

LU decomposition using Crout's method

- Example
  - BackwardSub.m
  - ForwardSub.m
  - LUdecompCrout.m
  - Program4_6.m

```
% This script file soves a system of equations bu using
% the LU Crout's decomposition method.
a = [9 -4 -2 0; -4 17 -6 -3; -2 -6 14 -6; 0 -3 -6 11];
b = [24; -16; 0; 18];
[L, U] = LUdecompCrout(a);
y = ForwardSub(L,b);
i = BackwardSub(U,y)
```

$$[L][U][x] = [b] \quad ; \quad [U][x] = [y] \quad \Rightarrow \quad [L][y] = [b]$$

LU Decomposition with Pivoting

- Pivoting might also be needed in LU decomposition
- If pivoting is used, then the matrices $[L]$ and $[U]$ that are obtained are not the decomposition of the original matrix $[a]$.
- The product $[L][U]$ gives a matrix with rows that have the same elements as $[a]$, but due to the pivoting, the rows are in a different order.
- When pivoting is used in the decomposition procedure, the changes that are made have to be recorded and stored.
- This is done by creating a matrix $[P]$, called a such that:
- The order of the rows of $[b]$ have to be changed !

$Ax=b$

❖ LU Decomposition with Pivoting

- Pivoting might also be needed in LU decomposition

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{21} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{21} & a_{23} & a_{24} \\ a_{41} & a_{42} & a_{43} & a_{44} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}$$

- Sequential pivoting

- Characteristic of permutation matrix

$$a := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad a^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad a^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$b := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad b \cdot a = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

❖ Inverse of a square matrix $[a]$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

❖ Separate systems of equation

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \\ x_{42} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \\ x_{43} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_{14} \\ x_{24} \\ x_{34} \\ x_{44} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

LU decomposition
Gauss-Jordan elimination

❖ Calculating the inverse with the LU decomposition method

Example 4-7: Determining the inverse of a matrix using the *LU* decomposition method.

Determine the inverse of the matrix $[a]$ by using the LU decomposition method.

$$[a] = \begin{bmatrix} 0.2 & -5 & 3 & 0.4 & 0 \\ -0.5 & 1 & 7 & -2 & 0.3 \\ 0.6 & 2 & -4 & 3 & 0.1 \\ 3 & 0.8 & 2 & -0.4 & 3 \\ 0.5 & 3 & 2 & 0.4 & 1 \end{bmatrix} \quad (4.49)$$

Do the calculations by writing a MATLAB user-defined function. Name the function `invA = InverseLU(A)`, where `A` is the matrix to be inverted, and `invA` is the inverse. In the function, use the functions `LUdecompCrout`, `ForwardSub`, and `BackwardSub` that were written in Example 4-6.

$$\begin{bmatrix} 0.2 & -5 & 3 & 0.4 & 0 \\ -0.5 & 1 & 7 & -2 & 0.3 \\ 0.6 & 2 & -4 & 3 & 0.1 \\ 3 & 0.8 & 2 & -0.4 & 3 \\ 0.5 & 3 & 2 & 0.4 & 1 \end{bmatrix}\begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \\ x_{51} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0.2 & -5 & 3 & 0.4 & 0 \\ -0.5 & 1 & 7 & -2 & 0.3 \\ 0.6 & 2 & -4 & 3 & 0.1 \\ 3 & 0.8 & 2 & -0.4 & 3 \\ 0.5 & 3 & 2 & 0.4 & 1 \end{bmatrix}\begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \\ x_{42} \\ x_{52} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0.2 & -5 & 3 & 0.4 & 0 \\ -0.5 & 1 & 7 & -2 & 0.3 \\ 0.6 & 2 & -4 & 3 & 0.1 \\ 3 & 0.8 & 2 & -0.4 & 3 \\ 0.5 & 3 & 2 & 0.4 & 1 \end{bmatrix}\begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \\ x_{43} \\ x_{53} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0.2 & -5 & 3 & 0.4 & 0 \\ -0.5 & 1 & 7 & -2 & 0.3 \\ 0.6 & 2 & -4 & 3 & 0.1 \\ 3 & 0.8 & 2 & -0.4 & 3 \\ 0.5 & 3 & 2 & 0.4 & 1 \end{bmatrix}\begin{bmatrix} x_{14} \\ x_{24} \\ x_{34} \\ x_{44} \\ x_{54} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0.2 & -5 & 3 & 0.4 & 0 \\ -0.5 & 1 & 7 & -2 & 0.3 \\ 0.6 & 2 & -4 & 3 & 0.1 \\ 3 & 0.8 & 2 & -0.4 & 3 \\ 0.5 & 3 & 2 & 0.4 & 1 \end{bmatrix}\begin{bmatrix} x_{15} \\ x_{25} \\ x_{35} \\ x_{45} \\ x_{55} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

❖ Calculating the inverse with the LU decomposition method

```
function invA = InverseLU(A)
% The function calculates the inverse of a matrix
% Input variables:
% A  The matrix to be inverted.
% Output variable:
% invA  The inverse of A.

[nR nC] = size(A);
I=eye(nR);
[L U]= LUdecompCrout(A);
for j=1:nC
    y=ForwardSub(L,I(:,j));
    invA(:,j)=BackwardSub(U,y);
end
```

A=[0.2 -5 3 0.4 0;-0.5 1 7 -2 0.3; 0.6 2 -4 3 0.1; 3 0.8 2 -0.4 3; 0.5 3 2 0.4 1]
InverseLU(A)

Review

Direct method

- Gauss elimination
- Gauss-Jordan elimination
- LU decomposition
  - Using Gauss elimination
  - Crout's method
  - Pivoting

$$[P][a] = [L][U]$$

Crout's method vs. Gauss elimination

$$L_{ij} = a_{ij} - \sum_{k=1}^{k=j-1} L_{ik}U_{kj}$$

$$U_{ij} = \frac{a_{ij} - \sum_{k=1}^{k=i-1} L_{ik}U_{kj}}{L_{ii}}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{bmatrix} \quad \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{bmatrix}$$

❖ Pivoting

- $M_1 P_1 A = A_2$
- $M_2 P_2 (M_1 P_1 A) = A_3$

$$\vdots$$

- $(M_{n-1} P_{n-1})(M_{n-2} P_{n-2}) \cdots (M_2 P_2)(M_1 P_1 A) = A_n = U$

- $(P_{n-1} P_{n-2} \cdots P_1) A = LU$

- $M_2 P_2 (M_1 P_1 A) = A_3$

  $M_1 P_1 A = P_2 L_2 A_3$

  $P_1 A = L_1 P_2 L_2 A_3$

  $P_2 P_1 A = (P_2 L_1 P_2) L_2 A_3$

$$\text{P2} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{L1} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix} \quad \text{P2}\cdot\text{L1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix} \quad \text{L1}\cdot\text{P2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix}$$

$P_3 P_2 P_1 A = (P_3 L_1' L_2 P_3) L_3 A_3$

$$\text{P2}\cdot\text{L1}\cdot\text{P2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 \end{pmatrix}$$

Pivoting

$$A = \begin{bmatrix} 0 & 0 & -1 & 1 \\ 1 & 1 & -1 & 2 \\ -1 & -1 & 2 & 0 \\ 1 & 2 & 0 & 2 \end{bmatrix} \qquad P_1A = \begin{pmatrix} 1 & 1 & -1 & 2 \\ 0 & 0 & -1 & 1 \\ -1 & -1 & 0 & 0 \\ 1 & 2 & 0 & 2 \end{pmatrix} \qquad \mathrm{P1} := \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathrm{L1} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathrm{A2} := \begin{pmatrix} 1 & 1 & -1 & 2 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 1 & 0 \end{pmatrix} \qquad P_2P_1A = \begin{pmatrix} 1 & 1 & -1 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & -1 & 1 \end{pmatrix} \qquad \mathrm{P2} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \qquad \mathrm{L2} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3 = U = \begin{pmatrix} 1 & 1 & -1 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{pmatrix} \qquad \mathrm{L3} := \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

$$U = \begin{bmatrix} 1 & 1 & -1 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{bmatrix} \qquad P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad PA = \begin{bmatrix} 1 & 1 & -1 & 2 \\ 1 & 2 & 0 & 2 \\ -1 & -1 & 2 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$PA = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & 2 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \end{bmatrix} = LU$$

Floating point operation counts (FLOP)

- Gauss elimination: forward elimination

	$i = 1$	i
Division:	$(n-1)$	$(n-i)$
Multiplication:	$(n-1)(n)$	$(n-i)(n-i+1)$
+ / -:	$(n-1)(n)$	$(n-i)(n-i+1)$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & b_n \end{bmatrix}$$

$$\sum_{i=1}^{n-1}(n-i)(n-i+2) = \sum_{i=1}^{n-1}(n^2 - 2ni + i^2 + 2n - 2i)$$

$$= \sum_{i=1}^{n-1}(n-i)^2 + 2\sum_{i=1}^{n-1}(n-i) = \sum_{i=1}^{n-1} i^2 + 2\sum_{i=1}^{n-1} i$$

$$= \frac{(n-1)n(2n-1)}{6} + 2\frac{(n-1)n}{2} = \frac{2n^3 + 3n^2 - 5n}{6}$$

$$\sum_{i=1}^{n-1}(n-i)(n-i+1) = \sum_{i=1}^{n-1}(n^2 - 2ni + i^2 + n - i)$$

$$= \sum_{i=1}^{n-1}(n-i)^2 + \sum_{i=1}^{n-1}(n-i) = \sum_{i=1}^{n-1} i^2 + \sum_{i=1}^{n-1} i$$

$$= \frac{(n-1)n(2n-1)}{6} + \frac{(n-1)n}{2} = \frac{n^3 - n}{3}.$$

Floating point operation counts (FLOP)

- Gauss elimination: back-substitution

	$i = n$	i
Division:	1	1
Multiplication:	0	$(n-i)$
+ / -:	0	$(n-i-1)$

↑ With $(n-i)$ terms

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ 0 & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ 0 & 0 & a_{33} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{nn} & b_n \end{bmatrix}$$

Multiplications/divisions

$$1 + \sum_{i=1}^{n-1} ((n-i)+1) = 1 + \left(\sum_{i=1}^{n-1} (n-i) \right) + n - 1$$

$$1 = n + \sum_{i=1}^{n-1} (n-i) = n + \sum_{i=1}^{n-1} i = \frac{n^2+n}{2}$$

Additions/subtractions

$$\sum_{i=1}^{n-1} ((n-i-1)+1) = \sum_{i=1}^{n-1} (n-i) = \sum_{i=1}^{n-1} i = \frac{n^2-n}{2}$$

Floating point operation counts (FLOP)

- Gauss elimination: in total

Multiplications/divisions

$$\frac{2n^3 + 3n^2 - 5n}{6} + \frac{n^2 + n}{2} = \frac{n^3}{3} + n^2 - \frac{n}{3}.$$

Additions/subtractions

$$\frac{n^3 - n}{3} + \frac{n^2 - n}{2} = \frac{n^3}{3} + \frac{n^2}{2} - \frac{5n}{6}.$$

The amount of computation and the time required increases with n in proportion to n^3!

Floating point operation counts (FLOP)

- LU decomposition
  - Forward/Backward substitution: $O(n^2)$
- Repeated solution of $Ax = b$ with several bs
  - Significantly less than elimination, particularly for large n.

Multiplications/divisions

$$1 + \sum_{i=1}^{n-1}((n-i)+1) = 1 + \left(\sum_{i=1}^{n-1}(n-i)\right) + n - 1$$

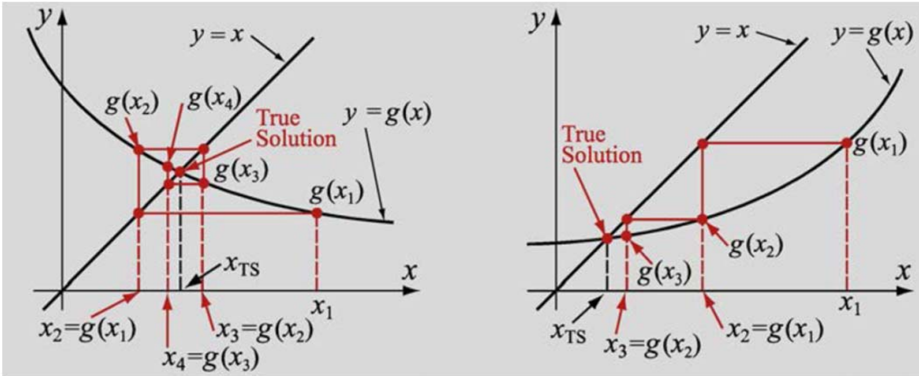
$$1 = n + \sum_{i=1}^{n-1}(n-i) = n + \sum_{i=1}^{n-1} i = \frac{n^2+n}{2}$$

Additions/subtractions

$$\sum_{i=1}^{n-1}((n-i-1)+1) = \sum_{i=1}^{n-1}(n-i) = \sum_{i=1}^{n-1} i = \frac{n^2-n}{2}$$

Iterative approach

- Same as in the fixed-point iteration method



$$f(x) = 0 \qquad x = x + f(x) = g(x)$$

- Explicit form for a system of four equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3 \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= b_4 \end{aligned}$$

Writing the equations in an explicit form. →

$$\begin{aligned} x_1 &= [b_1 - (a_{12}x_2 + a_{13}x_3 + a_{14}x_4)]/a_{11} \\ x_2 &= [b_2 - (a_{21}x_1 + a_{23}x_3 + a_{24}x_4)]/a_{22} \\ x_3 &= [b_3 - (a_{31}x_1 + a_{32}x_2 + a_{34}x_4)]/a_{33} \\ x_4 &= [b_4 - (a_{21}x_1 + a_{42}x_2 + a_{43}x_3)]/a_{44} \end{aligned}$$

❖ Iterative approach

- Solution process
  - Initial value assumption (**the first estimated solution**)

$$
\begin{aligned}
x_1 &= [b_1 - (a_{12}x_2 + a_{13}x_3 + a_{14}x_4)]/a_{11} \\
x_2 &= [b_2 - (a_{21}x_1 + a_{23}x_3 + a_{24}x_4)]/a_{22} \\
x_3 &= [b_3 - (a_{31}x_1 + a_{32}x_2 + a_{34}x_4)]/a_{33} \\
x_4 &= [b_4 - (a_{21}x_1 + a_{42}x_2 + a_{43}x_3)]/a_{44}
\end{aligned}
$$

  - In the first iteration, the first assumed solution is substituted on the right-hand side of the equations $\Rightarrow$ **the second estimated solution**.
  - In the second iteration, **the second solution** is substituted back $\Rightarrow$ **the third estimated solution**
  - The iterations continue until solutions converge toward the actual solution.

$$i = 1, 2, \ldots, n$$

Iterative approach

- Condition for convergence
  - A sufficient condition for convergence (not necessary)
    - The absolute value of the diagonal element is greater than the sum of the absolute values of the off-diagonal elements.

- Two specific iterative methods
  - Jacobi
    - Updated all at once at the end of each iteration
  - Gauss-Seidel
    - Updated when a new estimated is calculated

$$\begin{aligned}
x_1 &= [b_1 - (a_{12}x_2 + a_{13}x_3 + a_{14}x_4)]/a_{11} \\
x_2 &= [b_2 - (a_{21}x_1 + a_{23}x_3 + a_{24}x_4)]/a_{22} \\
x_3 &= [b_3 - (a_{31}x_1 + a_{32}x_2 + a_{34}x_4)]/a_{33} \\
x_4 &= [b_4 - (a_{21}x_1 + a_{42}x_2 + a_{43}x_3)]/a_{44}
\end{aligned}$$

4.7 Iterative Methods

❖ Jacobi iterative method

$$x_i^{(2)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1,\, j\neq i}^{j=n} a_{ij}x_j^{(1)}\right] \qquad i = 1, 2, \ldots, n$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1,\, j\neq i}^{j=n} a_{ij}x_j^{(k)}\right] \qquad i = 1, 2, \ldots, n$$

- Convergence check
  - Absolute value of relative error of all unknowns

$$\left|\frac{x_i^{(k+1)} - x_i^{(k)}}{x_i^{(k)}}\right| < \varepsilon \qquad i = 1, 2, \ldots, n$$

$$\begin{aligned}
x_1 &= [b_1 - (a_{12}x_2 + a_{13}x_3 + a_{14}x_4)]/a_{11}\\
x_2 &= [b_2 - (a_{21}x_1 + a_{23}x_3 + a_{24}x_4)]/a_{22}\\
x_3 &= [b_3 - (a_{31}x_1 + a_{32}x_2 + a_{34}x_4)]/a_{33}\\
x_4 &= [b_4 - (a_{21}x_1 + a_{42}x_2 + a_{43}x_3)]/a_{44}
\end{aligned}$$

$$\begin{aligned}
x_1 &= [b_1 - (a_{12}x_2 + a_{13}x_3 + a_{14}x_4)]/a_{11}\\
x_2 &= [b_2 - (a_{21}x_1 + a_{23}x_3 + a_{24}x_4)]/a_{22}\\
x_3 &= [b_3 - (a_{31}x_1 + a_{32}x_2 + a_{34}x_4)]/a_{33}\\
x_4 &= [b_4 - (a_{21}x_1 + a_{42}x_2 + a_{43}x_3)]/a_{44}
\end{aligned}$$

$$\begin{aligned}
x_1 &= [b_1 - (a_{12}x_2 + a_{13}x_3 + a_{14}x_4)]/a_{11}\\
x_2 &= [b_2 - (a_{21}x_1 + a_{23}x_3 + a_{24}x_4)]/a_{22}\\
x_3 &= [b_3 - (a_{31}x_1 + a_{32}x_2 + a_{34}x_4)]/a_{33}\\
x_4 &= [b_4 - (a_{21}x_1 + a_{42}x_2 + a_{43}x_3)]/a_{44}
\end{aligned}$$

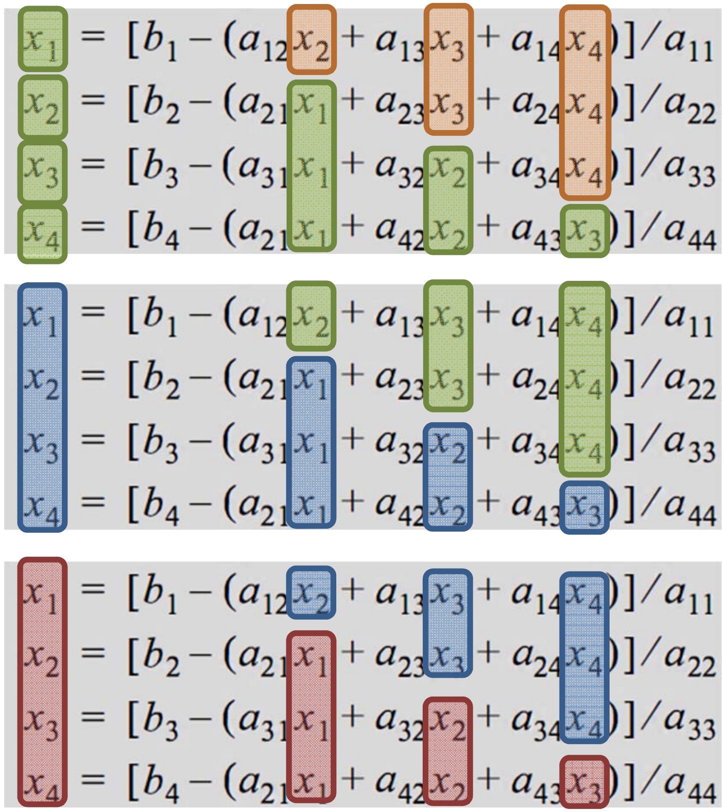
Gauss-Siedel iterative method

$$x_1^{(k+1)} = \frac{1}{a_{11}}\left[b_1 - \sum_{j=2}^{j=n} a_{i1} x_j^{(k)}\right]$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \qquad\qquad\qquad\qquad \right]$$

$i = 2, 3, \ldots, n-1$

$$x_n^{(k+1)} = \frac{1}{a_{nn}}\left[b_n - \sum_{j=1}^{j=n-1} a_{nj} x_j^{(k+1)}\right]$$



- Comment
  - Gauss-Siedel method converges faster than the Jacobi method and requires less computer memory.

Example

Example 4-8: Solving a set of four linear equations using Gauss–Seidel method.

Solve the following set of four linear equations using the Gauss–Seidel iteration method.

$$9x_1 - 2x_2 + 3x_3 + 2x_4 = 54.5$$
$$2x_1 + 8x_2 - 2x_3 + 3x_4 = -14$$
$$-3x_1 + 2x_2 + 11x_3 - 4x_4 = 12.5$$
$$-2x_1 + 3x_2 + 2x_3 + 10x_4 = -21$$

```
k = 1; x1 = 0; x2 = 0; x3 = 0; x4 = 0;
disp('  k          x1         x2         x3           x4')
fprintf(' %2.0f        %-8.5f  %-8.5f  %-8.5f  %-8.5f \n', k, x1, x2, x3, x4)
for k=2 : 8
    x1 = (54.5 - (-2*x2 + 3*x3 + 2*x4))/9;
    x2 = (-14 - (2*x1 - 2*x3 + 3*x4))/8;
    x3 = (12.5 - (-3*x1 + 2*x2 - 4*x4))/11;
    x4 = (-21 - (-2*x1 + 3*x2 + 2*x3))/10;
    fprintf(' %2.0f        %-8.5f  %-8.5f  %-8.5f  %-8.5f \n', k, x1, x2, x3, x4)
end
```

4.8 Use of MATLAB Built-in Functions

MATLAB operators

- Left division \
  - To solve a system of n equations written in matrix form $[a][x] = [b]$

```
x = a\b
```

- Right division /
  - To solve a system of equations written in matrix form $[x][a] = [b]$

```
x = b/a
```

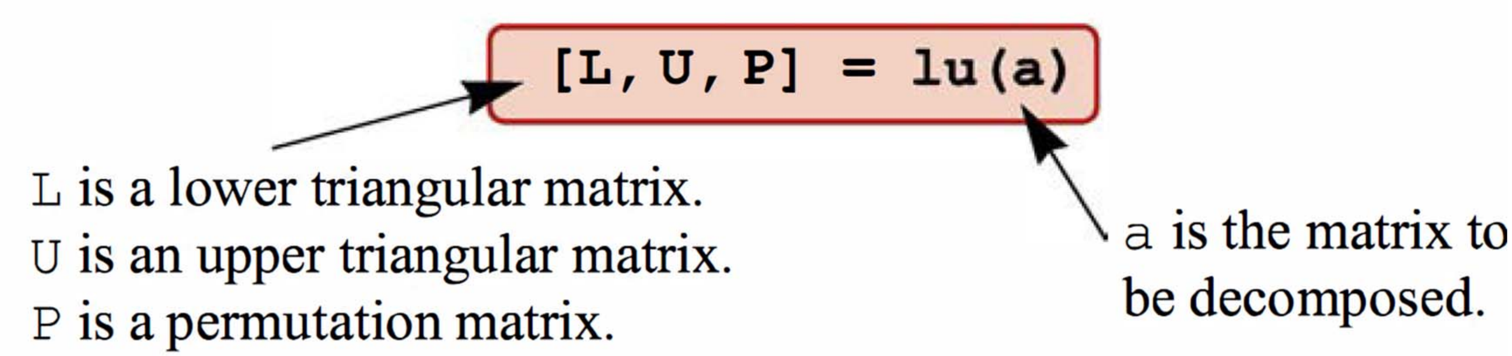
- Matrix inversion
  - `inv(a)`
  - `a^-1`

```
>> a=[4 -2 -3 6; -6 7 6.5 -6; 1 7.5 6.25 5.5; -12 22 15.5 -1];
>> b=[12; -6.5; 16; 17];
>> x=a^-1*b
x =
    2.0000
    4.0000
   -3.0000
    0.5000
```

The same result is obtained by typing >> x = inv(a)*b.

4.8 Use of MATLAB Built-in Functions

❖ MATLAB's built-in function for LU decomposition



$$[L][U] = [P][a]$$

- Without pivoting, $[P] = [I]$

MATLAB's built-in function for LU decomposition

- Example

```
>> a=[4 -2 -3 6; -6 7 6.5 -6; 1 7.5 6.25 5.5; -12 22 15.5 -1];
>> b=[12; -6.5; 16; 17];
>> [L, U, P]=lu(a)
L =
    1.0000         0         0         0
   -0.0833    1.0000         0         0
   -0.3333    0.5714    1.0000         0
    0.5000   -0.4286   -0.9250    1.0000
U =
  -12.0000   22.0000   15.5000   -1.0000
         0    9.3333    7.5417    5.4167
         0         0   -2.1429    2.5714
         0         0         0   -0.800
```

Decomposition of $[a]$ using MATLAB's `lu` function.

MATLAB's built-in function for LU decomposition

- Example

```
P =
     0     0     0     1
     0     0     1     0
     1     0     0     0
     0     1     0     0
>> y=L\(P*b)

y =
   17.0000
   17.4167
    7.7143
   -0.4000
>> x=U\y
x =
    2.0000
    4.0000
   -3.0000
    0.5000
```

Multiplying $[P][a]$ gives the pivoted $[a]$

```
>> P*a
ans =
  -12.0000   22.0000   15.5000   -1.0000
    1.0000    7.5000    6.2500    5.5000
    4.0000   -2.0000   -3.0000    6.0000
   -6.0000    7.0000    6.5000   -6.0000
```

Solve for y in Eq. (4.23).

Vector $[b]$ is multiplied by the permutation matrix.

Solve for x in Eq. (4.22).

$Ax=b$

$LU=PA$

$P^T LUx=b$

$L[Ux]=Pb$

Solve $Ly=Pb$

Then $Ux=y$

❖ Additional MATLAB built-in functions

- Example

Function	Description	Example

`inv(A)`	Inverse of a matrix.
	A is a square matrix. Returns the inverse of A.
		`>> A=[-3 1 0.6; 0.2 -4 3; 0.1 0.5 2];`
		`>> Ain=inv(A)`
		`Ain =`
		`-0.3310 -0.0592 0.1882`
		`-0.0035 -0.2111 0.3178`
		`0.0174 0.0557 0.4111`

Function	Description	Example

`d=det(A)`	Determinant of a matrix
	A is a square matrix, d is the determinant of A.
		`>> A=[-3 1 0.6; 0.2 -4 3; 0.1 0.5 2];`
		`>> d=det(A)`
		`d =`
		`28.7000`

4.9 Tri-diagonal Systems of Equations

❖ Tri-diagonal systems of linear equations

- Zero matrix coefficients except along the except along the diagonal, above-diagonal, and below-diagonal elements

$$
\begin{bmatrix}
A_{11} & A_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\
A_{21} & A_{22} & A_{23} & 0 & 0 & 0 & 0 & 0 \\
0 & A_{32} & A_{33} & A_{34} & 0 & 0 & 0 & 0 \\
 & & \cdots & \cdots & \cdots & & & \\
 & & & \cdots & \cdots & \cdots & & \\
0 & 0 & 0 & 0 & A_{n-2,\,n-3} & A_{n-2,\,n-2} & A_{n-2,\,n-1} & 0 \\
0 & 0 & 0 & 0 & 0 & A_{n-1,\,n-2} & A_{n-1,\,n-1} & A_{n-1,\,n} \\
0 & 0 & 0 & 0 & 0 & 0 & A_{n,\,n-1} & A_{n,\,n}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ \cdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix}
=
\begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \cdots \\ \cdots \\ B_{n-2} \\ B_{n-1} \\ B_n \end{bmatrix}
$$

- The system can be solved with the standard methods.
  - A large number of zero elements are stored and a large number of needles operations are executed.
- To save computer memory and computing time, special numerical methods have been developed.
  - Ex)

Thomas algorithm for solving tri-diagonal systems

- Similar to the Gaussian elimination method
  - Upper triangular matrix $\Rightarrow$ back substitution
- Much more efficient because only the nonzero elements of the matrix of coefficients are stored, and only the necessary operations are executed.
- Procedure
  - Assigning the non-zero elements of the TDM $[A]$ to three vectors
  - Diagonal vector d , above diagonal vector a, below diagonal vector b
    - $d_i = A_{ii}$, $a_i = A_{i,i+1}$, $b_i = A_{i,i-1}$

$$\begin{bmatrix} d_1 & a_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_2 & d_2 & a_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_3 & d_3 & a_3 & 0 & 0 & 0 & 0 \\ & & \cdots & \cdots & \cdots & & & \\ & & & \cdots & \cdots & \cdots & & \\ 0 & 0 & 0 & 0 & b_{n-2} & d_{n-2} & a_{n-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & b_{n-1} & d_{n-1} & a_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & b_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ \cdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \cdots \\ \cdots \\ B_{n-2} \\ B_{n-1} \\ B_n \end{bmatrix}$$

Only vectors b, d and a are stored!

Thomas algorithm for solving tri-diagonal systems

- Procedure
  - First row is normalized by dividing the row by d_1.
  - Element b_2 is eliminated.
  - Second row is normalized by dividing the row by d'_2.
  - Element b_3 is eliminated.

$$a'_1 = a_1/d_1 \text{ and } B'_1 = B_1/d_1$$

$$d'_2 = d_2 - b_2 a'_1 \text{, and } B'_2 = B_2 - B_1 b_2$$

$$\begin{bmatrix} 1 & a'_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ b_2 & d_2 & a_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_3 & d_3 & a_3 & 0 & 0 & 0 & 0 \\ & & \cdots & \cdots & \cdots & & & \\ & & & \cdots & \cdots & \cdots & & \\ 0 & 0 & 0 & 0 & b_{n-2} & d_{n-2} & a_{n-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & b_{n-1} & d_{n-1} & a_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & b_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ \cdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} B'_1 \\ B_2 \\ B_3 \\ \cdots \\ \cdots \\ B_{n-2} \\ B_{n-1} \\ B_n \end{bmatrix}$$

$$\begin{bmatrix} 1 & a'_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d'_2 & a_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_3 & d_3 & a_3 & 0 & 0 & 0 & 0 \\ & & \cdots & \cdots & \cdots & & & \\ & & & \cdots & \cdots & \cdots & & \\ 0 & 0 & 0 & 0 & b_{n-2} & d_{n-2} & a_{n-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & b_{n-1} & d_{n-1} & a_{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & b_n & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ \cdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} B'_1 \\ B'_2 \\ B_3 \\ \cdots \\ \cdots \\ B_{n-2} \\ B_{n-1} \\ B_n \end{bmatrix}$$

Thomas algorithm for solving tri-diagonal systems

- Procedure
  - This process continues row after row until the matrix is transformed to be upper triangular one.

$$
\begin{bmatrix}
1 & a'_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & a'_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & a'_3 & 0 & 0 & 0 & 0 \\
 & & \cdots & \cdots & \cdots & & & \\
 & & & \cdots & \cdots & \cdots & & \\
0 & 0 & 0 & 0 & 0 & 1 & a'_{n-2} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & a'_{n-1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ \cdots \\ x_{n-2} \\ x_{n-1} \\ x_n \end{bmatrix}
=
\begin{bmatrix} B'_1 \\ B''_2 \\ B''_3 \\ \cdots \\ \cdots \\ B''_{n-2} \\ B''_{n-1} \\ B''_n \end{bmatrix}
$$

  - Back substitution

Thomas algorithm for solving tri-diagonal systems

- Mathematical form
  - Step 1
    - Define the vectors $b = [0, b_2, b_3, \ldots, b_n]$, $d = [d_1, d_2, \ldots, d_n]$, $a = [a_1, a_2, \ldots, a_{n-1}]$, and $B = [B_1, B_2, \ldots, B_n]$.
  - Step 2
    - Calculate: $a_1 = \dfrac{a_1}{d_1}$ and $B_1 = \dfrac{B_1}{d_1}$
  - Step 3
    - For $i = 2, 3, \ldots, n-1$ and

1 $\quad a_{i-1}$ $\quad B_{i-1}$

b_i $\quad d_i$ $\quad a_i$ $\quad B_i$

  - Step 4

$$B_n = \frac{B_n - b_n B_{n-1}}{d_n - b_n a_{n-1}}$$

  - Back substitution

$$x_n = B_n \qquad x_i = B_i - a_i x_{i+1}$$

Thomas algorithm for solving tri-diagonal systems

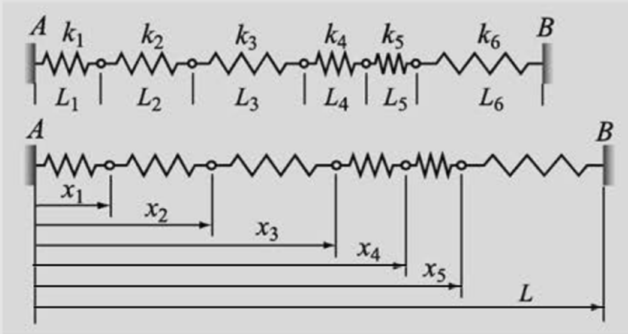
- Example

Example 4-9: Solving a tridiagonal system of equations using the Thomas algorithm.

Six springs with different spring constants k_i and unstretched lengths L_i are attached to each other in series. The endpoint B is then displaced such that the distance between points A and B is $L = 1.5$ m. Determine the positions $x_1, x_2, \ldots, x_5$ of the endpoints of the springs.

The spring constants and the unstretched lengths of the springs are:

spring	1	2	3	4	5	6
k (kN/m)	8	9	15	12	10	18
L (m)	0.18	0.22	0.26	0.19	0.15	0.30



$$F = k\delta$$

$$
\begin{aligned}
k_1(x_1 - L_1) &= k_2[(x_2 - x_1) - L_2] \\
k_2[(x_2 - x_1) - L_2] &= k_3[(x_3 - x_2) - L_3] \\
k_3[(x_3 - x_2) - L_3] &= k_4[(x_4 - x_3) - L_4] \\
k_4[(x_4 - x_3) - L_4] &= k_5[(x_5 - x_4) - L_5] \\
k_5[(x_5 - x_4) - L_5] &= k_6[(L - x_5) - L_6]
\end{aligned}
$$

$$
\begin{bmatrix}
k_1 + k_2 & -k_2 & 0 & 0 & 0 \\
-k_2 & k_2 + k_3 & -k_3 & 0 & 0 \\
0 & -k_3 & k_3 + k_4 & -k_4 & 0 \\
0 & 0 & -k_4 & k_4 + k_5 & -k_5 \\
0 & 0 & 0 & -k_5 & k_5 + k_6
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
=
\begin{bmatrix}
k_1 L_1 - k_2 L_2 \\
k_2 L_2 - k_3 L_3 \\
k_3 L_3 - k_4 L_4 \\
k_4 L_4 - k_5 L_5 \\
k_5 L_5 + k_6 L - k_6 L_6
\end{bmatrix}
$$

Thomas algorithm for solving tri-diagonal systems

- Example

```
function x = Tridiagonal(A,B)
% The function solve a tridiagonal system of linear equations [a][x]=[b]
% using Thomas algorithm.
% Input variables:
% A  The matrix of coefficients.
% B  A column vector of constants.
% Output variable:
% x  A colum vector with the solution.

[nR, nC] = size(A);
for i = 1:nR
    d(i) = A(i,i);
end
for i = 1:nR-1
    ad(i) = A(i,i+1);
end
for i = 2:nR
    bd(i) = A(i,i-1);
end
ad(1) = ad(1)/d(1);
B(1) = B(1)/d(1);
for i = 2:nR-1
    ad(i) = ad(i)/(d(i)-bd(i)*ad(i-1));
    B(i)=(B(i)-bd(i)*B(i-1))/(d(i)-bd(i)*ad(i-1));
end
B(nR)=(B(nR)-bd(nR)*B(nR-1))/(d(nR)-bd(nR)*ad(nR-1));
x(nR,1) = B(nR);
for i = nR-1:-1:1
    x(i,1) = B(i)-ad(i)*x(i+1);
end
```

$$a_i = \frac{a_i}{d_i - b_i a_{i-1}} \text{ and } B_i = \frac{B_i - b_i B_{i-1}}{d_i - b_i a_{i-1}}$$

$$B_n = \frac{B_n - b_n B_{n-1}}{d_n - b_n a_{n-1}}$$

$$x_n = B_n \qquad x_i = B_i - a_i x_{i+1}$$

Thomas algorithm for solving tri-diagonal systems

- Example

```
clear all
% Example 4-9
k1 = 8000; k2 = 9000; k3 = 15000; k4 = 12000; k5 = 10000; k6 = 18000;
L = 1.5; L1 = 0.18; L2 = 0.22; L3 = 0.26; L4 = 0.19; L5 = 0.15; L6 = 0.30;
a = [k1 + k2, -k2, 0, 0, 0; -k2, k2+k3, -k3, 0, 0; 0, -k3, k3+k4, -k4, 0
    0, 0, -k4, k4+k5, -k5; 0, 0, 0, -k5, k5+k6]
b = [k1*L1 - k2*L2; k2*L2 - k3*L3; k3*L3 - k4*L4; k4*L4 - k5*L5; k5*L5 + k6*L - k6*L6]
Xs = Tridiagonal(a,b)
```

4.10 Error, Residual, Norms, and Condition Number

Error and residual

- True error

  $$[e] = [x_{TS}] - [x_{NS}]$$

  - True error cannot be calculated because the true solution is not known.

- Residual
  - An alternative measure of the accuracy of a solution

  $[r] \;=$

  - This does not really indicate how small the error is.
  - It shows how well the right-hand side of the equations is satisfied when $[x_{NS}]$ is substituted for $[x]$ in the original equations.
  - It is possible to have an approximate numerical solution that has a large true error but gives a small residual.

- **Norm**

4.10 Error, Residual, Norms, and Condition Number

Error and residual

- Example

Example 4-10: Error and residual.

The true (exact) solution of the system of equations:

$$1.02x_1 + 0.98x_2 = 2$$
$$0.98x_1 + 1.02x_2 = 2$$

is $x_1 = x_2 = 1$.

Calculate the true error and the residual for the following two approximate solutions:

(a) $x_1 = 1.02, \quad x_2 = 1.02.$

(b) $x_1 = 2, \quad x_2 = 0.$

$$[e] = [x_{TS}] - [x_{NS}] = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1.02 \\ 1.02 \end{bmatrix} = \begin{bmatrix} -0.02 \\ -0.02 \end{bmatrix}$$

$$[r] = [b] - [a][x_{NS}] = [b] - [a][x_{NS}] = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 1.02 & 0.98 \\ 0.98 & 1.02 \end{bmatrix} \begin{bmatrix} 1.02 \\ 1.02 \end{bmatrix} = \begin{bmatrix} -0.04 \\ -0.04 \end{bmatrix}$$

$$[e] = [x_{TS}] - [x_{NS}] = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$[r] = [b] - [a][x_{NS}] = [b] - [a][x_{NS}] = \begin{bmatrix} 2 \\ 2 \end{bmatrix} - \begin{bmatrix} 1.02 & 0.98 \\ 0.98 & 1.02 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.04 \\ 0.04 \end{bmatrix}$$

Small residual does not necessarily guarantee a small error.
Whether or not a small residual implies a small error depends on the "magnitude" of the matrix $[a]$.

Norms and condition number

- Norm
  - A real number assigned to a matrix or vector that satisfies the following four properties;

$$\|[a]\| \geq 0 \quad \text{and} \quad \|[a]\| = 0 \quad \text{only if} \quad [a] = 0$$

$$\|\alpha[a]\| = |\alpha|\,\|[a]\|$$ $[a]$ and $[-a]$: same "magnitude" $[10a]$: 10 times the magnitude of $[a]$

$$\|[a][x]\| \leq \|[a]\|\,\|[x]\|$$

$$\|[a+b]\| \leq \|[a]\| + \|[b]\|$$ Triangle inequality

- Vector norms
  - Infinity norm $\|v\|_\infty = \max_{1 \leq i \leq n} |v_i|$
  - 1-norm $\|v\|_1 = \sum_{i=1}^{n} |v_i|$
  - Euclidean 2-norm $\|v\|_2 = \left(\sum_{i=1}^{n} v_i^2\right)^{1/2}$

Norms and condition number

- Matrix norms
  - Infinity norm

$$\|[a]\|_\infty = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}|$$

Summation is done for each row

  - 1-norm

$$\|[a]\|_1 = \max_{1 \le j \le n} \sum_{i=1}^{n} |a_{ij}|$$

Summation is done for each column

  - 2-norm

$$\|[a]\|_2 = max\left(\frac{\|[a][v]\|}{\|[v]\|}\right)$$

← Eigenvector

$[a] = [u][d][v]$ The largest value of the diagonal elements of $[d]$

$[u]^{-1} = [u]^T$

  - Euclidean norm for an $m \times n$ matrix $[a]$

Frobenius norm

$$\|[a]\|_{Euclidean} = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2\right)^{1/2}$$

4.10 Error, Residual, Norms, and Condition Number

Norms and condition number

- Using norms to determine bounds on the error of numerical solutions
  - Residual written in terms of the error

$$[r] = [a][x_{TS}] - [a][x_{NS}] = [a]([x_{TS}] - [x_{NS}]) = [a][e]$$

  - Error

$$[e] = [a]^{-1}[r]$$

  - By $\|[a][x]\| \le \|[a]\|\,\|[x]\|$

$$\|[r]\| = \|[a][e]\| \le \|[a]\|\,\|[e]\| \qquad \|[e]\| = \left\|[a]^{-1}[r]\right\| \le \left\|[a]^{-1}\right\|\,\|[r]\|$$

$$\frac{\|[r]\|}{\|[a]\|} \le \|[e]\| = \left\|[a]^{-1}[r]\right\| \le \left\|[a]^{-1}\right\|\,\|[r]\|$$

  - Relative error and relative residual $\|[e]\|/\|[x_{TS}]\|$ $\quad$ $\|[r]\|/\|[b]\|$

$$\frac{1}{\|[a]\|}\frac{\|b\|}{\|[x_{TS}]\|}\frac{\|[r]\|}{\|[b]\|} \le \frac{\|[e]\|}{\|[x_{TS}]\|} \le \left\|[a]^{-1}\right\|\frac{\|[b]\|}{\|[x_{TS}]\|}\frac{\|[r]\|}{\|[b]\|}$$

Norms and condition number

- Using norms to determine bounds on the error of numerical solutions

$$\frac{1}{\|[a]\|}\frac{\|b\|}{\|[x_{TS}]\|}\frac{\|[r]\|}{\|[b]\|} \le \frac{\|[e]\|}{\|[x_{TS}]\|} \le \|[a]^{-1}\|\frac{\|[b]\|}{\|[x_{TS}]\|}\frac{\|[r]\|}{\|[b]\|}$$

- Definition of true solution

$$[a][x_{TS}] = [b] \qquad [x_{TS}] = [a]^{-1}[b]$$

  - By $\|[a][x]\| \le \|[a]\|\|[x]\|$ $\Rightarrow$ $\|[b]\| \le \|[a]\|\|[x_{TS}]\|$ $\Rightarrow$ $\frac{\|[b]\|}{\|[x_{TS}]\|} \le \|[a]\|$

    $\Rightarrow$ $\|[x_{TS}]\| \le \|[a]^{-1}\|\|[b]\|$ $\Rightarrow$ $\frac{1}{\|[a]^{-1}\|} \le \frac{\|[b]\|}{\|[x_{TS}]\|}$

$$\frac{1}{\|[a]\|\|[a]^{-1}\|}\frac{\|[r]\|}{\|[b]\|} \le \frac{\|[e]\|}{\|[x_{TS}]\|} \le \|[a]^{-1}\|\|[a]\|\frac{\|[r]\|}{\|[b]\|}$$

4.10 Error, Residual, Norms, and Condition Number

Norms and condition number

- Condition number

$$Cond[a] = \|[a]\| \, \|[a]^{-1}\|$$

  - The condition number of the identity matrix is 1.
  - The condition number of any other matrix is 1 or greater.
  - If the condition number is approximately 1, then the true relative error is of the same order of magnitude as the relative residual.
  - If the condition number is much larger than 1, then a small relative residual does not necessarily imply a small true relative error.
  - For a given matrix, the value of the condition number depends on the matrix norm that is used.
  - The inverse of a matrix has to be known in order to calculate the condition number of the matrix.

Norms and condition number

- Example

Example 4-11: Calculating error, residual, norm and condition number.

Consider the following set of four equations (the same that was solved in Example 4-8).

$$9x_1 - 2x_2 + 3x_3 + 2x_4 = 54.5$$
$$2x_1 + 8x_2 - 2x_3 + 3x_4 = -14$$
$$-3x_1 + 2x_2 + 11x_3 - 4x_4 = 12.5$$
$$-2x_1 + 3x_2 + 2x_3 + 10x_4 = -21$$

The true solution of this system is $x_1 = 5$, $x_2 = -2$, $x_3 = 2.5$, and $x_4 = -1$. When this system was solved in Example 4-8 with the Gauss–Seidel iteration method, the numerical solution in the sixth iteration was $x_1 = 4.98805$, $x_2 = -1.99511$, $x_3 = 2.49806$, and $x_4 = -1.00347$.

(*a*) Determine the true error, $[e]$, and the residual, $[r]$.

(*b*) Determine the infinity norms of the true solution, $[x_{TS}]$, the error, $[e]$, the residual, $[r]$, and the vector $[b]$.

(*c*) Determine the inverse of $[a]$, the infinity norm of $[a]$ and $[a]^{-1}$, and the condition number of the matrix $[a]$.

(*d*) Substitute the quantities from parts (*b*) and (*c*) in Eq. (4.85) and discuss the results.

$$\frac{1}{\|[a]\| \|[a]^{-1}\|} \frac{\|[r]\|}{\|[b]\|} \le \frac{\|[e]\|}{\|[x_{TS}]\|} \le \|[a]^{-1}\| \|[a]\| \frac{\|[r]\|}{\|[b]\|}$$

Meaning

- System in which small variations in the coefficients cause large changes in the solution.
- Ill-conditioned systems generally has a condition number that is significantly greater than 1.

$$\begin{aligned} 6x_1 - 2x_2 &= 10 \\ 11.5x_1 - \boxed{3.85}x_2 &= 17 \end{aligned}$$

$$x_1 = \frac{a_{12}b_2 - a_{22}b_1}{a_{12}a_{21} - a_{11}a_{22}} = \frac{-2 \cdot 17 - (-3.85 \cdot 10)}{-2 \cdot 11.5 - (6 \cdot -3.85)} = \frac{4.5}{0.1} = 45$$

$$x_2 = \frac{a_{21}b_1 - a_{11}b_2}{a_{12}a_{21} - a_{11}a_{22}} = \frac{11.5 \cdot 10 - (6 \cdot 17)}{-2 \cdot 11.5 - (6 \cdot -3.85)} = \frac{13}{0.1} = 130$$

$$\begin{aligned} 6x_1 - 2x_2 &= 10 \\ 11.5x_1 - \boxed{3.84}x_2 &= 17 \end{aligned}$$

$$x_1 = \frac{a_{12}b_2 - a_{22}b_1}{a_{12}a_{21} - a_{11}a_{22}} = \frac{-2 \cdot 17 - (-3.84 \cdot 10)}{-2 \cdot 11.5 - (6 \cdot -3.84)} = \frac{4.4}{0.04} = 110$$

$$x_2 = \frac{a_{21}b_1 - a_{11}b_2}{a_{12}a_{21} - a_{11}a_{22}} = \frac{11.5 \cdot 10 - (6 \cdot 17)}{-2 \cdot 11.5 - (6 \cdot -3.84)} = \frac{13}{0.04} = 325$$

- Large difference between denominators of the two equations.

Determinant of $[a]$

Example

- Condition number

$$\begin{aligned} 6x_1 - 2x_2 &= 10 \\ 11.5x_1 - 3.85x_2 &= 17 \end{aligned}$$

$$[a] = \begin{bmatrix} 6 & -2 \\ 11.5 & -3.85 \end{bmatrix} \quad \text{and} \quad [a]^{-1} = \begin{bmatrix} 38.5 & -20 \\ 115 & -60 \end{bmatrix}$$

  - Using the infinity norm and 1-norm

$$Cond[a] = \|[a]\|\left\|[a]^{-1}\right\| = 15.35 \cdot 175 = 2686.25$$

$$Cond[a] = \|[a]\|\left\|[a]^{-1}\right\| = 17.5 \cdot 153.5 = 2686.25$$

  - 2-norm

$$Cond[a] = \|[a]\|\left\|[a]^{-1}\right\| = 13.6774 \cdot 136.774 = 1870.7$$

With any norm used,
the condition number is much larger than 1!

Comment

- Numerical solution of an ill-conditioned system of equations
  - High probability of large error
  - Difficult to quantify the value of the condition number criterion
- Need to check only
  - Whether or not the condition number is much larger than 1