

ORF307 – Optimization

11. The simplex method

Ed Forum

- For the 1-norm optimal control problem I don't quite understand the significance of adding the infinity-norm (max input) and the 1 norm (max-input variation) as the extra inequality constraints and how that translates to the optimal state trajectory and optimal input trajectory graphs.

Recap

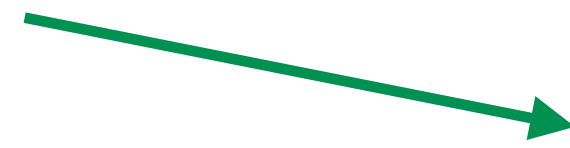
Vehicle example with output tracking

1-norm with constraints

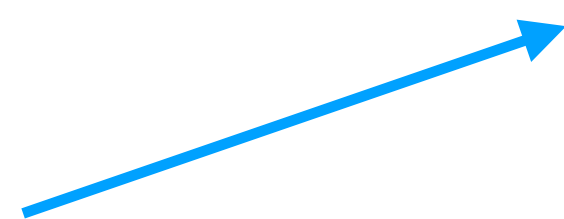
Linear optimization can have more interesting constraints

$$\begin{aligned} &\text{minimize} && \sum_{t=1}^T \|y_t - y_t^{\text{des}}\|_1 + \rho \sum_{t=1}^{T-1} \|u_t\|_1 \\ &\text{subject to} && x_{t+1} = Ax_t + Bu_t, \quad t = 1, \dots, T-1 \\ & && y_t = Cx_t, \quad t = 1, \dots, T \\ & && \|u_t\|_\infty \leq u^{\text{max}}, \quad t = 1, \dots, T-1 \\ & && \|u_t - u_{t-1}\|_1 \leq s^{\text{max}}, \quad t = 1, \dots, T-1 \\ & && x_1 = x^{\text{init}} \end{aligned}$$

max-input



max-input variation

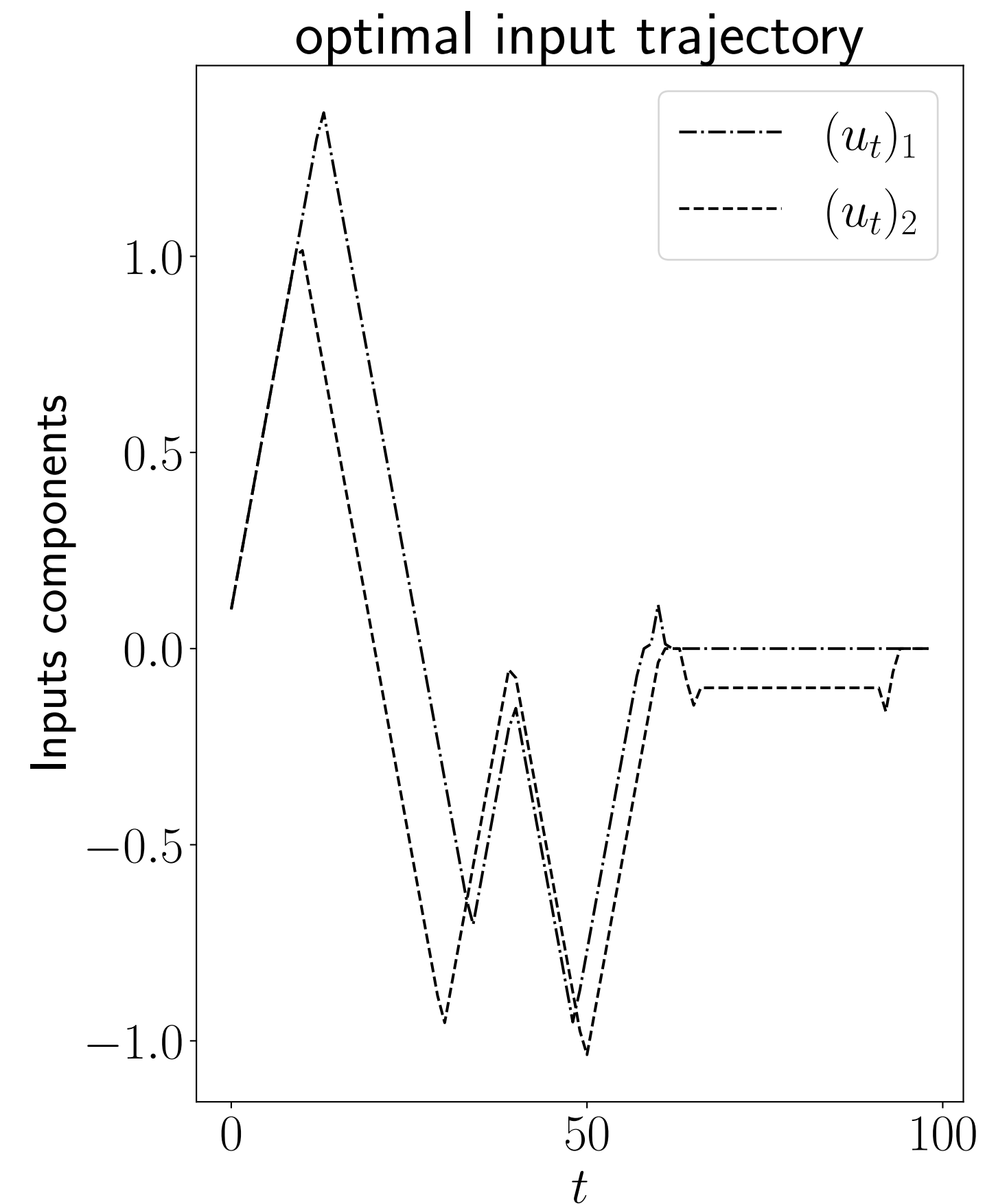
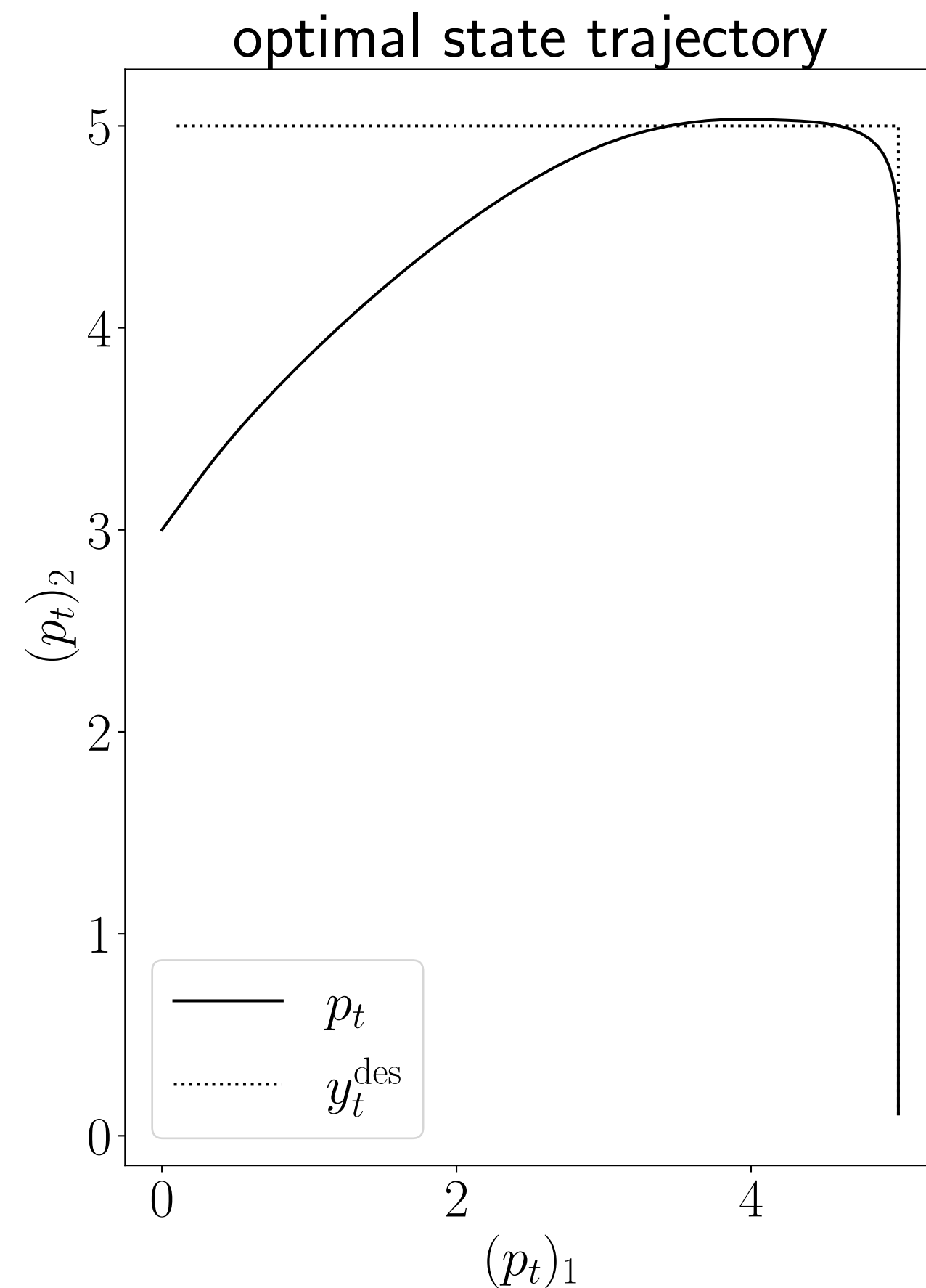


Vehicle example with output tracking

1-norm with constraints results

Parameters

$$u^{\max} = 10, \quad s^{\max} = 0.1$$



Constructing a basic solution

Two equalities ($m = 2, n = 3$)

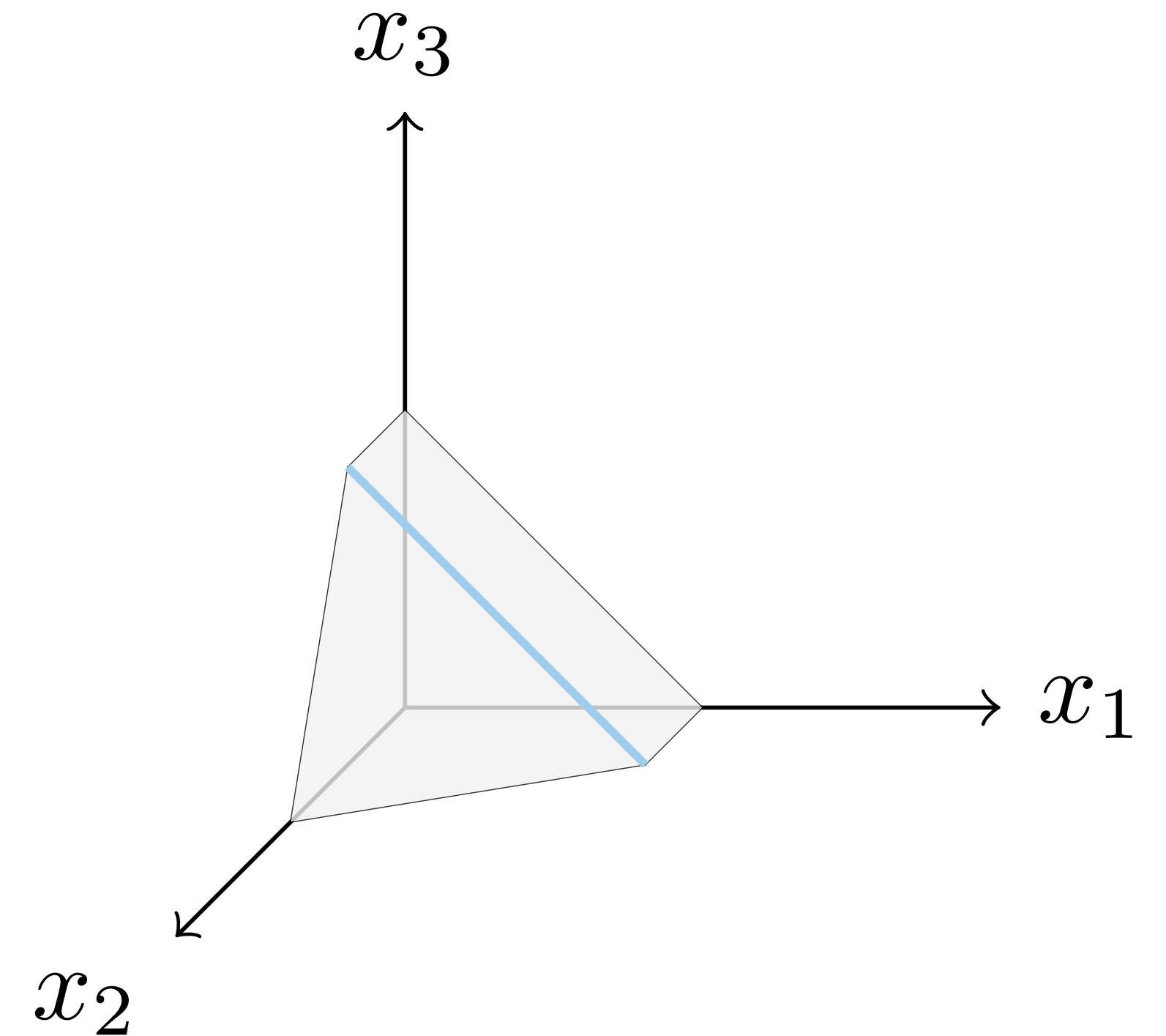
minimize $c^T x$

subject to $x_1 + x_3 = 1$

$(1/2)x_1 + x_2 + (1/2)x_3 = 1$

$x_1, x_2, x_3 \geq 0$

$n - m = 1$ inequalities have to be tight: $x_i = 0$



Set $x_1 = 0$ and solve

$$\begin{bmatrix} 1 & 0 & 1 \\ 1/2 & 1 & 1/2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 1 \\ 1 & 1/2 \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \longrightarrow (x_2, x_3) = (0.5, 1)$$

Standard form polyhedra

Standard form LP

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

Assumption

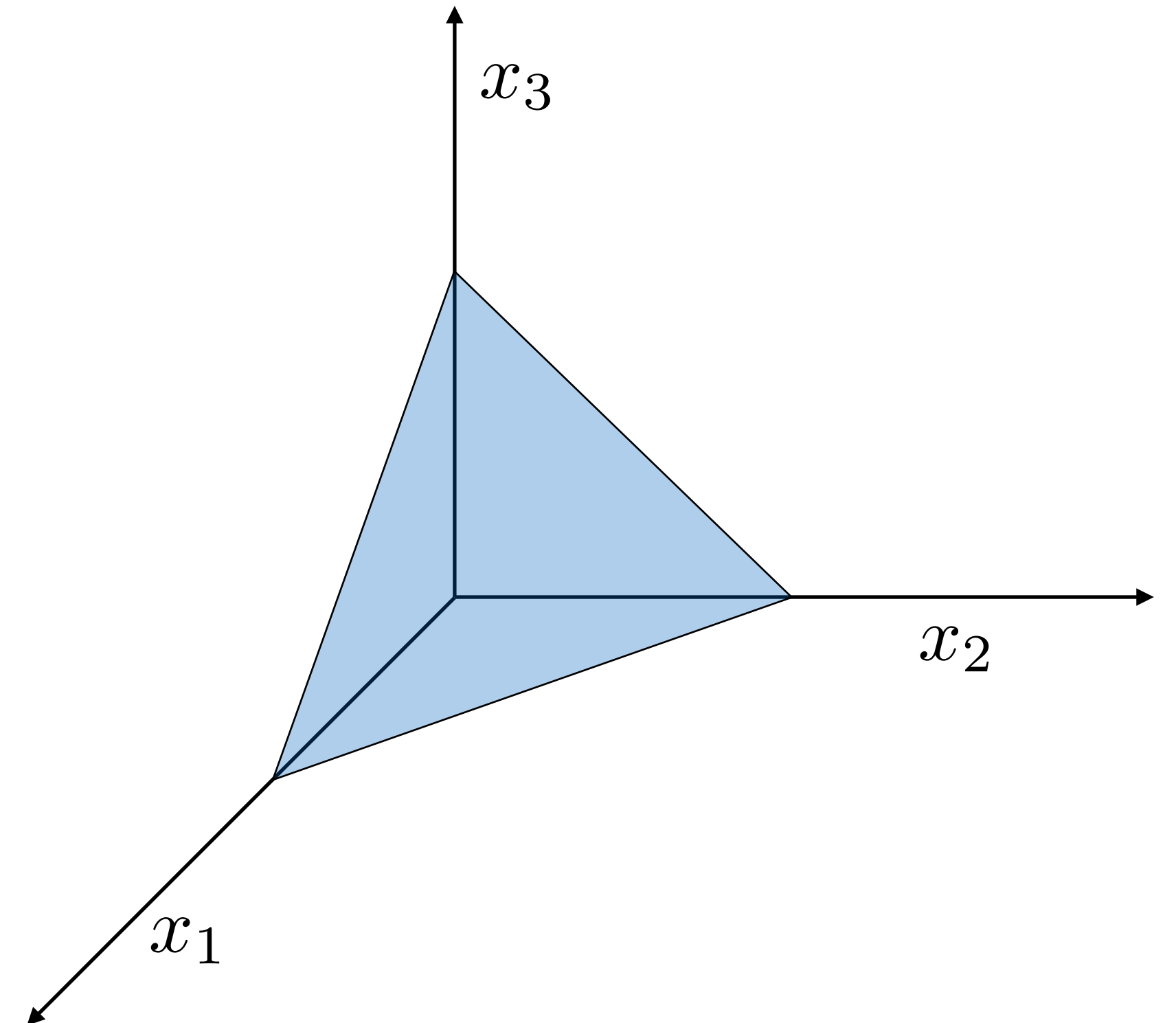
$A \in \mathbf{R}^{m \times n}$ has full row rank $m \leq n$

Interpretation

P is an $(n - m)$ -dimensional surface

Standard form polyhedron

$$P = \{x \mid Ax = b, x \geq 0\}$$



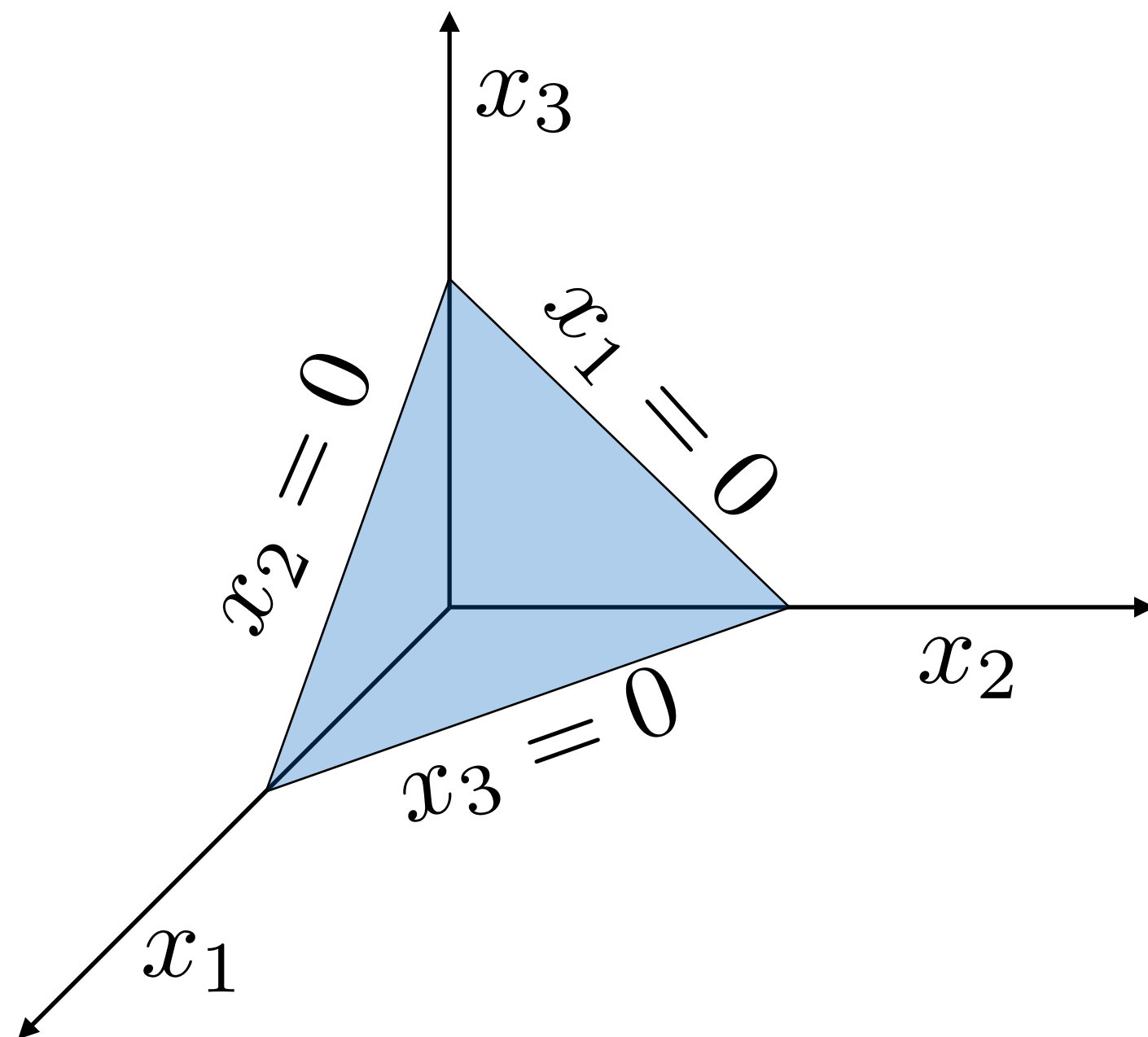
$$n = 3, m = 1$$

Standard form polyhedra

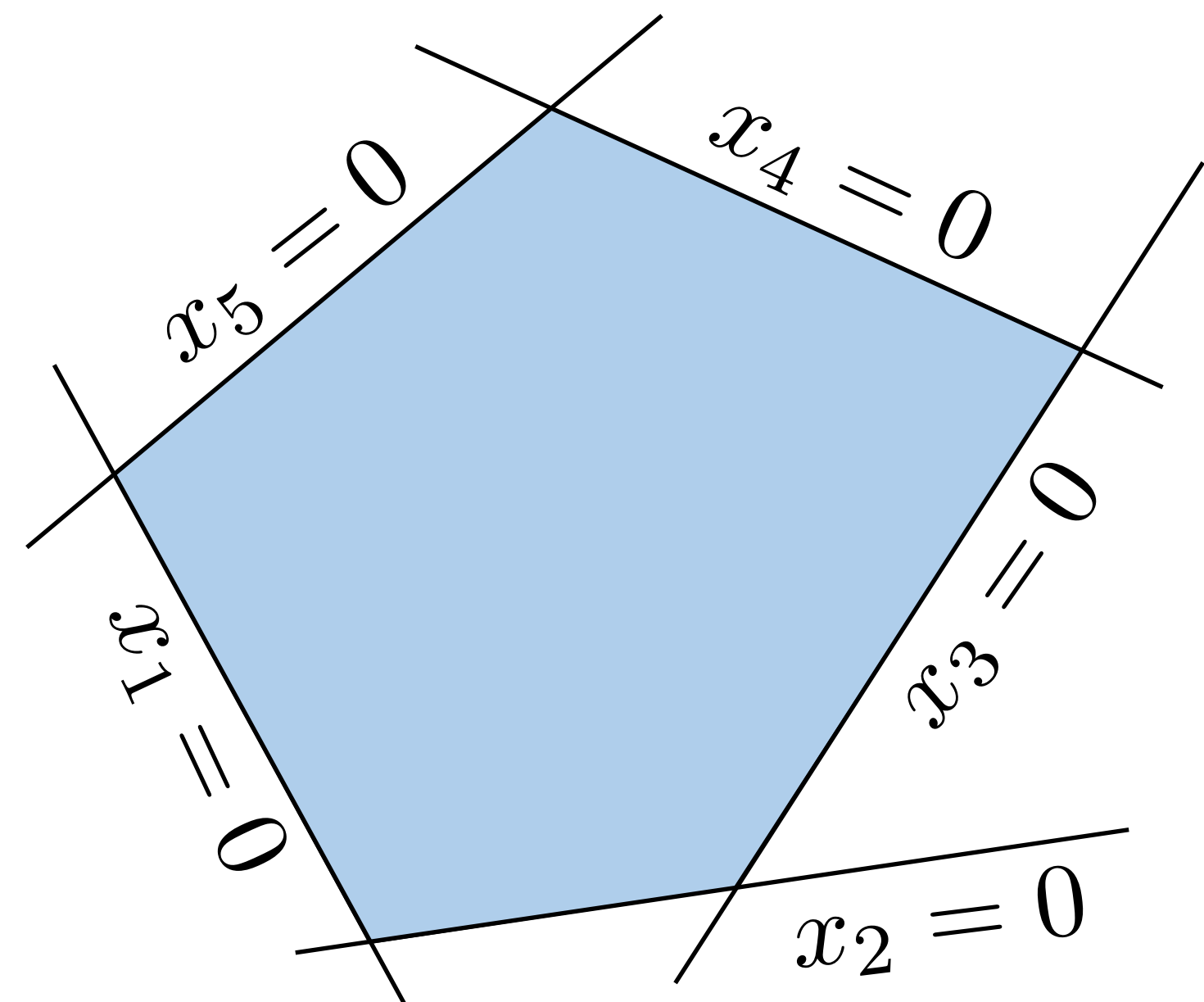
Visualization

$$P = \{x \mid Ax = b, x \geq 0\}, \quad n - m = 2$$

Three dimensions

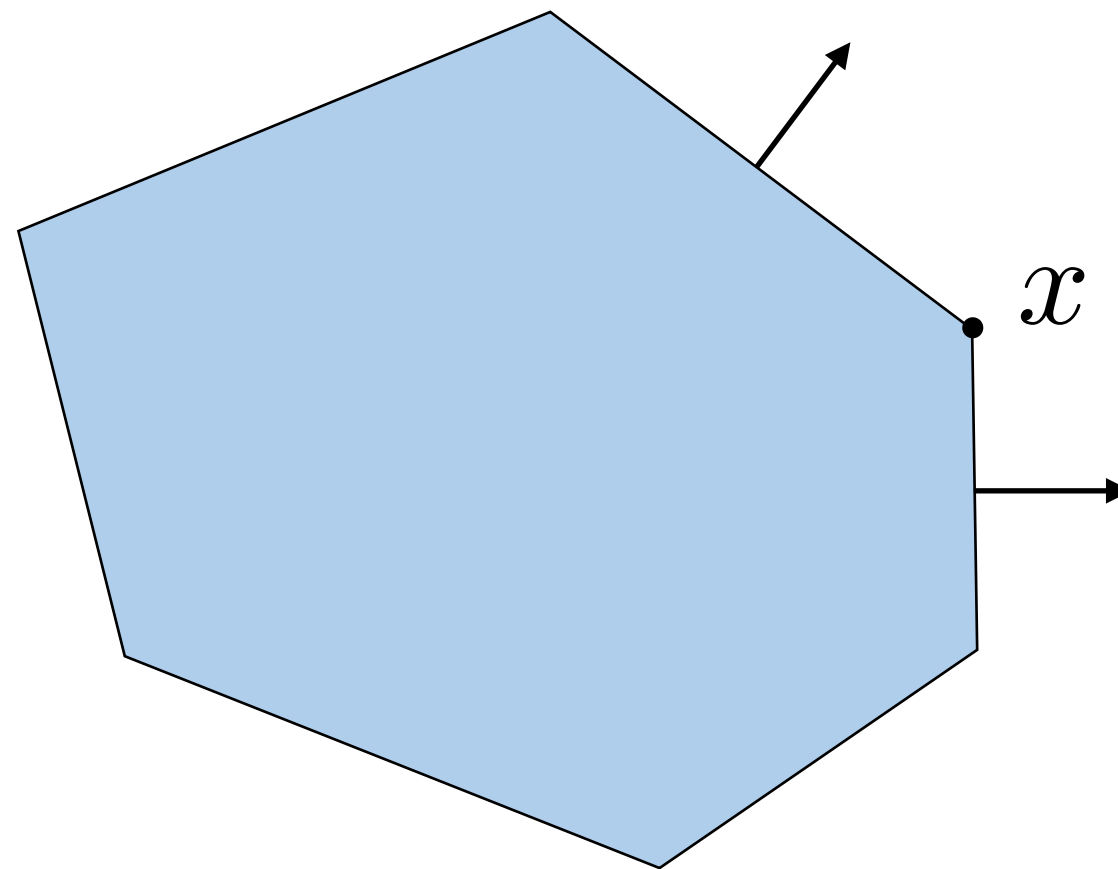


Higher dimensions



Equivalence Theorem

Given a nonempty polyhedron $P = \{x \mid Ax \leq b\}$



Let $x \in P$

x is a **vertex** $\iff x$ is an **extreme point** $\iff x$ is a **basic feasible solution**

Constructing basic solution

1. Choose any m independent columns of A : $A_{B(1)}, \dots, A_{B(m)}$
2. Let $x_i = 0$ for all $i \neq B(1), \dots, B(m)$
3. Solve $Ax = b$ for the remaining $x_{B(1)}, \dots, x_{B(m)}$

Basis
matrix

Basis columns

Basic variables

$$A_B = \begin{bmatrix} | & | & & | \\ A_{B(1)} & A_{B(2)} & \dots & A_{B(m)} \\ | & | & & | \end{bmatrix}, \quad x_B = \begin{bmatrix} x_{B(1)} \\ \vdots \\ x_{B(m)} \end{bmatrix} \longrightarrow \text{Solve } A_B x_B = b$$

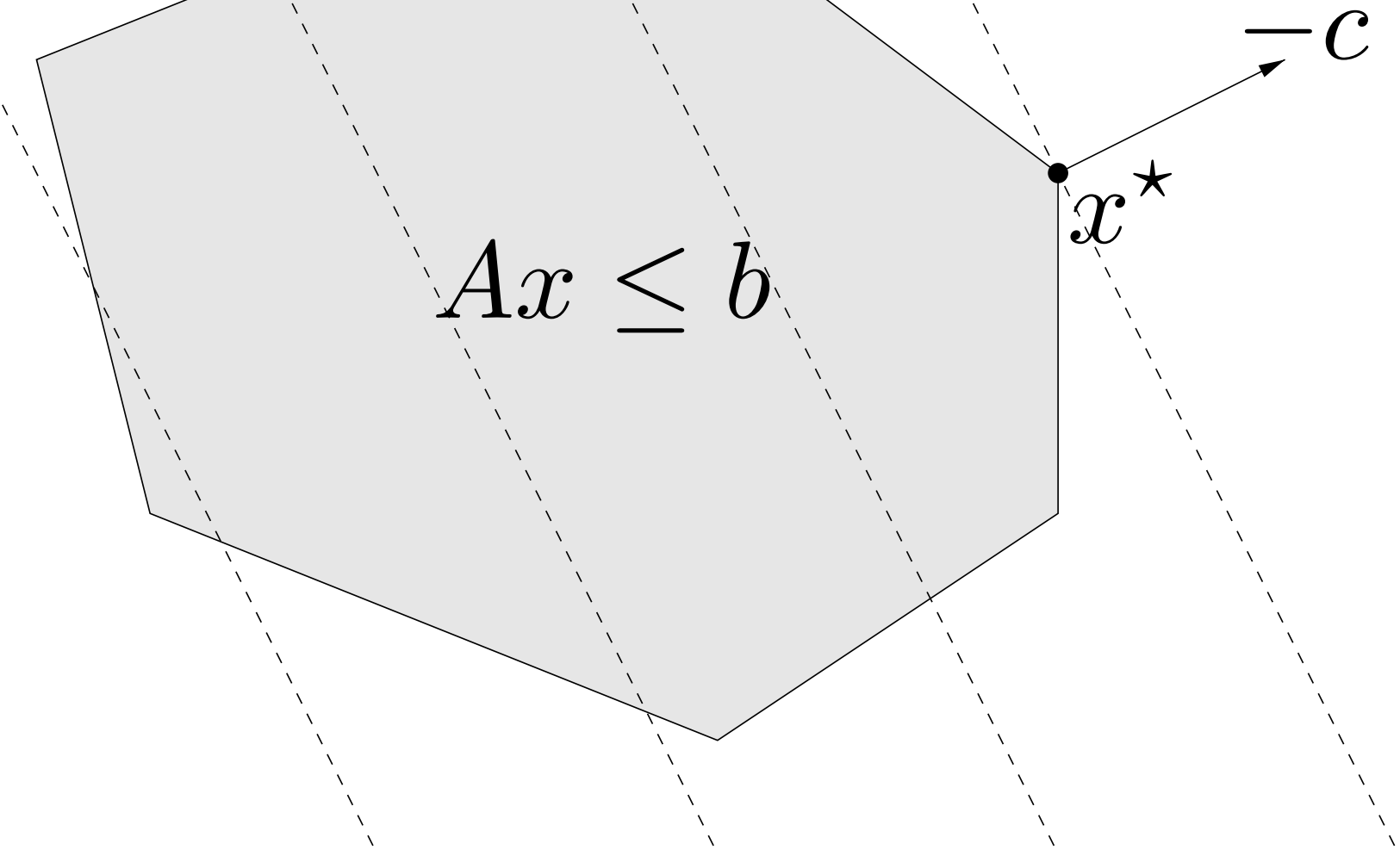
If $x_B \geq 0$, then x is a **basic feasible solution**

Optimality of extreme points

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

- If
- P has at least one extreme point
 - There exists an optimal solution x^*

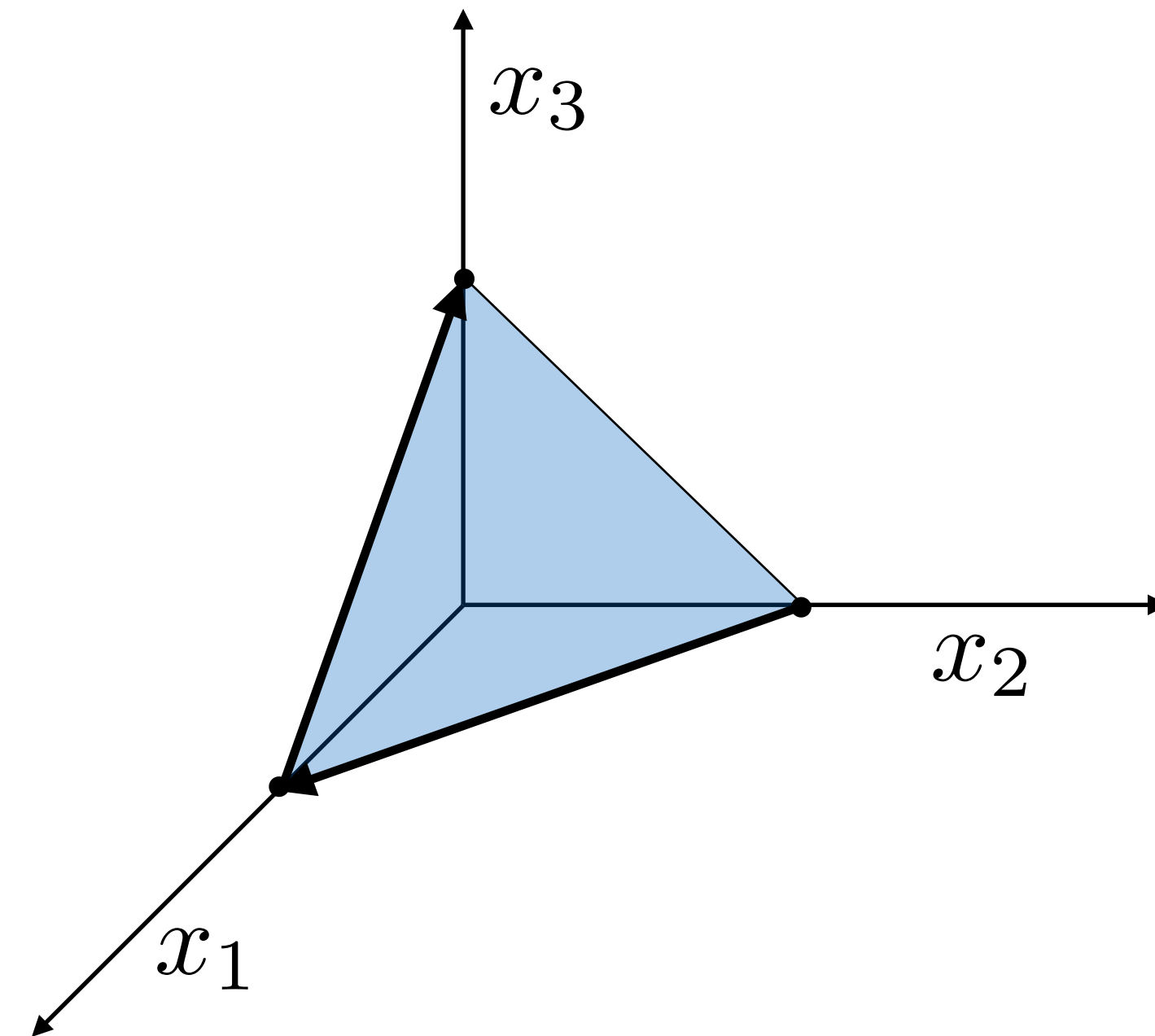
Then, there exists an optimal solution which is an **extreme point** of P



We only need to search between **extreme points**

Conceptual algorithm

- Start at corner
- Visit neighboring corner that improves the objective



Today's agenda

The simplex method

- Iterate between neighboring basic solutions
- Optimality conditions
- Simplex iterations

The simplex method

Top 10 algorithms of the 20th century

1946: Metropolis algorithm

1947: Simplex method →

1950: Krylov subspace method

1951: The decompositional approach to matrix computations

1957: The Fortran optimizing compiler

1959: QR algorithm

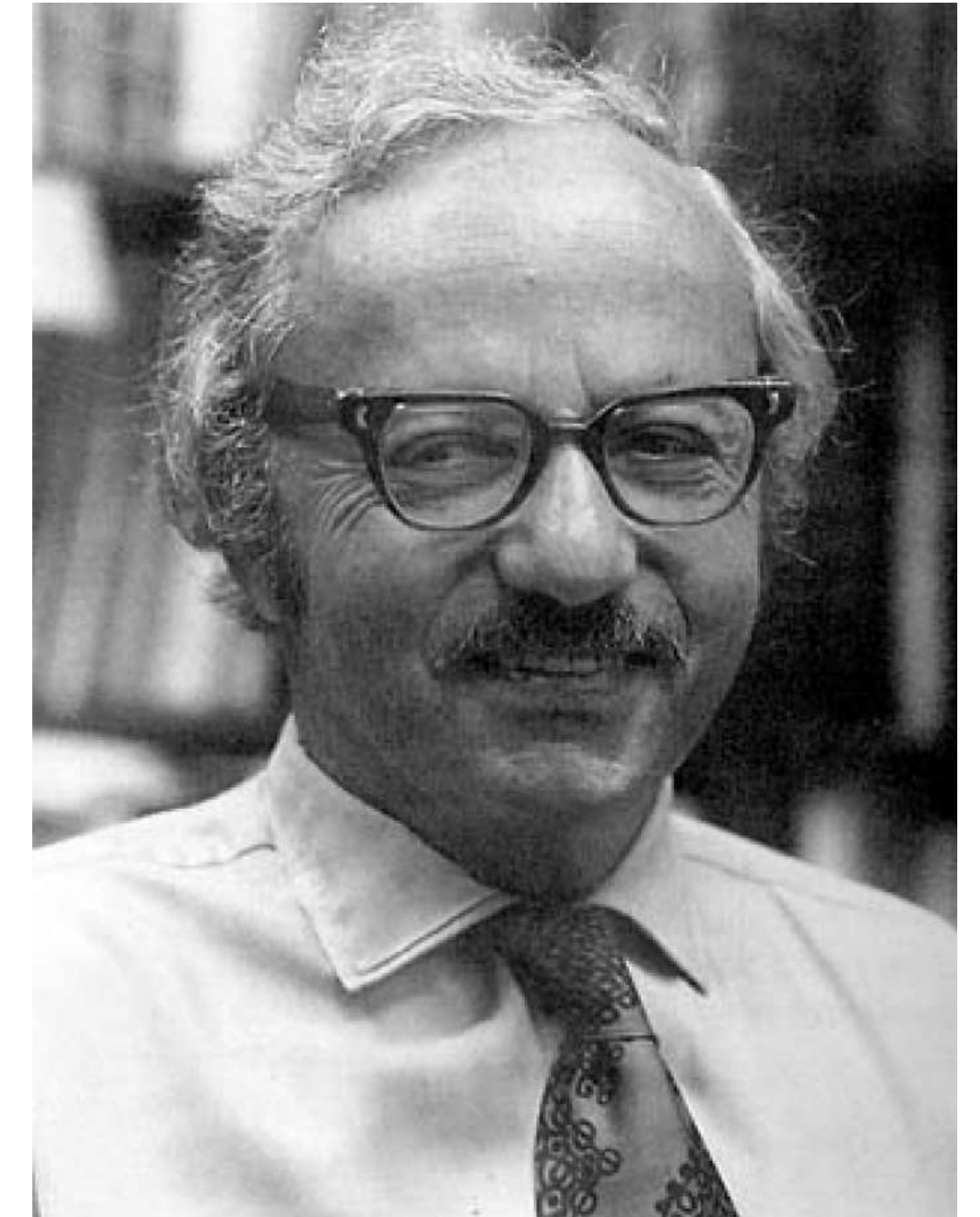
1962: Quicksort

1965: Fast Fourier transform

1977: Integer relation detection

1987: Fast multipole method

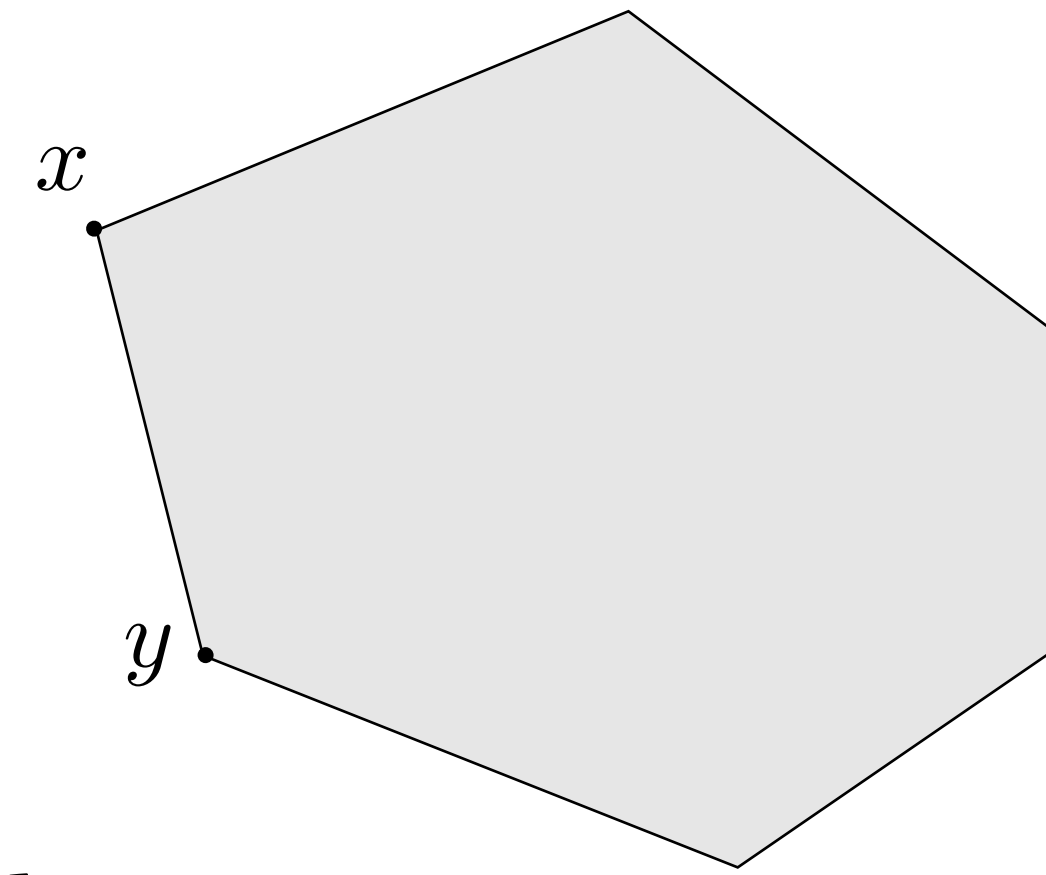
George Dantzig



Neighboring basic solutions

Neighboring solutions

Two basic solutions are **neighboring** if their basic indices differ by exactly one variable



Example

$$A \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = b$$

$$\begin{bmatrix} 1 & -1 & 0 & 3 & -2 \\ 2 & 0 & -1 & -1 & 0 \\ 0 & 2 & 4 & -1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} -5 \\ -1 \\ 14 \end{bmatrix}$$

$$B = \{1, 3, 5\} \quad x_2 = x_4 = 0$$

$$A_B x_B = b \longrightarrow x_B = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2.5 \end{bmatrix}$$

$$\bar{B} = \{1, 3, 4\} \quad y_2 = y_5 = 0$$

$$A_{\bar{B}} y_{\bar{B}} = b \longrightarrow y_{\bar{B}} = \begin{bmatrix} y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 3.0 \\ -1.7 \end{bmatrix}^{16}$$

Feasible directions

Conditions

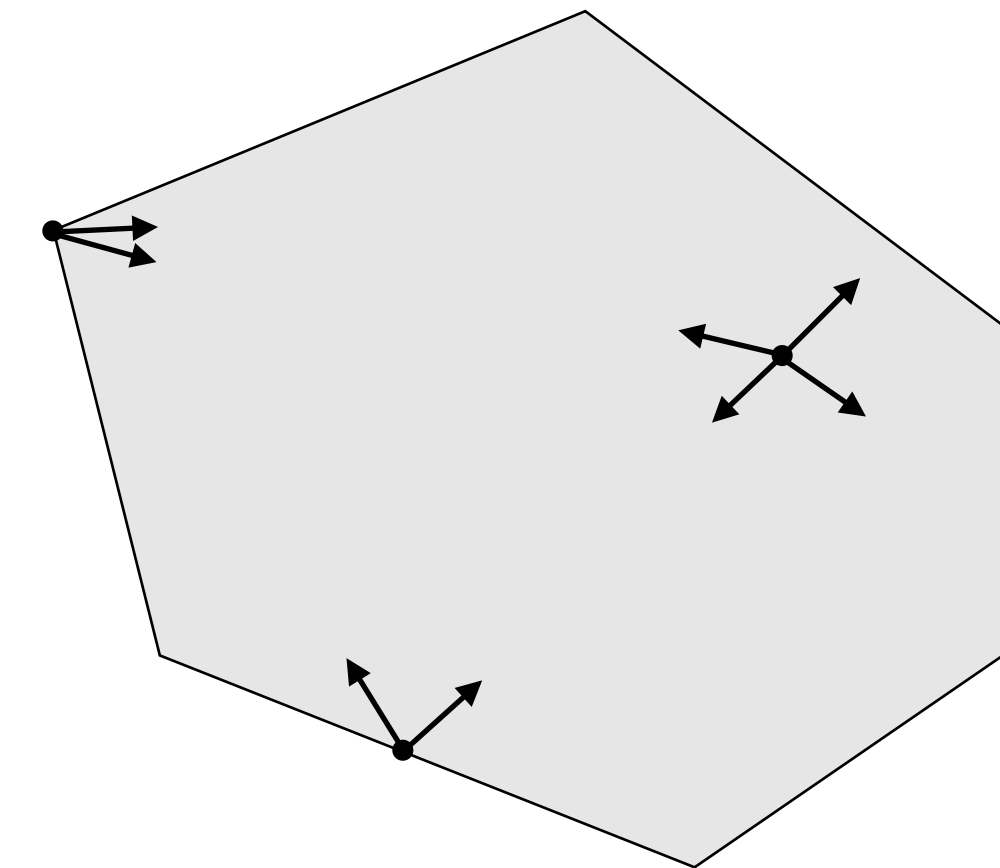
$$P = \{x \mid Ax = b, x \geq 0\}$$

Given a basis matrix $A_B = \begin{bmatrix} A_{B(1)} & \dots & A_{B(m)} \end{bmatrix}$

we have basic feasible solution x :

- x_B solves $A_B x_B = b$
- $x_i = 0, \forall i \neq B(1), \dots, B(m)$

Let $x \in P$, a vector d is a **feasible direction** at x if $\exists \theta > 0$ for which $x + \theta d \in P$



Feasible direction d

- $A(x + \theta d) = b \implies Ad = 0$
- $x + \theta d \geq 0$

Feasible directions

Computation

$$P = \{x \mid Ax = b, x \geq 0\}$$

Feasible direction d

- $A(x + \theta d) = b \implies Ad = 0$
- $x + \theta d \geq 0$

Nonbasic indices ($x_i = 0$)

- $d_j = 1 \longrightarrow$ Add j to basis B
- $d_k = 0, \forall k \notin \{j, B(1), \dots, B(m)\}$

Basic indices ($x_B > 0$)

$$Ad = 0 = \sum_{i=1}^n A_i d_i = A_B d_B + A_j = 0 \implies d_B \text{ solves } A_B d_B = -A_j$$

Non-negativity (non-degenerate assumption)

- Non-basic variables: $x_i = 0$. Nonnegative direction $d_i \geq 0$
- Basic variables: $x_B > 0$. Therefore $\exists \theta > 0$ such that $x_B + \theta d_B \geq 0$

Feasible directions

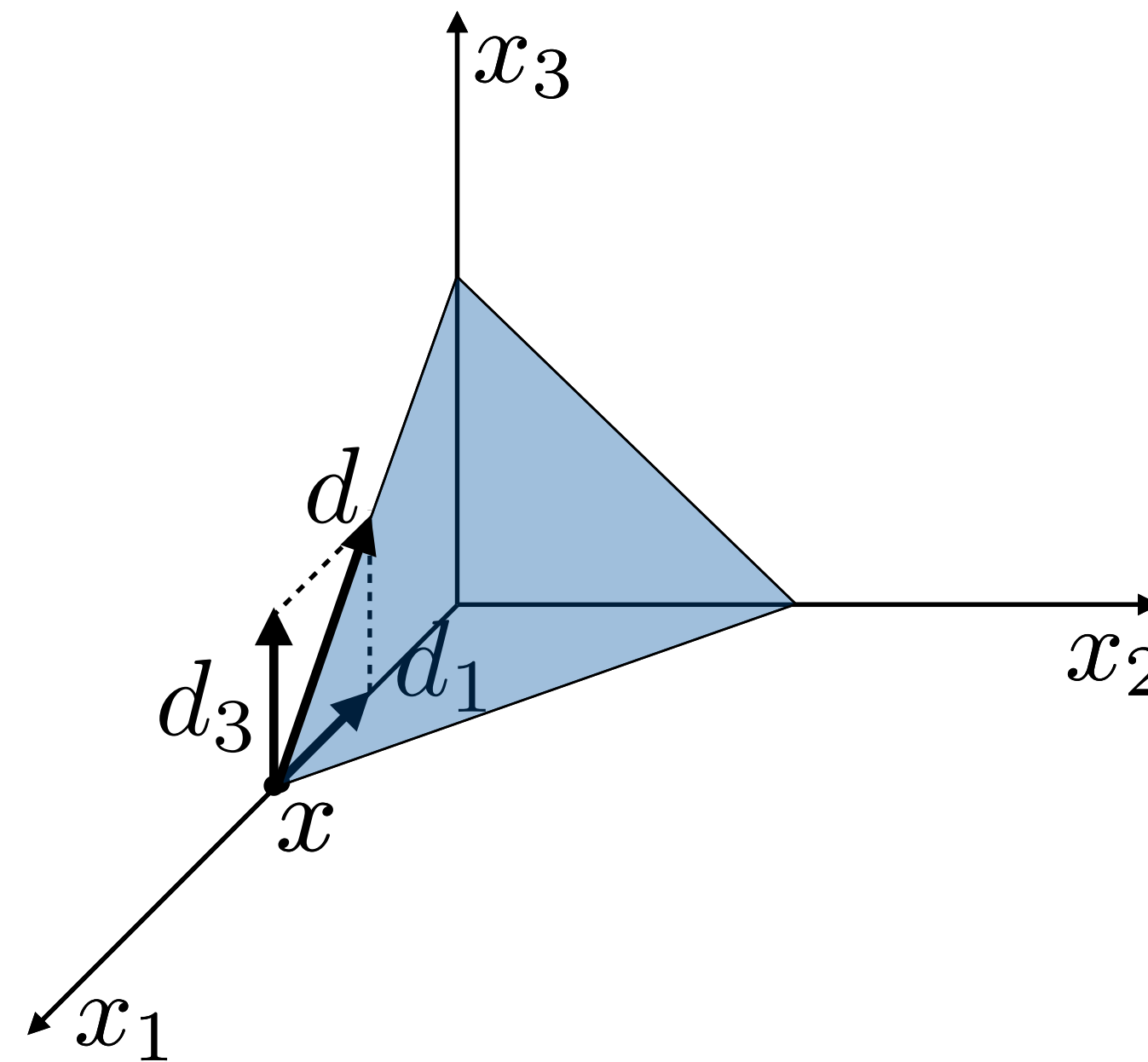
Example

$$P = \{x \mid x_1 + x_2 + x_3 = 2, \quad x \geq 0\}$$

$$x = (2, 0, 0) \quad B = \{1\}$$

Basic index $j = 3$ $\longrightarrow d = (-1, 0, 1)$

$$A_B d_B = -A_j \quad \Rightarrow \quad d_j = 1$$
$$A_B d_B = -A_j \quad \Rightarrow \quad d_B = -1$$



How does the cost change?

Cost improvement

$$c^T(x + \theta d) - c^T x = \theta c^T d$$

New cost

Old cost

We call \bar{c}_j the **reduced cost** of (introducing) variable x_j in the basis

$$\bar{c}_j = c^T d = \sum_{j=1}^n c_j d_j = c_j + c_B^T d_B = c_j - c_B^T A_B^{-1} A_j$$

Reduced costs

Interpretation

Change in objective/marginal cost of adding x_j to the basis

$$\bar{c}_j = c_j - c_B^T A_B^{-1} A_j$$

Cost per-unit increase
of variable x_j

Cost to change other variables
compensating for x_j
to enforce $Ax = b$

- $\bar{c}_j > 0$: adding x_j will increase the objective (bad)
- $\bar{c}_j < 0$: adding x_j will decrease the objective (good)

Reduced costs for basic variables is 0

$$\begin{aligned}\bar{c}_{B(i)} &= c_{B(i)} - c_B^T A_B^{-1} A_{B(i)} = c_{B(i)} - c_B^T (A_B^{-1} A_B) e_i \\ &= c_{B(i)} - c_B^T e_i = c_{B(i)} - c_{B(i)} = 0\end{aligned}$$

Vector of reduced costs

Reduced costs

$$\bar{c}_j = c_j - c_B^T A_B^{-1} A_j$$

Full vector in one shot?

$$\bar{c} = (\bar{c}_1, \dots, \bar{c}_n)$$

Isolate basis B -related components p
(they are the same across j)

$$\bar{c}_j = c_j - A_j^T (A_B^{-1})^T c_B = c_j - A_j^T p$$

Obtain p by solving linear system

$$p = (A_B^{-1})^T c_B \quad \Rightarrow \quad A_B^T p = c_B$$

Note: $(M^{-1})^T = (M^T)^{-1}$
for any square invertible M

Computing reduced cost vector

1. Solve $A_B^T p = c_B$
2. $\bar{c} = c - A^T p$

Optimality conditions

Optimality conditions

Theorem

Let x be a basic feasible solution associated with basis B

Let \bar{c} be the vector of reduced costs.

If $\bar{c} \geq 0$, then x is **optimal**

Remark

This is a **stopping criterion** for the simplex algorithm.

If the **neighboring solutions** do not improve the cost, we are done

Optimality conditions

Proof

For a basic feasible solution x with basis B the reduced costs are $\bar{c} \geq 0$.

Consider any feasible solution y and define $d = y - x$

Since x and y are feasible, then $Ax = Ay = b$ and $Ad = 0$

$$Ad = A_B d_B + \sum_{i \in N} A_i d_i = 0 \quad \Rightarrow \quad d_B = - \sum_{i \in N} A_B^{-1} A_i d_i$$

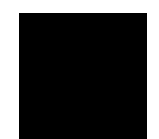
N are the
nonbasic indices

The change in objective is

$$c^T d = c_B^T d_B + \sum_{i \in N} c_i d_i = \sum_{i \in N} (c_i - c_B^T A_B^{-1} A_i) d_i = \sum_{i \in N} \bar{c}_i d_i$$

Since $y \geq 0$ and $x_i = 0, i \in N$, then $d_i = y_i - x_i \geq 0, i \in N$

$$c^T d = c^T (y - x) \geq 0 \quad \Rightarrow \quad c^T y \geq c^T x.$$



Simplex iterations

Stepsize

What happens if some $\bar{c}_j < 0$?

We can decrease the cost by bringing x_j into the basis

How far can we go?

$$\theta^* = \max\{\theta \mid \theta \geq 0 \text{ and } x + \theta d \geq 0\} \quad d \text{ is the } j\text{-th basic direction}$$

Unbounded

If $d \geq 0$, then $\theta^* = \infty$. The LP is unbounded.

Bounded

If $d_i < 0$ for some i , then

$$\theta^* = \min_{\{i \mid d_i < 0\}} \left(-\frac{x_i}{d_i} \right) = \min_{\{i \in B \mid d_i < 0\}} \left(-\frac{x_i}{d_i} \right)$$

(Since $d_i \geq 0$, $i \notin B$)

Moving to a new basis

Next feasible solution

$$x + \theta^* d$$

Let $B(\ell) \in \{B(1), \dots, B(m)\}$ be the index such that $\theta^* = -\frac{x_{B(\ell)}}{d_{B(\ell)}}$. Then,

$$x_{B(\ell)} + \theta^* d_{B(\ell)} = 0$$

New solution

- $x_{B(\ell)}$ becomes 0 (exits)
- x_j becomes θ^* (enters)

New basis

$$A_{\bar{B}} = \left[A_{B(1)} \quad \dots \quad A_{B(\ell-1)} \quad A_j \quad A_{B(\ell+1)} \quad \dots \quad A_{B(m)} \right]$$

An iteration of the simplex method

First part

We start with

- a basic feasible solution x
- a basis matrix $A_B = \begin{bmatrix} A_{B(1)} & \dots, & A_{B(m)} \end{bmatrix}$

1. Compute the reduced costs \bar{c}

- Solve $A_B^T p = c_B$
- $\bar{c} = c - A^T p$

2. If $\bar{c} \geq 0$, x **optimal. break**

3. Choose j such that $\bar{c}_j < 0$

An iteration of the simplex method

Second part

4. Compute search direction d with $d_j = 1$ and $A_B d_B = -A_j$
5. If $d_B \geq 0$, the problem is **unbounded** and the optimal value is $-\infty$. **break**
6. Compute step length $\theta^* = \min_{\{i \in B \mid d_i < 0\}} \left(-\frac{x_i}{d_i} \right)$
7. Define y such that $y = x + \theta^* d$
8. Get new basis \bar{B} (i exits and j enters)

Example

$$P = \{x \mid x_1 + x_2 + x_3 = 2, \quad x \geq 0\}$$

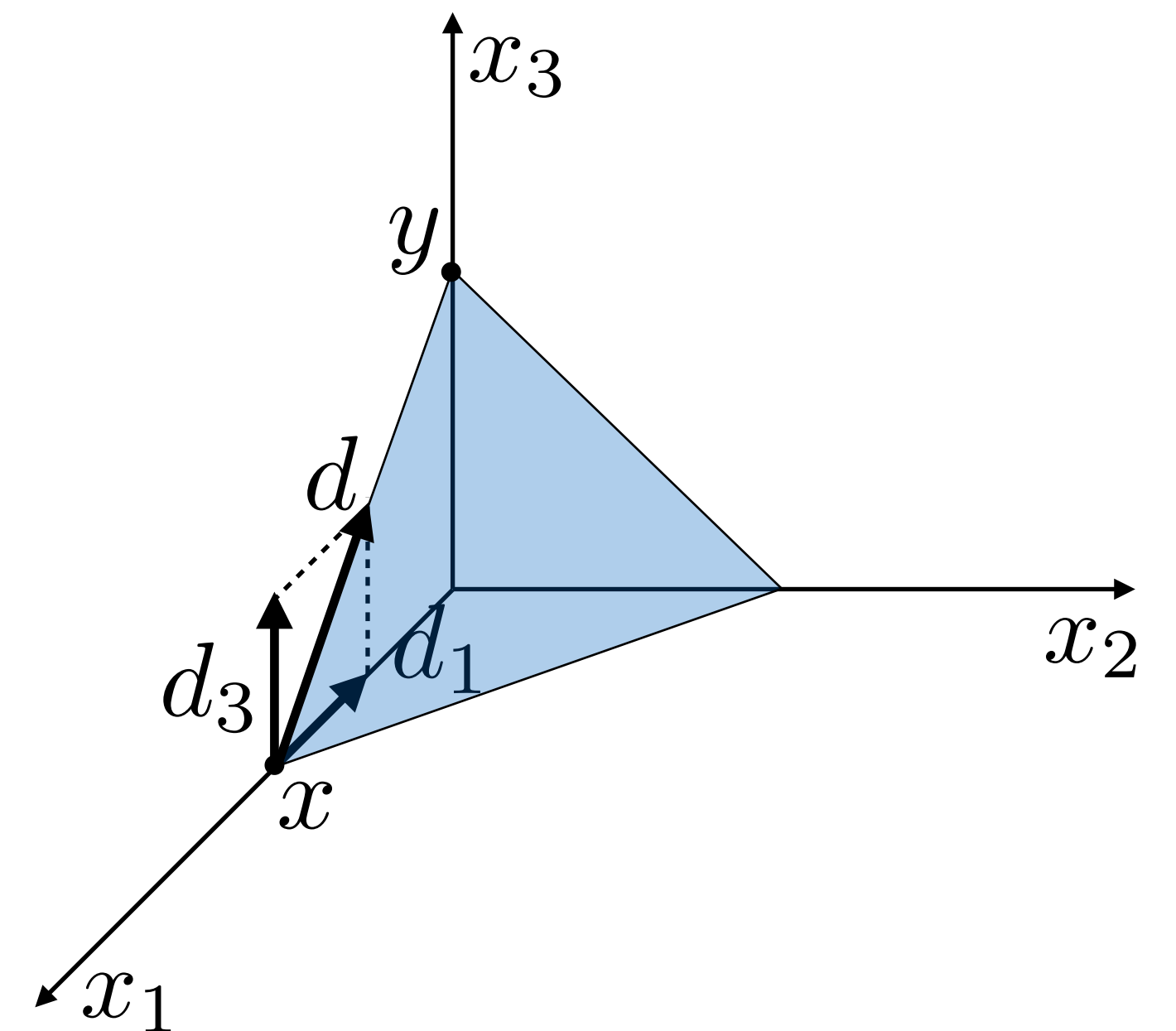
$$x = (2, 0, 0) \quad B = \{1\}$$

$$\text{Basic index } j = 3 \longrightarrow d = (-1, 0, 1) \\ d_j = 1$$

$$A_B d_B = -A_j \quad \Rightarrow \quad d_B = -1$$

$$\text{Stepsize } \theta^* = -\frac{x_1}{d_1} = 2$$

$$\text{New solution } y = x + \theta^* d = (0, 0, 2) \quad \bar{B} = \{3\}$$



Finite convergence

Assume that

- $P = \{x \mid Ax = b, x \geq 0\}$ not empty
- Every basic feasible solution **non degenerate**

Then

- The simplex method **terminates after a finite number of iterations**
- At termination we either have one of the following
 - an **optimal basis** B
 - a **direction** d such that $Ad = 0$, $d \geq 0$, $c^T d < 0$ and the optimal cost is $-\infty$

Finite convergence

Proof sketch

At each iteration the algorithm improves

- by a **positive** amount θ^*
- along the **direction** d such that $c^T d < 0$

Therefore

- The cost strictly decreases
- No basic feasible solution can be visited twice

Since there is a **finite number of basic feasible solutions**

The algorithm **must eventually terminate**



The simplex method

Today, we learned to:

- **Iterate** between basic feasible solutions
- **Verify** optimality and unboundedness conditions
- **Apply** a single iteration of the simplex method
- **Prove** finite convergence of the simplex method in the non-degenerate case

References

- Bertsimas and Tsitsiklis: Introduction to Linear Optimization
 - Chapter 3: The simplex method
- R. Vanderbei: Linear Programming — Foundations and Extensions
 - Chapter 2 : The simplex method
 - Chapter 6: The simplex method in matrix notation

Next lecture

- Finding initial basic feasible solution
- Degeneracy
- Complexity