ORF307 — Optimization

1. Introduction

Bartolomeo Stellato - Spring 2025

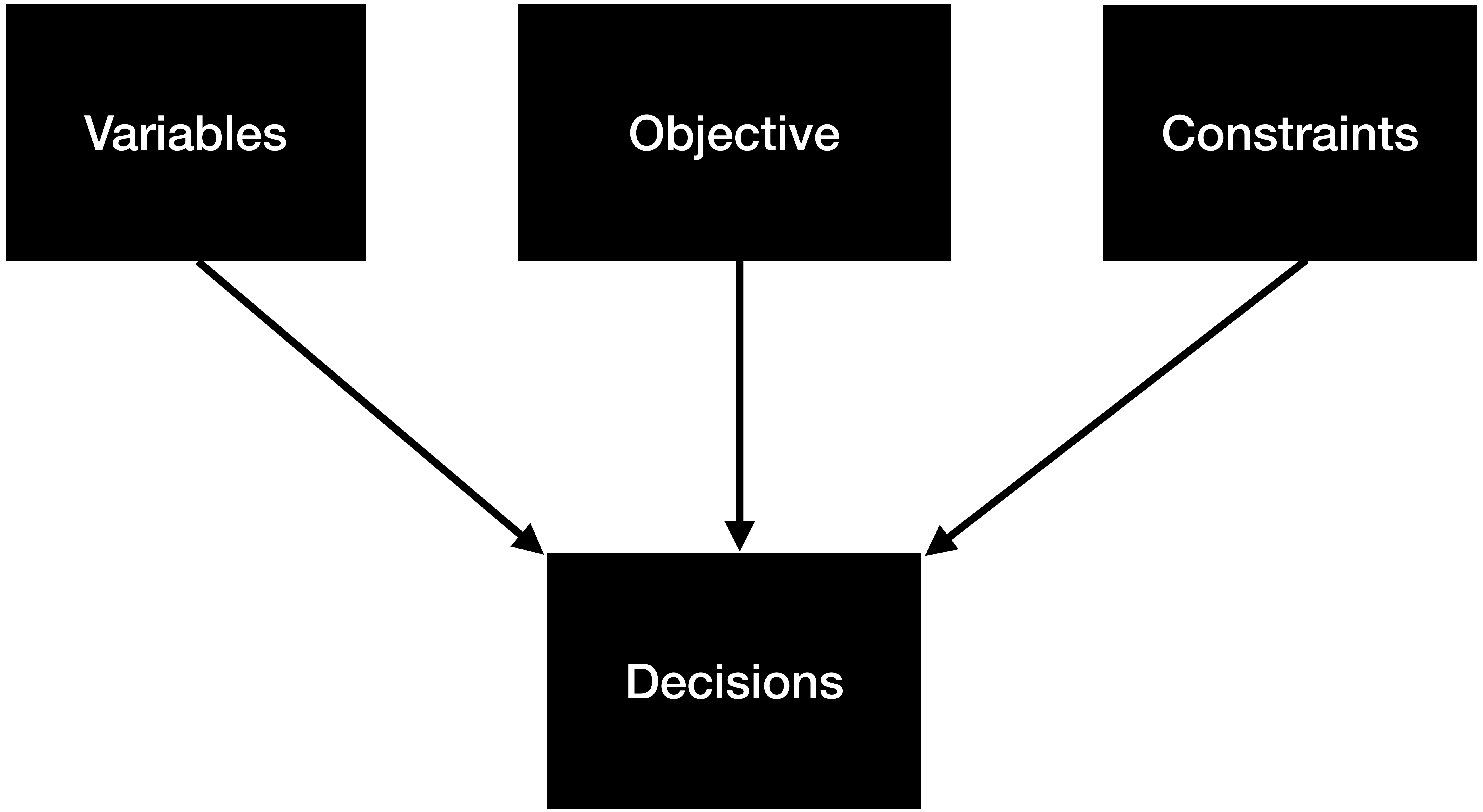
Meet your classmates!

What is your department?

https://www.menti.com/bldoiyg1sj3r

What is this course about?

The mathematics behind making optimal decisions



Mathematical optimization

The problem

minimize	**objective**
subject to	**constraints**

with respect to **variables**

Finance

Variables
Amounts invested in each asset

Constraints
Budget, investment per asset,
minimum return, etc.

Objective
Maximize profit, minus risk

Optimal control

Variables
Inputs: thrust, flaps, etc.

Constraints
System limitations, obstacles, etc.

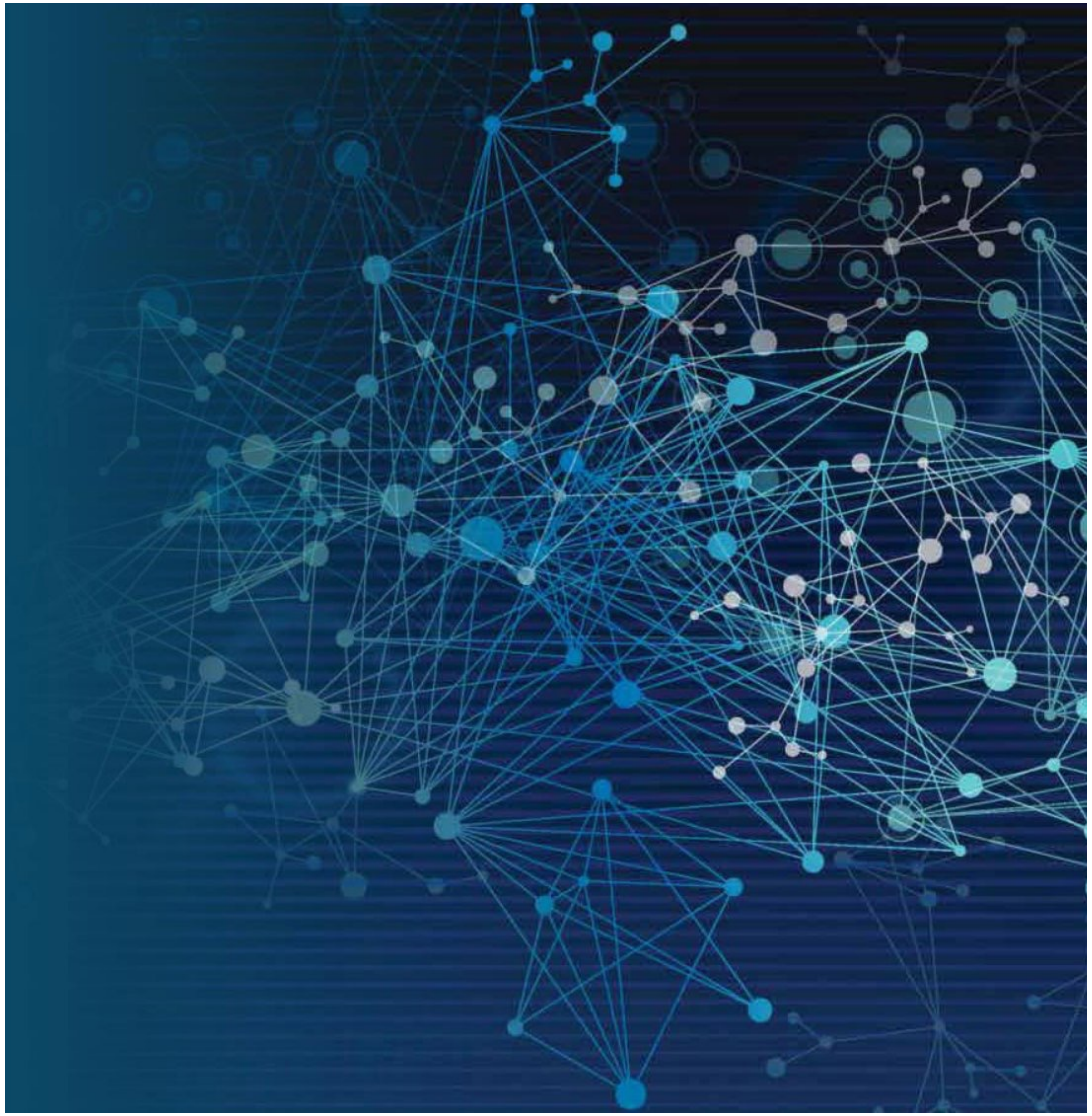
Objective
Minimize distance to target and fuel consumption

Machine learning

Variables
Model parameters

Constraints
Prior information, parameter limits

Objective
Minimize prediction error, plus regularization



Most optimization problems cannot be solved

Solving optimization problems

General case ⟶ **Very hard!**

Compromises

- Long computation times
- Not finding the solution
  (in practice it may not matter)

Exceptions

- Least squares
- Linear optimization
- Convex optimization

⟶ **Can be solved very efficiently and reliably**

Meet your instructors

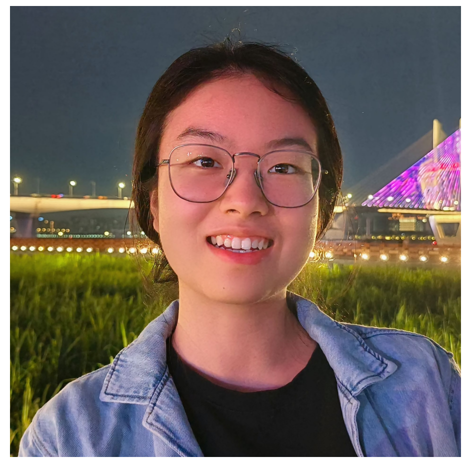
Bartolomeo Stellato
I am an Assistant Professor at ORFE. I obtained my PhD from the University of Oxford and I was a postdoc at MIT.

email: bstellato@princeton.edu

office hours: Tue 2:00pm-3:30pm EST, Sherrerd 123

website: stellato.io

Meet your assistants in instruction (AIs)



Yanjun Liu

email: yanjun.liu@princeton.edu

OHs: Thu 1:30pm-3:00pm EST, Sherrerd 123

Qishuo Yin

email: qy1448@princeton.edu

OHs: Tue 6:00pm-7:30pm EST, Sherrerd 107

Chenyu Yu

email: cy6247@princeton.edu

OHs: Wed 6:00pm-7:30pm EST, Sherrerd 107

Kaiwen Zhang

email: kz5126@princeton.edu

OHs: Thu 6:00pm-7:30pm EST, Sherrerd 122

Today's agenda

- Technological innovation
- A bit of history
- Course contents and information
- Notation and basic definitions

Technological innovations

Lots of data

easy storage
and
transmission

Massive computations

computers
are
super fast

High-level programming languages

easy to
do complex
stuff

History of optimization

Age of computers

Algorithms

Origin of linear optimization (Kantorovich, Koopmans, von Neumann)
1930s

Simplex algorithm (Dantzig)
1947

Interior-point methods (Karmarkar)
1984

Large-scale optimization
2000s

Applications

Operations Research
Economics

Engineering
Statistics
1990s

Machine learning
Image processing
Communication systems
Embedded intelligent systems
2000s

Contents of this course

Least-squares

- Solving linear systems in practice
- Modelling and applications
- Multiobjective least squares
- Constrained least squares

Linear optimization

- Modelling and applications
- Geometry
- The simplex method
- Duality
- Network optimization
- Interior point methods

Extensions

- Mixed-integer optimization
- Branch and bound algorithms

Weekly schedule

Lectures

Tuesday and Thursdays 11:00am—12:20pm, CS Building 104

Precepts

- P01: Tuesday 7:30pm—8:20pm, Sherrerd 001
- P02: Tuesday 7:30pm—8:20pm, Sherrerd 101
- P03: Wednesday 7:30pm — 8:20pm, Sherrerd 001
- P04: Wednesday 7:30pm — 8:20pm, Sherrerd 101

Course information

Grading

- **30% Homeworks**
  8 bi-weekly homeworks with coding component. Available on Thursday, deadline Friday 9pm of the following week. Collaborations are encouraged!

- **40% Two midterms**
  80 minutes written exam in class. No collaborations.

- **30% Final project**
  24 hours take-home project with coding component. No collaborations.

Generative AI Policy

Generative AI tools (like Chat GPT/GitHub Copilot) are **allowed but discouraged**

They often make subtle mathematical/logical mistakes and could hinder your learning

If you use a generative AI tool, you must do the following (in your solution sheet):

- **Declare it**
- **Describe how you used it**
- **Include prompt and relevant output**

Failure to do so is a violation of the ***University's academic regulations*** (Sec. 2.4.6)

Course information

Materials:

Prerequisites

- Linear algebra (MAT202 and/or MAT204)
- Basic computer programming knowledge.

Materials

Lecture slides and readings.

- Main course website: `stellato.io/teaching/orf307`
- Github repo: `github.com/ORF307/companion`

Readings

The following books are useful references (all digitally available):

- Boyd, Vandenberghe: *Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares*
- Vanderbei: *Linear Programming: Foundations & Extensions*
- Bertsimas, Tsitsiklis: *Introduction to Linear Optimization*

Software (open-source)

python™ jupyter

Numerical computations

Numerical computations on *numpy* and *scipy*.

CVXPY

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \end{array} \longrightarrow$$

```
x = cp.Variable(n)
prob = cp.Problem(
    cp.Minimize(c.T@x),
    [A @ x <= b]
)
prob.solve()
print("The optimal value is", prob.value)
print("The solution x is", x.value)
```

Learning goals

- **Model** decision-making problems across different disciplines as least squares, linear and integer optimization problems.
- **Apply** the most appropriate optimization tools when faced with a concrete problem.
- **Understand** which algorithms are slower or faster, and which problems are easier or harder to solve.

Notation and basic definitions

Vectors

vector of length n

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- We also use the notation $x = (x_1, \dots, x_n)$
- x_i is the i-th *element component*
- The set of real n-vectors is denoted as $\mathbf{R}^n$

Special vectors

- $x = 0$ (zero vector): $x_i = 0, \quad i = 1, \dots, n$
- $x = \mathbf{1}$ (vector of all ones): $x_i = 1, \quad i = 1, \dots, n$
- $x = e_i$ (unit vector): $x_i = 1,\ x_k = 0$ for $k \neq i$

Vector operations

addition

$$x + y = \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{bmatrix}$$

scalar multiplication

$$\alpha x = (\alpha x_1, \ldots, \alpha x_n)$$

inner-product (dot product)

$$x^T y = \sum_{i=1}^{n} x_i y_i = x_1 y_1 + \cdots + x_n y_n$$

Vector inner product examples

Special cases

- $e_i^T a = a_i$ (pick i-th entry)
- $\mathbf{1}^T a = \sum_{i=1}^n a_i = a_1 + \cdots + a_n$ (sum all the entries)
- $a^T a = a_1^2 + \cdots + a_n^2$ (sum of squares of entries)

Total cost

- p vector of prices
- q vector of quantities

$\downarrow$

$p^T q$ is total cost

Portfolio value

- s portfolio holdings (in shares)
- p asset prices

$\downarrow$

$p^T s$ is portfolio value

More inner product examples

Portfolio returns

- r vector of (fractional) returns

$$r_i = \frac{p_i^{\text{final}} - p_i^{\text{initial}}}{p_i^{\text{initial}}} \longrightarrow$$

$r^T w$ is the (fractional) return

- w fractional holdings

Vector norms

Euclidean norm

$$\|x\|_2 = \sqrt{x_1^2 + \cdots + x_n^2} = \sqrt{x^T x}$$

ℓ_1-norm

$$\|x\|_1 = |x_1| + \cdots + |x_n|$$

ℓ_∞-norm

$$\|x\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$$

Properties

- $\|\alpha x\| = |\alpha|\|x\|$ (homogeneous)
- $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality)
- $\|x\| \geq 0$ (nonnegativity)
- $\|x\| = 0$ if and only if $x = 0$ (definiteness)

Angle between vectors

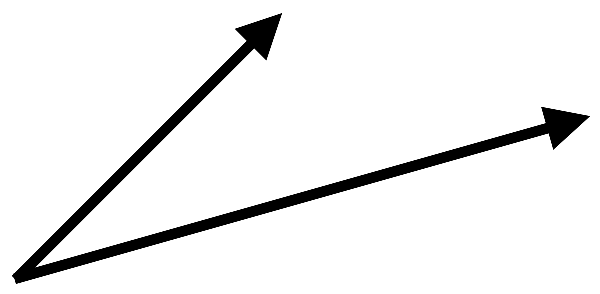
The angle $\theta = \angle(x, y)$ between x and y is the number in $[0, \pi]$ such that

$$\theta = \arccos \frac{x^T y}{\|x\| \|y\|} \qquad \text{(i.e., } x^T y = \|x\| \|y\| \cos\theta\text{)}$$

acute angle

$x^T y > 0$

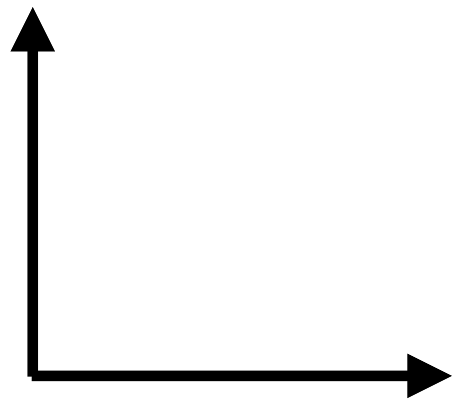
$\theta < \pi/2$



orthogonal vectors

$x^T y = 0$

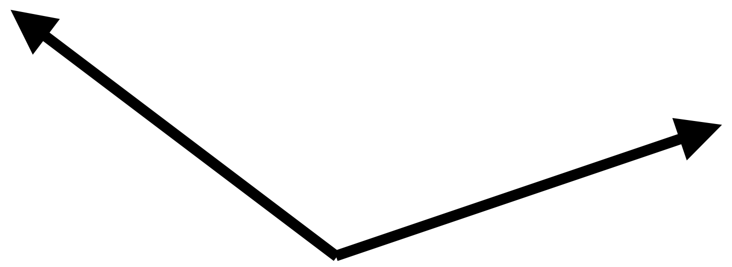
$\theta = \pi/2$



obtuse angle

$x^T y < 0$

$\theta > \pi/2$



Cauchy-Schwarz inequality

$$|x^T y| \leq \|x\| \|y\|$$

Properties

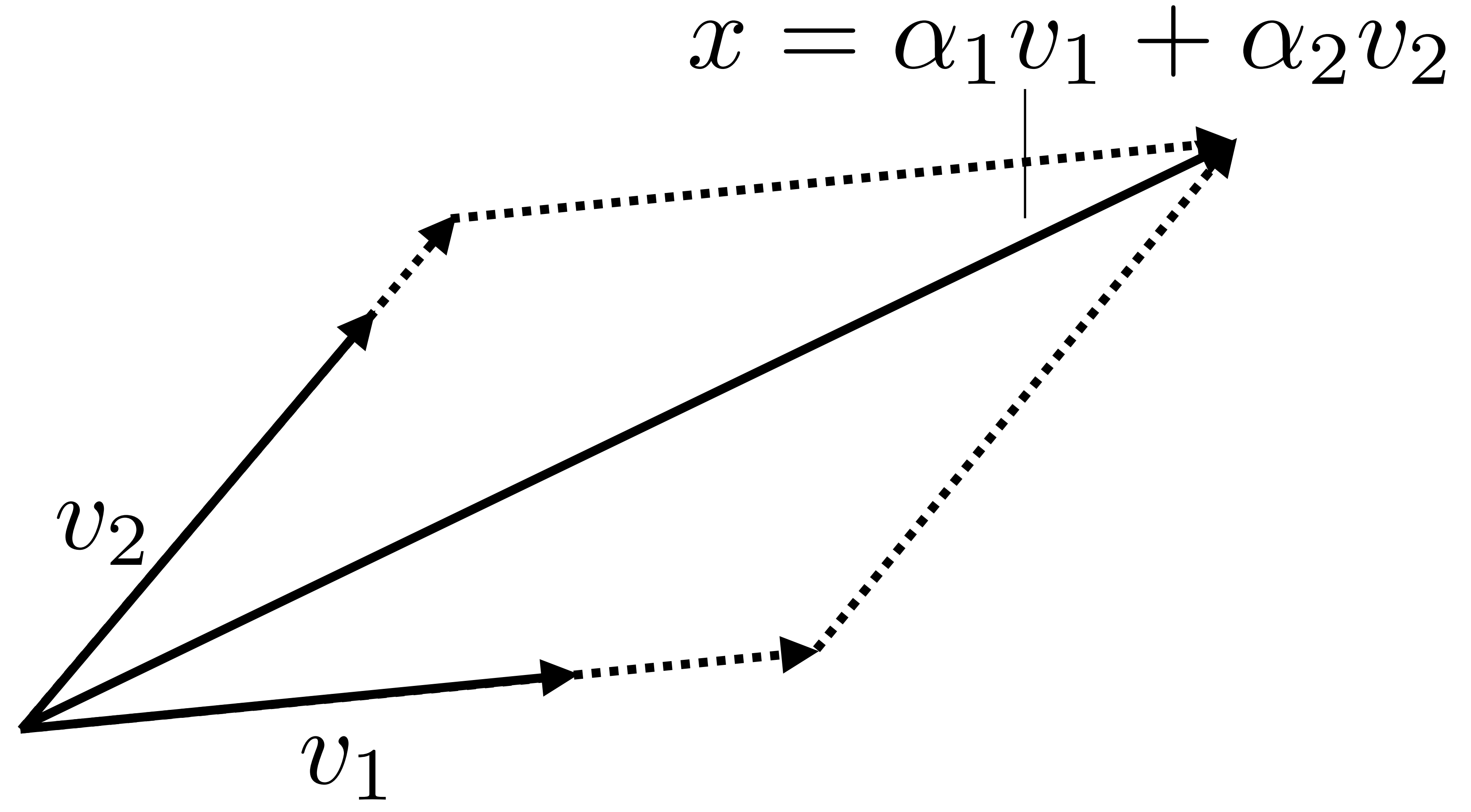
- It holds for all vectors x and y of same length
- $|x^T y| = \|x\| \|y\|$ if and only if x and y aligned

Linear independence

A nonempty set of vectors $\{v_1, \ldots, v_k\}$ is **linearly independent** if

$$\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_k v_k = 0$$

hold only for $\alpha_1 = \alpha_2 = \cdots = \alpha_k = 0$



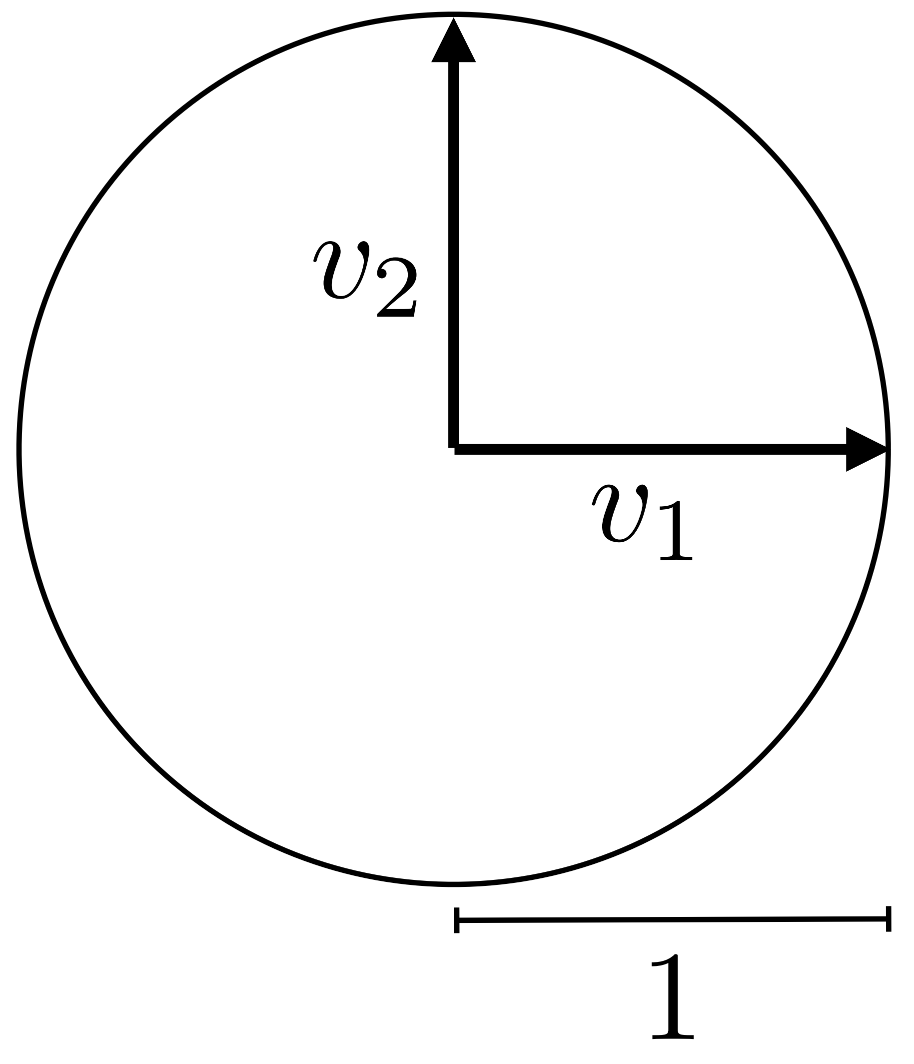
Properties

- Linear combinations have **unique coefficients** α_i
  $x = \alpha_1 v_1 + \cdots + \alpha_k v_k$
- Noone of the v_i is a linear combination of the others
- A set of n linearly independent n vectors $v_1, \ldots, v_n$ is called **basis**:
  (any n-vector x can be expressed as their linear combination)

Orthonormal vectors

A set of n-vectors $v_1, \ldots, v_k$ that

- mutually orthogonal: $v_i^T v_j = 0$ for $i \neq j$
- normalized: $\|v_i\| = 1$ for $i = 1, \ldots, k$



If $k = n$ then $v_1, \ldots, v_n$ form an **orthonormal basis**

Flop counts

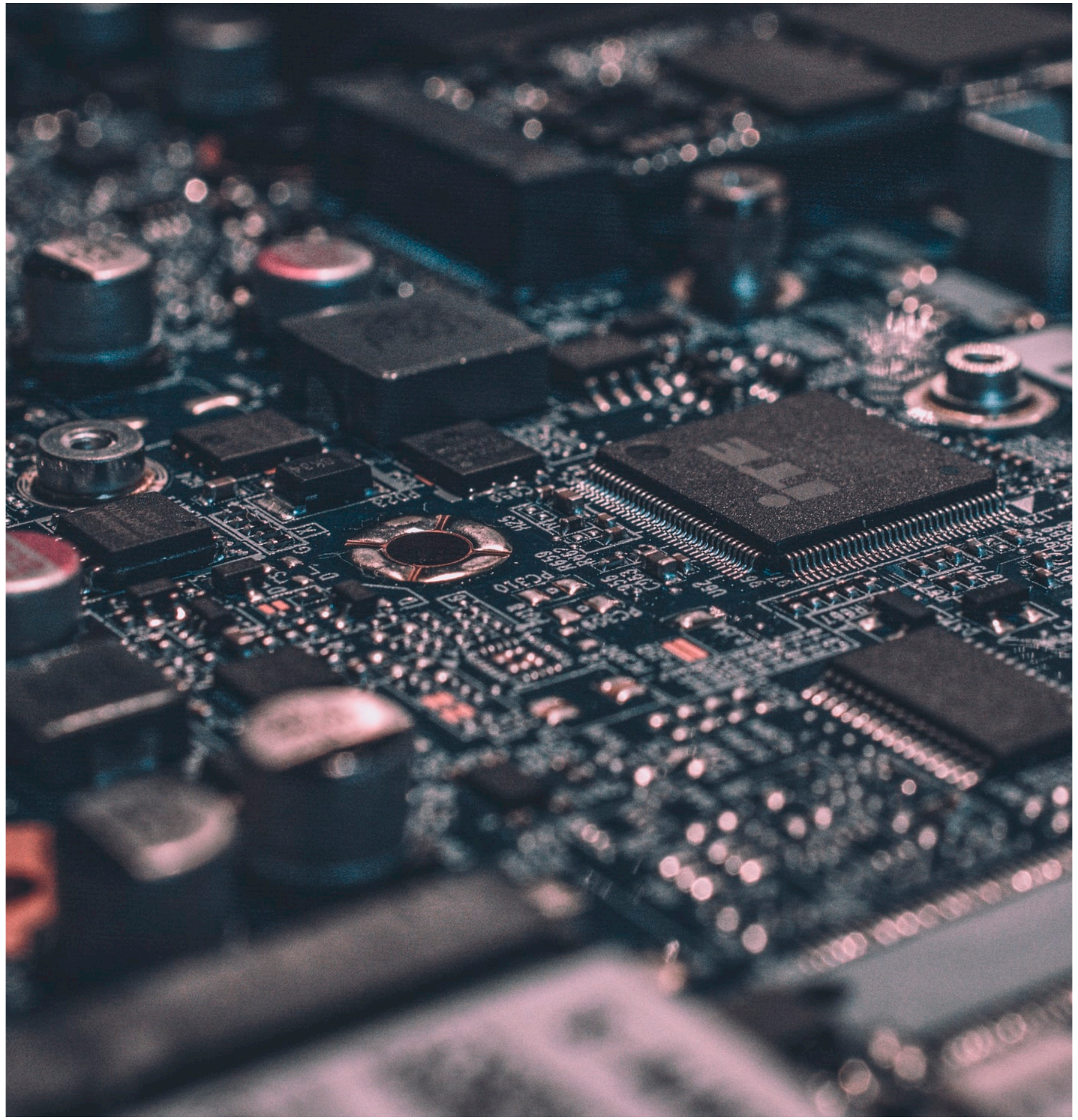
- Computers store real numbers in **floating-point format**

- Basic arithmetic operations (addition, multiplication, etc…) are called **floating point operations (flops)**

- **Algorithm complexity**: total number of flops needed as function of dimensions

- **Execution time** $\approx$ (flops)/(computer speed) *[Very grossly approximated]*

- Modern computers can go at 1 Gflop/sec (10^9 flops/sec)



Complexity of vector operations

Examples

- $x + y$ needs n addition: n flops
- $x^T y$ needs n multiplications and $n - 1$ additions: $2n - 1$ flops
  (Usually simplified as $2n$ or even n, i.e., leading term without coefficients)

Most vector operations have complexity n

Next lecture

Matrix operations and solving linear systems in practice

- Matrices operations on computers
- Solving linear systems
- Matrix factorization