Linear Algebra Week 5

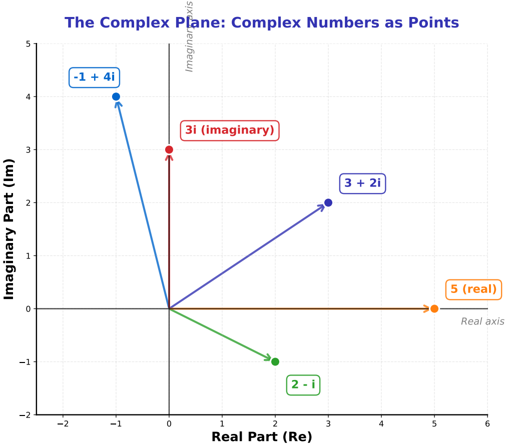
Summary: Eigenvalues and Applications

16 Key Concepts with Visualizations

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

The Complex Plane (Argand Diagram)

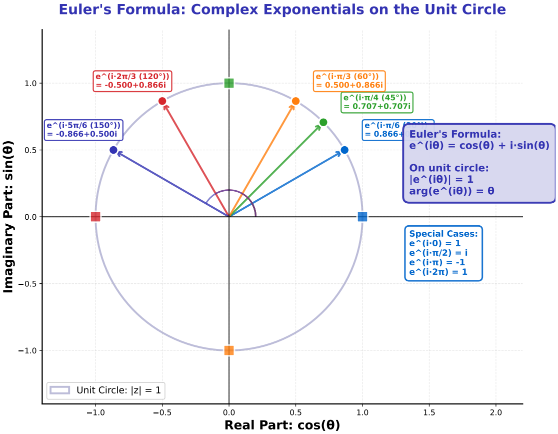
- Complex number $z = a + bi$ visualized as point (a, b) in 2D plane
- Key property: $i^2 = -1$ extends real numbers to solve all polynomials



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

Polar Form and Euler's Formula

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

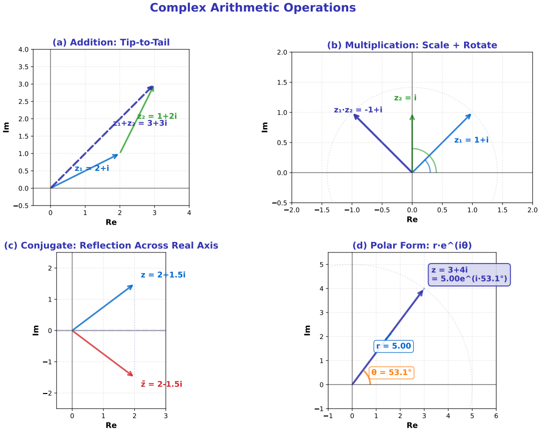


Example: $z = 1 + i$ **in polar:** $r = \sqrt{2}$, $\theta = \pi/4$, **so** $z = \sqrt{2} \cdot e^{i\pi/4}$

Complex Arithmetic: Four Fundamental Operations

Operations in the Complex Plane

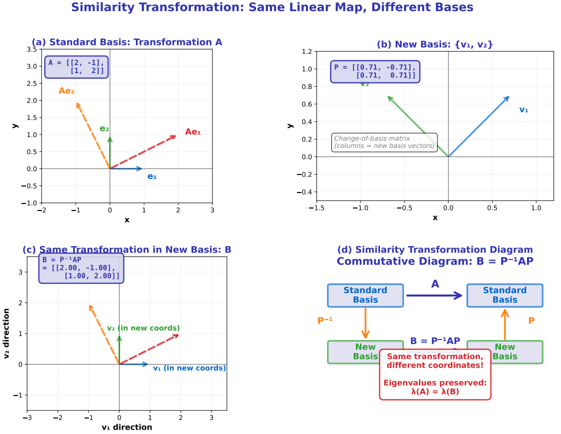
- Addition: vector addition (component-wise)
- Multiplication: scale moduli, add angles (rotation + scaling)



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

Similarity Transformation

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

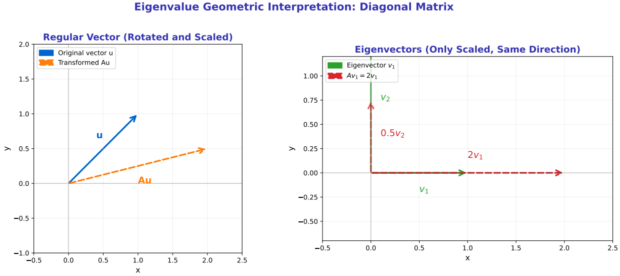


P **is the change-of-basis matrix; diagonalization is a special case of similarity**

Eigenvectors: Special Directions That Only Scale

The Fundamental Equation: $A\mathbf{v} = \lambda\mathbf{v}$

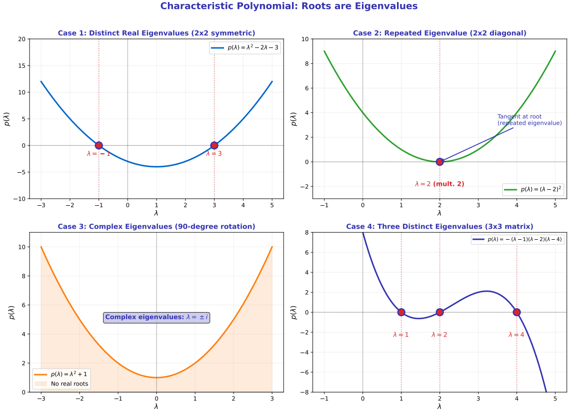
- Eigenvector $\mathbf{v}$: direction preserved by transformation A
- Eigenvalue λ: scaling factor ($\lambda > 1$ stretch, $0 < \lambda < 1$ shrink, $\lambda < 0$ flip)



Example: $A = \begin{pmatrix} 2 & 0 \\ 0 & 0.5 \end{pmatrix}$ **has** $\mathbf{v}_1 = (1, 0)$ **with** $\lambda_1 = 2$, $\mathbf{v}_2 = (0, 1)$ **with** $\lambda_2 = 0.5$

Finding Eigenvalues

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

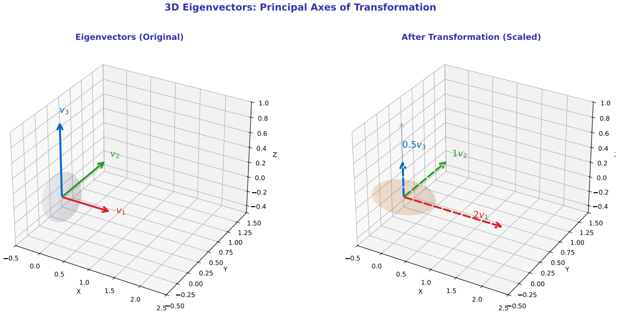


Example: $A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \Rightarrow p(\lambda) = \lambda^2 - 2\lambda - 3 = (\lambda - 3)(\lambda + 1)$, **so** $\lambda = 3, -1$

Finding Eigenvectors: Solving the Null Space Problem

Eigenvectors from Eigenvalues

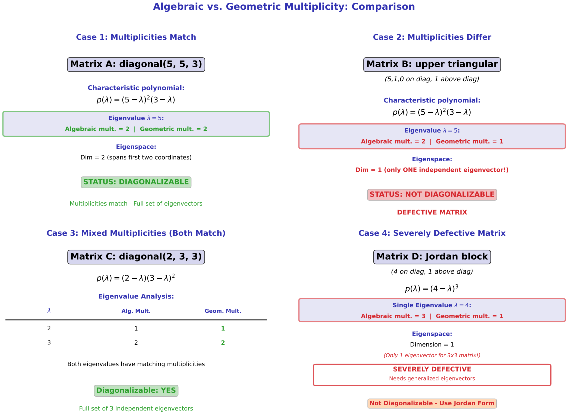
- For each eigenvalue λ, solve $(A - \lambda I)\mathbf{v} = \mathbf{0}$
- Eigenspace $E_\lambda = \ker(A - \lambda I)$ contains all eigenvectors for λ



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

When Can We Diagonalize?

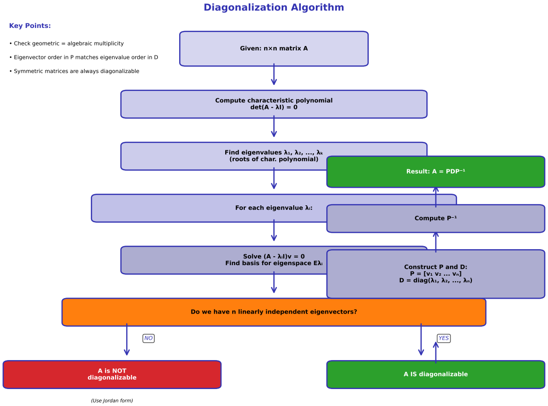
- Algebraic multiplicity: times λ appears as root of $\det(A - \lambda I)$
- Geometric multiplicity: dimension of eigenspace $\dim(E_\lambda)$



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

Finding $A = PDP^{-1}$

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

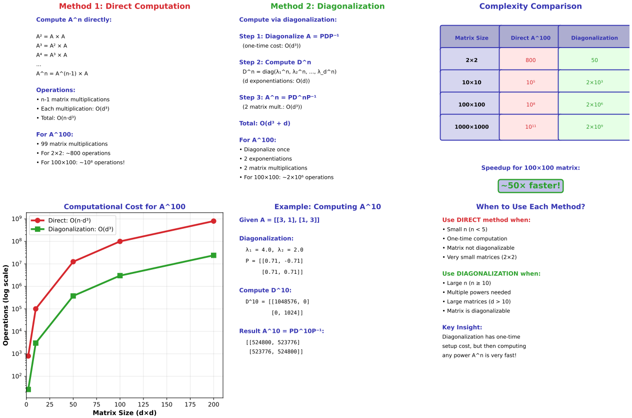


Requires n linearly independent eigenvectors; symmetric matrices always diagonalizable

Computing Matrix Powers: Huge Speedup

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

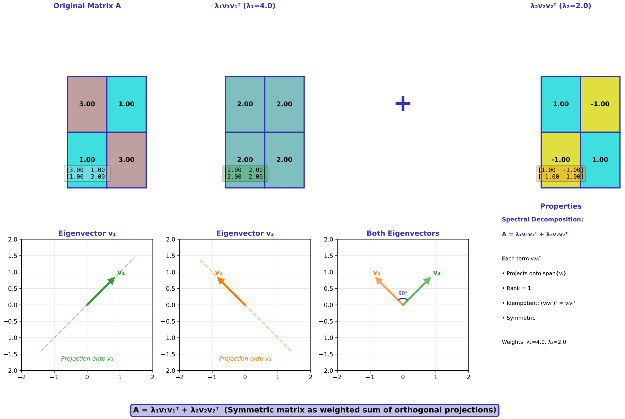
- Direct method: A^n requires $n-1$ matrix multiplications
- Via diagonalization: $D^n = \text{diag}(\lambda_1^n, \ldots, \lambda_n^n)$ is trivial



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

For Symmetric Matrices

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

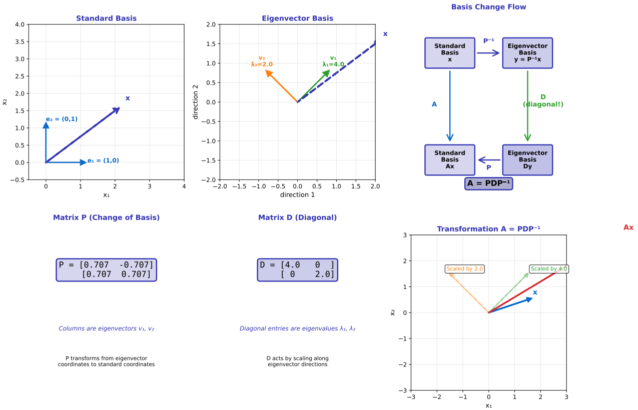


Orthonormal eigenvectors for symmetric matrices; basis for PCA in data science

Diagonalization as Basis Change

Standard → Eigenbasis → Scale → Standard

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

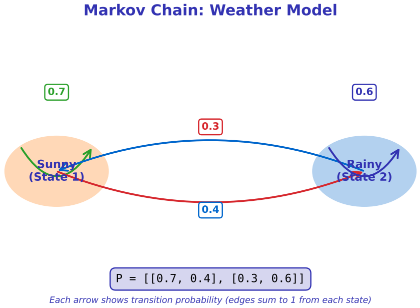


Workflow: $\mathbf{x} \xrightarrow{P^{-1}} \mathbf{y} \xrightarrow{D} D\mathbf{y} \xrightarrow{P} PD\mathbf{y} = A\mathbf{x}$

Markov Chains: Probabilistic State Transitions

Stochastic Matrices

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

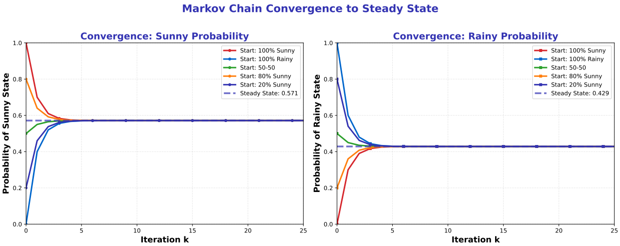


Weather example: $P = \begin{pmatrix} 0.7 & 0.4 \\ 0.3 & 0.6 \end{pmatrix}$ **— Sunny→Sunny: 70%, Rainy→Sunny: 40%**

Steady State: The Dominant Eigenvector

Long-Term Behavior

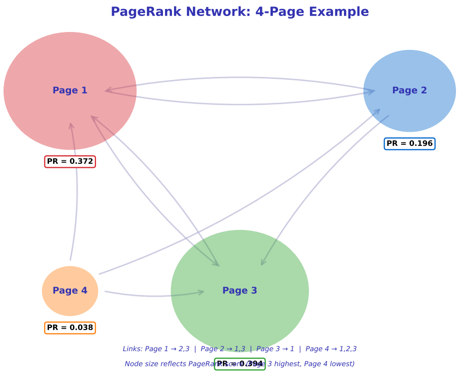
- Steady state $\boldsymbol{\pi}$ satisfies $P\boldsymbol{\pi} = \boldsymbol{\pi}$ (eigenvector with $\lambda = 1$)
- Every stochastic matrix has eigenvalue $\lambda = 1$



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

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



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

PageRank Example: 3 Web Pages

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:

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

Computing PageRank:

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

After iterations:

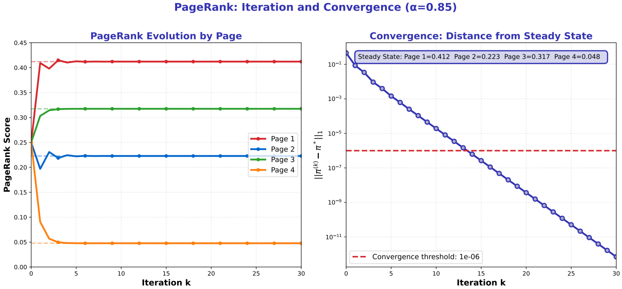
$$\boldsymbol{\pi}^{(\infty)} = (0.4, 0.2, 0.4)$$

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

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

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 $\boldsymbol{\pi}$ where $G\boldsymbol{\pi} = \boldsymbol{\pi}$ (eigenvector with $\lambda = 1$!)
- Compute by iteration: start anywhere, keep multiplying by G, converges to answer



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