Lecture 24

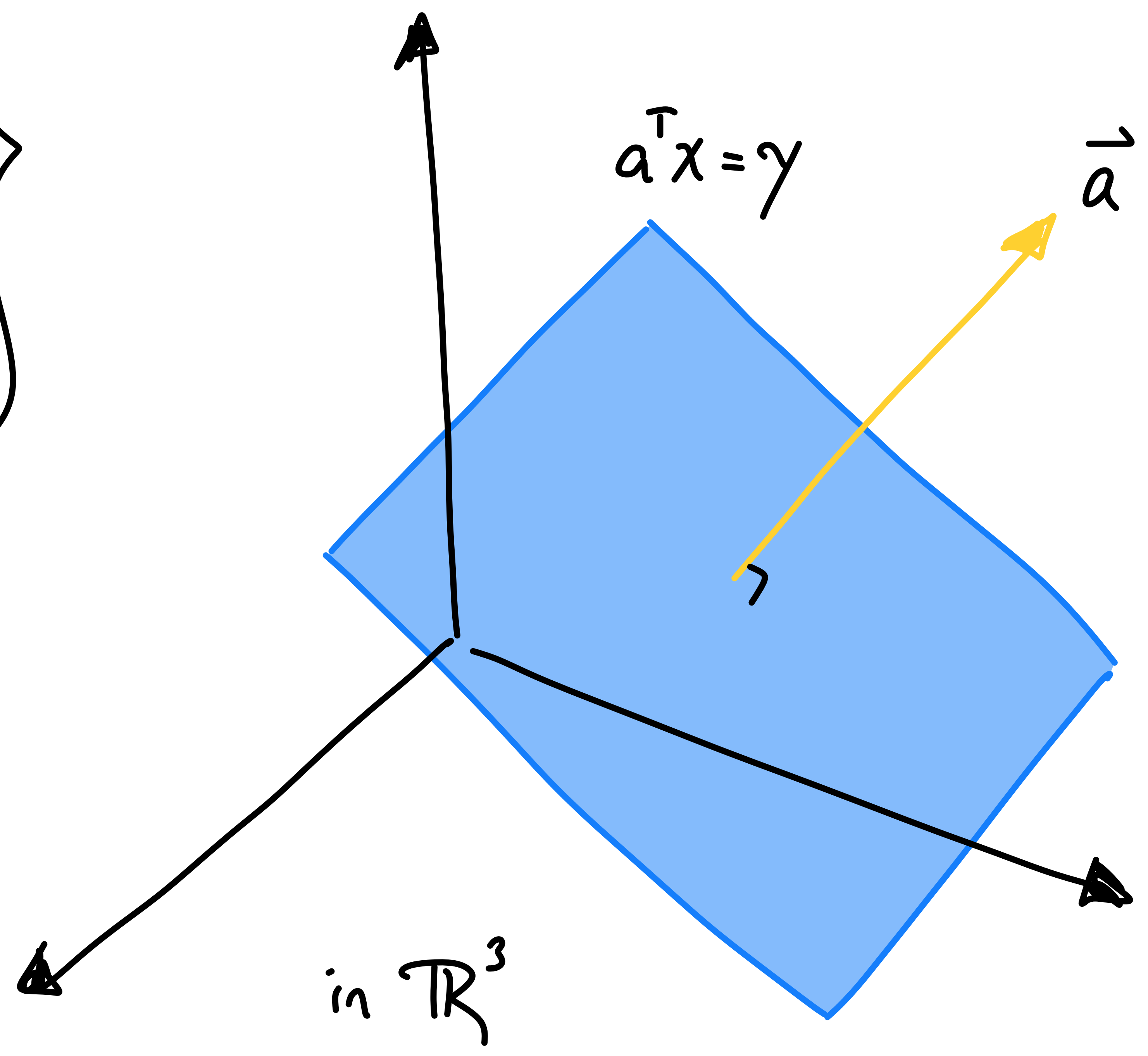
Linear programming I

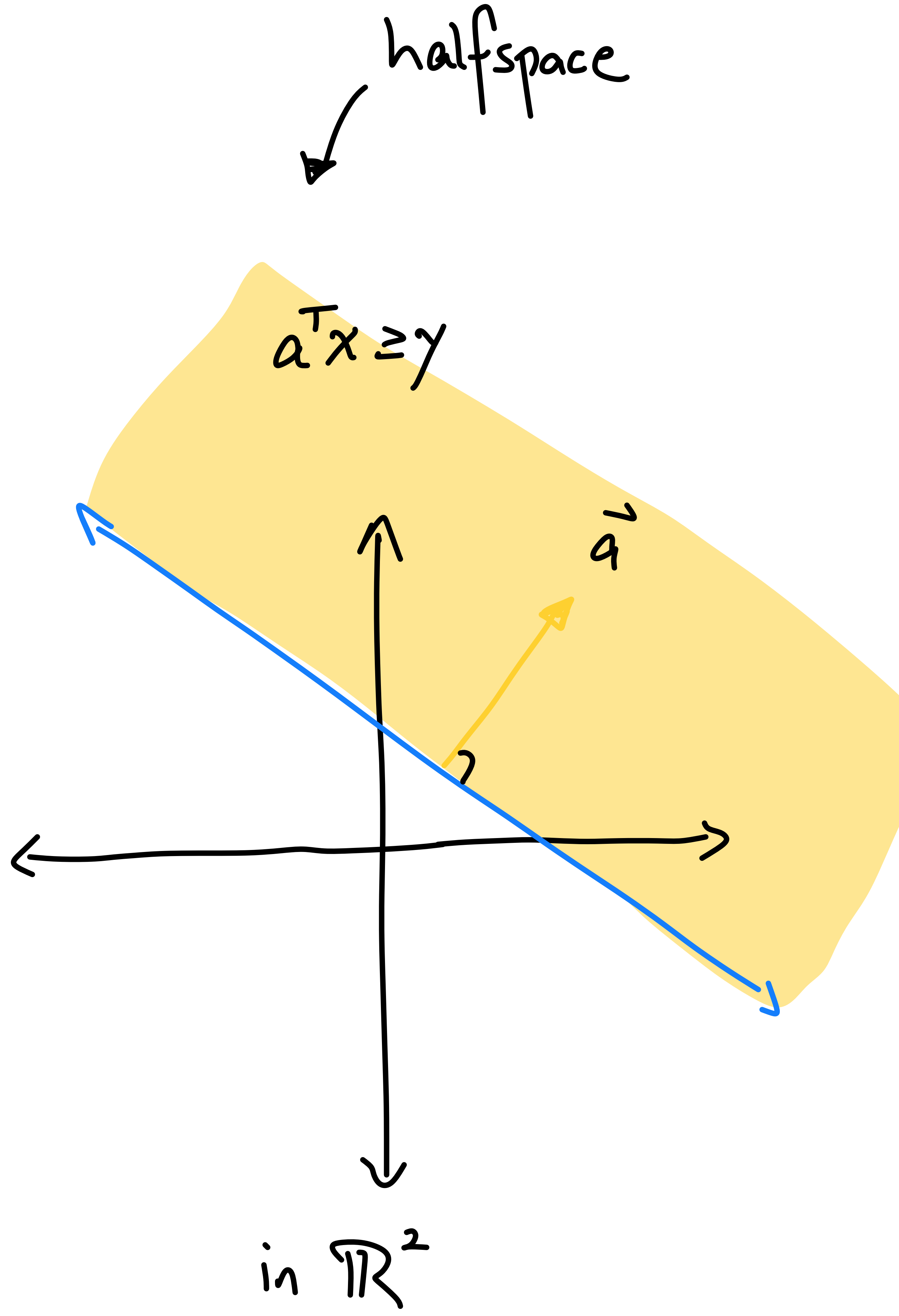
Chinmay Nirkhe | CSE 421 Winter 2026

Linear algebra/geometry review

Within the space $\mathbb{R}^n$, an affine subspace of dim $n-1$ is defind by

$$\left\{ \begin{bmatrix} | \\ x \\ | \end{bmatrix} : \begin{bmatrix} — a — \end{bmatrix} \cdot \begin{bmatrix} | \\ x \\ | \end{bmatrix} = y \right\}$$

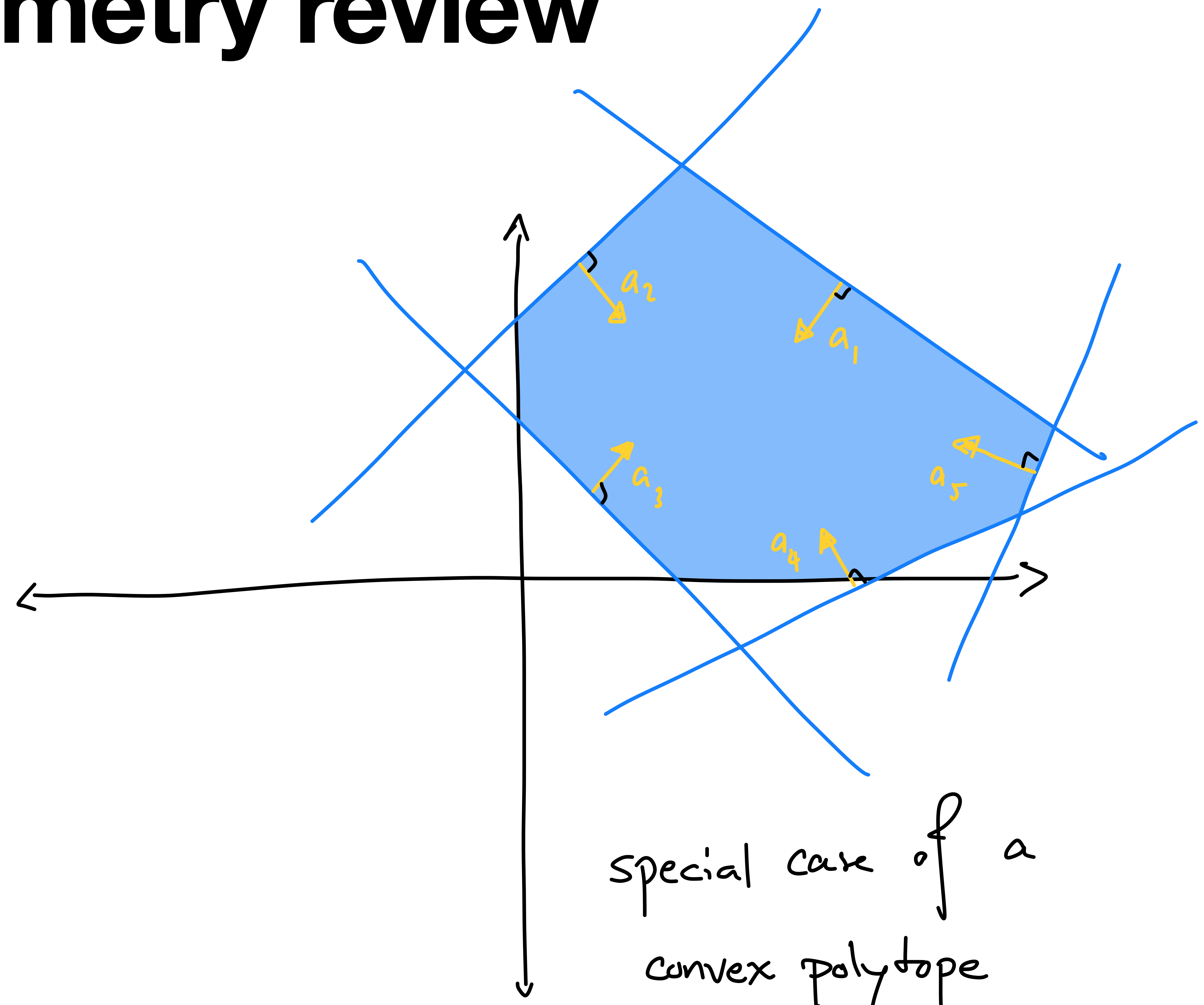




Linear algebra/geometry review

$$\begin{bmatrix} \text{—} \; a_1 \; \text{—} \\ \text{—} \; a_2 \; \text{—} \\ \vdots \\ \text{—} \; a_m \; \text{—} \end{bmatrix} \cdot \begin{bmatrix} | \\ x \\ | \end{bmatrix} \le \begin{bmatrix} | \\ b \\ | \end{bmatrix}$$

and $\begin{bmatrix} | \\ x \\ | \end{bmatrix} \ge 0.$

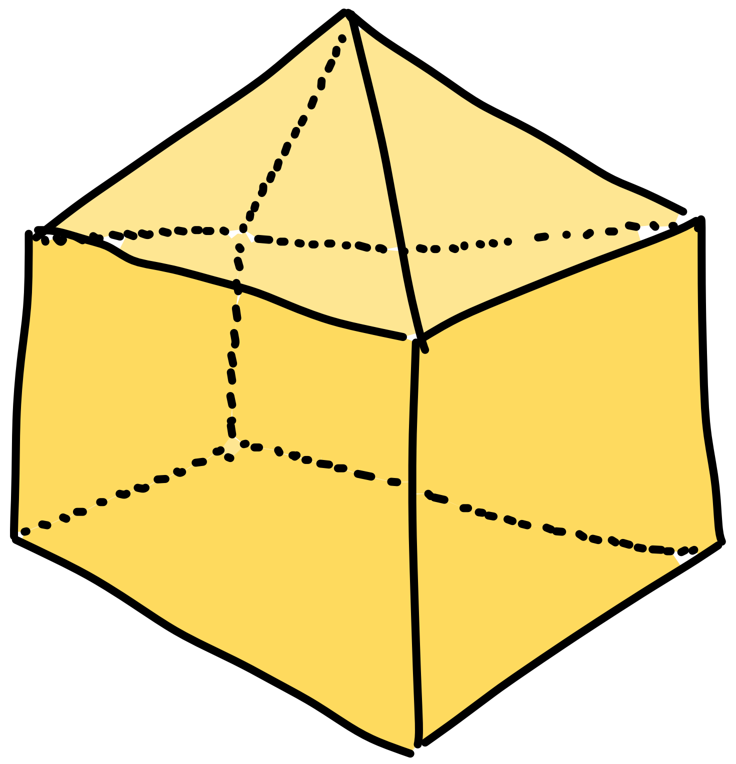


special case of a convex polytope

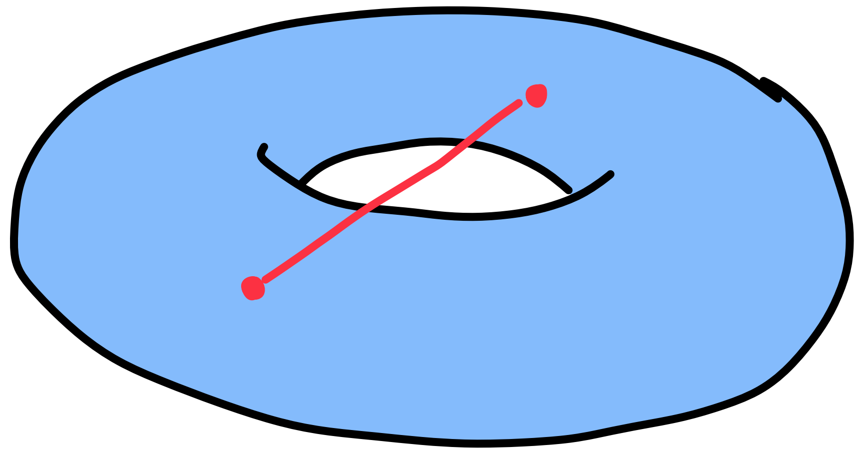
Meaning of convexity

- **Definition:** $F \subseteq \mathbb{R}^n$ is a **convex region** if for all $x, y \in F$, the line segment $\overline{xy}$ is contained in F – i.e. for $\lambda \in [0,1], \lambda x + (1-\lambda)y \in F$.

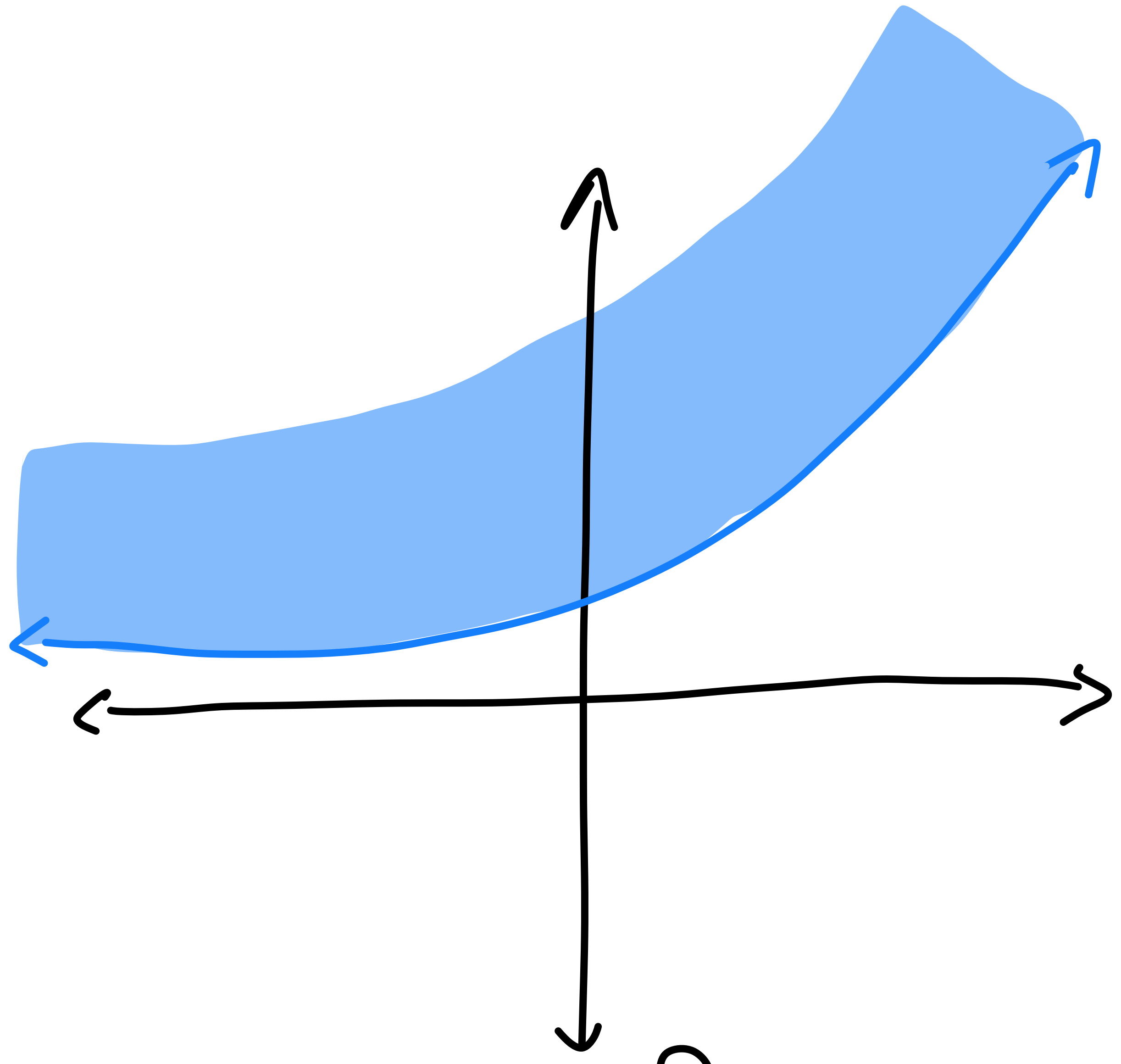
- **Definition:** A function $f : \mathbb{R}^n \to \mathbb{R}$ is **convex** if $\{(x, y) \in \mathbb{R}^{n+1} : y \geq f(x)\}$ is a convex region.



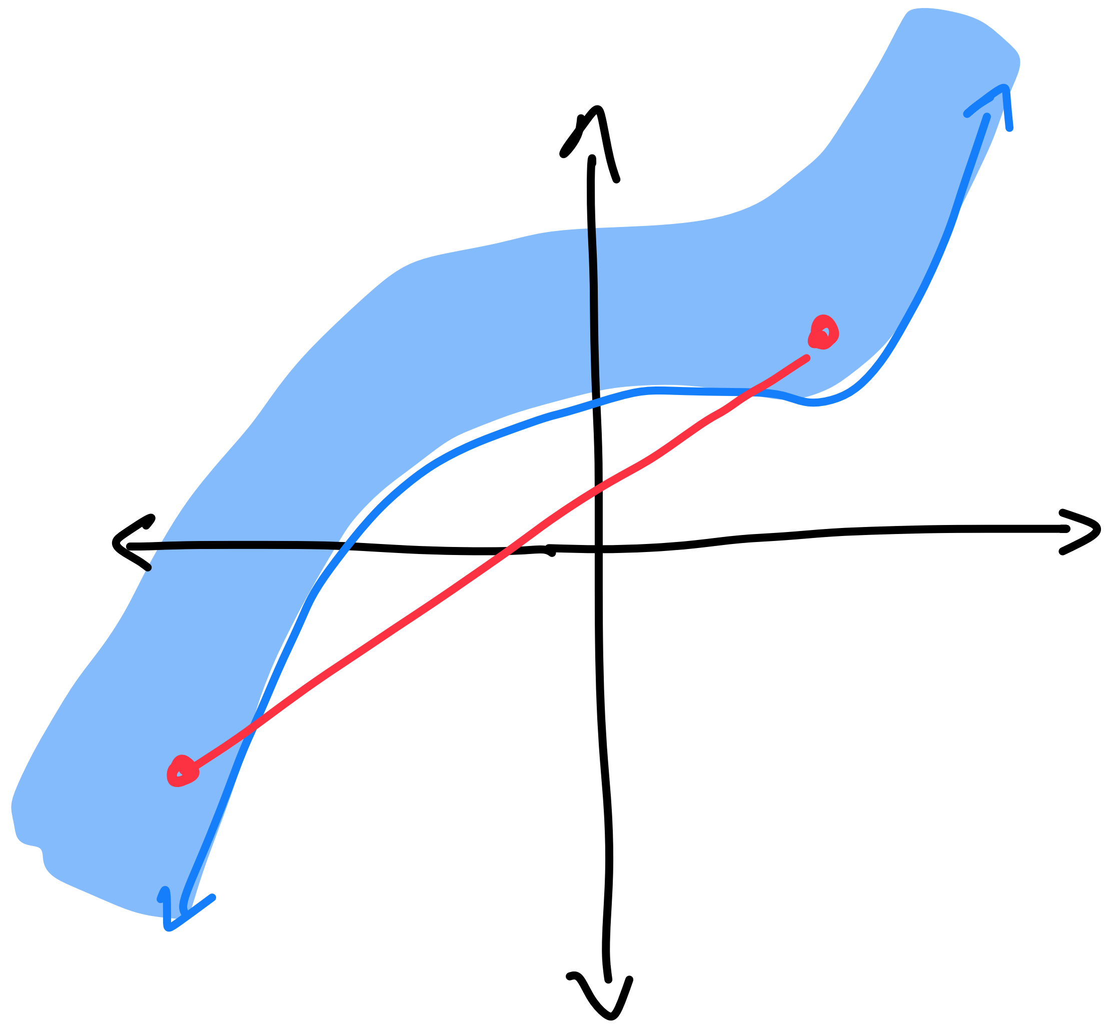
convex region



not convex



convex fn

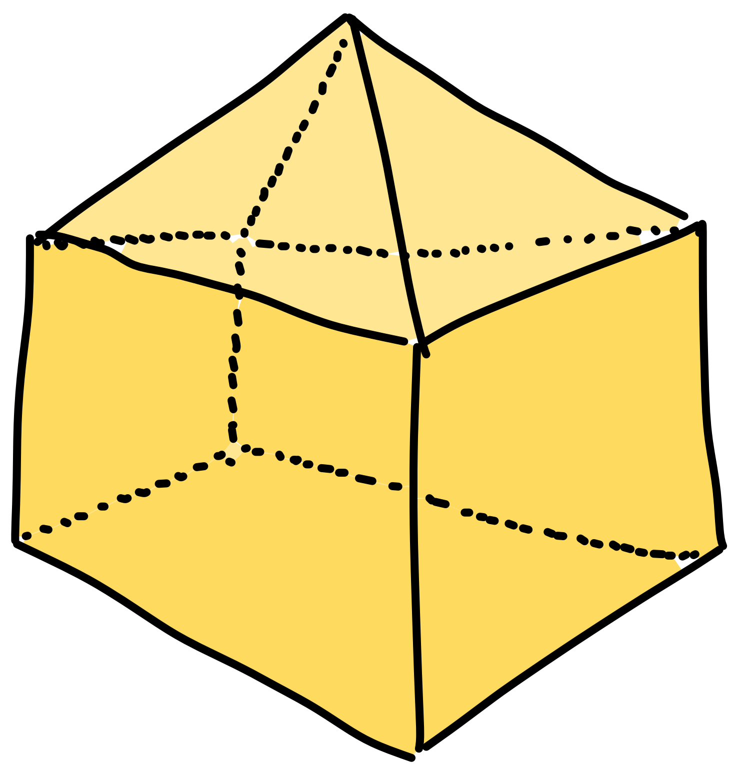


non - convex fn

Convex polytope

Generalization of a convex polygon/polyhedron to n-dimensions

- **Definition:** Given a matrix $A \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$, the set of $x \in \mathbb{R}^n$ such that $Ax \leq b$ is a convex polytope.
- **Definition:** If a polytope is contained in a ball, it said to be *bounded*.
- Example: Given a set of points $y_1, \ldots, y_k \in \mathbb{R}^n$, the convex hull $\mathbf{conv}(y_1, \ldots, y_k)$ is a bounded convex polytope. A convex hull $\mathbf{conv}(y_1, \ldots, y_k)$ is the intersection of all convex sets containing the points $y_1, \ldots, y_k$.
- **Theorem**: Every bounded polytope is the convex hulls of finite set of points.



convex polytope

Optimization problems

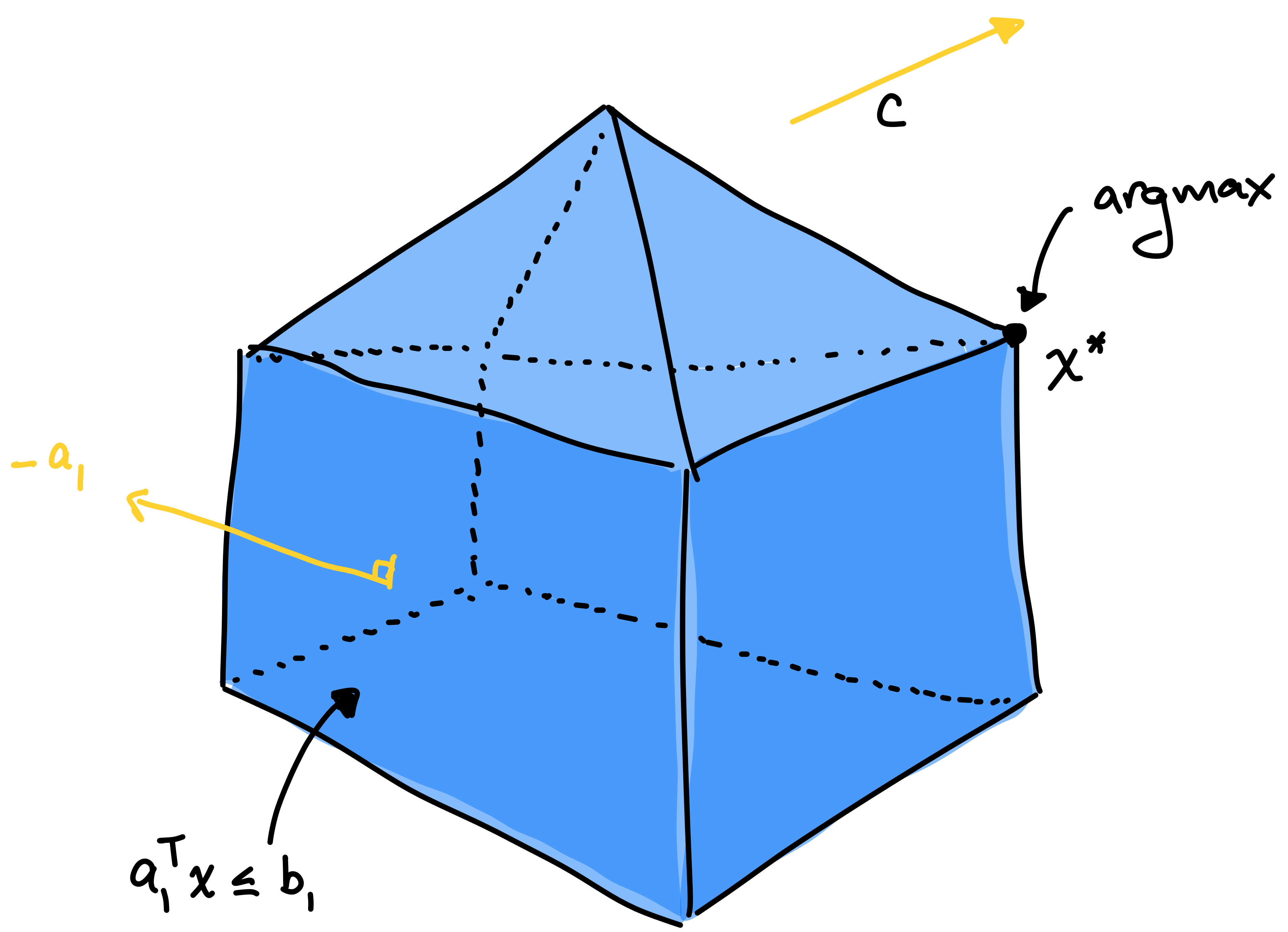
- Optimization problems are most of the problems we have seen
- An optimization problem is described by some function $\underbrace{f : \Sigma \to \mathbb{R}}_{\text{optimization function}}$ and a subset $\Gamma \subseteq \Sigma$.

  feasible region domain

  - Goal is to find $x \in \Gamma$ such that for all $y \in \Gamma, f(x) \geq f(y)$ – i.e. x is the argmax of f with respect to Γ.
  - Ex.: Knapsack. $\Sigma = \{S : S \subseteq [n]\}, \Gamma = \{S : \text{weight}(S) \leq W\}, f(S) = \text{value}(S)$
  - Ex. Shortest path $s \to t$. $\Sigma = \{\text{seq . of edges}\}, \Gamma = \{\text{paths}\}, f(p) = \sum_{e \in p} w(e)$
  - Ex. Greedy. $\Sigma = \{\text{job assignments}\}, \Gamma = \{\text{non} - \text{overlapping}\}, f(x) = \text{value}(x)$

Linear programming

maximizing $c^T x$ subject to $x \in \Gamma$
and Γ being a convex polytope.



Pictorally, the optimal pt will be the point $\in \Gamma$ most in the $\vec{c}$ direction.

Linear programming:

Optimizing a linear fn subject to a convex polytope.

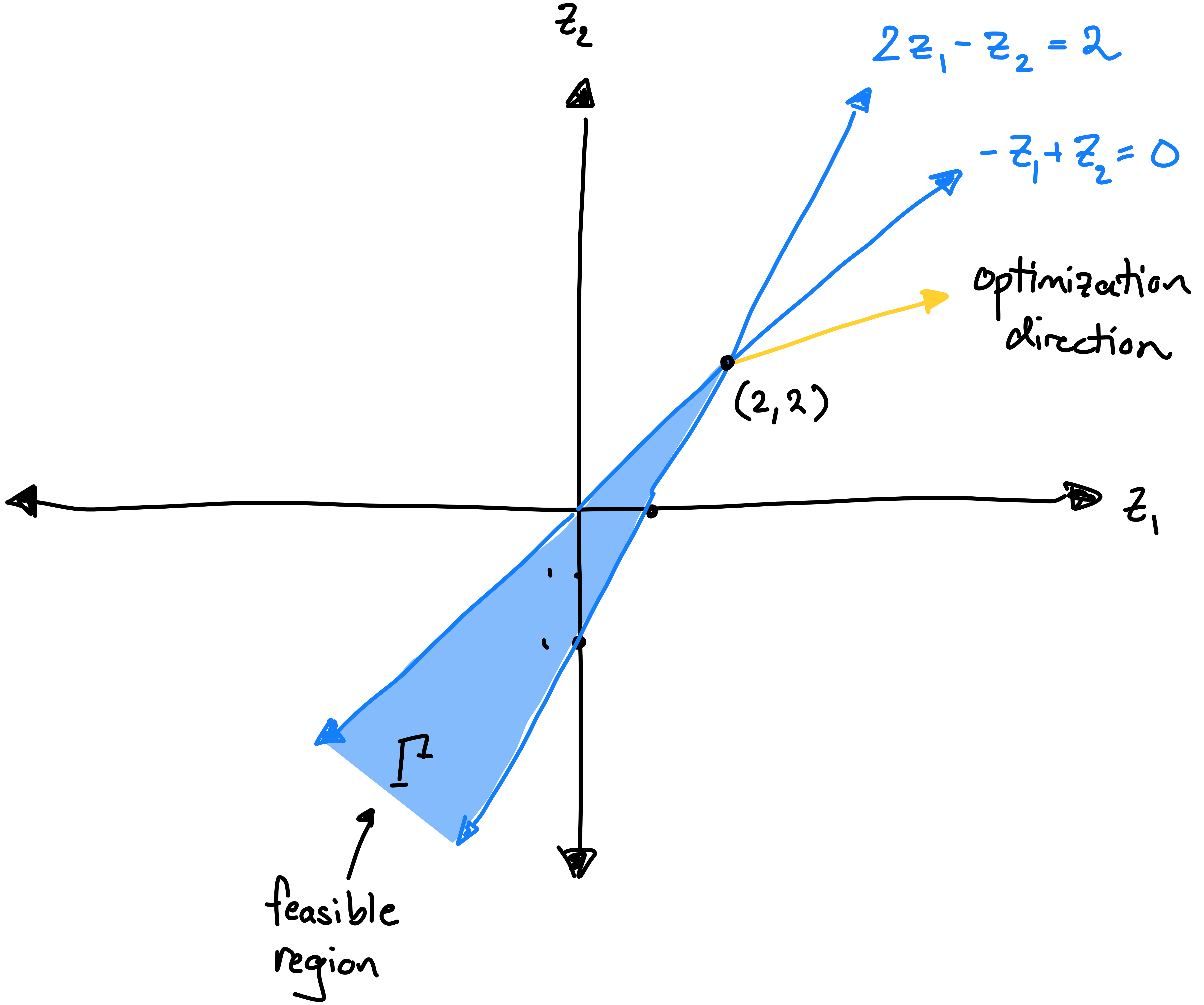
Linear programming example

General linear programming example:

$$\text{maximize} \quad 10 z_1 + z_2$$

$$\text{subject to} \quad 2z_1 - z_2 \leq 2$$

$$-z_1 + z_2 \leq 0$$



Linear programming

- An optimization problem paradigm
- Both the optimization function f and feasible region Γ are linear.

$$\max \underbrace{\begin{bmatrix} - \; c \; - \end{bmatrix}}_{\mathbb{R}^n} \cdot \underbrace{\begin{bmatrix} | \\ x \\ | \end{bmatrix}}_{\mathbb{R}^n} \quad \text{subject to} \quad \underbrace{\begin{bmatrix} \\ A \\ \\ \end{bmatrix}}_{\mathbb{R}^{m \times n}} \cdot \begin{bmatrix} | \\ x \\ | \end{bmatrix} \le \underbrace{\begin{bmatrix} | \\ b \\ | \end{bmatrix}}_{\mathbb{R}^m} \quad \text{and} \quad \begin{bmatrix} | \\ x \\ | \end{bmatrix} \ge 0.$$

Global space $\Sigma = \mathbb{R}^n$, feasible region $\Gamma = \{ x : Ax \le b, x \ge 0 \}$, opt. fn. $f(x) = c^T x$

Linear programming examples

- Some we have seen
  - Max flow / min cut
  - Shortest paths
- Some we have not
  - Zero-sum games
  - Linear regression
  - Approximation algorithms for some NP-complete problems

Advertising campaign

- A political candidate has realized that if she spends \$1 on advertising on policy p she wins or loses voters according to table
    - Ex. If she spends \$1 on train advertising, 8 urban voters will vote for her, 4 suburban voters will switch to opponent, and 2 rural voters will vote for her.
- Also listed is the number of votes she needs from each demographic
- What is the minimum amount she needs to spend to win?

	Urban	Suburban	Rural
Number of votes needed	**50**	**100**	**82**
Build roads	-2	5	3
Build trains	8	-4	2
Farms	0	0	10
Gas Tax	10	0	-2

Advertising campaign

$$\min \quad x_r + x_t + x_f + x_g$$

$$\text{s.t.} \quad -2x_r + 8x_t + 0x_f + 10x_g \geq 50$$

$$5x_r - 4x_t + 0x_f + 0x_g \geq 100$$

$$3x_r + 2x_t + 10x_f - 2x_g \geq 82$$

$$x_r, x_t, x_f, x_g \geq 0.$$

	Urban	Suburban	Rural
Number of votes needed	**50**	**100**	**82**
Build roads	-2	5	3
Build trains	8	-4	2
Farms	0	0	10
Gas Tax	10	0	-2

Advertising campaign

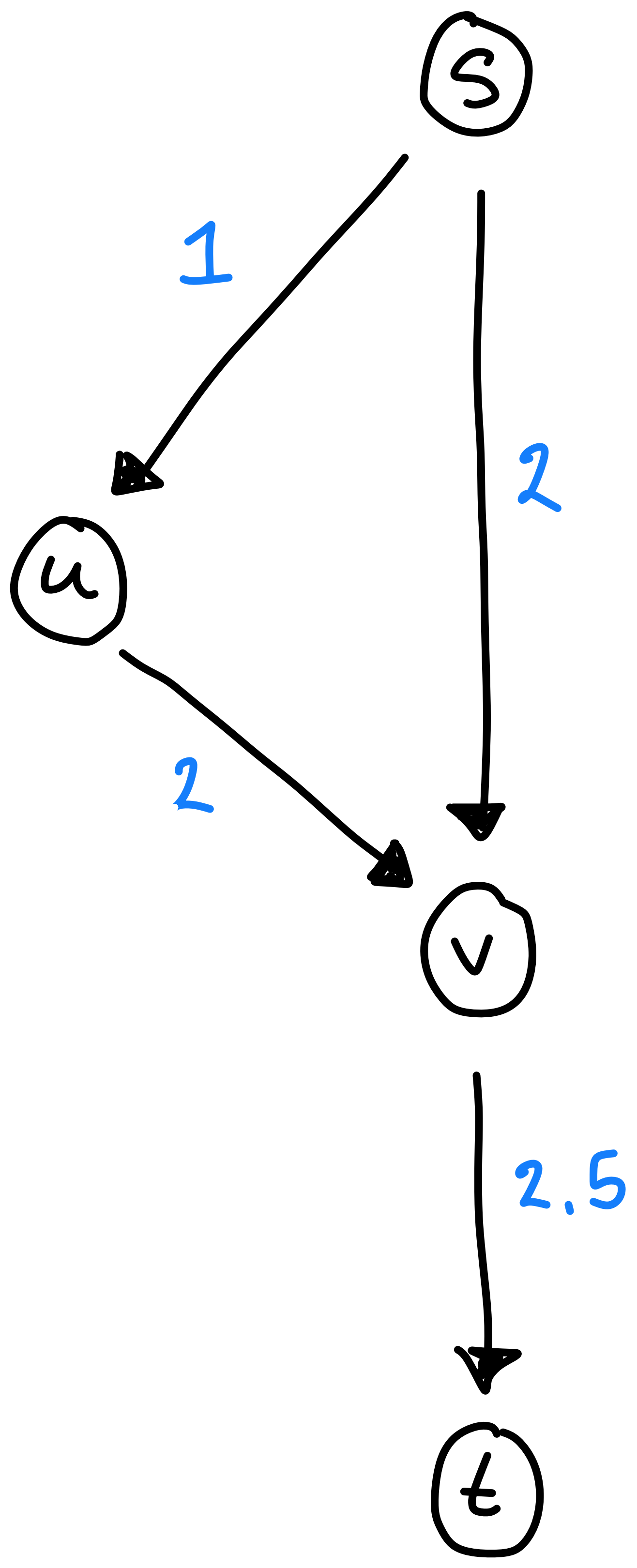
$$\min \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_r \\ x_t \\ x_f \\ x_g \end{pmatrix}$$

$$\text{s.t.} \begin{pmatrix} -2 & 8 & 0 & 10 \\ 5 & -4 & 0 & 0 \\ 3 & 2 & 10 & -2 \end{pmatrix} \begin{pmatrix} x_r \\ x_t \\ x_f \\ x_g \end{pmatrix} \geq \begin{pmatrix} 50 \\ 100 \\ 82 \end{pmatrix}$$

$$x \geq 0$$

	Urban	Suburban	Rural
Number of votes needed	**50**	**100**	**82**
Build roads	-2	5	3
Build trains	8	-4	2
Farms	0	0	10
Gas Tax	10	0	-2

Max flow as a linear program



max flow is equivalent to

$$\max \quad f_{su} + f_{sv}$$

s.t.

capacity constraints

$$0 \le f_{su} \le 1$$
$$0 \le f_{sv} \le 2$$
$$0 \le f_{uv} \le 2$$
$$0 \le f_{vt} \le 2.5$$

conservation of flow

$$f_{su} - f_{uv} = 0$$
$$f_{sv} + f_{uv} - f_{vt} = 0$$

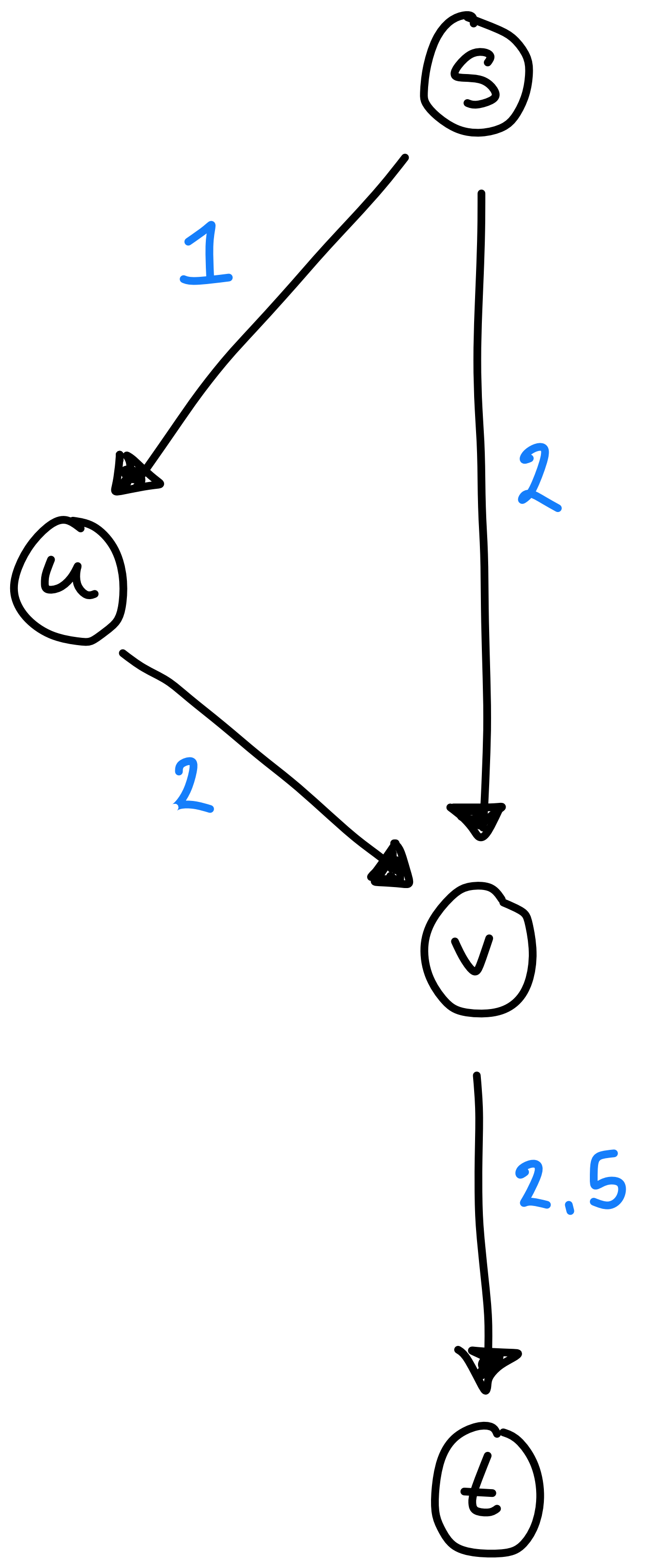
$$\begin{cases} f_{su} - f_{uv} \le 0 \\ -f_{su} + f_{uv} \le 0 \end{cases}$$

$$\begin{cases} f_{sv} + f_{uv} - f_{vt} \le 0, \\ -f_{sv} - f_{uv} + f_{vt} \le 0 \end{cases}$$

This is a linear program.

We can also express it in standard form (next slide).

Max flow as a linear program



Let $f = \begin{bmatrix} f_{su} \\ f_{sv} \\ f_{uv} \\ f_{vt} \end{bmatrix}$. Then, max flow is equivalent to:

$$\max \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \cdot f$$

$$\text{s.t.} \quad \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \\ \hline 1 & 0 & -1 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 1 & -1 \\ 0 & -1 & -1 & 1 \end{bmatrix} \cdot f \leq \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2.5 \\ \hline 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$f \geq 0.$$

Max flow as a linear program (generalized)

- Let (G, c, s, t) be a flow network. Then the max flow $f \in \mathbb{R}^E$ is the vector optimizing the following LP:
    - Let $g = \mathbf{1}_{\{e \text{ out of } s\}}$
    - For each vertex $v \in V \backslash \{s, t\}$, let $h_v = +\mathbf{1}_{\{e \text{ out of } v\}} - \mathbf{1}_{\{e \text{ into } v\}}$.

max flow equals

$$\max \quad g^T f$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbb{I}_E \\ \hline h_{v_1} \\ -h_{v_1} \\ \vdots \\ h_{v_n} \\ -h_{v_n} \end{bmatrix} \cdot f \leq \begin{bmatrix} c \\ \hline 0 \end{bmatrix}, \quad \begin{array}{l} \} \text{ capacity constraints} \\ \} \text{ conservation of flow} \end{array}$$

$$f \geq 0.$$

Max flow as a linear program (generalized)

- Max flow on a graph with $|V| = n, |E| = m$ is equivalent to a linear program over m variables and $m + 2(n-2) = O(m+n)$ constraints
- This is a reduction!
- Therefore, fast algorithms for solving linear programs imply fast algorithms for max flow.
- We will see an algorithm for LPs soon!

Linear programming standard form

Standard Form:

$$\max \quad c^T x$$

$$\text{s.t.} \quad \begin{cases} Ax \le b \\ x \ge 0 \end{cases}$$

Claim: Any optimization over convex polytope Γ is equiv. to an optimization over a LP of standard form

Proof: $\left\{ \begin{array}{ll} \max & c^T z \\ \text{s.t.} & Az \le b \end{array} \right\} = \left\{ \begin{array}{ll} \max & c_1 z_1 + c_2 z_2 + \dots + c z_n \\ \text{s.t.} & a_1^T z \le b_1, \dots, a_m^T z \le b_m \end{array} \right\}$

Replace z_i with $x_i^{(+)} - x_i^{(-)}$ with $x_i^{(+)}, x_i^{(-)} \ge 0$. $\underbrace{z = x^{(+)} - x^{(-)}}_{\text{vector eq.}}$

$$\left\{ \begin{array}{ll} \max & c_1(x_1^{(+)} - x_1^{(-)}) + \dots + c_n(x_n^{(+)} - x_n^{(-)}) \\ \text{s.t.} & a_1^T(x^{(+)} - x^{(-)}) \le b_1, \dots, a_m^T(x^{(+)} - x^{(-)}) \le b_m \\ & x^{(+)} \ge 0, \ x^{(-)} \ge 0 \end{array} \right\}$$

The value of expressing problems as LPs

- Due to the prevalence of LPs, many optimizations are known
- We know LPs can be solved in polynomial time
    - Makes writing down a problem as an LP a good first step
- Writing a problem as a linear program, can make a solution apparent
- Arguing correctness of an LP can be easier
- Applying duality (next!) can give a different perspective on the problem

Minimization linear programs

$$\left\{ \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \le b \\ & x \ge 0 \end{array} \right\} \quad \text{is equivalent to} \quad - \left\{ \begin{array}{ll} \max & (-c)^T x \\ \text{s.t.} & Ax \le b \\ & x \ge 0 \end{array} \right\}$$

Shortest paths as an LP

- **Input**: Directed graph $G = (V, E)$ and vertices s, t
- **Output**: (Length) of shortest path $s \rightsquigarrow t$
- **Claim**: The length of the shortest path is the solution to the following “flow-like” LP.
- **Proof (sketch)**:
- $(\Rightarrow)$:A path of length ℓ corresponds to a valid flow.
- $(\Leftarrow)$: A flow is the sum of $\leq m$ flows along paths. Since total flow is 1, the flow can be thought of as a probability distribution over paths. So, the LP’s feasible solution is an expectation over paths.

minimize $\mathbf{1}^T \cdot f$

s.t. $\sum_{e \text{ out of } s} f(e) = 1,$

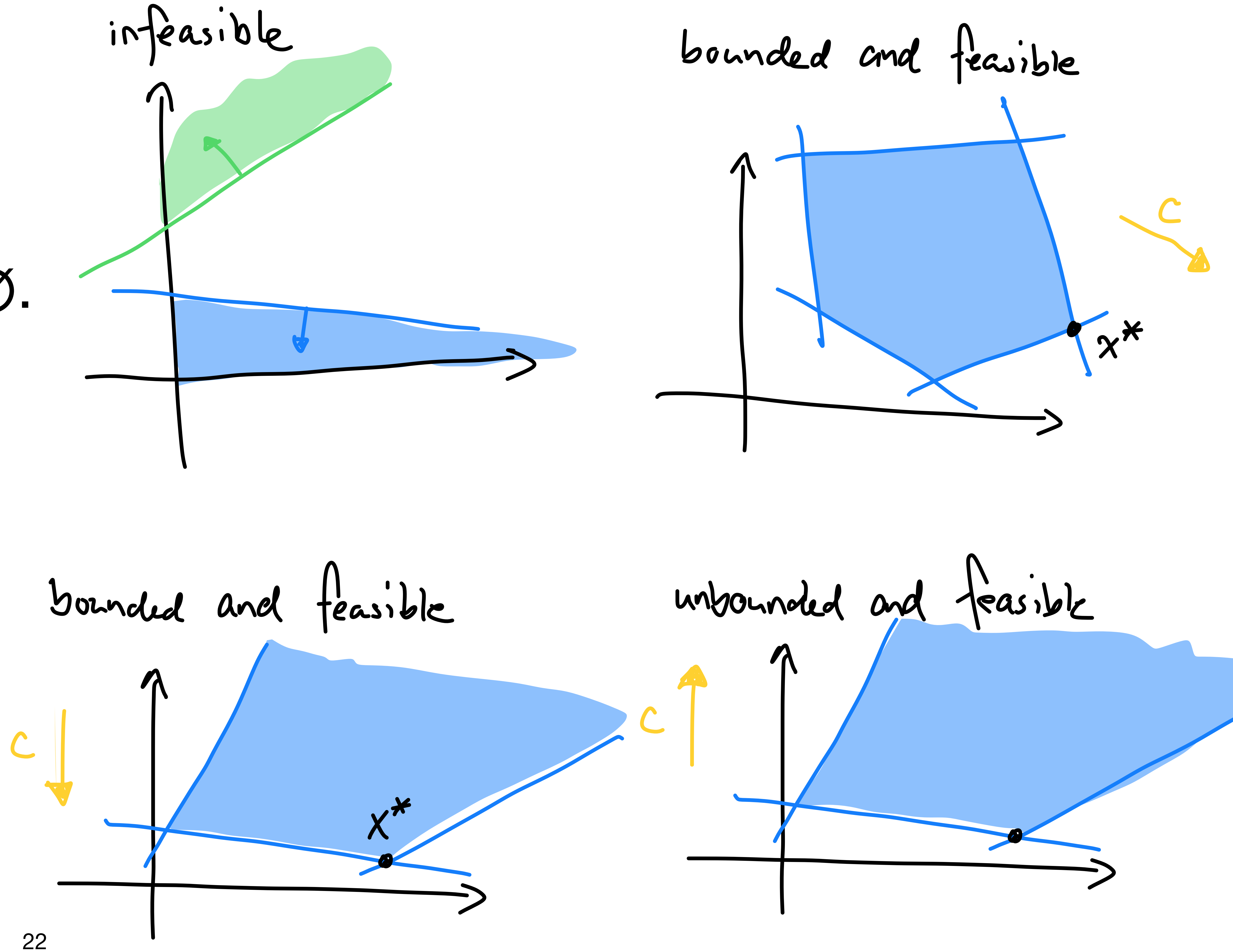
$\sum_{e \text{ into } t} f(e) = 1,$

$\forall v \in V \setminus \{s,t\}, \sum_{e \text{ out of } v} f(e) = \sum_{e \text{ into } v} f(e),$

$f \geq 0$

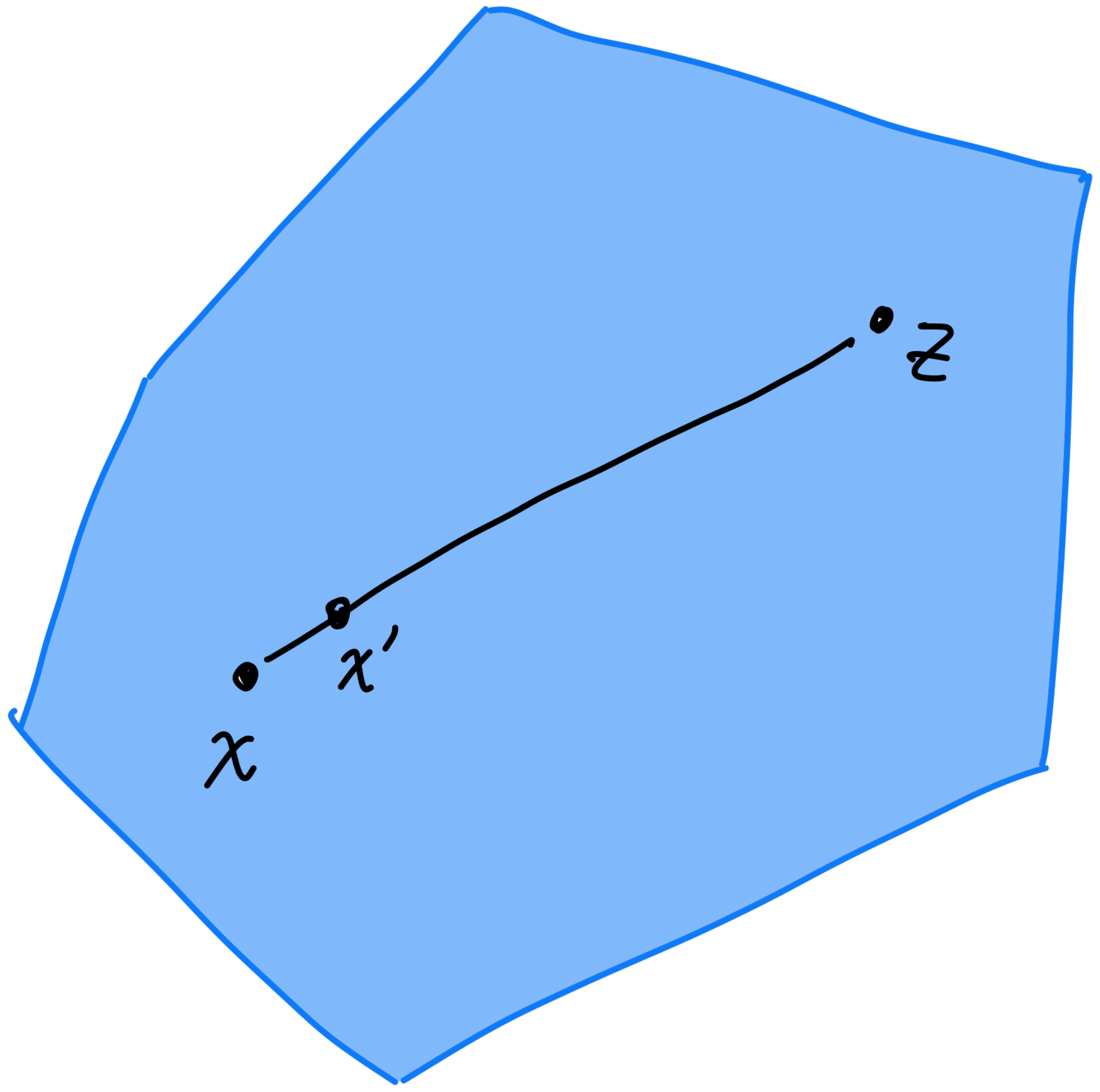
Linear programming feasibility

- Recall, the feasible region of a standard LP is $\Gamma = \{x : Ax \leq b, x \geq 0\}$.
- **Definition**: The LP is *infeasible* if $\Gamma = \emptyset$.
- **Definition**: The LP is *unbounded* if $c^\top x$ can be arbitrarily large for some $x \in \Gamma$.
- Even just deciding if a LP is feasible or not, seems like a challenging problem.



Where are the optimums of LPs

- **Theorem:** If a local optimum exists for an LP, it is a global optimum.
- **Proof:** Recall we are maximizing $c^\top x$ subject to $x \in \Gamma$ and Γ is convex. Assume x is a local optimum but not a global optimum.
  - Then $c^\top x < c^\top z$ for some $z \in \Gamma$ as x is not a global optimum.
  - Consider the line $\overline{xz} \in \Gamma$. Then $x' := x + \epsilon(z - x) \in \Gamma$ for small $\epsilon > 0$ and
  - $c^\top x' = c^\top x + \epsilon c^\top (z - x) > c^\top x$.
  - So x is not a local optimum.
  - This proves the contrapositive.

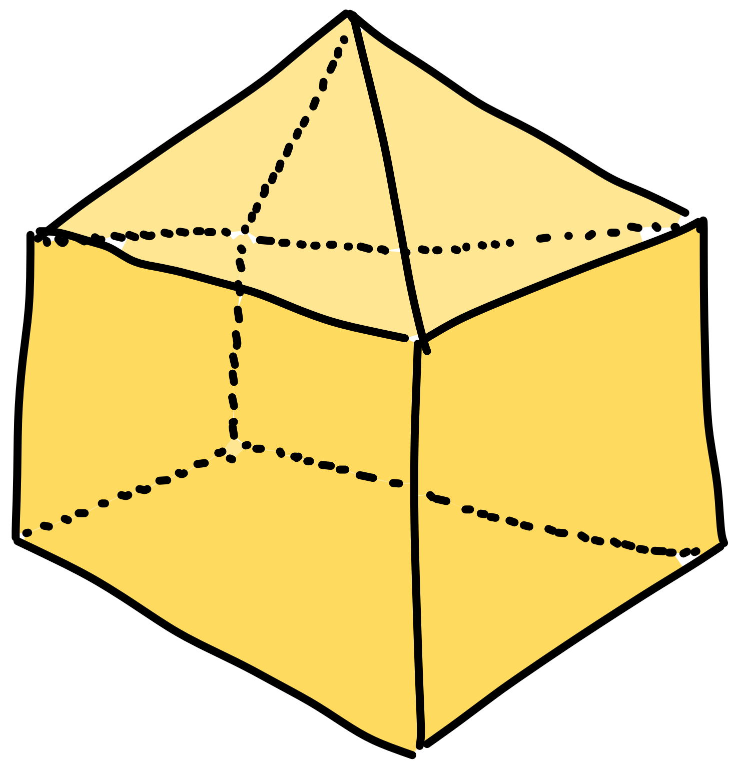


Bounded Convex polytopes

- **Definition:** A **vertex** z of a convex polytope Γ is any point such that z is not the midpoint of any line segment $\overline{xy} \in \Gamma$ for $x \neq y$.

- **Remark:** If $v_1, \ldots, v_k$ are **all** the vertices of a bounded convex polytope Γ, then $\Gamma = \mathbf{conv}(v_1, \ldots, v_k)$, the convex hull of the vertices.

- **Theorem:** If the optimum is finite for a standard linear program whose feasible region, then the optimum must be achieved at some vertex.



Bounded Convex polytope

- **Theorem:** If the optimum is finite for a standard linear program whose feasible region, then the optimum must be achieved at some vertex.

- **Proof:** Let $v_1, \ldots, v_k$ be the vertices of the feasible region Γ.
  - Then every point $x \in \Gamma$ equals $\sum_{i=1}^{k} \lambda_i v_i$ for $\lambda \geq 0$ and $\sum_{i=1}^{k} \lambda_i = 1$.
  - By linearity of objective function,
    - $c^\top x = \sum_{i=1}^{k} \lambda_i c^\top v_i \leq \max_{i=1}^{k} c^\top v_i$
    - So one of the vertices must do better than the vertex x.