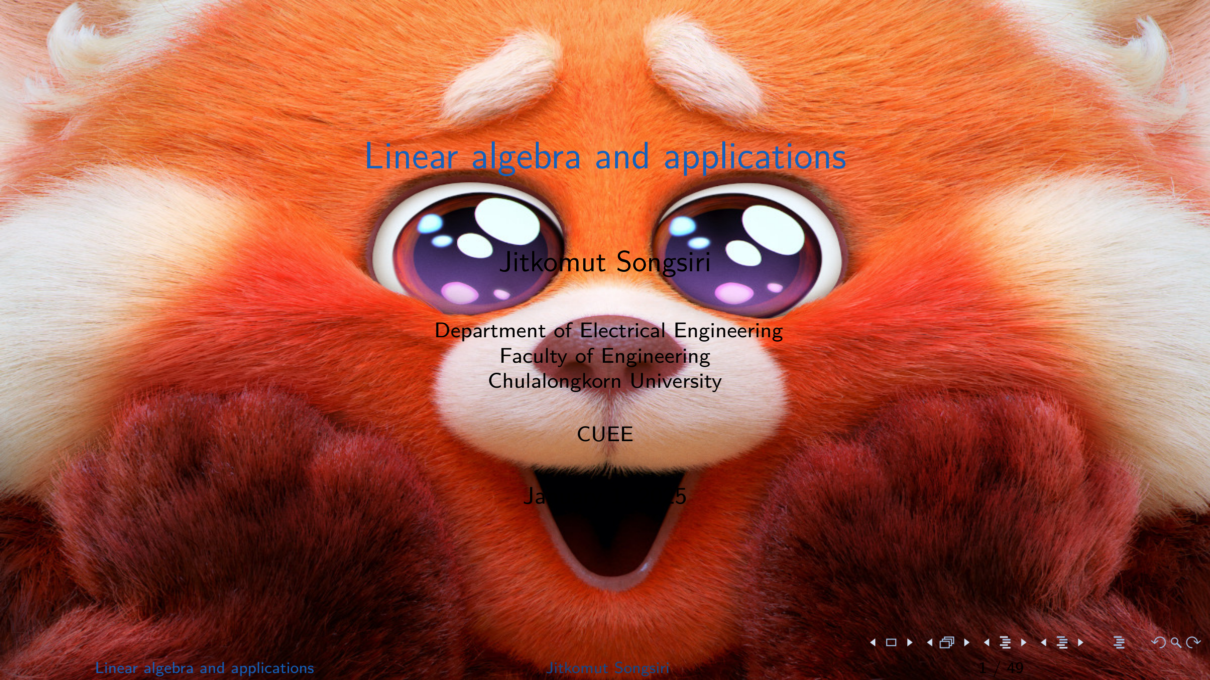


# Linear algebra and applications


Jitkomut Songsiri

Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University

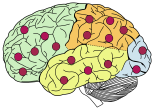
CUEE


Ja[illegible]5

# Outline

# How to read this handout

1. the note is used with lecture in EE205 (you cannot master this topic just by reading this note) – class activities include
   - graphical concepts, math derivation of details/steps in between
   - computer codes to illustrate examples
2. always read 'textbooks' after lecture
3. pay attention to the symbol ✎; you should be able to prove such ✎ result
4. each chapter has a list of references; find more formal details/proofs from in-text citations
5. almost all results in this note can be Googled; readers are encouraged to 'stimulate neurons' in your brain by proving results without seeking help from the Internet first
6. typos and mistakes can be reported to `jitkomut@gmail.com`



# Matrix decomposition

# Decompositions

- LU
- Cholesky
- SVD

# Factor-solve approach

to solve $Ax = b$, first write $A$ as a product of 'simple' matrices

$$A = A_1 A_2 \cdots A_k$$

then solve $(A_1 A_2 \cdots A_k)x = b$ by solving $k$ equations

$$A_1 z_1 = b, \qquad A_2 z_2 = z_1, \quad \ldots, \quad A_{k-1} z_{k-1} = z_{k-2}, \qquad A_k x = z_{k-1}$$

**complexity** of factor-solve method: flops $= f + s$

- $f$ is cost of factoring $A$ as $A = A_1 A_2 \cdots A_k$ (factorization step)
- $s$ is cost of solving the $k$ equations for $z_1$, $z_2$, ...$z_{k-1}$, $x$ (solve step)
- usually $f \gg s$

# Forward substitution

solve $Ax = b$ when $A$ is lower triangular with nonzero diagonal elements

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

**algorithm**:

$$\begin{aligned} x_1 &:= b_1/a_{11} \\ x_2 &:= (b_2 - a_{21}x_1)/a_{22} \\ x_3 &:= (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \\ &\vdots \\ x_n &:= (b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{n,n-1}x_{n-1})/a_{nn} \end{aligned}$$

**cost**: $1 + 3 + 5 + \cdots + (2n - 1) = n^2$ flops

# LU decomposition (w/o row pivoting)

**Theorem:** if $A$ can be lower reduced (w/o row interchanged) to a row-echelon matrix $U$, then $A = LU$ where $L$ is lower triangular and invertible and $U$ is upper triangular and row-echelon

- suppose $A$ can be reduced to $A \to E_1 A \to E_2 E_1 A \to E_k E_{k-1} \cdots E_2 E_1 A = U$
- $A = LU$ where $L = E_1^{-1} E_2^{-1} \cdots E_k^{-1}$
  - $E_j$ corresponds to scaling operation or $R_i + \alpha R_j \to R_i$ for $i > j$
  - $E_j$ is lower triangular (and invertible)
  - $E_j^{-1}$ is also lower triangular, hence, $L$ is lower triangular

# Example

find LU for $A = \begin{bmatrix} 2 & 4 & 2 \\ 1 & 1 & 2 \\ -1 & 0 & 2 \end{bmatrix}$

$$\begin{array}{llll}
R_1/2, & E_1 = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & E_1^{-1} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \Rightarrow \begin{bmatrix} 1 & 2 & 1 \\ 1 & 1 & 2 \\ -1 & 0 & 2 \end{bmatrix} \\
R_2 - R_1 \to R_2, & E_2 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & E_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \Rightarrow \begin{bmatrix} 1 & 2 & 1 \\ 0 & -1 & 1 \\ -1 & 0 & 2 \end{bmatrix} \\
R_3 + R_1 \to R_3, & E_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, & E_3^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, & \Rightarrow \begin{bmatrix} 1 & 2 & 1 \\ 0 & -1 & 1 \\ 0 & 2 & 3 \end{bmatrix} \\
R_2/-1 \to R_2, & E_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & E_4^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \Rightarrow \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & -1 \\ 0 & 2 & 3 \end{bmatrix} \\
R_3 - 2R_2 \to R_3, & E_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}, & E_5^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}, & \Rightarrow \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 5 \end{bmatrix} = U
\end{array}$$

we have $A = E_1^{-1}E_2^{-1}E_3^{-1}E_4^{-1}E_5^{-1}U = \begin{bmatrix} 2 & 0 & 0 \\ 1 & -1 & 0 \\ -1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 5 \end{bmatrix}$

each column in $L$ can be read from the leading column in $A$ while performing Gaussian elimination

# LU algorithm

let $A \in \mathbf{R}^{m \times n}$ of rank $r$ and suppose $A$ can be lower reduced to $U$ (without row interchanged) then $A = LU$ where the lower triangular, invertible $L$ is constructed as follows

1. if $A = 0$ then $L = I_m$ and $U = 0$
2. if $A \neq 0$, write $A_1 = A$ and let $c_1$ be the leading column of $A_1$
3. use $c_1$ to create the first leading 1 and create zero below it; denote $A_2$ the matrix consisting of rows 2 to $m$
4. if $A_2 \neq 0$ let $c_2$ be the leading column of $A_2$ and repeat step 2-3 to create $A_3$
5. continue until $U$ is found where all rows below the last leading 1 consist of zeros; this happen after $r$ steps
6. create $L$ by placing $c_1, c_2, \ldots, c_r$ at the bottom of the first $r$ columns of $I_m$

# Example

find LU for $A = \begin{bmatrix} 2 & 6 & -2 & 0 & 2 \\ 3 & 9 & -3 & 3 & 1 \\ -1 & 3 & 1 & -3 & 1 \end{bmatrix}$

$$R_1/2 \quad \begin{bmatrix} 1 & 3 & -1 & 0 & 1 \\ 3 & 9 & -3 & 3 & 1 \\ -1 & 3 & 1 & -3 & 1 \end{bmatrix}, \qquad R_2 - 3R_1 \to R_2, R_3 + R_1 \to R_3 \quad \begin{bmatrix} 1 & 3 & -1 & 0 & 1 \\ 0 & 0 & 0 & 3 & -2 \\ 0 & 0 & 0 & -3 & 2 \end{bmatrix}$$

$$R_2/3 \quad \begin{bmatrix} 1 & 3 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -2/3 \\ 0 & 0 & 0 & -3 & 2 \end{bmatrix}, \quad R_3 + 3R_2 \to R_3 \begin{bmatrix} 1 & 3 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -2/3 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = U$$

we obtain

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & -3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -2/3 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Is LU decomposition unique?

from the previous page

$$A = \begin{bmatrix} 2 & 6 & -2 & 0 & 2 \\ 3 & 9 & -3 & 3 & 1 \\ -1 & 3 & 1 & -3 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & -3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -2/3 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = L_1 U_1$$

we can make $L$ **the unit lower triangular** (all diagonals are 1) (standard choice)

$$A = \begin{bmatrix} 2 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & -3 & 1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -2/3 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 3/2 & 1 & 0 \\ -1/2 & -1 & 1 \end{bmatrix} \begin{bmatrix} 3 & 9 & -3 & 0 & 3 \\ 0 & 0 & 0 & 3 & -2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = L_2 U_2$$

# Not every matrix has an LU factor

without row pivoting, LU factor may not exist even when $A$ is invertible

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \Rightarrow \quad LU = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}$$

from this example,

- if $A$ could be factored as LU, it would require that $l_{11}u_{11} = a_{11} = 0$
- one of $L$ or $U$ would be singular, contradicting to the fact that $A = LU$ is nonsingular

# Existence and uniqueness

- **existence**

  **Theorem:** suppose $A$ is invertible; then $A$ has LU factorization $A = LU$ if and only if all leading principle minors are nonzero

  $$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{is non-singular but has no LU factorization}$$

- **uniqueness**

  **Theorem:** if an invertible $A$ has an LU factorization then $L$ and $U$ are uniquely determined (if we require the diagonals of $L$ (or $U$) are all 1)

(Horn, Corollary 3.5.6)

# LU decomposition with row pivoting

find LU of $A = \begin{bmatrix} 0 & 0 & -1 \\ -1 & -1 & 1 \\ 2 & 1 & -2 \end{bmatrix}$

- the first row has a leading zero, so row operations require a row interchange, here choose $R_1 \Leftrightarrow R_3$ corresponding to $P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$
- note that $P^2 = I$ (permutation property), we can write

$$A = P^2 A = PPA = P \begin{bmatrix} 2 & 1 & -2 \\ -1 & -1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

- perform LU decomposition on the resulting $PA$

# LU decomposition with row pivoting

- perform $R_1/2$, $R_2 + 2R_1 \to R_1$

$$A = P \begin{bmatrix} 2 & & \\ -1 & 1 & \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1/2 & -3/2 \\ 0 & -1/2 & -1/2 \\ 0 & 0 & -1 \end{bmatrix}$$

- perform $R_2 \times -2 \to R_2$

$$A = P \begin{bmatrix} 2 & & \\ -1 & -1/2 & \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1/2 & -3/2 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

- perform $R_3 \times -1 \to R_3$

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & & \\ -1 & -\frac{1}{2} & \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & -\frac{3}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & & \\ -\frac{1}{2} & 1 & \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -3 \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -1 \end{bmatrix} \triangleq PLU$$

# LU decomposition with row pivoting

same $A$ on page 15 but swap $R_1 \Leftrightarrow R_2$ using $P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

perform LU decomposition and we get different factors

$$A = \begin{bmatrix} 0 & 0 & -1 \\ -1 & -1 & 1 \\ 2 & 1 & -2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 3 \\ 0 & 1 & 1/2 \\ 0 & 0 & 9/2 \end{bmatrix}$$

# Common pivoting strategy

permute rows so that the largest entry of the first column is on the top left

$$A = \begin{bmatrix} 2 & 4 & 2 \\ 1 & 1 & 2 \\ -1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & -1 & 1 \\ 0 & 2 & 3 \end{bmatrix} \quad \begin{array}{l} R_1/2 \rightarrow R_1 \\ R_2 - R_1 \rightarrow R_2 \\ R_3 + R_1 \rightarrow R_3 \end{array}$$

$$= P_1 P_1 \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} P_1 P_1 \begin{bmatrix} 1 & 2 & 1 \\ 0 & -1 & 1 \\ 0 & 2 & 3 \end{bmatrix} \quad \text{(swap row 2 and 3)}, \; P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \because P_1^2 = I$$

$$= P_1 \left( P_1 \begin{bmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} P_1 \right) \begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 3 \\ 0 & -1 & 1 \end{bmatrix} = P_1 \begin{bmatrix} 2 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 3 \\ 0 & -1 & 1 \end{bmatrix}$$

$$= P_1 \begin{bmatrix} 2 & 0 & 0 \\ -1 & 2 & 0 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 3/2 \\ 0 & 0 & 5/2 \end{bmatrix} \quad \begin{array}{l} R_2/2 \rightarrow R_2 \\ R_3 + R_2 \rightarrow R_3 \end{array}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 1/2 & -1/2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 2 \\ 0 & 2 & 3 \\ 0 & 0 & 5/2 \end{bmatrix}$$

# Conclusion

any square matrix $A$ can be factorized as (with row pivoting)

$$A = PLU$$

factorization:

- $P$ permutation matrix, $L$ unit lower triangular, $U$ upper triangular
- **factorization cost**: $(2/3)n^3$ if $A$ has order $n$
- not unique; there may be several possible choices for $P$, $L$, $U$
- interpretation: permute the rows of $A$ and factor $P^TA$ as $P^TA = LU$
- also known as *Gaussian elimination with partial pivoting* (GEPP)

# Example

- a singular $A$ (no row pivoting)

$$A = \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1/2 & 1 \end{bmatrix} \begin{bmatrix} 4 & 2 \\ 0 & 0 \end{bmatrix}$$

- nonsingular $A$ (that requires row pivoting)

$$A = \begin{bmatrix} 0 & 2 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

- nonsingular $A$ (showing two choices of $(P, L, U)$)

$$A = \begin{bmatrix} 2 & 4 & 2 \\ 1 & 1 & 2 \\ -1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ -1/2 & -2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 2 \\ 0 & -1 & 1 \\ 0 & 0 & 5 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 1/2 & -1/2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 2 \\ 0 & 2 & 3 \\ 0 & 0 & 5/2 \end{bmatrix}$$

# Solving a linear system with LU factor

solving linear system: $(PLU)x = b$ in three steps

- permutation: $z_1 = P^T b$ (0 flops)
- forward substitution: solve $Lz_2 = z_1$ ($n^2$ flops)
- back substitution: solve $Ux = z_2$ ($n^2$ flops)

**total cost**: $(2/3)n^3 + 2n^2$ flops, or roughly $(2/3)n^3$

# Softwares

**MATLAB**

- `[L,U,P] = lu(A)` find LU decomposition: $A = P^T LU$ where $L$ is unit lower triangular and $U$ is upper triangular

**Python**

- `P,L,U = scipy.linalg.lu(A)` find LU decomposition: $A = PLU$ where $L$ is unit lower triangular and $U$ is upper triangular

# Exercises

1. find LU factorization (explain if row pivoting is required) and compare the results with coding

$$A_1 = \begin{bmatrix} 2 & 4 & 2 \\ 1 & -1 & 3 \\ -1 & 7 & -7 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & -1 & 2 \\ 0 & 0 & 3 \\ -1 & 2 & 2 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 3 & 3 & 0 \\ -2 & 0 & 2 \\ 3 & 2 & -1 \end{bmatrix}$$

2. suppose we aim to solve $Ax = b^{(k)}$ for $k = 1, \ldots, 1000$ where $A \in \mathbf{R}^{2000 \times 2000}$ and $b^{(k)}$'s can be randomized as examples, write computer code to solve the linear system using factor approach and measure the computation time in each process

# Cholesky factorization

every positive definite matrix $A$ can be factored as

$$A = LL^T$$

where $L$ is lower triangular with positive diagonal elements

- **cost**: $(1/3)n^3$ flops if $A$ is of order $n$
- $L$ is called the *Cholesky factor* of $A$
- can be interpreted as 'square root' of a positive define matrix
- $L$ is invertible (its diagonal elements are nonzero)
- $A$ is invertible and

$$A^{-1} = L^{-T}L^{-1}$$

# Cholesky factorization algorithm

partition matrices in $A = LL^T$ as

$$\begin{bmatrix} a_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} l_{11} & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix} = \begin{bmatrix} l_{11}^2 & l_{11}L_{21}^T \\ l_{11}L_{21} & L_{21}L_{21}^T + L_{22}L_{22}^T \end{bmatrix}$$

**algorithm:**

1. determine $l_{11}$ and $L_{21}$:

$$l_{11} = \sqrt{a_{11}}, \qquad L_{21} = \frac{1}{l_{11}} A_{21}$$

2. compute $L_{22}$ from

$$A_{22} - L_{21}L_{21}^T = L_{22}L_{22}^T$$

   this is a Cholesky factorization of order $n-1$

# Proof of Cholesky algorithm

**proof** that the algorithm works for positive definite $A$ of order $n$

- step 1: if $A$ is positive definite then $a_{11} > 0$
- step 2: if $A$ is positive definite, then

$$A_{22} - L_{21}L_{21}^T = A_{22} - \frac{1}{a_{11}}A_{21}A_{21}^T$$

  is positive definite (by Schur complement)
- hence the algorithm works for $n = m$ if it works for $n = m - 1$
- it obviously works for $n = 1$; therefore it works for all $n$

# Example of Cholesky algorithm

$$\begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

- first column of $L$

$$\begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 \\ 3 & l_{22} & 0 \\ -1 & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} 5 & 3 & -1 \\ 0 & l_{22} & l_{32} \\ 0 & 0 & l_{33} \end{bmatrix}$$

- second column of $L$

$$\begin{bmatrix} 18 & 0 \\ 0 & 11 \end{bmatrix} - \begin{bmatrix} 3 \\ -1 \end{bmatrix} \begin{bmatrix} 3 & -1 \end{bmatrix} = \begin{bmatrix} l_{22} & 0 \\ l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{22} & l_{32} \\ 0 & l_{33} \end{bmatrix}$$

$$\begin{bmatrix} 9 & 3 \\ 3 & 10 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 1 & l_{33} \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 0 & l_{33} \end{bmatrix}$$

- third column of $L$: $10 - 1 = l_{33}^2$, *i.e.*, $l_{33} = 3$

conclusion:

$$\begin{bmatrix} 25 & 15 & -5 \\ 15 & 18 & 0 \\ -5 & 0 & 11 \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 \\ 3 & 3 & 0 \\ -1 & 1 & 3 \end{bmatrix} \begin{bmatrix} 5 & 3 & -1 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{bmatrix}$$

# Solving equations with positive definite $A$

$$Ax = b \qquad (A \text{ positive definite of order } n)$$

**algorithm**

- factor $A$ as $A = LL^T$
- solve $LL^Tx = b$
  - forward substitution $Lz = b$
  - back substitution $L^Tx = z$

**cost**: $(1/3)n^3$ flops

- factorization: $(1/3)n^3$
- forward and backward substitution: $2n^2$

# Softwares

**MATLAB**

- `U = chol(A)` returns Cholesky decomposition $A = U^T U$ where $U$ is upper triangular

**Python**

- `L = scipy.linalg.cholesky(A)` returns Cholesky decomposition $A = LL^T$ or $A = U^T U$ where $L$ is lower (`lower=True`) and $U$ is upper triangular

# Exercises

1. find Cholesky factorization and compare the results with coding

$$A_1 = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 12 & 4 & 3 \\ 4 & 2 & -1 \\ 3 & -1 & 7 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 20 & 4 & 5 \\ 4 & 2 & 3 \\ 5 & 3 & 5 \end{bmatrix}$$

2. suggest a method to randomize $A$ and guarantee that $A \succ 0$
3. suppose we aim to solve $Ax = b^{(k)}$ for $k = 1, \ldots, 1000$ where $A \in \mathbf{S}_{++}^{2000 \times 2000}$ (pdf) and $b^{(k)}$'s can be randomized as examples, write computer code to solve the linear system using factor approach and measure the computation time in each process

# SVD decomposition

- recall that $A^T A \succeq 0$ and eigenvalues are non-negative
- singular values
- left and right singular vectors
- applications: pseudo inverse

# Singular values and vectors

let $A \in \mathbf{R}^{m\times n}$, we form eigenvalue problem of $A^TA$

$$A^TAv_i = \sigma_i^2 v_i, \quad i = 1, 2, \ldots, n$$

- $\sigma_i = \sqrt{\lambda_i(A^TA)} > 0$ is called **singular value** of $A$
- $v_i$ (orthogonal and has unit-norm) is called **right singular vector**
- fact: if rank of $A$ is $r$ then $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_r > 0$ and $\sigma_i = 0$ for $i > r$

rank of $A$ is the number of non-zero singular values of $A$

- there exist **left singular vector** $u_1, u_2, \ldots, u_m$ that are orthogonal such that

$$Av_1 = \sigma_1 u_1, \quad Av_2 = \sigma_2 u_2, \ldots, Av_r = \sigma_r u_r, \quad Av_{r+1} = \cdots = Av_n = 0$$

# Matrix form

$$Av_1 = \sigma_1 u_1, \quad Av_2 = \sigma_2 u_2, \ldots, \quad Av_r = \sigma_r u_r, \quad Av_{r+1} = \cdots = Av_n = 0$$

or in matrix form: $AV = U\Sigma$ (where $U$ and $V$ are orthogonal matrices)

$$A \left[\begin{array}{ccc|ccc} v_1 & \cdots & v_r & v_{r+1} & \cdots & v_n \end{array}\right] = \left[\begin{array}{ccc|ccc} u_1 & \cdots & u_r & u_{r+1} & \cdots & u_m \end{array}\right] \left[\begin{array}{ccc|c} \sigma_1 & & & 0 \\ & \ddots & & 0 \\ & & \sigma_r & 0 \\ \hline 0 & 0 & 0 & \mathbf{0} \end{array}\right]$$

it can be shown that

- $v_1, \ldots, v_r, v_{r+1}, \ldots, v_n$ are orthogonal (eigenvectors of $A^T A$, which is symmetric)
- $u_{r+1}, \ldots, u_m$ can be chosen such that $\{u_1, \ldots, u_m\}$ are orgothogonal
- hence, $V, U$ are orthogonal matrices, $V^V = I, U^T U = I$

unlike eigenvalue decomposition: $AX = X\Lambda$, SVD needs two sets of singular vectors

# SVD decomposition

let $A \in \mathbf{R}^{m \times n}$ be a rectangular matrix; there exists the SVD form of $A$

$$A = U \Sigma V^T$$


- $U \in \mathbf{R}^{m \times m}, V \in \mathbf{R}^{n \times n}$ are orthogonal matrices
- $\Sigma \in \mathbf{R}^{m \times n}$ with $\Sigma_{ii} = \sigma_i \geq 0$ and $\Sigma_{ij} = 0$ for $i \neq j$
- for a rectangular $A$, $\Sigma$ has a diagonal submatrix $\Sigma_1$ with dimension of $\min(m, n)$

$$A_{\text{tall}} = \begin{bmatrix} U_1 & | & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 \\ \hline 0 \end{bmatrix} V^T = U_1 \Sigma_1 V^T, \quad A_{\text{fat}} = U \begin{bmatrix} \Sigma_1 & | & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ \hline V_2^T \end{bmatrix} = U \Sigma_1 V_1^T$$

# Square $A$

$$\begin{bmatrix} 2 & 1 \\ -1 & -2 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}^T, \mathbf{rank}(A) = 2$$

$$\begin{bmatrix} 2 & 4 & -2 \\ -2 & 0 & -2 \\ 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} -0.94 & -0.27 & -0.20 \\ 0.11 & -0.80 & 0.59 \\ -0.31 & 0.53 & 0.78 \end{bmatrix}\begin{bmatrix} 5.10 & 0 & 0 \\ 0 & 3.46 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} -0.53 & 0.62 & 0.58 \\ -0.80 & -0.15 & -0.58 \\ 0.27 & 0.77 & -0.58 \end{bmatrix}^T, \mathbf{rank}(A) = 2$$

$$\begin{bmatrix} -2 & 1 & 3 \\ 4 & -2 & -6 \\ 2 & -1 & -3 \end{bmatrix} = \begin{bmatrix} -0.41 & -0.91 & 0 \\ 0.82 & -0.37 & -0.45 \\ 0.41 & -0.18 & 0.89 \end{bmatrix}\begin{bmatrix} 9.17 & 0 & 9 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0.53 & -0.85 & 0 \\ -0.27 & -0.17 & 0.95 \\ -0.80 & -0.51 & -0.32 \end{bmatrix}^T, \mathbf{rank}(A) = 1$$

- check the singular values and eigenvalues of $A^T A$
- confirm the rank and the number of nonzero singular values
- if $A$ is invertible, so is $\Sigma$

# Fat $A$

$$A_1 = \begin{bmatrix} 2 & 0 & 2 \\ 0 & -2 & 1 \end{bmatrix} = \begin{bmatrix} -0.89 & -0.45 \\ -0.45 & 0.89 \end{bmatrix} \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} -0.60 & -0.45 & -0.67 \\ 0.30 & -0.89 & 0.33 \\ -0.75 & 0 & 0.67 \end{bmatrix}^T, \mathbf{rank}(A) = 2$$

$$A_2 = \begin{bmatrix} 2 & -1 & 1 & 0 \\ 2 & 0 & 1 & -2 \\ -2 & 0 & -1 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.42 & 0.91 & 0 \\ 0.64 & -0.30 & 0.71 \\ -0.64 & 0.30 & 0.71 \end{bmatrix} \begin{bmatrix} 4.6100 & 0 & 0 & 0 \\ 0 & 1.65 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.74 & 0.38 & 0.40 & -0.38 \\ -0.09 & -0.55 & 0.82 & 0.14 \\ 0.37 & 0.19 & 0.01 & 0.91 \\ -0.56 & 0.72 & 0.41 & 0.07 \end{bmatrix}^T, \mathbf{rank}(A) = 1$$

- $A_2$ is low rank, the SVD form can be reduced to $A_2 = U\Sigma V^T = U_r \Sigma_r V_r^T$ where $U_r, V_r$ have the first $r$ columns of $U$ and $V$ respectively and $\Sigma_r$ is the leading $r$-diagonal block of $\Sigma$ $\quad (r = \mathbf{rank}(A))$

# Tall $A$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ -2 & -2 & 0 \\ 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1.00 \\ 0.33 & -0.63 & -0.71 & 0 \\ 0.89 & 0.46 & 0 & 0 \\ -0.33 & 0.63 & -0.71 & 0 \end{bmatrix} \begin{bmatrix} 3.080 & 0 & 0 \\ 0 & 1.59 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.58 & -0.58 & 0.58 \\ -0.79 & 0.21 & -0.58 \\ 0.21 & -0.79 & -0.58 \end{bmatrix}^T$$

- $\mathbf{rank}(A) = 2$ and there are two nonzero singular values
- $A$ can be reduced to

$$A = U\Sigma V^T = U_r \Sigma_r V_r^T, \quad r = \mathbf{rank}(A) = 2$$

# Softwares

**MATLAB**

- `[U,S,V] = svd(A)` returns SVD decomposition: $A = USV^T$

**Python**

- `U,S,Vt = scipy.linalg.svd(A)`
- `U,S,Vt = numpy.linalg.svd(A)`

  returns SVD decomposition: $A = USV^T$ where `S` is returned as a vector of singular values and `Vt` as $V^T$

# Pseudo-inverse (Penrose Theorem)

one can have a notion of 'inverse' for a non-square matrix

**Penrose's Theorem:** given $A \in \mathbf{R}^{m \times n}$, there is exactly one $n \times m$ matrix $B$ such that

1. $ABA = A$ and $BAB = B$
2. both $AB$ and $BA$ are symmetric

**definition:** the **pseudo inverse** of $A \in \mathbf{R}^{m \times n}$ is the unique $n \times m$ matrix $A^\dagger$ such that

1. $AA^\dagger A = A$ and $A^\dagger A A^\dagger = A^\dagger$
2. both $AA^\dagger$ and $A^\dagger A$ are symmetric

# Pseudo-inverse

consider a full rank matrix $A \in \mathbf{R}^{m \times n}$ in three cases

- **tall matrix:** $A$ is full rank $\Leftrightarrow$ columns of $A$ are LI $\Leftrightarrow$ $A^TA$ is invertible

$$((A^TA)^{-1}A^T)A = (A^TA)^{-1}(A^TA) = I$$

  the **pseudo-inverse** of $A$ (or left-inverse) is $A^\dagger = (A^TA)^{-1}A^T$

- **wide matrix:** $A$ is full rank $\Leftrightarrow$ row of $A$ are LI $\Leftrightarrow$ $AA^T$ is invertible

$$A(A^T(AA^T)^{-1}) = (AA^T)(AA^T)^{-1} = I$$

  the **pseudo-inverse** of $A$ (or right-inverse) is $A^\dagger = A^T(AA^T)^{-1}$

- **square matrix:** $A$ is full rank $\Leftrightarrow$ $A$ is invertible and both formula of pseudo-inverses reduce to the ordinary inverse $A^{-1}$

✎ the pseudo inverses of the three cases have the same dimension ?

# Example

$$A = \begin{bmatrix} 0 & 2 & 1 \\ -2 & 1 & -2 \end{bmatrix}, \quad A^\dagger = A^T(AA^T)^{-1} = \begin{bmatrix} 0 & -2/9 \\ 2/5 & 1/9 \\ 1/5 & -2/9 \end{bmatrix}$$

$$A = \begin{bmatrix} -2 & -1 \\ 2 & -1 \\ -1 & 0 \end{bmatrix}, \quad A^\dagger = (A^TA)^{-1}A^T = \begin{bmatrix} -2/9 & 2/9 & 1/9 \\ -1/2 & -1/2 & 0 \end{bmatrix}$$

however, when rentangular $A$ has low rank, we can use SVD to find the pseudo inverse

# Pseudo-inverse via SVD

the pseudo-inverse $A^\dagger$ can be computed from any SVD for $A \in \mathbf{R}^{n\times m}$

- from $A = U_{n\times n}\Sigma_{n\times m}V_{m\times m}^T$ if $A$ has rank $r$ then

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}_{m\times n}, \quad \text{and that } \Sigma_r \text{ is invertible}$$

- define $\Sigma^\dagger = \begin{bmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{bmatrix}_{n\times m}$ and we can verify that

$$\Sigma\Sigma^\dagger\Sigma = \Sigma, \;\; \Sigma^\dagger\Sigma\Sigma^\dagger = \Sigma^\dagger, \;\; \Sigma\Sigma^\dagger = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}_{m\times m}, \;\; \Sigma^\dagger\Sigma = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}_{n\times n}$$

  proving that $\Sigma^\dagger$ is the pseudoinverse of $\Sigma$

# Pseudo-inverse via SVD

given $A = U\Sigma V^T$, then the pseudo-inverse of $A$ is

$$A^\dagger = V\Sigma^\dagger U^T$$

by verifying Penrose's Theorem from page 39 that

- $AA^\dagger A = (U\Sigma V^T)(V\Sigma^\dagger U^T)(U\Sigma V^T) = U\Sigma\Sigma^\dagger\Sigma V^T = U\Sigma V^T = A$
- $A^\dagger AA^\dagger = (V\Sigma^\dagger U^T)(U\Sigma V^T)(V\Sigma^\dagger U^T) = V\Sigma^\dagger\Sigma\Sigma^\dagger U^T = V\Sigma^\dagger U^T = A^\dagger$
- $AA^\dagger = U\Sigma\Sigma^\dagger U^T$ which is symmetric
- $A^\dagger A = V\Sigma^\dagger\Sigma V^T$ which is symmetric

# Example

a tall full rank $A$

$$A = \begin{bmatrix} -2 & -1 \\ 2 & -1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} -0.6667 & -0.7071 & -0.2357 \\ 0.6667 & -0.7071 & 0.2357 \\ -0.3333 & -0.0000 & 0.9428 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1.4142 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T$$

$$A^\dagger = V\Sigma^\dagger U^T = V \begin{bmatrix} 0.3333 & 0 & 0 \\ 0 & 0.7071 & 0 \end{bmatrix} U^T$$

$$= \begin{bmatrix} -0.22 & 0.22 & -0.1100 \\ -0.50 & -0.50 & 0 \end{bmatrix}$$

# Example

a fat low rank $A$

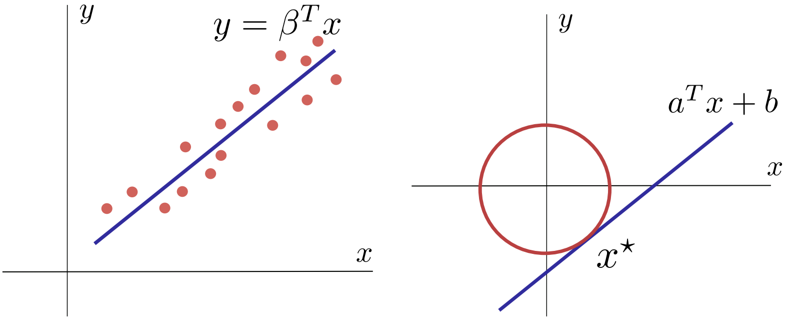
$$A = \begin{bmatrix} -2 & -1 & -3 & 0 \\ 0 & -3 & -3 & -2 \\ 2 & -2 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 0.47 & 0.67 & -0.58 \\ 0.81 & -0.08 & 0.58 \\ 0.34 & -0.74 & -0.58 \end{bmatrix} \begin{bmatrix} 5.76 & 0 & 0 & 0 \\ 0 & 3.85 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -0.05 & -0.73 & 0.51 & -0.45 \\ -0.62 & 0.27 & -0.27 & -0.68 \\ -0.67 & -0.46 & -0.25 & 0.53 \\ -0.40 & 0.43 & 0.78 & 0.23 \end{bmatrix}^T$$

$$A^\dagger = V\Sigma^\dagger U^T = V \begin{bmatrix} 0.1736 & 0 & 0 \\ 0 & 0.2596 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T$$

$$= \begin{bmatrix} -0.13 & 0.01 & 0.14 \\ 0 & -0.09 & -0.09 \\ -0.13 & -0.09 & 0.05 \\ 0.04 & -0.07 & -0.11 \end{bmatrix}$$

- $\mathbf{rank}(A) = 2 < n$ and there are two non-zero singular values
- $\Sigma \in \mathbf{R}^{3\times 4}$ and $\Sigma^\dagger \in \mathbf{R}^{4\times 3}$ with $2 \times 2$ invertible block

# Applications of pseudo-inverse




- **least-square problem:** find a straight line that fit best in 2-norm sense to data points
- **least-norm problem:** find a point $x$ on the given hyperplane that has the smallest norm

# Least-square problem

given $X \in \mathbf{R}^{N\times p}, y \in \mathbf{R}^N$ where typically $N > p$, a least-square problem is

$$\underset{\beta}{\text{minimize}} \quad \|y - X\beta\|_2^2$$

- it generalizes solving an overdetermined linear system that cannot be solved exactly by allowing the system to have the smallest residual
- if $X$ is full rank, and from zero-gradient condition, the optimal solution is

$$\beta = (X^T X)^{-1} X^T y$$

- the solution is linear in $y$ where the coefficient is the **left inverse** of $X$

# Least-norm problem

given $A \in \mathbf{R}^{m \times n}, b \in \mathbf{R}^m$ where $m < n$ and $A$ is full rank, the least-norm problem is

$$\underset{x}{\text{minimize}} \quad \|x\|_2 \quad \text{subject to} \quad Ax = y$$

- find a point on hyperplane $Ax = b$ while keeping the 2-norm of $x$ smallest
- it extends from solving an under-determined system that has many solutions and we aim to find the solution with smallest norm
- it can be shown that the optimal solution is

$$x^\star = A^T(AA^T)^{-1}y, \quad \text{provided that } A \text{ is full row rank}$$

- the solution is linear in $y$ where the coefficient is the **right inverse** of $A$

# References


1. W.K. Nicholson, *Linear Algebra with Applications*, McGraw-Hill, 2006
2. S. Boyd and L. Vandenberghe, *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least squares*, Cambridge, 2018
3. Lecture notes of EE133A, L. Vandenberhge, UCLA `https://www.seas.ucla.edu/~vandenbe/133A`