

# Linear Algebra Week 5

Summary: Eigenvalues and Applications

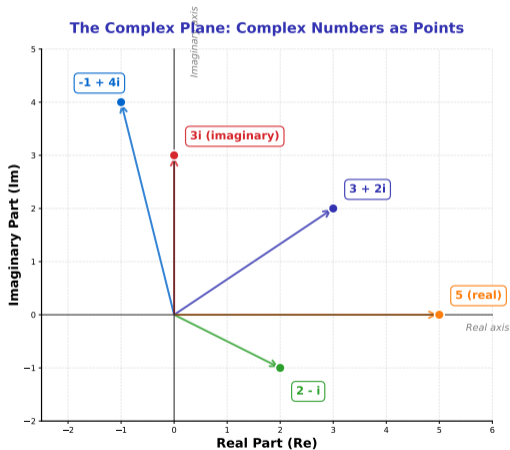
16 Key Concepts with Visualizations

Part 1: Complex Numbers — Part 2: Eigenvalues — Part 3: Diagonalization — Part 4: Applications

# Complex Numbers: The Complete Number System

## The Complex Plane (Argand Diagram)

- Complex number  $z = a + bi$  visualized as point  $(a, b)$  in 2D plane
- Key property:  $i^2 = -1$  extends real numbers to solve all polynomials



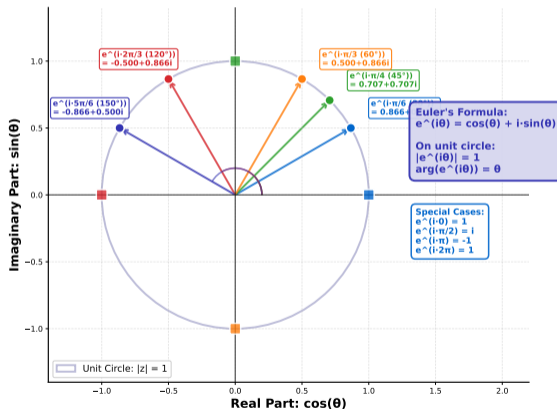
Example:  $z = 3 + 2i$  is at point  $(3, 2)$  with  $|z| = \sqrt{13} \approx 3.6$

# Euler's Formula: Exponentials Meet Trigonometry

## Polar Form and Euler's Formula

- Every complex number has polar form:  $z = r \cdot e^{i\theta}$
- Euler's formula:  $e^{i\theta} = \cos \theta + i \sin \theta$

### Euler's Formula: Complex Exponentials on the Unit Circle



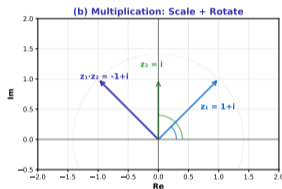
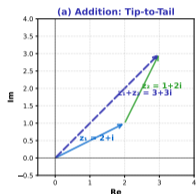
Example:  $z = 1 + i$  in polar:  $r = \sqrt{2}$ ,  $\theta = \pi/4$ , so  $z = \sqrt{2} \cdot e^{i\pi/4}$

# Complex Arithmetic: Four Fundamental Operations

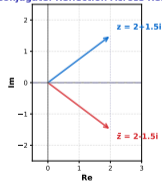
## Operations in the Complex Plane

- Addition: vector addition (component-wise)
- Multiplication: scale moduli, add angles (rotation + scaling)

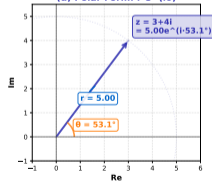
### Complex Arithmetic Operations



(c) Conjugate: Reflection Across Real Axis



(d) Polar Form:  $r \cdot e^{i\theta}$



Division:  $\frac{z_1}{z_2} = \frac{z_1 \cdot \bar{z}_2}{|z_2|^2}$  using conjugate  $\bar{z} = a - bi$

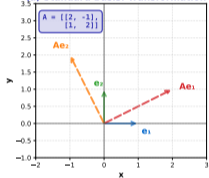
# Similar Matrices: Same Transformation, Different Coordinates

## Similarity Transformation

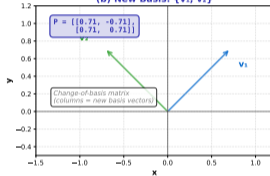
- $B = P^{-1}AP$  means  $A$  and  $B$  are similar matrices
- Similar matrices preserve: eigenvalues, trace, determinant, rank

### Similarity Transformation: Same Linear Map, Different Bases

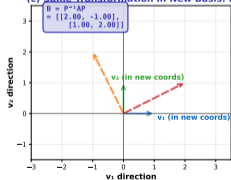
(a) Standard Basis: Transformation A



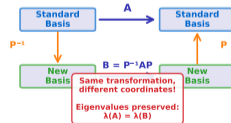
(b) New Basis:  $\{v_1, v_2\}$



(c) Same Transformation in New Basis: B



(d) Similarity Transformation Diagram  
Commutative Diagram:  $B = P^{-1}AP$



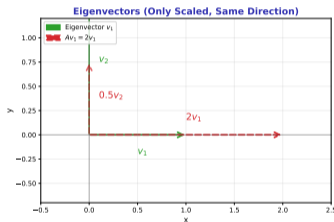
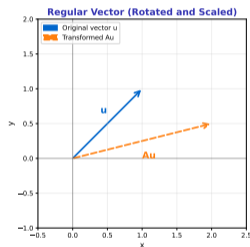
$P$  is the change-of-basis matrix; diagonalization is a special case of similarity

# Eigenvectors: Special Directions That Only Scale

**The Fundamental Equation:**  $Av = \lambda v$

- Eigenvector  $v$ : direction preserved by transformation  $A$
- Eigenvalue  $\lambda$ : scaling factor ( $\lambda > 1$  stretch,  $0 < \lambda < 1$  shrink,  $\lambda < 0$  flip)

**Eigenvalue Geometric Interpretation: Diagonal Matrix**



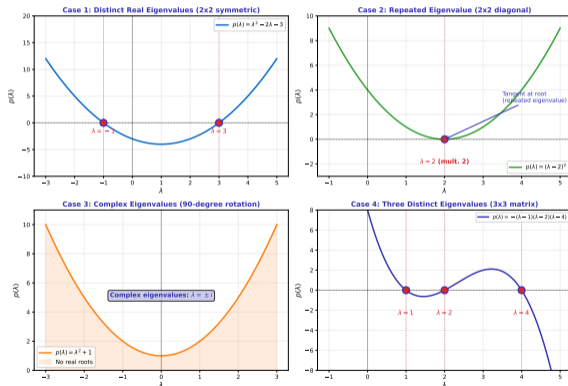
**Example:**  $A = \begin{pmatrix} 2 & 0 \\ 0 & 0.5 \end{pmatrix}$  has  $v_1 = (1, 0)$  with  $\lambda_1 = 2$ ,  $v_2 = (0, 1)$  with  $\lambda_2 = 0.5$

# Characteristic Polynomial: Roots Are Eigenvalues

## Finding Eigenvalues

- Characteristic polynomial:  $p(\lambda) = \det(A - \lambda I)$
- Eigenvalues are roots of  $p(\lambda) = 0$

### Characteristic Polynomial: Roots are Eigenvalues



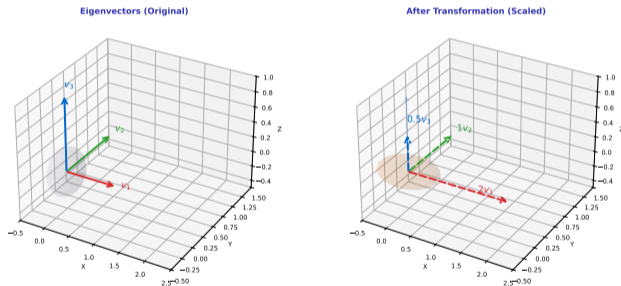
**Example:**  $A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \Rightarrow p(\lambda) = \lambda^2 - 2\lambda - 3 = (\lambda - 3)(\lambda + 1)$ , so  $\lambda = 3, -1$

# Finding Eigenvectors: Solving the Null Space Problem

## Eigenvectors from Eigenvalues

- For each eigenvalue  $\lambda$ , solve  $(A - \lambda I)\mathbf{v} = \mathbf{0}$
- Eigenspace  $E_\lambda = \ker(A - \lambda I)$  contains all eigenvectors for  $\lambda$

### 3D Eigenvectors: Principal Axes of Transformation



Use Gaussian elimination on  $(A - \lambda I)$  to find basis vectors for each eigenspace

## When Can We Diagonalize?

- Algebraic multiplicity: times  $\lambda$  appears as root of  $\det(A - \lambda I)$
- Geometric multiplicity: dimension of eigenspace  $\dim(E_\lambda)$

### Algebraic vs. Geometric Multiplicity: Comparison

#### Case 1: Multiplicities Match

**Matrix A: diagonal(5, 5, 3)**

Characteristic polynomial:

$$p(\lambda) = (5 - \lambda)^2(3 - \lambda)$$

Eigenvalue  $\lambda = 5$ :

Algebraic mult. = 2 | Geometric mult. = 2

Eigenspace:

Dim = 2 (spans first two coordinates)

**STATUS: DIAGONALIZABLE**

Multiplicities match - Full set of eigenvectors

#### Case 3: Mixed Multiplicities (Both Match)

**Matrix C: diagonal(2, 3, 3)**

$$p(\lambda) = (2 - \lambda)(3 - \lambda)^2$$

Eigenvalue Analysis:

$\lambda$	Alg. Mult.	Geom. Mult.
2	1	1
3	2	2

Both eigenvalues have matching multiplicities

**Diagonalizable: YES**

Full set of 3 independent eigenvectors

#### Case 2: Multiplicities Differ

**Matrix B: upper triangular**

(5,1,0 on diag, 1 above diag)

Characteristic polynomial:

$$p(\lambda) = (5 - \lambda)^2(3 - \lambda)$$

Eigenvalue  $\lambda = 5$ :

Algebraic mult. = 2 | Geometric mult. = 1

Eigenspace:

Dim = 1 (only ONE independent eigenvector!)

**STATUS: NOT DIAGONALIZABLE**

DEFECTIVE MATRIX

#### Case 4: Severely Defective Matrix

**Matrix D: Jordan block**

(4 on diag, 1 above diag)

$$p(\lambda) = (4 - \lambda)^3$$

Single Eigenvalue  $\lambda = 4$ :

Algebraic mult. = 3 | Geometric mult. = 1

Eigenspace:

Dimension = 1

(Only 1 eigenvector for 3x3 matrix)

**SEVERELY DEFECTIVE**

Needs generalized eigenvectors

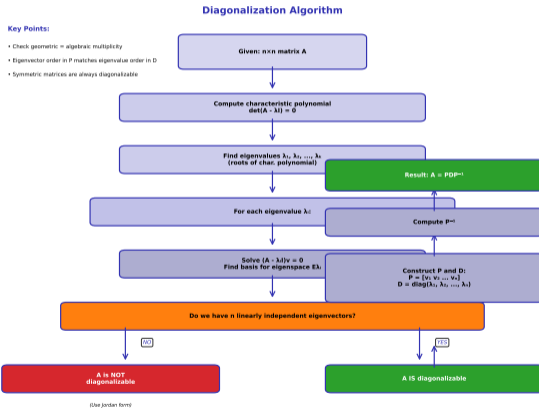
**Not Diagonalizable - Use Jordan Form**

Key inequality:  $1 \leq \text{geo} \leq \text{alg}$ . Diagonalizable  $\Leftrightarrow$  equality for all eigenvalues

# Diagonalization Algorithm: Step-by-Step

Finding  $A = PDP^{-1}$

- $D = \text{diag}(\lambda_1, \dots, \lambda_n)$  contains eigenvalues on diagonal
- $P = [\mathbf{v}_1 | \mathbf{v}_2 | \dots | \mathbf{v}_n]$  has eigenvectors as columns



Requires  $n$  linearly independent eigenvectors; symmetric matrices always diagonalizable

# Computing Matrix Powers: Huge Speedup

$$A^n = PD^nP^{-1}$$

- Direct method:  $A^n$  requires  $n - 1$  matrix multiplications
- Via diagonalization:  $D^n = \text{diag}(\lambda_1^n, \dots, \lambda_n^n)$  is trivial

## Method 1: Direct Computation

Compute  $A^n$  directly:

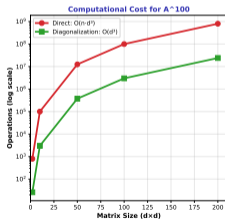
$A^1 = A \times A$   
 $A^2 = A^2 \times A$   
 $A^3 = A^3 \times A$   
 ...  
 $A^n = A^{n-1} \times A$

Operations:

- $n - 1$  matrix multiplications
- Each multiplication:  $O(d^3)$
- Total:  $O(n \cdot d^3)$

For  $A^{100}$ :

- 99 matrix multiplications
- For  $2 \times 2$ :  $\sim 800$  operations
- For  $100 \times 100$ :  $\sim 10^8$  operations!



## Method 2: Diagonalization

Compute via diagonalization:

**Step 1: Diagonalize  $A = PDP^{-1}$**   
 (one-time cost:  $O(d^3)$ )

**Step 2: Compute  $D^n$**   
 $D^n = \text{diag}(\lambda_1^n, \dots, \lambda_n^n)$   
 (d exponentiations:  $O(d)$ )

**Step 3:  $A^n = PD^nP^{-1}$**   
 (2 matrix mult.:  $O(d^3)$ )

Total:  $O(d^3 + d)$

For  $A^{100}$ :

- Diagonalize once
- 2 exponentiations
- 2 matrix multiplications
- For  $100 \times 100$ :  $\sim 2 \times 10^6$  operations

## Complexity Comparison

Matrix Size	Direct $A^{100}$	Diagonalization
2x2	800	50
10x10	$10^3$	$2 \times 10^3$
100x100	$10^8$	$2 \times 10^6$
1000x1000	$10^{11}$	$2 \times 10^9$

Speedup for  $100 \times 100$  matrix:

**$\sim 50 \times$  faster!**

When to Use Each Method?

**Use DIRECT method when:**

- Small  $n$  ( $n < 5$ )
- One-time computation
- Matrix not diagonalizable
- Very small matrices ( $2 \times 2$ )

**Use DIAGONALIZATION when:**

- Large  $n$  ( $n \geq 10$ )
- Multiple powers needed
- Large matrices ( $d > 10$ )
- Matrix is diagonalizable

**Key Insight:**

Diagonalization has one-time setup cost, but then computing any power  $A^n$  is very fast!

Example: Computing  $A^{10}$

Given  $A = \begin{bmatrix} 1 & 1 \\ 3 & 1 \end{bmatrix}$

**Diagonalization:**

$\lambda_1 = 4.8, \lambda_2 = 2.8$   
 $P = \begin{bmatrix} 1 & 0.71 \\ 0.71 & 0.71 \end{bmatrix}$

**Compute  $D^{10}$ :**

$D^{10} = \begin{bmatrix} 11048576 & 0 \\ 0 & 36241 \end{bmatrix}$

**Result  $A^{10} = PD^{10}P^{-1}$ :**

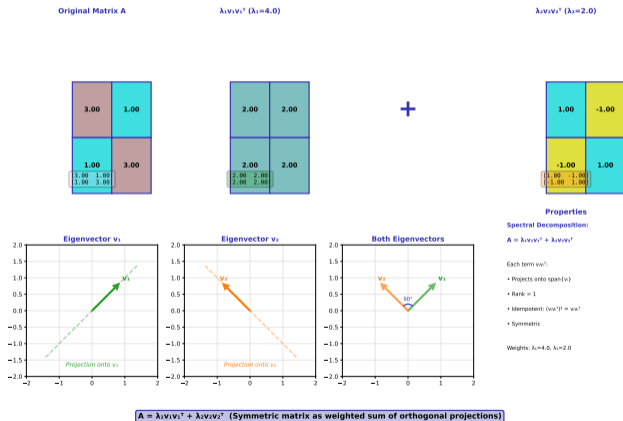
$\begin{bmatrix} 11524809 & 523776 \\ 523776 & 5248601 \end{bmatrix}$

For  $100 \times 100$  matrix: direct  $\approx 10^8$  ops vs diagonalization  $\approx 2 \times 10^6$  ops (50x faster)

# Spectral Decomposition: Matrix as Sum of Projections

## For Symmetric Matrices

- $A = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T + \dots$
- Each term  $\mathbf{v}_i \mathbf{v}_i^T$  projects onto the eigenspace



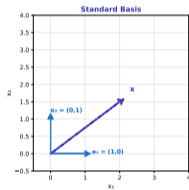
$$A = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^T + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^T \quad (\text{Symmetric matrix as weighted sum of orthogonal projections})$$

Orthonormal eigenvectors for symmetric matrices; basis for PCA in data science

# Diagonalization as Basis Change

Standard  $\rightarrow$  Eigenbasis  $\rightarrow$  Scale  $\rightarrow$  Standard

- $P^{-1}$ : convert to eigenvector coordinates
- $D$ : simple scaling along eigenvector directions

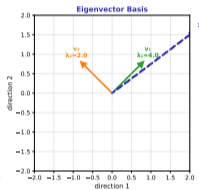


Matrix P (Change of Basis)

$$P = \begin{bmatrix} 0.707 & -0.707 \\ 0.707 & 0.707 \end{bmatrix}$$

Columns are eigenvectors  $v_1, v_2$

$P$  transforms from eigenvector coordinates to standard coordinates



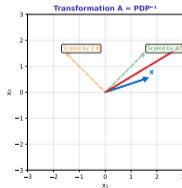
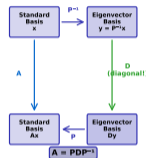
Matrix D (Diagonal)

$$D = \begin{bmatrix} 4.0 & 0 \\ 0 & 2.0 \end{bmatrix}$$

Diagonal entries are eigenvalues  $\lambda_1, \lambda_2$

$D$  acts by scaling along eigenvector directions

Basis Change Flow

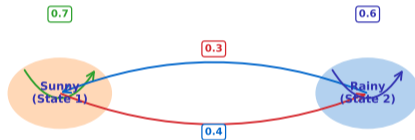


Workflow:  $x \xrightarrow{P^{-1}} y \xrightarrow{D} Dy \xrightarrow{P} PDy = Ax$

## Stochastic Matrices

- Stochastic matrix  $P$ : columns sum to 1, all entries  $\geq 0$
- State evolution:  $\mathbf{x}^{(k+1)} = P \cdot \mathbf{x}^{(k)}$

### Markov Chain: Weather Model



$$P = \begin{bmatrix} 0.7 & 0.4 \\ 0.3 & 0.6 \end{bmatrix}$$

Each arrow shows transition probability (edges sum to 1 from each state)

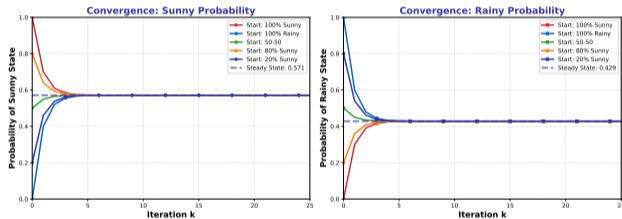
Weather example:  $P = \begin{pmatrix} 0.7 & 0.4 \\ 0.3 & 0.6 \end{pmatrix}$  — Sunny  $\rightarrow$  Sunny: 70%, Rainy  $\rightarrow$  Sunny: 40%

# Steady State: The Dominant Eigenvector

## Long-Term Behavior

- Steady state  $\pi$  satisfies  $P\pi = \pi$  (eigenvector with  $\lambda = 1$ )
- Every stochastic matrix has eigenvalue  $\lambda = 1$

Markov Chain Convergence to Steady State



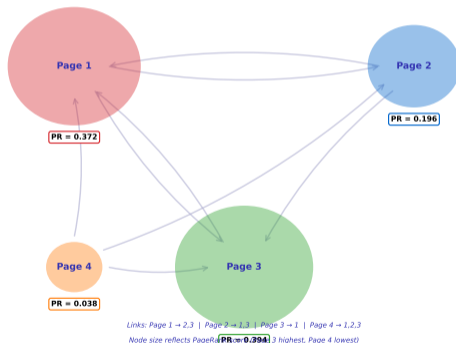
Weather model steady state:  $\pi = (4/7, 3/7) \approx (0.57, 0.43)$  — Long-term: 57% sunny

# PageRank: How Google Ranks Web Pages

## The Key Idea: Links = Votes of Importance

- A link from page A to page B is like A “voting” for B
- Pages with many incoming links are more important
- Links from important pages count more than links from unimportant pages

PageRank Network: 4-Page Example



Random surfer model: imagine clicking random links — where do you end up most often?

## A Tiny Web with 3 Pages

### The Network:

- Page A links to B and C
- Page B links to C only
- Page C links to A only

### Transition Matrix:

$$G = \begin{pmatrix} 0 & 0 & 1 \\ 0.5 & 0 & 0 \\ 0.5 & 1 & 0 \end{pmatrix}$$

Page C wins because it gets a link from important page A; B loses because only unimportant A links to it

### Computing PageRank:

Start:  $\pi^{(0)} = (0.33, 0.33, 0.33)$

After iterations:

$$\pi^{(\infty)} = (0.4, 0.2, 0.4)$$

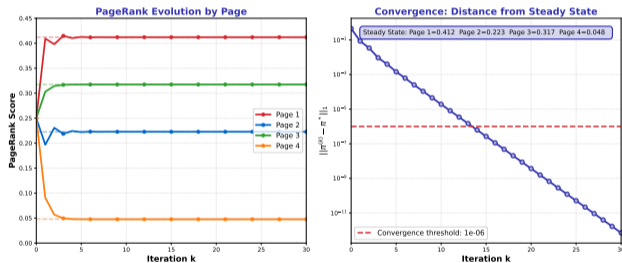
**Result:** Pages A and C tie for most important (40% each), B is least important (20%)

# PageRank = Eigenvalue Problem in Disguise

## From Random Surfing to Linear Algebra

- Build matrix  $G$ : entry  $G_{ij}$  = probability of going from page  $j$  to page  $i$
- PageRank scores = steady state  $\pi$  where  $G\pi = \pi$  (eigenvector with  $\lambda = 1$ )
- Compute by iteration: start anywhere, keep multiplying by  $G$ , converges to answer

PageRank: Iteration and Convergence ( $\alpha=0.85$ )



Google (1998): This simple eigenvalue idea revolutionized web search for billions of pages