# Matrices and Linear Systems of Equations

- ♣ Matrix notations and operations
- ♣ A Linear System of Equations
- ♣ Matlab Solutions for A Linear System of Equations
- ♣ Elementary row operations
- ♣ LU-decomposition
- ♣ Gaussian Elimination with Partial Pivoting
- ♣ Row-echelon form
- ♣ Matrix inverse and Transpose of a Matrix
- ♣ Computing An Inverse Matrix By Elementary Row Operations
- ♣ Some special matrices
- ♣ Analysis of Gaussian Elimination and Back Substitution

# Matrix Notations and Operations

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \cdot & \cdot & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_m \end{bmatrix}$$

- $A = [a_{ij}]$ *or* $(a_{ij})$, $i = 1, 2, \cdots, m$; $j = 1, 2, \cdots, n$ or denote $A \in R^{m \times n}$
- An $m \times 1$ matrix is called a column vector, denote $\mathbf{b} \in R^m$
- A $1 \times n$ matrix is called a row vector, denote $\mathbf{y}^t$ *or* $\mathbf{y}' \in R^n$, where $\mathbf{y} = [y_1, \ y_2, \ \cdots, \ y_n]$
- Let $X, Y \in R^{m \times n}$, $X = [x_{ij}]$, $Y = [y_{ij}]$, define $X + Y = [x_{ij} + y_{ij}]$
- $C = AB$ for $A \in R^{m \times n}$, $B \in R^{p \times q}$ is well-defined only when $n = p$

  $C = [c_{ij}] \in R^{m \times q}$, where $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$
- $C = A[\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_q] = [A\mathbf{b}_1, A\mathbf{b}_2, \cdots, A\mathbf{b}_q]$
- Matrix multiplication is associative $(AB)C = A(BC)$ but not commutative $AB \neq BA$

◇ The following linear system of equations can be written as $A\mathbf{x} = \mathbf{b}$ with the *augmented* matrix $[A \mid \mathbf{b}]$.

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdot & \cdot & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdot & \cdot & + & a_{2n}x_n & = & b_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdot & \cdot & + & a_{mn}x_n & = & b_m \end{array}$$

The system is *homogeneous* if $b_1 = b_2 = \cdots = b_m = 0$, *overdetermined* if $m > n$, and *underdetermined* if $m < n$

# Overdetermined, Underdetermined, Homogeneous Systems

$$
\begin{array}{ccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\
. & & . & & . & & . & & . \\
. & & . & & . & & . & & . \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m
\end{array}
$$

$A\mathbf{x} = \mathbf{b}$

**Definition:** A linear system is said to be *overdetermined* if there are more equations than unknowns $(m > n)$, *underdetermined* if $m < n$, *homogeneous* if $b_i = 0, \ \forall\ 1 \leq i \leq m$.

$$
(A) \quad \begin{array}{rcrcr}
x & + & y & = & 1 \\
x & - & y & = & 3 \\
-x & + & 2y & = & -2
\end{array}
\qquad
(B) \quad \begin{array}{rcrcr}
x & + & y & = & 3 \\
x & - & y & = & 1 \\
2x & + & y & = & 5
\end{array}
\qquad
(C) \quad \begin{array}{rcrcr}
x & + & y & = & 2 \\
2x & + & 2y & = & 4 \\
-x & - & y & = & -2
\end{array}
$$

(A) has no solution, (B) has unique solution, (C) has infinitely many solutions

$$
(D) \quad \begin{array}{rcrcrcr}
x & + & 2y & + & z & = & -1 \\
2x & + & 4y & + & 2z & = & 3
\end{array}
\qquad
(E) \quad \begin{array}{rcrcrcr}
x & + & 2y & + & z & = & 5 \\
2x & - & y & + & z & = & 3
\end{array}
$$

(D) has no solution, (E) has infinitely many solutions

# A Direct Solution of Linear Systems

A linear system

$$
\begin{array}{rcrcrcr}
2x & + & y & + & z & = & 5 \\
4x & - & 6y & & & = & -2 \\
-2x & + & 7y & + & 2z & = & 9
\end{array}
$$

A matrix representation

$$
A\mathbf{x} = \mathbf{b}, \quad or \quad \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 5 \\ -2 \\ 9 \end{bmatrix}.
$$

♣ Solution using MATLAB

$\rangle\rangle$ $A = [2, 1, 2;\ 4, -6, 0;\ \ -2, 7, 2];$

$\rangle\rangle$ $b = [5,\ \ -2,\ 9]';$

$\rangle\rangle$ $x = A\backslash b \quad (x = [1;\ 1;\ 2])$

# Elementary Row Operations

**(1)** Interchange two rows: $A_r \leftrightarrow A_s$

**(2)** Multiply a row by a nonzero real number: $A_r \leftarrow \alpha A_r$

**(3)** Replace a row by its sum with a multiple of another row: $A_s \leftarrow \alpha A_r + A_s$

$$E_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \quad E_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix},$$

♣ *Example*

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix}, \quad E_1 A = \begin{bmatrix} 4 & -6 & 0 \\ 2 & 1 & 1 \\ -2 & 7 & 2 \end{bmatrix}, \quad E_2 A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ 4 & -14 & -4 \end{bmatrix}, \quad E_3 A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ 0 & 8 & 3 \end{bmatrix}$$

Let

$$L_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix}$$

Then

$$L_3 L_2 L_1 A = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix} = U \quad (Upper - \Delta)$$

$$A = (L_1^{-1} L_2^{-1} L_3^{-1}) U = LU, \quad \textit{where } L \textit{ is unit lower} - \Delta$$

# LU-Decomposition

If a matrix $A$ can be decomposed into the product of a unit lower-$\Delta$ matrix $L$ and an upper-$\Delta$ matrix $U$, then the linear system $A\mathbf{x} = \mathbf{b}$ can be written as $LU\mathbf{x} = \mathbf{b}$. The problem is reduced to solving two simpler triangular systems $L\mathbf{y} = \mathbf{b}$ and $U\mathbf{x} = \mathbf{y}$ by forward and back substitutions.

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix}, \quad L_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then

$$L_3 L_2 L_1 A = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix} = U \quad \Rightarrow \quad A = L_1^{-1} L_2^{-1} L_3^{-1} U = LU$$

where

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix} = LU$$

◇ If $\mathbf{b} = [5, -2, 9]^t$, then $\mathbf{y} = [5, -12, 2]^t$ and $\mathbf{x} = [1, 1, 2]^t$

$$B = \begin{bmatrix} 2 & -2 & 3 \\ -2 & 3 & -4 \\ 4 & -3 & 7 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ 5 \end{bmatrix} \quad \Rightarrow \quad \mathbf{x} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

# Gaussian Elimination with Partial Pivoting

Not every matrix $A$ (even if $A$ is nonsingular) can be decomposed into the product of a unit lower-$\Delta$ matrix $L$ and an upper-$\Delta$ matrix $U$ by directly using Gaussian elimination. Whereas, for a nonsingular matrix $A$, Gaussian Elimination with Partial Pivoting will be introduced to overcome this problem later.
A linear system

$$\begin{array}{rcrcrcr} & & y & + & 2z & = & 0 \\ x & + & y & + & 2z & = & 1 \\ 2x & + & 2y & + & z & = & -1 \end{array}$$

A matrix representation

$$A\mathbf{x} = \mathbf{b}, \quad or \quad \begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}.$$

A corresponding augmented matrix for $A\mathbf{x} = \mathbf{b}$ is

$$[A|\mathbf{b}] = \begin{bmatrix} 0 & 1 & 2 & | & 0 \\ 1 & 1 & 2 & | & 1 \\ 2 & 2 & 1 & | & -1 \end{bmatrix}$$

or equivalently

$$E_{13} \times [A|\mathbf{b}] = \begin{bmatrix} 2 & 2 & 1 & | & -1 \\ 1 & 1 & 2 & | & 1 \\ 0 & 1 & 2 & | & 0 \end{bmatrix}$$

♣ Solution by using MATLAB

$\rangle\rangle$ $A = [0, 1, 2; 1, 1, 2; 2, 2, 1];$

$\rangle\rangle$ $b = [0, 1, -1]';$

$\rangle\rangle$ $x = A\backslash b \quad (x = [1; \ -2; \ 1])$

# Row Echelon Form

*Definition:* A matrix is said to be in row echelon form if

**(a)** The first nonzero entry in each row is 1

**(b)** If row k does not consist entirely of 0s, the number of leading zero entries in row k+1 is greater than that in row k

**(c)** If there are rows whose entries are all zero, they are below the rows having nonzero entries

♣ *Example:* Matrices are in row echelon form

$$\begin{bmatrix} 1 & 4 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 3 & 0 & 1 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

♣ *Example:* Matrices which are not in row echelon form

$$\begin{bmatrix} 2 & 4 & 6 \\ 0 & 3 & 5 \\ 0 & 0 & 4 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

# Matrix Inverse and Transpose

A matrix $A \in R^{n\times n}$ is invertible iff $\exists$ $B$ such that $AB = BA = I_n$. If the inverse $B$ exists, $B$ is unique and is denoted by $A^{-1}$

**(1)** A matrix is *nonsingular* if it is invertible

**(2)** If there exists two inverses B and C, then B=C

**(3)** The inverse of $A^{-1}$ is $A$

**(4)** A diagonal matrix is invertible if none of its diagonal entries is 0

**(5)** If $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, then $A^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

**(6)** If both $A$ and $B$ are invertible, then $(AB)^{-1} = B^{-1}A^{-1}$

**(7)** The transpose of $A = [a_{ij}]$ is $A^t = [a_{ji}]$

**(8)** $A \in R^{m\times n} \Rightarrow A^t \in R^{n\times m}$, and $(AB)^t = B^t A^t$

# Computing An Inverse Matrix By Elementary Row Operations

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \\ 2 & 4 & 7 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I$$

$$E_1 A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 2 & 4 & 7 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = E_1 I$$

$$E_2 E_1 A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} = E_2 E_1 I$$

$$E_3 E_2 E_1 A = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 3 & -2 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} = E_3 E_2 E_1 I$$

$$E_4 E_3 E_2 E_1 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 9 & -2 & -3 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} = E_4 E_3 E_2 E_1 I = A^{-1}$$

where the elementary matrices are

$$E_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}, \quad E_3 = \begin{bmatrix} 1 & -2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad E_4 = \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# Computing Matrix Inverse

◇ Gauss-Jordan Method for Computing $A^{-1}$ with $O(n^3)$

◇ By solving $A[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n]$

*Example*

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix} \quad \Rightarrow \quad A^{-1} = \begin{bmatrix} \frac{3}{4} & \frac{-5}{16} & \frac{-3}{8} \\ \frac{1}{2} & \frac{-3}{8} & \frac{-1}{4} \\ -1 & 1 & 1 \end{bmatrix}$$

◇ The inverse of a unit lower-$\Delta$ matrix is also unit lower-$\Delta$

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{4} & 1 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix} \quad \Rightarrow \quad L^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{4} & 1 & 0 & 0 \\ -\frac{1}{4} & -\frac{1}{3} & 1 & 0 \\ -\frac{1}{4} & -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}$$

# Some Special Matrices

$A = [a_{ij}] \in R^{n \times n}$

- *Diagonal* if $a_{ij} = 0 \; \forall \; i \neq j$
- $Lower - \Delta$ if $a_{ij} = 0$ if $j > i$
- $Unit\ lower - \Delta$ if $A$ is lower-$\Delta$ with $a_{ii} = 1$
- $Lower\ Hessenberg$ if $a_{ij} = 0$ for $j > i + 1$
- *Band* matrix with *bandwidth* $2k + 1$ if $a_{ij} = 0$ for $|i - j| > k$

□ A band matrix with *bandwidth* 1 is *diagonal*

□ A band matrix with *bandwidth* 3 is *tridiagonal*

□ A band matrix with *bandwidth* 5 is *pentadiagonal*

□ A lower and upper Hessenberg matrix is *tridiagonal*

$$A_1 = \begin{bmatrix} 7 & 0 & 0 \\ 1 & 8 & 0 \\ 2 & 3 & 9 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 5 & 2 & 0 & 0 \\ 1 & 6 & 4 & 0 \\ 2 & 3 & 7 & 3 \\ 1 & 2 & 0 & 8 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 5 & 2 & 0 & 0 \\ 1 & 6 & 4 & 0 \\ 0 & 3 & 7 & 3 \\ 0 & 0 & 2 & 8 \end{bmatrix}$$

# Elementary Matrices (Row Operations)

**(1)** Interchange two rows $E_{ij}: \quad R_i \longleftrightarrow R_j$

**(2)** Multiply a row by a nonzero real number $E_k(\alpha): \quad \alpha R_k$

**(3)** Replace a row by its sum with a multiple of another row $E_{ij}(\gamma): \quad \gamma R_i + R_j \longrightarrow R_j$

$$E_{12} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad E_3(-2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \quad E_{13}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix},$$

♣ *Example*

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix}, \quad E_{12}A = \begin{bmatrix} 4 & -6 & 0 \\ 2 & 1 & 1 \\ -2 & 7 & 2 \end{bmatrix}, \quad E_3(-2)A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ 4 & -14 & -4 \end{bmatrix}, \quad E_{13}(1)A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ 0 & 8 & 3 \end{bmatrix}$$

For

$$E_{23}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad E_{13}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad E_{12}(-2) = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix}$$

We have

$$E_{23}(1)E_{13}(1)E_{12}(-2)A = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix} = U \ (Upper{-}\Delta) \Rightarrow A = [E_{12}(-2)]^{-1}[E_{13}(1)]^{-1}[E_{23}(1)]^{-1}U$$

# LU-Decomposition

If a matrix $A$ can be decomposed into the product of a unit lower-$\Delta$ matrix $L$ and an upper-$\Delta$ matrix $U$, then the linear system $A\mathbf{x} = \mathbf{b}$ can be written as $LU\mathbf{x} = \mathbf{b}$. The problem is reduced to solving two simpler triangular systems $L\mathbf{y} = \mathbf{b}$ and $U\mathbf{x} = \mathbf{y}$ by forward and back substitutions.

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix}, \quad E_{23}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad E_{13}(1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \quad E_{12}(-2) = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then

$$E_{23}(1)E_{13}(1)E_{12}(-2)A = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix} = U \;\Rightarrow\; A = [E_{12}(-2)]^{-1}[E_{13}(1)]^{-1}[E_{23}(1)]^{-1}U = LU$$

where

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix} = LU$$

◇ If $\mathbf{b} = [5, -2, 9]^t$ and use $B = [A|\mathbf{b}]$ instead of $A$, the above processes will obtain $\mathbf{y} = [5, -12, 2]^t$ and $\mathbf{x} = [1, 1, 2]^t$

□ The inverse of a (unit) $lower - \Delta$ matrix is (unit) $lower - \Delta$.

□ The product of (unit) $lower - \Delta$ matrices is (unit) $lower - \Delta$.

# Analysis of Gaussian Elimination

♣ *Algorithm*

for $i = 1, 2, \cdots, n-1$

for $k = i+1, i+2, \cdots, n$

$m_{ki} \leftarrow a_{ki}/a_{ii}$ if $a_{ii} \neq 0$

$a_{ki} \leftarrow m_{ki}$

for $j = i+1, i+2, \cdots, n$

$a_{kj} \leftarrow a_{kj} - m_{ki} * a_{ij}$

endfor

endfor

endfor

- *The Worst Computational Complexity is* $O(\frac{2}{3}n^3)$

1. # of divisions are $(n-1) + (n-2) + \cdots + 1 = \frac{n(n-1)}{2}$
2. # of multiplications are $(n-1)^2 + (n-2)^2 + \cdots + 1^2 = \frac{n(n-1)(2n-1)}{6}$
3. # of subtractions are $(n-1)^2 + (n-2)^2 + \cdots + 1^2 = \frac{n(n-1)(2n-1)}{6}$

# The Analysis of Gaussian Elimination and Back Substitution to solve Ax=b

$\frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{7}{6}n$

$$
\begin{array}{lccccccccc}
R_1: & a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\
R_2: & a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\
 & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
R_i: & a_{i1}x_1 & + & a_{i2}x_2 & + & \cdots & + & a_{in}x_n & = & b_i \\
 & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
R_n: & a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & b_n
\end{array}
$$

By Gaussian Elimination, we need $C_1 = [\sum_{k=1}^{n}(k+1)(k-1) \ + \ \sum_{k=1}^{n} k(k-1)]$ flops to reduce the above linear system of equations equivalent to the following upper triangular system.

$$
\begin{array}{lccccccccc}
R_1: & u_{11}x_1 & + & u_{12}x_2 & + & \cdots & + & u_{1n}x_n & = & c_1 \\
 & & & \vdots & & \vdots & & \vdots & & \vdots \\
R_i: & & & u_{ii}x_i & + & \cdots & + & u_{in}x_n & = & c_i \\
 & & & & & \vdots & & \vdots & & \vdots \\
R_n: & & & & & & & u_{nn}x_n & = & c_n
\end{array}
$$

We need $C_2 = \sum_{k=1}^{n}(2k-1) = n^2$ flops to solve an upper triangular linear system of equations. Therefore, the total number of flops of solving $A\mathbf{x} = \mathbf{b}$ is summarized as

$$C_1 + C_2 = \frac{2}{3}n^3 + \frac{3}{2}n^2 - \frac{7}{6}n$$

# PA=LU

Let $A \in R^{4\times 4}$, by Gaussian Elimination with Partial Pivoting, we might have

$$L_3P_3L_2P_2L_1P_1A = U$$

where $P_1$ corresponds to $R_1 \longleftrightarrow R_4$, $P_2$ corresponds to $R_2 \longleftrightarrow R_4$, $P_3$ corresponds to $R_3 \longleftrightarrow R_4$, and

$$L_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha_1 & 1 & 0 & 0 \\ \alpha_2 & 0 & 1 & 0 \\ \alpha_3 & 0 & 0 & 1 \end{bmatrix}, \quad L_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \alpha_4 & 1 & 0 \\ 0 & \alpha_5 & 0 & 1 \end{bmatrix}, \quad L_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \alpha_6 & 1 \end{bmatrix}$$

Here

$$P_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

If we rewrite $L_3P_3L_2P_2L_1P_1A = U$ as

$$U = L_3P_3L_2P_2L_1(P_2^{-1}P_2)P_1A = L_3P_3L_2(P_2L_1P_2^{-1})P_2P_1A,$$

we have

$$P_2L_1P_2^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha_1 & 1 & 0 & 0 \\ \alpha_2 & 0 & 1 & 0 \\ \alpha_3 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha_3 & 1 & 0 & 0 \\ \alpha_2 & 0 & 1 & 0 \\ \alpha_1 & 0 & 0 & 1 \end{bmatrix} = L_1^{(1)}$$

Note that $P_2^{-1} = P_2$. Similarly, $P_3^{-1} = P_3$, and thus we have

$$U = L_3P_3L_2L_1^{(1)}P_2P_1A = L_3(P_3L_2P_3)(P_3L_1^{(1)}P_3)P_3P_2P_1A = L_3L_2^{(1)}L_1^{(2)}P_3P_2P_1A$$

where

$$L_2^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & \alpha_5 & 1 & 0 \\ 0 & \alpha_4 & 0 & 1 \end{bmatrix}, \quad L_1^{(2)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha_3 & 1 & 0 & 0 \\ \alpha_1 & 0 & 1 & 0 \\ \alpha_2 & 0 & 0 & 1 \end{bmatrix},$$

It follows that

$$PA = (P_3P_2P_1)A = (L_3L_2^{(1)}L_1^{(2)})^{-1}U = (L_1^{(2)})^{-1}(L_2^{(1)})^{-1}L_3^{-1}U = LU$$

Here $L$ is the *unit lower triangular matrix* (unit lower-$\Delta$) and $P$ is a permutation matrix given below.

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\alpha_3 & 1 & 0 & 0 \\ -\alpha_1 & -\alpha_5 & 1 & 0 \\ -\alpha_2 & -\alpha_4 & -\alpha_6 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

**Theorem:** For any $A \in R^{n\times n}$, there exists a permutation matrix $P$ such that $PA = LU$, where $L$ is unit $lower - \Delta$ and $U$ is $upper - \Delta$.

# Gaussian Elimination with partial Pivoting

♣ *Algorithm*

for $i = 1, 2, \cdots, n$

p(i)=i

endfor

for $i = 1, 2, \cdots, n-1$

$(a)$ select a pivotal element $a_{p(j),i}$ such that $|a_{p(j),i}| = max_{i \le k \le n}|a_{p(k),i}|$

$(b)$ $p(i) \longleftrightarrow p(j)$

$(c)$ for $k = i+1, i+2, \cdots, n$

$m_{p(k),i} = a_{p(k),i}/a_{p(i),i}$

for $j = i+1, i+2, \cdots, n$

$a_{p(k),j} = a_{p(k),j} - m_{p(k),i} * a_{p(i),j}$

endfor

endfor

endfor

• *An example*

$$A = \begin{bmatrix} 0 & 9 & 1 \\ 1 & 2 & -2 \\ 2 & -5 & 4 \end{bmatrix} \quad \Rightarrow \quad P_{23}P_{13}A = LU = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 2 & -5 & 4 \\ 0 & 9 & 1 \\ 0 & 0 & \frac{-9}{2} \end{bmatrix}$$

# Matlab Codes for Gaussian Elimination with Partial Pivoting

```
% function [x,P]=GaussPP(A,b) - Solving Ax=b by PA=LUx=b,
%                 Gaussian Elimination with partial pivoting
function [x,P]=GaussPP(A,b)
[m n]=size(A);
if (m~=n)
    error('matrix A must be square');
end
P=1:n;
Aug=[A, b];            % the augmented matrix
% Forward Elimination
for i=1:n-1
    kmax=i;
    t=abs(Aug(i,i));
    for k=i+1:n
        if (abs(Aug(k,i))>t)
            t=abs(A(k,i)); kmax=k;
        end
    end
    if (kmax~=i)
      tv=Aug(i,i:n+1);
      Aug(i,i:n+1)=Aug(kmax,i:n+1);
      Aug(kmax,i:n+1)=tv;
      kr=P(i);  P(i)=P(kmax);  P(kmax)=kr;
    end
    for k=i+1:n
        r=Aug(k,i)/Aug(i,i);
        Aug(k,i+1:n+1)=Aug(k,i+1:n+1)-r*Aug(i,i+1:n+1);
        Aug(k,i)=r;
    end
end
% Back Substitution
x=zeros(n,1);
x(n)=Aug(n,n+1)/Aug(n,n);
for j=n-1:-1:1
    x(j)=(Aug(j,n+1)-Aug(j,j+1:n)*x(j+1:n))/Aug(j,j);
end
```

# Matlab Codes for Gaussian Elimination with Partial Pivoting

```
%%-----------------------------------------------------------------------------%%
%% gausspp.m - drive of Gaussian Elimination wit Partial Pivoting             %%
%%-----------------------------------------------------------------------------%%
fin=fopen('gaussmat.dat','r');
n=fscanf(fin,'%d',1);
A=fscanf(fin,'%f',[n n]);   A=A';
b=fscanf(fin,'%f',n);
X=gausspivot(A,b,n)

%%-----------------------------------------------------------------------------%%
%% gausspivot.m - Gaussian elimination with Partial Pivoting                  %%
%%-----------------------------------------------------------------------------%%
function X=gausspivot(A,b,n)

if (abs(det(A))<eps)
  disp(sprintf('A is singular with det=%f\n',det(A)))
end

C=[A, b];
%------- Gaussian Elimination with Partial Pivoting ------%
for i=1:n-1
  [pivot, k]=max(abs(C(i:n,i)));
  if (k>1)
    temp=C(i,:);
    C(i,:)=C(i+k-1,:);
    C(i+k-1,:)=temp;
  end
  m(i+1:n,i)= -C(i+1:n,i)/C(i,i);
  C(i+1:n,:)=C(i+1:n,:)+m(i+1:n,i)*C(i,:);
end
%------- Back substitution -------%
X=zeros(n,1); %% Let X be a column vector of size n
X(n)=C(n,n+1)/C(n,n);
for i=n-1:-1:1
  X(i)=(C(i,n+1)-C(i,i+1:n)*X(i+1:n))/C(i,i);
end
```

# Matlab Codes for A=LU Decomposition

```
% function [L,U]=LUdecomp(A) - A=LU decomposition
% Compute the LU decomposition of A such that (L,U) appeared in output A
function [L,U]=LUdecomp(A)
% A = [ 2,  1, -2;
%      -4, -1,  5;
%       2,  2,  2];
% b = [-2;  3; 0];
% x = [ 1; -2; 1];  Solution for A*x=b
[m n]=size(A);
if (m~=n)
    error('matrix A must be square');
end
A
% Forward Elimination
for i=1:n-1
    for k=i+1:n
        r=A(k,i)/A(i,i);
        A(k,i+1:n)=A(k,i+1:n)-r*A(i,i+1:n);
        A(k,i)=r;
    end
end
L=zeros(m,n);
U=zeros(m,n);
for i=1:m
    for j=1:n
        if (i>j)
            L(i,j)=A(i,j);
        elseif (i==j)
            L(i,j)=1;
            U(i,j)=A(i,j);
        elseif (i<j)
            U(i,j)=A(i,j);
        end
    end
end
OutputMatrixA=A
%  2  1 -2              1  0  0        2  1 -2
% -2  1  1, where  L = -2  1  0 ;  U = 0  1  1
%  1  1  3              1  1  1        0  0  3
```

# Matlab Codes for [L,U,P]=PALU(A)

```
% function [L,U,P]=PALU(A) for (solving Gaussian Elimination with
%                                                Partial Pivoting)
function [L,U,P]=PALU(A)
[m n]=size(A);
if (m~=n)
    error('matrix A must be square');
end
P=eye(n); Q=1:n-1;
% Forward Elimination with Partial Pivoting
for i=1:n-1
    t=abs(A(i,i));
    for k=i+1:n
        if (abs(A(k,i))>t)
            t=abs(A(k,i)); kmax=k;
        end
    end
    Q(i)=kmax;
    tv=A(i,i:n);
    A(i,i:n)=A(kmax,i:n);
    A(kmax,i:n)=tv;
    rowvec=P(i,:);  P(i,:)=P(kmax,:);  P(kmax,:)=rowvec;  % compute P
    for k=i+1:n
        r=A(k,i)/A(i,i);
        A(k,i+1:n)=A(k,i+1:n)-r*A(i,i+1:n);
        A(k,i)=r;
    end
end
% Switch Q(j+1) with L(j) in PA=LU
for i=2:n-1
    k=Q(i);
    if (k~=i)
        rv=A(i,1:i-1);
        A(i,1:i-1)=A(k,1:i-1);
        A(k,1:i-1)=rv;
    end
end
L=eye(n); U=zeros(n,n);
for i=1:n
    for j=1:n
```

```
        if (i==j)
            U(i,j)=A(i,j);
        elseif (i>j)
            L(i,j)=A(i,j);
        elseif (i<j)
            U(i,j)=A(i,j);
        end
    end
end
Q
% A=[ 0  1  2  3;
%     1  2  4  5;
%     2 -2  3 -1;
%     4  2  1  0];
% 4.0000  2.0000  1.0000  0.0000
% 0.2500 -3.2500  2.5000 -1.0000
% 0.5000 -0.5000  5.0000  4.5000
% 0.0000 -0.3333  0.5667  0.1167
%---------------------------------
% P =[ 0 0 0 1;
%      0 0 1 0;
%      0 1 0 0;
%      1 0 0 0];
```