Overview

- Course mechanics
- What this course is about
- Vectors and matrices: a geometric view
- The central equation: $y = Ax$
- Motivating examples
- The three fundamental questions, and where we are headed

Lecture notes and course materials originally by Stephen Boyd and Sanjay Lall. Revisions by Amirhossein Afsharrad.

Course mechanics

- Course website: `ee263.stanford.edu`
  - Course description, schedule, lecture slides, homework, exams, office hours
  - We will go over it together briefly today
- Three platforms you should have access to:
  - **Canvas**: lecture recordings
  - **Ed Discussion**: questions and course discussion
  - **Gradescope**: homework submission and grades

Course requirements

- Weekly homework assignments, due Thursdays at 11:59pm
- **Midterm exam**: in-person, Thursday July 16, 9:30am–11:00am
- **Final exam**: in-person, Friday August 14, 9:00am–11:30am
- CGOE/SCPD students may take exams remotely via Exam Monitor, following Stanford CGOE procedures
- No electronics permitted during exams; one formula sheet allowed
- Grading weights: homework 25%, midterm 25%, final 50%

Prerequisites

- Linear algebra, $e.g.$, Engr108 or Math104
- Basic probability

Not required, but increases appreciation:

- Signal processing
- Machine learning
- Control systems or circuits
- Optimization

Changes to EE263

- In previous years, EE263 was titled *Introduction to Linear Dynamical Systems*
- Starting Fall 2025, it was split into two courses:
  - **EE263**: *Matrix Methods: Singular Value Decomposition* (this course)
  - **EE363**: *Linear Dynamical Systems* (offered in Spring)
- This course covers an expanded version of the linear algebra from the old EE263, with the same emphasis on applications
- Good preparation for both EE363 and EE364a
- **If you took the old EE263**: you cannot and should not take this course; you may wish to take EE363 instead

What this course is about

This is an advanced course in applied linear algebra.

The central object is the linear map $y = Ax$. An enormous range of problems in engineering, science, and data analysis reduces to understanding this equation:

- *Estimation*: given measurements y, determine x
- *Control/design*: choose x to achieve a desired y
- *Approximation*: when no exact solution exists, find the best one

The goal is both **mathematical depth** (knowing what these tools are actually doing) and **practical fluency** (being able to use them in real problems from across engineering and science)

Course outline

1. **Linear functions**: $y = Ax$, interpretations, linearization
2. **Range and null space**: what outputs are reachable, what inputs are invisible
3. **Orthogonality and QR factorization**: geometry of inner products
4. **Least-squares**: best approximate solution when $Ax = y$ has no exact solution
5. **Least-norm solutions**: when $Ax = y$ has infinitely many solutions
6. **Recursive estimation and data fitting**
7. **Eigenvalues and symmetric matrices**: quadratic forms, positive definiteness
8. **Singular value decomposition (SVD)**: the workhorse of modern data analysis
9. **Gaussian distributions and MMSE estimation**: the linear statistical model
10. **Spectral graph embedding**

Applications throughout: signal processing, machine learning, circuits, navigation, control, finance, and more

Notation

- Scalars and column vectors: lowercase letters, $e.g.$, x, y, λ, σ
  - Row vector: transpose of a column vector, x^T
- Matrices: uppercase letters, $e.g.$, A, B, Σ
  - In terms of columns: $A = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \in \mathbb{R}^{m\times n}$, $\quad a_j \in \mathbb{R}^m$
  - In terms of rows: $A = \begin{bmatrix} \tilde{a}_1^\mathsf{T} \\ \vdots \\ \tilde{a}_m^\mathsf{T} \end{bmatrix}$, $\quad \tilde{a}_i \in \mathbb{R}^n$
- Special matrices and vectors:
  - Identity matrix: I or I_n (when size matters)
  - Zero matrix/vector: 0, or 0_n (vector), $0_{m\times n}$ (matrix)
  - Ones vector/matrix: $\mathbb{1}$, or $\mathbb{1}_n$ (vector), $\mathbb{1}_{m\times n}$ (matrix)
  - Diagonal matrix: $\mathbf{diag}(d_1, d_2, \ldots, d_n)$
  - Block matrix: assembled from submatrices, $e.g.$, $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ with A, B, C, D matrices of compatible sizes

$x \in \mathbb{R}^n \qquad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

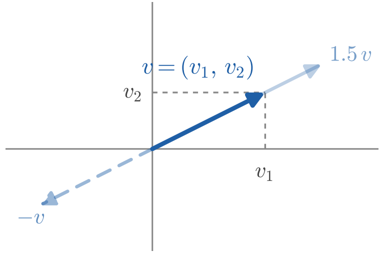
$I_n = \begin{bmatrix} 1 & & \emptyset \\ & 1 & \\ \emptyset & & \ddots \quad 1 \end{bmatrix}$

Vectors: beyond arrays of numbers

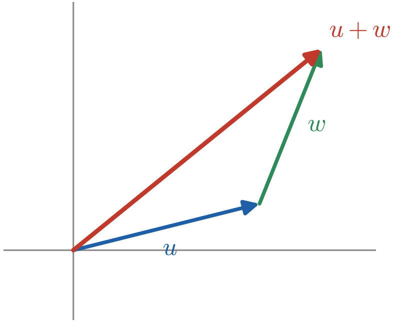
A vector $x \in \mathbb{R}^n$ is a point in n-dimensional space, but it can represent many different things:

- a position in 2D or 3D ($n = 2, 3$)
- a portfolio of n assets (long/short)
- a discrete-time signal at n times
- a feature vector in machine learning
- the pixels of a grayscale image
- the powers of n transmitters

A vector and its scalar multiples



Vector addition (tip to tail)



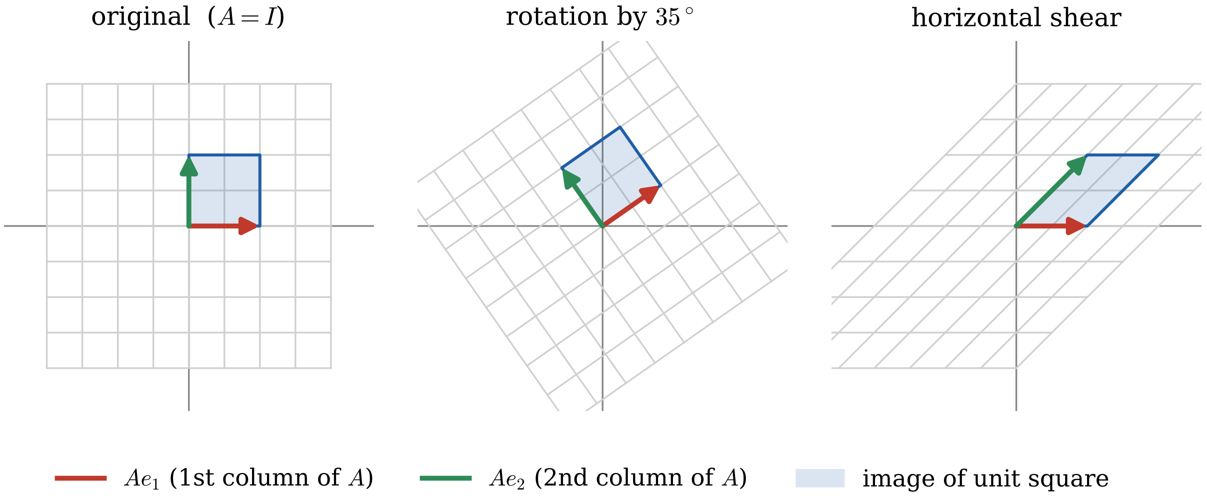
Matrices: functions on vectors

$A \in \mathbb{R}^{m \times n}$

$a \in \mathbb{R}^{mn}$

A matrix $A \in \mathbb{R}^{m\times n}$ defines a *function* $x \in \mathbb{R}^n \mapsto Ax \in \mathbb{R}^m$

Geometrically (for $m = n = 2$): A *transforms* the plane



Different matrices do very different things: stretch, rotate, reflect, project, collapse. This geometric view will be central all quarter

The central equation

$y = Ax$

$\min \| y - Ax \|$

$m > n$

A system of m linear equations in n unknowns:

$$
\begin{aligned}
y_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\
y_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\
&\vdots \\
y_m &= a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n
\end{aligned}
$$

$m = n$

$m < n$

Compact matrix form: $y = Ax$, where

$$
y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}, \qquad A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, \qquad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}
$$

$5 = x_1 + 2x_2$

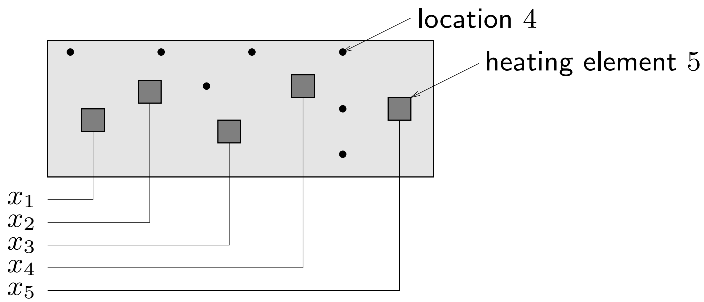
$6 = x_1 - x_2$

$-3 = x_1 + 8x_2$

$0 = 2x_1 - x_2$

a_{ij} is the *gain factor* from jth input x_j to ith output y_i

Example: Thermal system



- n heaters in a slab: x_j = power of heater j, y_i = steady-state temperature rise at location i
- $y = Ax$, where a_{ij} = temperature rise at location i per watt at heater j (in °C/W)

Problems that become linear algebra:

- *Control*: choose heater powers x to reach a target temperature profile y_{des}
- *Estimation*: infer heater powers from temperature measurements
- *Feasibility*: which temperature profiles are achievable at all?

Example: Data fitting and machine learning

Given N data points (z_i, w_i), fit a model of the form

$$w_i \approx \underline{x_1}\phi_1(z_i) + \underline{x_2}\phi_2(z_i) + \cdots + \underline{x_n}\phi_n(z_i)$$

In matrix form: $w \approx Ax$, where $A_{ij} = \phi_j(z_i)$

Fit a 3rd deg. polynomial

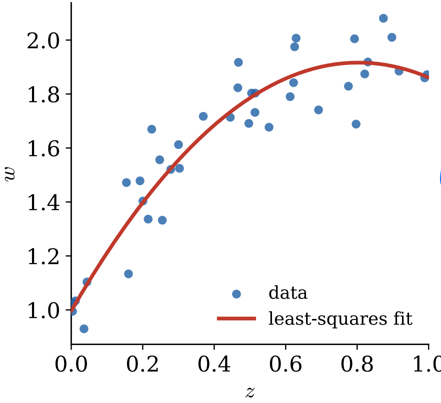
$f(t) = \underline{x_1} + \underline{x_2} t + \underline{x_3} t^2 + \underline{x_4} t^3$

$y = A\underline{x}$

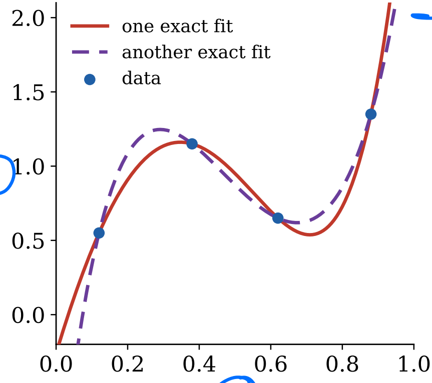
$\mathbb{R}^{100}$ $\qquad$ $\mathbb{R}^4$

$A \in \mathbb{R}^{100 \times 4}$

Overdetermined: $N \gg n$



Underdetermined: $N \ll n$



w	z
5	1
7	2
2	-1

$5 = f(1)$

$= x_1 + x_2 + x_3 + x_4$

$7 = x_1 + 2x_2 + 4x_3 + 8x_4$

- overdetermined ($N \gg n$): more data than parameters $\rightarrow$ **least-squares**
- underdetermined ($N \ll n$): more parameters than data $\rightarrow$ **least-norm, regularization**

Example: Portfolio returns

n assets, m time periods

- x_j = weight of asset j in portfolio (positive = long, negative = short)
- a_{ij} = return of asset j in period i
- $y_i = \sum_j a_{ij}x_j$ = portfolio return in period i: $\quad y = Ax$

Problems that become linear algebra:

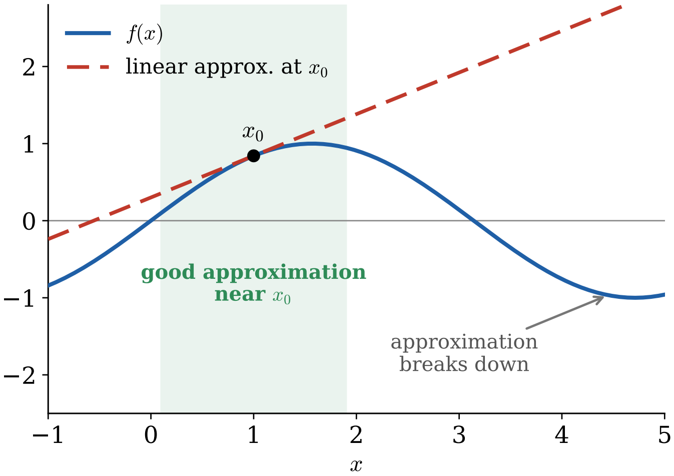
- *Design*: find a portfolio achieving target return profile y_{des}
- *Hedging*: among all portfolios achieving y_{des}, find the one with minimum total capital deployed
- *Factor analysis*: which combinations of assets explain the variance in returns? (this is SVD)

The same mathematical structure appears in circuits, signal processing, navigation, and climate modeling

Why linear? Linearization

Many real systems are nonlinear. Why study linear ones?

- Any smooth function looks linear locally
- Linear methods work surprisingly well even for nonlinear systems
- If you don't understand linear, you can't understand nonlinear



If $f : \mathbb{R}^n \to \mathbb{R}^m$ is differentiable at x_0:

$$f(x) \approx f(x_0) + Df(x_0)(x - x_0)$$

the *Jacobian* $Df(x_0) \in \mathbb{R}^{m \times n}$ has entries

$$Df(x_0)_{ij} = \left.\frac{\partial f_i}{\partial x_j}\right|_{x_0}$$

The three fundamental questions

Given $A \in \mathbb{R}^{m \times n}$ and $y \in \mathbb{R}^m$, we ask about $y = Ax$:

1. **Existence**: is there any x such that $Ax = y$?
   - equivalently: is y in the *range* of A?
2. **Uniqueness**: if such an x exists, is it the only one?
   - equivalently: is the *null space* of A trivial?
3. **Best approximation**: if no exact solution exists, what is the closest we can get?
   - find x minimizing $||Ax - y||$ (the **least-squares** problem)

These three questions drive the first half of the course. The SVD, which we build toward, gives a complete and unified answer to all three.