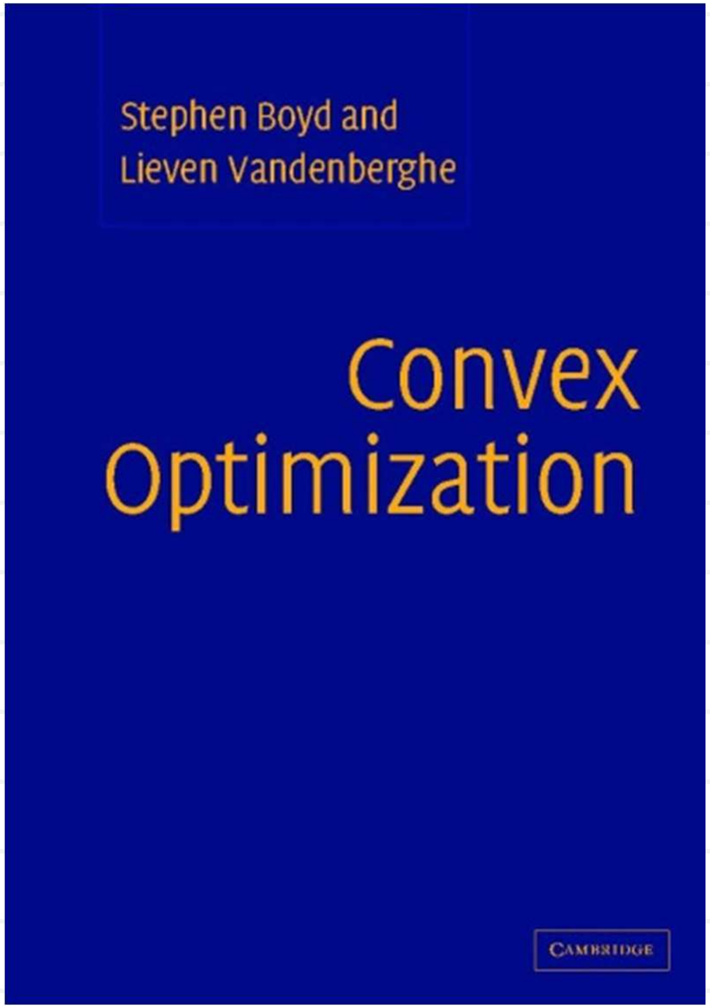
Lecture 13: Convex optimization, special cases

Today's reading



Convex programming

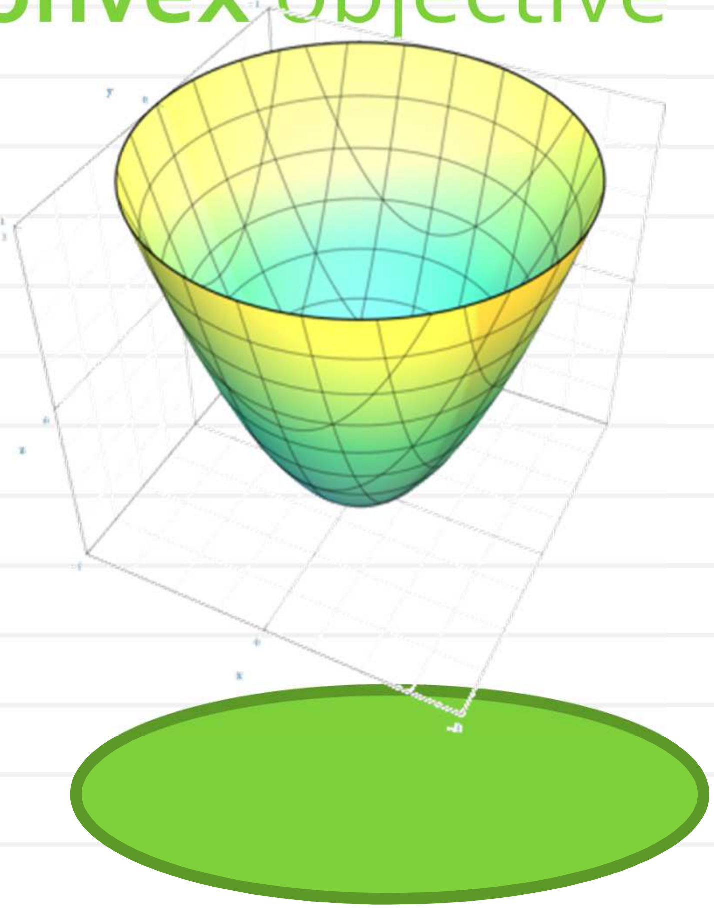
Convex objective

"minimize"

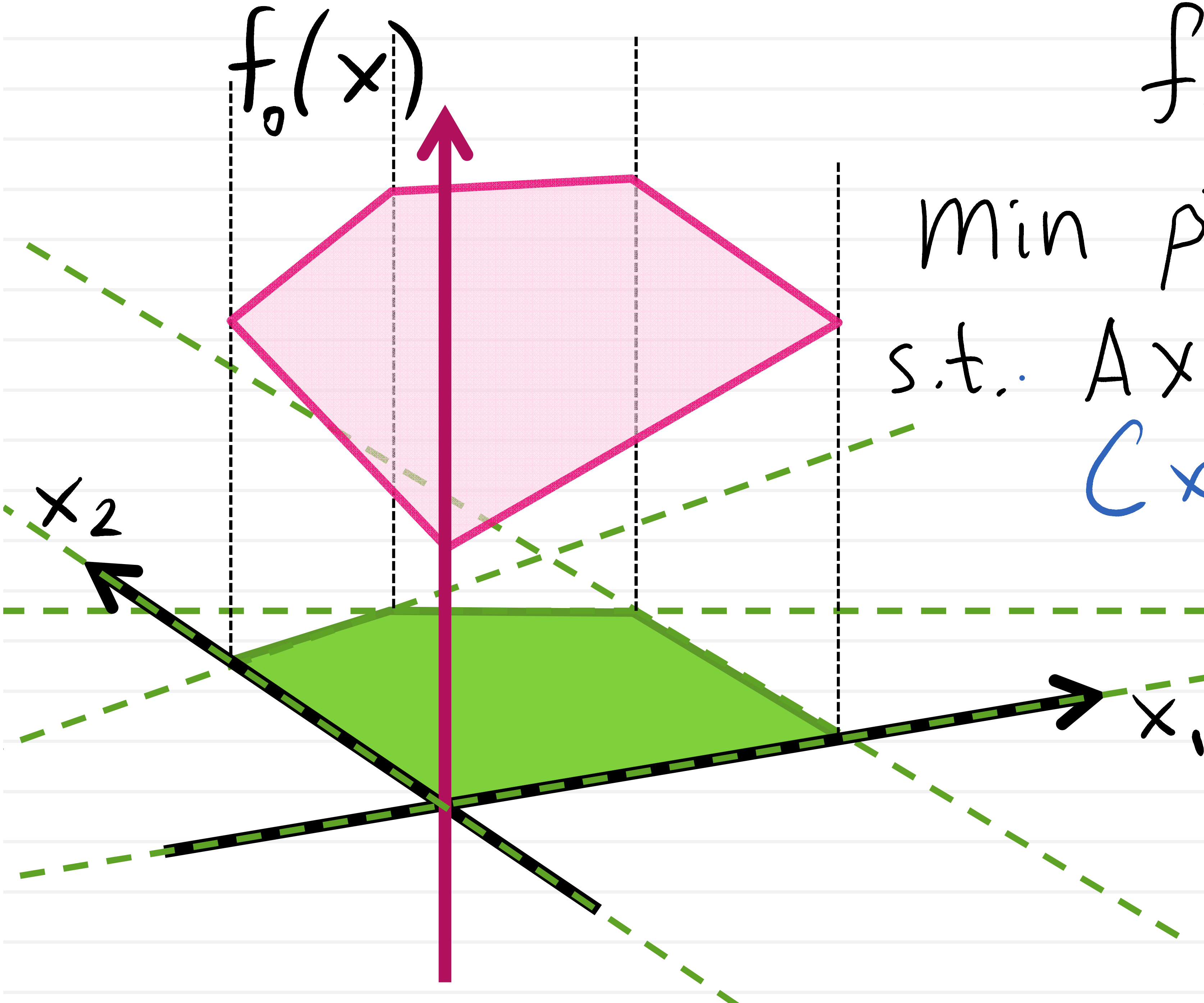
$$\min \; f_0(x)$$
$$s.t.: \; x \in D$$

"subject to"

Convex domain



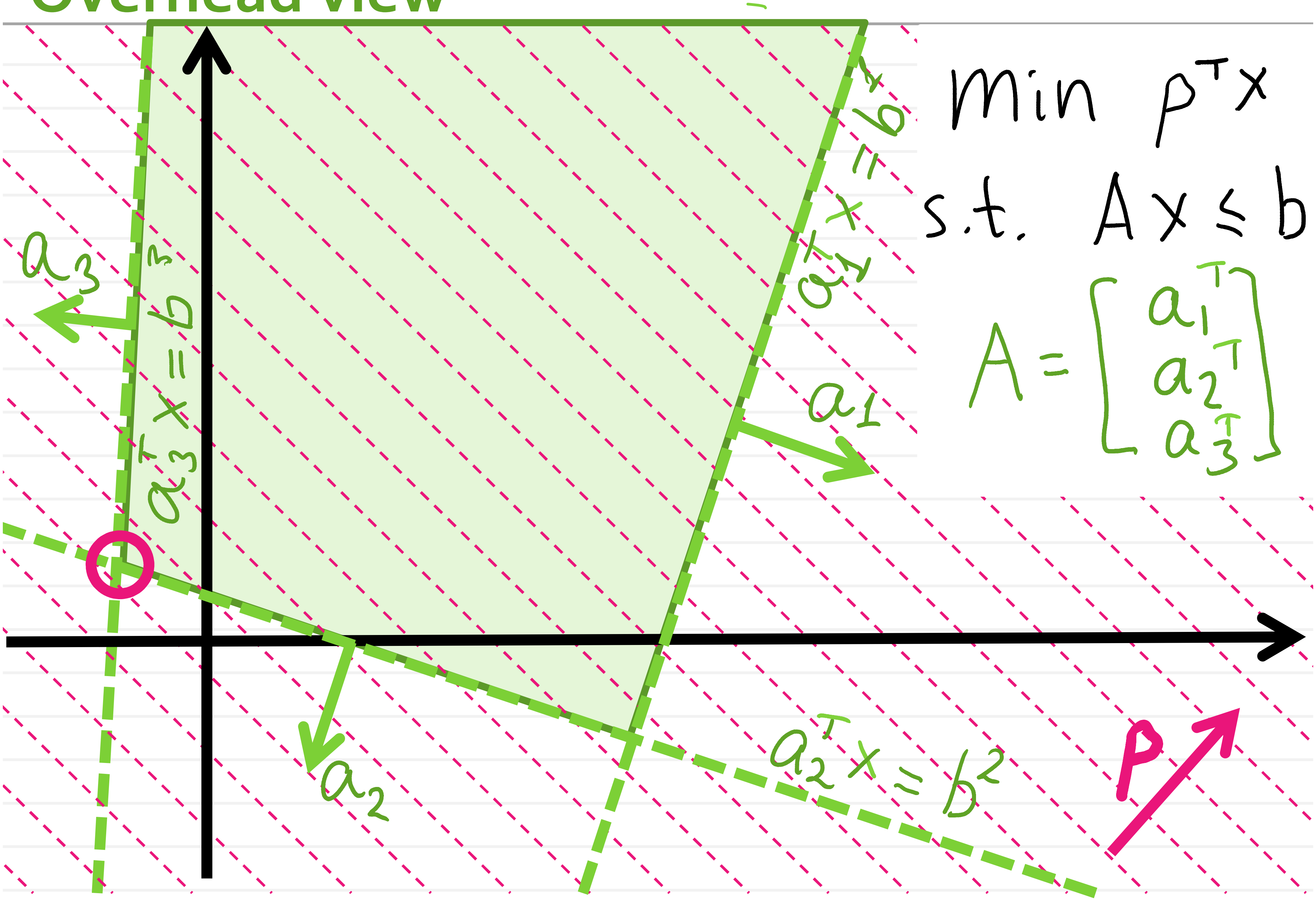
Linear program



$$f_0(x) = p^T x$$

$$\min \; p^T x$$
$$\text{s.t.} \; Ax \le b$$
$$Cx = d$$

Overhead view

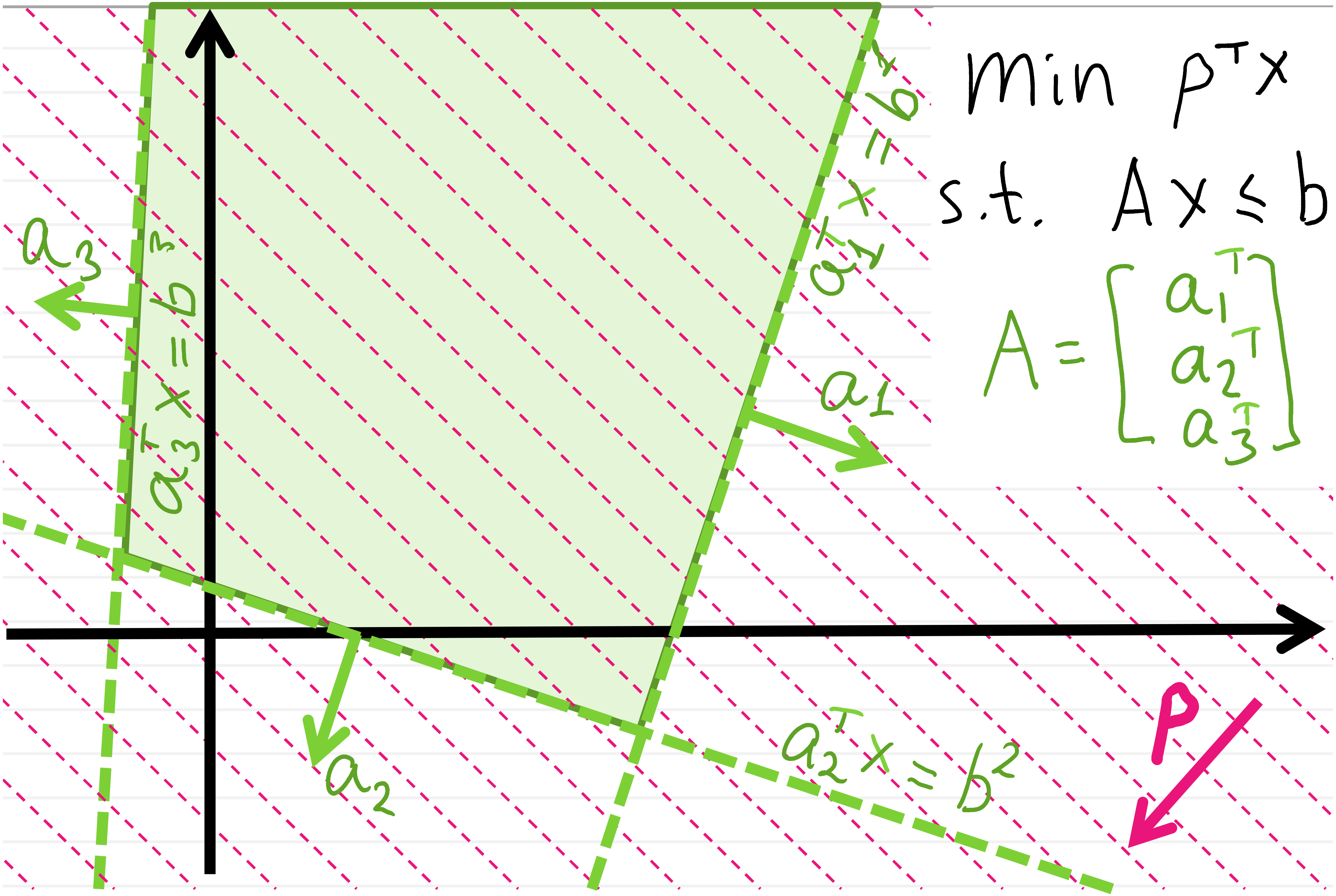


$\min \ p^T x$

s.t. $Ax \le b$

$A = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix}$

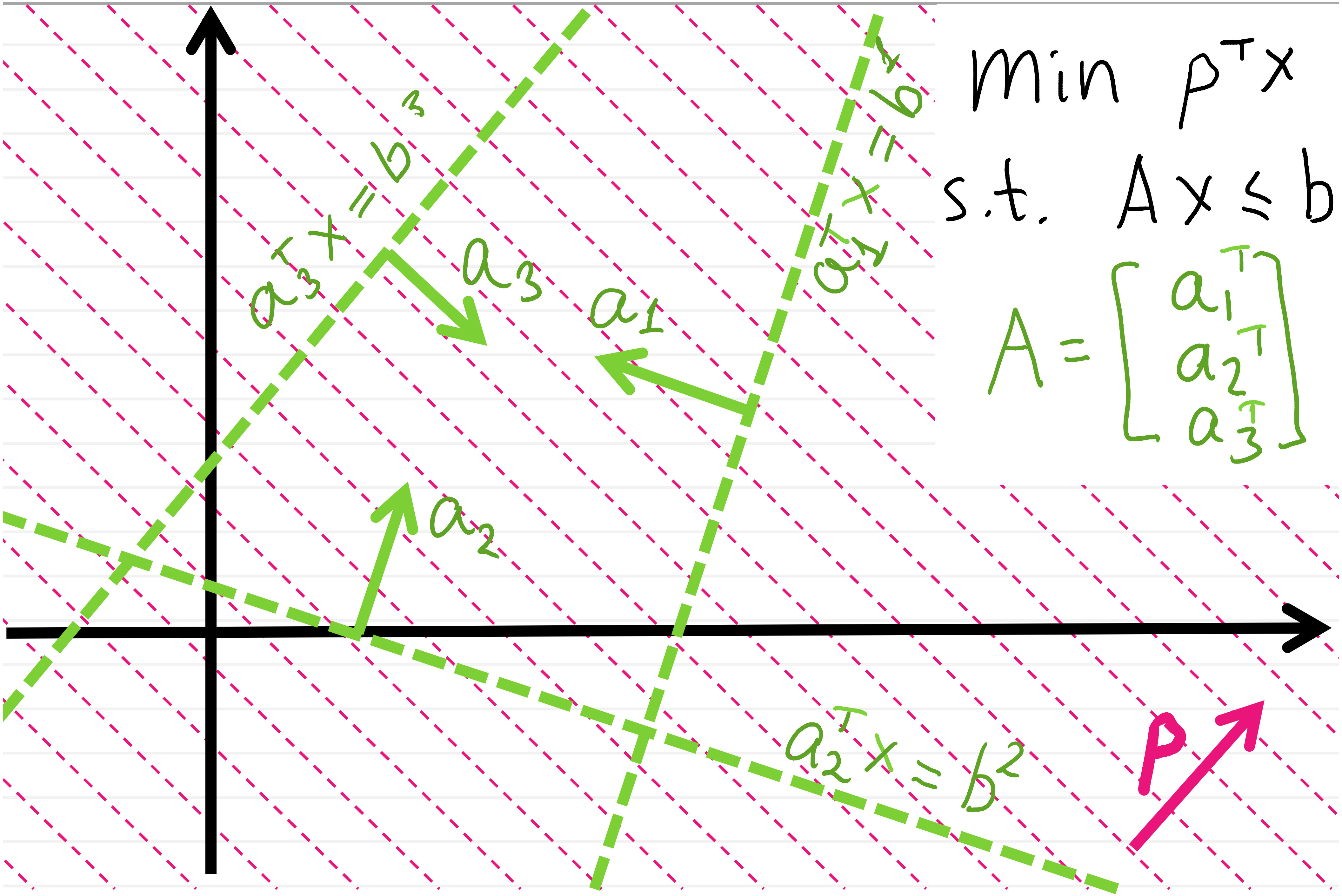
Overhead view



$$\min p^T x$$
$$\text{s.t. } Ax \le b$$

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix}$$

Overhead view



$$\min \; p^T x$$
$$\text{s.t. } \; Ax \le b$$
$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix}$$

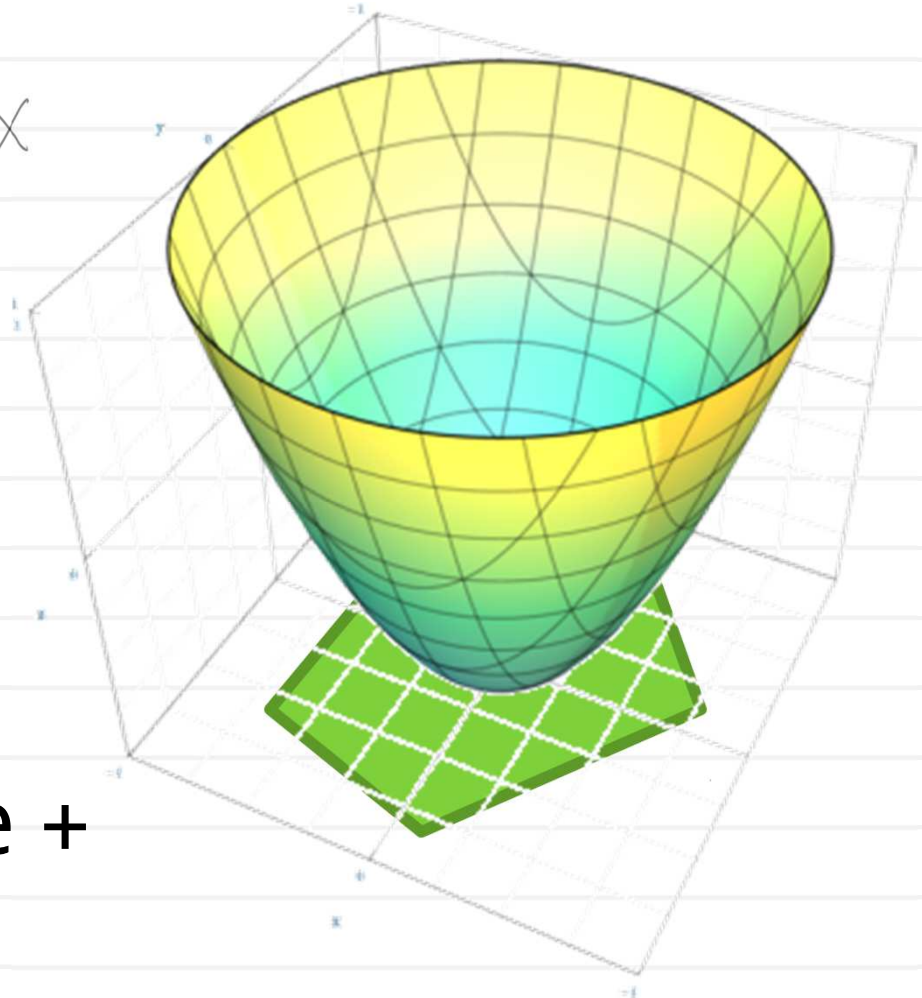
Convex Quadratic Program

Positive
Semi-definite

$$\min_x \frac{1}{2} x^T Q x + p^T x$$

$$\text{s.t. } Ax \leq b$$

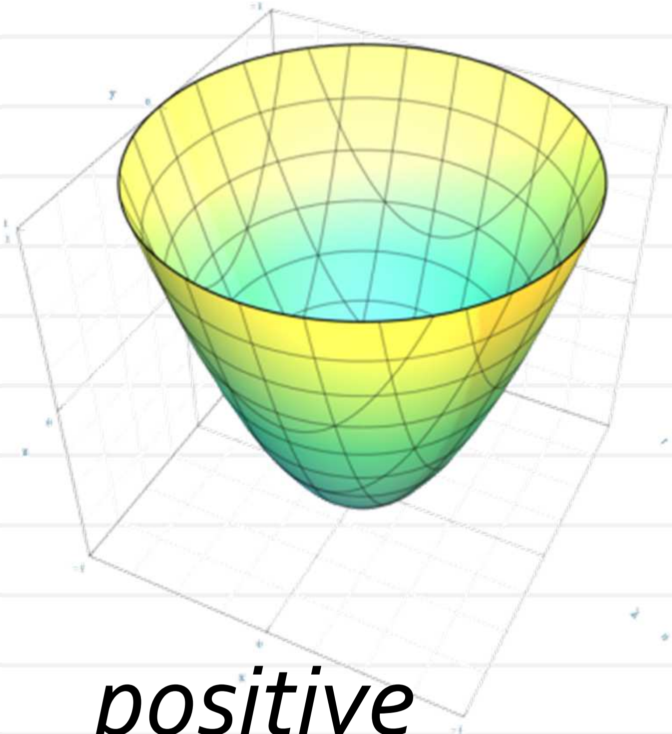
$$Cx = d$$



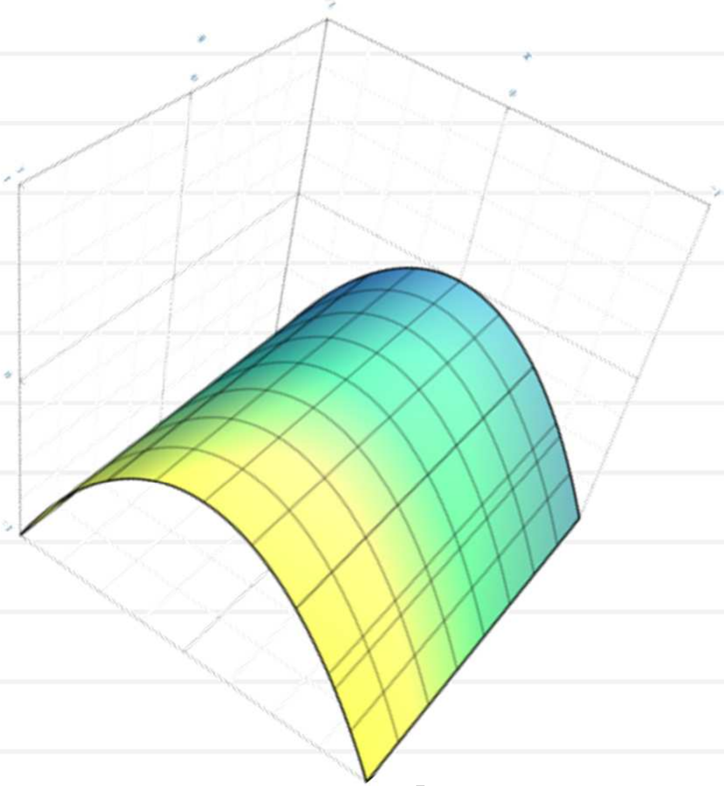
QP = quadratic objective + linear constraints

Reminder: quadratic functions

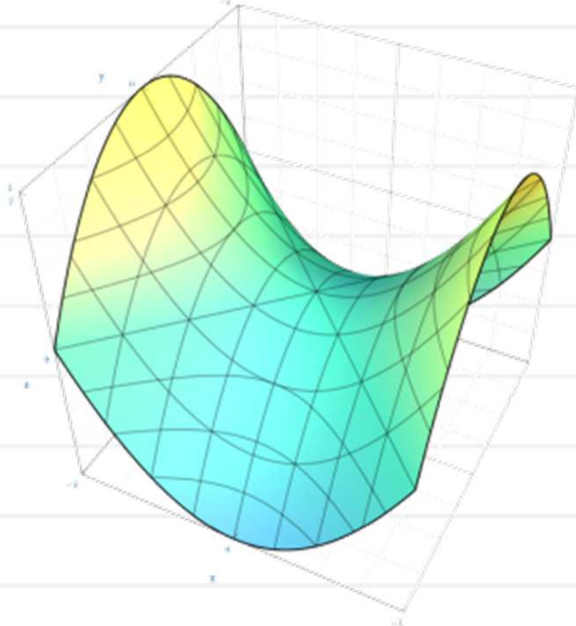
$$f(x) = \frac{1}{2} x^T A x + b^T x + c$$



positive definite



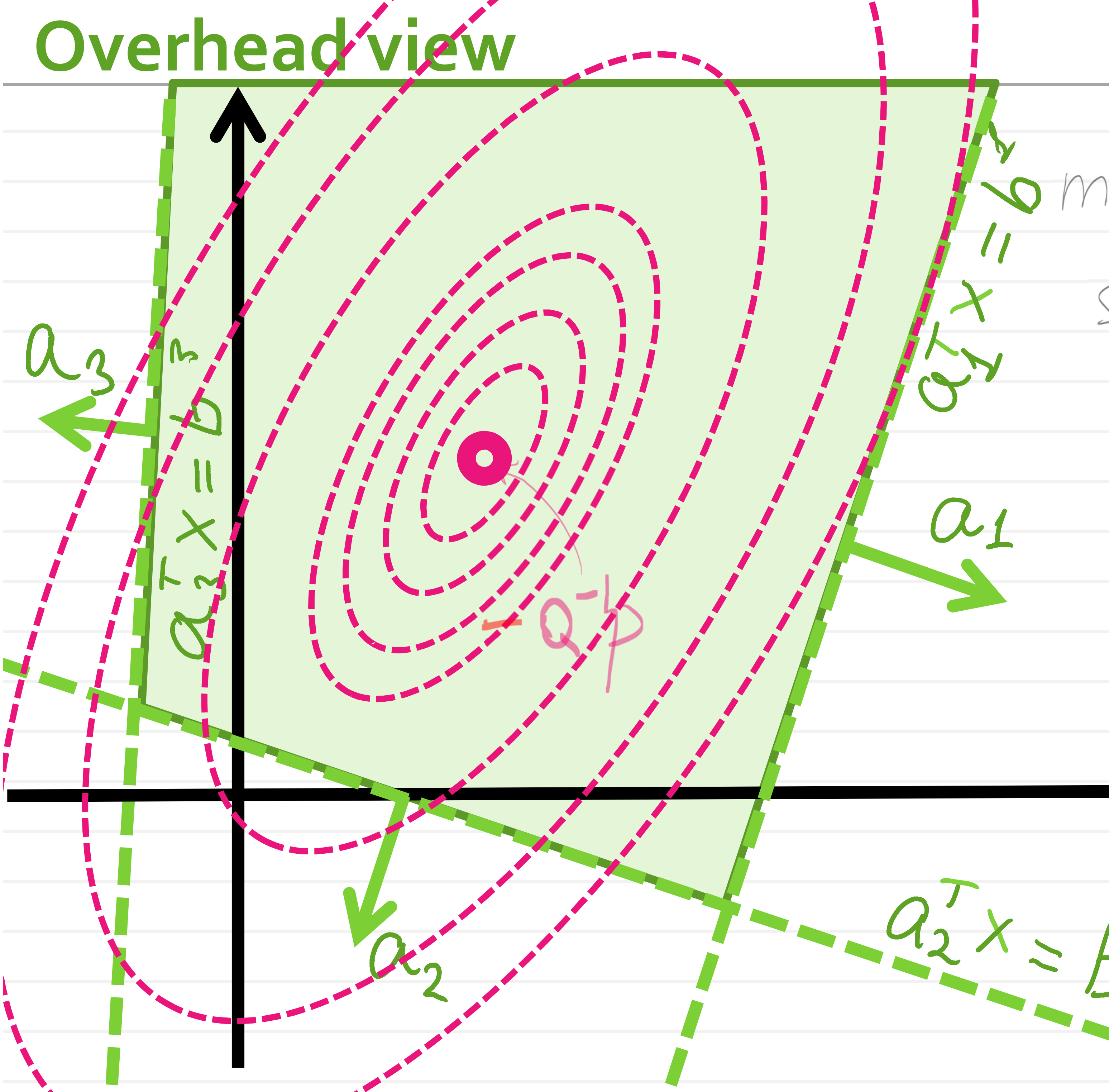
negative semi-definite



indefinite

Images from Wikipedia

Overhead view

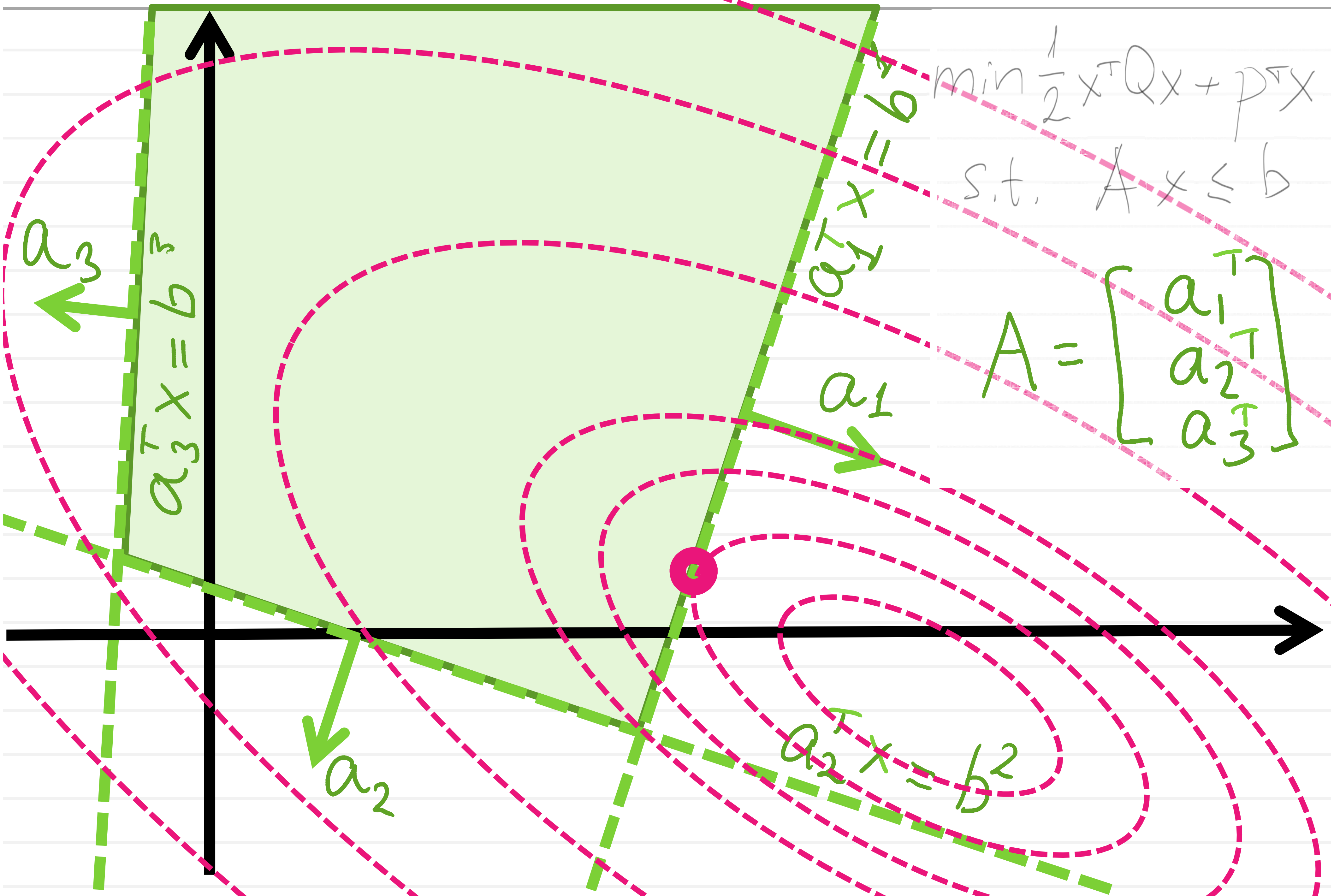


$$\min \frac{1}{2} x^T Q x + p^T x$$

$$\text{s.t. } Ax \leq b$$

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix}$$

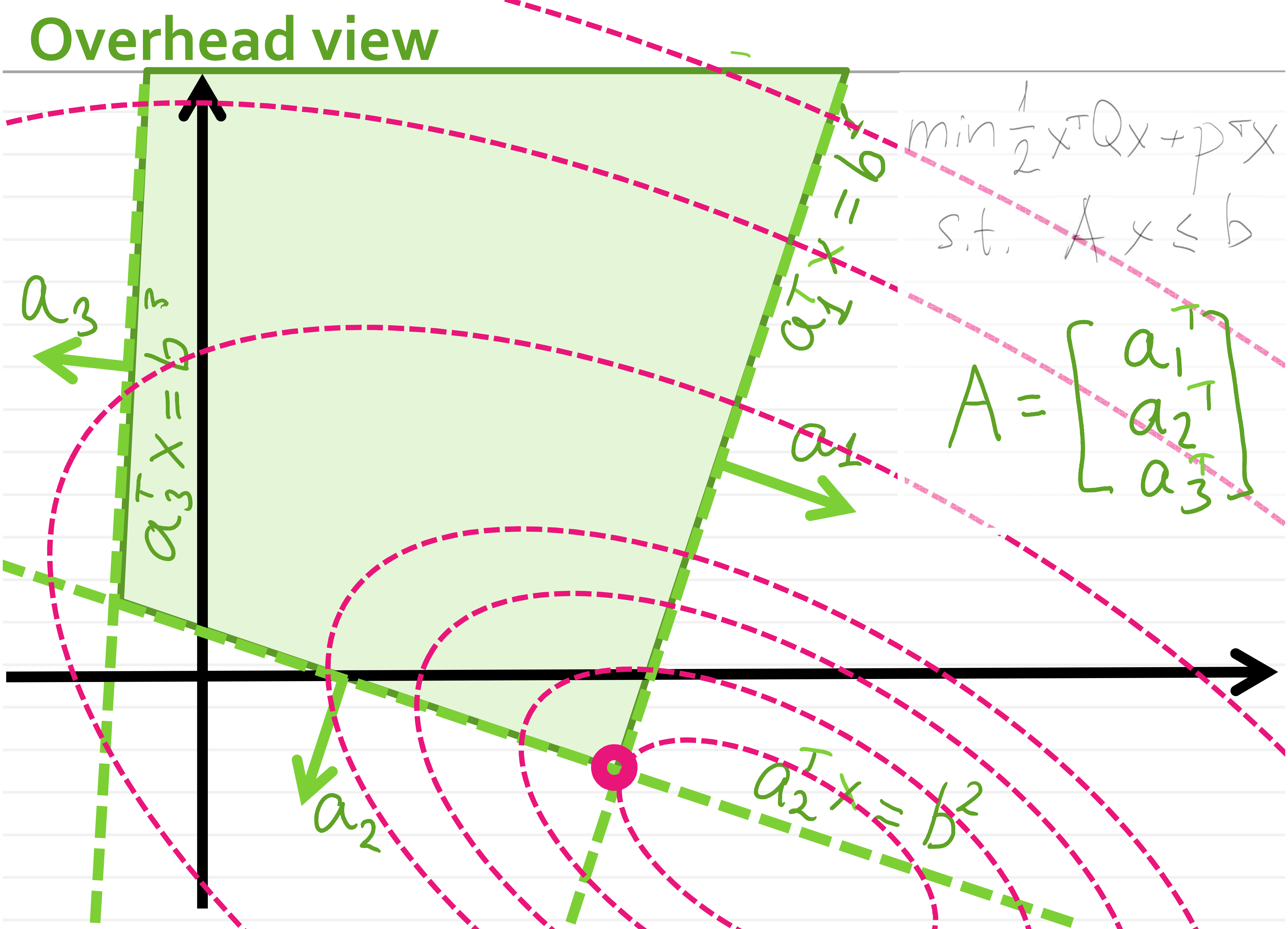
Overhead view



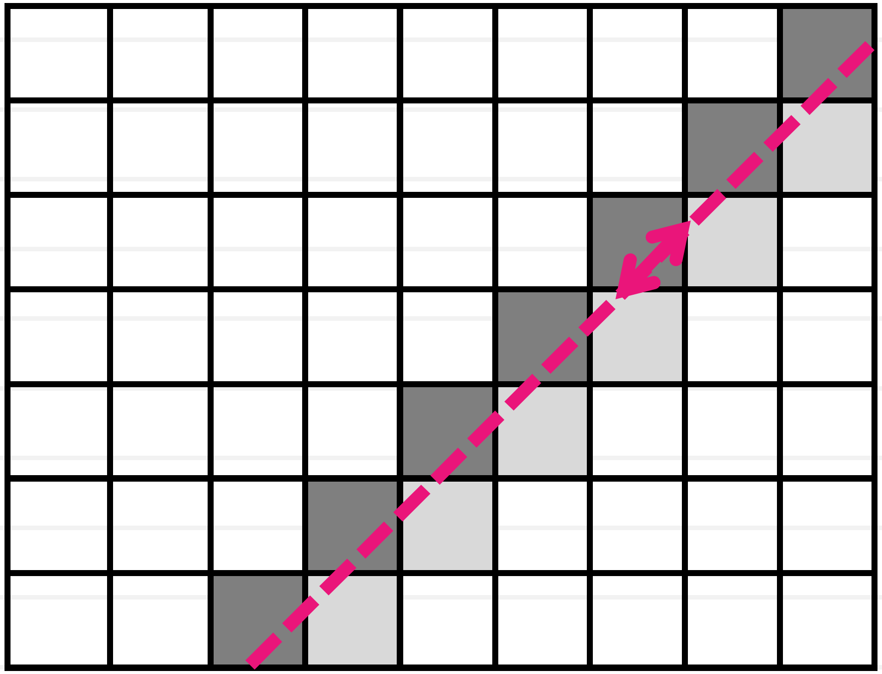
$$\min \frac{1}{2} x^T Q x + p^T x$$
$$\text{s.t. } Ax \le b$$

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix}$$

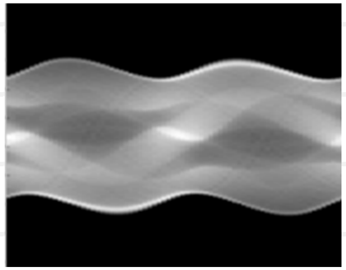
Overhead view

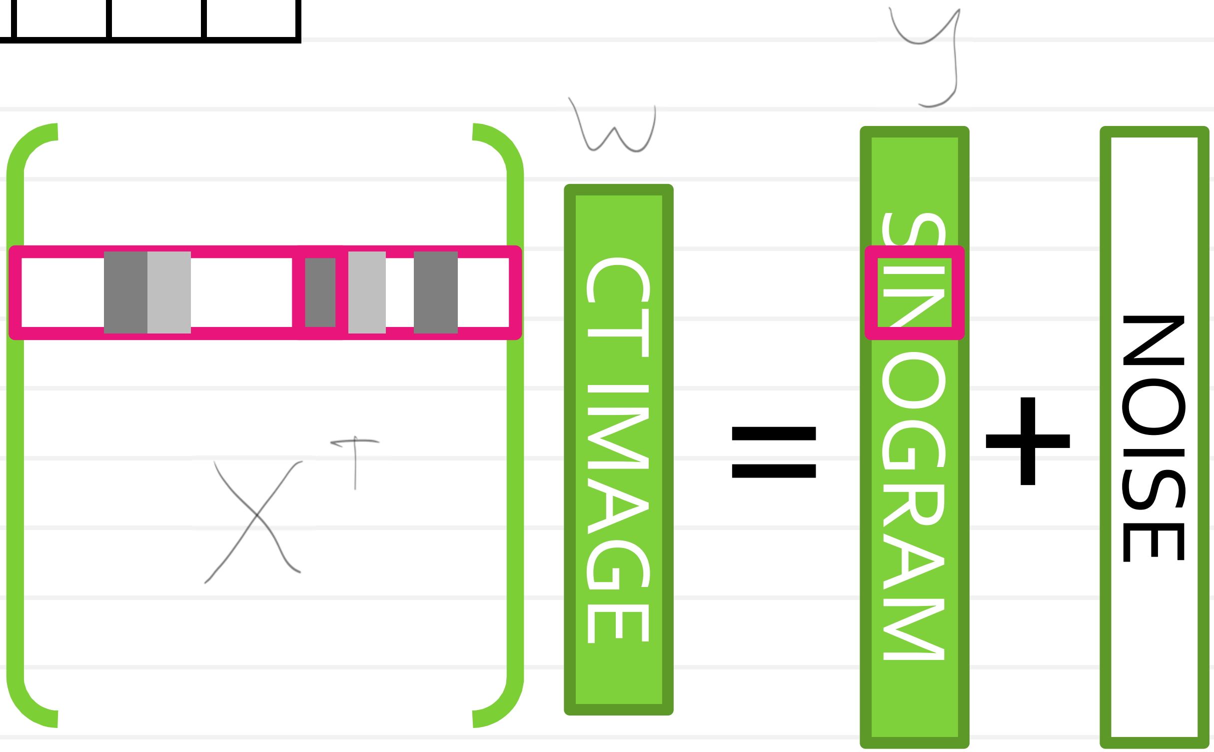


Reminder: least squares



$$\min_{\omega} \frac{1}{2}\|X^T\omega - y\|^2$$



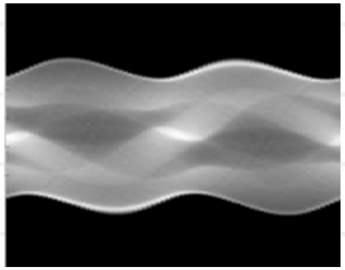


Constrained least squares

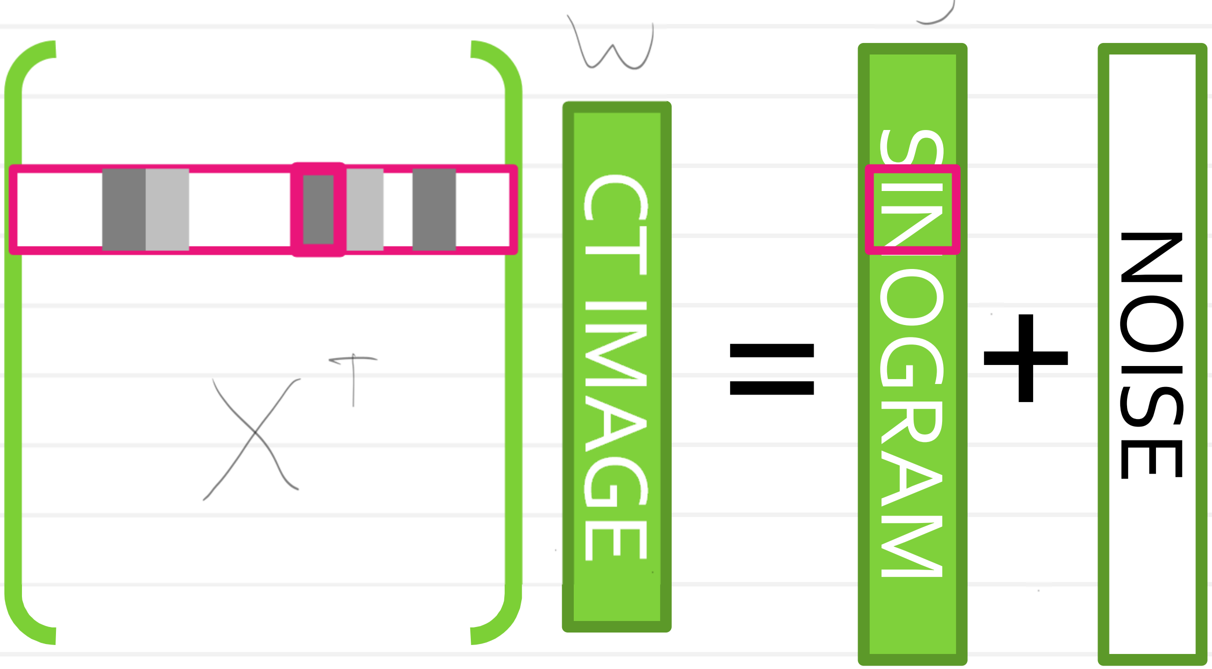
$$\min \frac{1}{2} \omega^T X X^T \omega - y^T X^T \omega \qquad \min_{\omega} \frac{1}{2}\|X^T\omega - y\|^2$$

$$XX^T = Q \qquad y^T X^T = p^T$$

$$\text{s.t. } 0 \le \omega_i \le W_{max}$$



y



Regularized empirical risk minimization

(x_i, y_i) annotated examples we learn from

ω parameters we learn

$\omega^T x$ our prediction

$$\min_{\omega} \frac{1}{2}\|\omega\|^2 + \frac{C}{N}\sum_{i=1}^{N} L(\omega^T x_i, y_i)$$

Regularization:
doing not much worse on other examples

Empirical risk:
doing well on training examples

Regularized empirical risk minimization

$$\min_{\omega} \frac{1}{2}\|\omega\|^2 + \frac{C}{N}\sum_{i=1}^{N} L(\omega^T x_i, y_i)$$

Example 1: regression

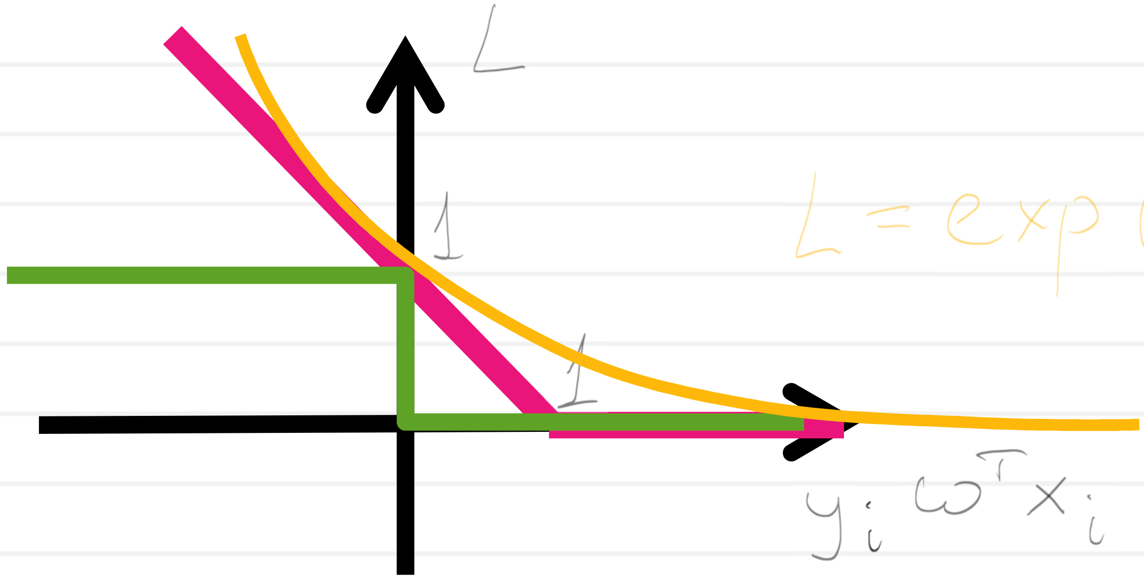
$$L(\omega^T x_i, y_i) = (\omega^T x_i - y_i)^2$$

Not really suitable for classification…

$$y_i \in \{-1; 1\}$$

Learning to classify

$$\min_{\omega} \frac{1}{2}\|\omega\|^2 + \frac{C}{N}\sum_{i=1}^{N} L(\omega^T x_i, y_i)$$



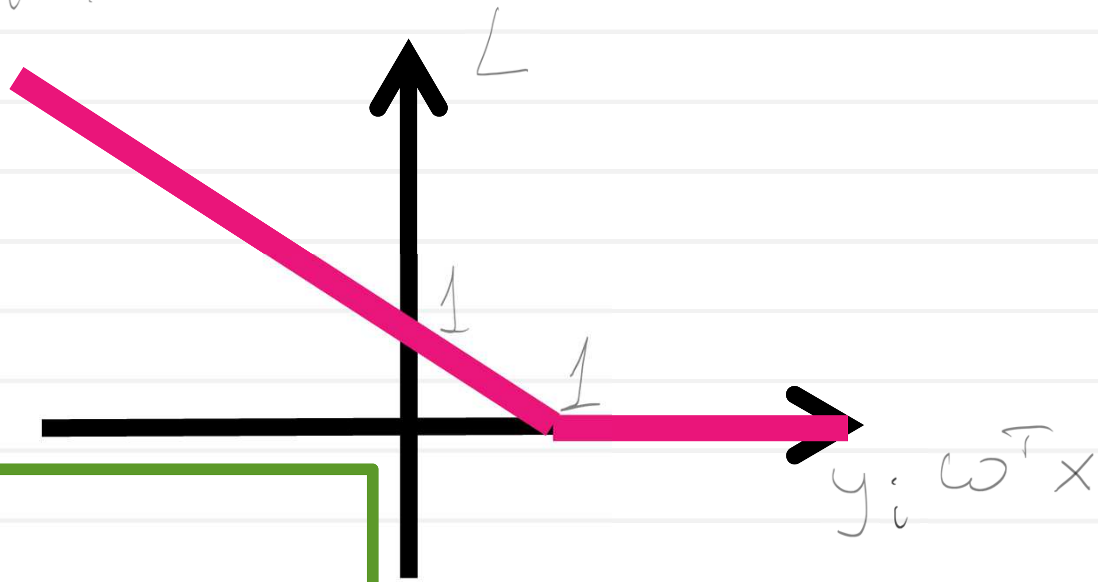
$$L = \exp(-y_i \omega^T x_i)$$

$$L(\omega^T x_i, y_i) = [\operatorname{sign} \omega^T x_i == y_i]$$

$$L(\omega^T x_i, y_i) = \max(0, 1 - y_i \omega^T x_i)$$

Learning to classify with the hinge loss

$$\min_{\omega} \frac{1}{2}\|\omega\|^2 + \frac{C}{N}\sum_{i=1}^{N} \max(0, 1 - y_i \omega^T x_i)$$



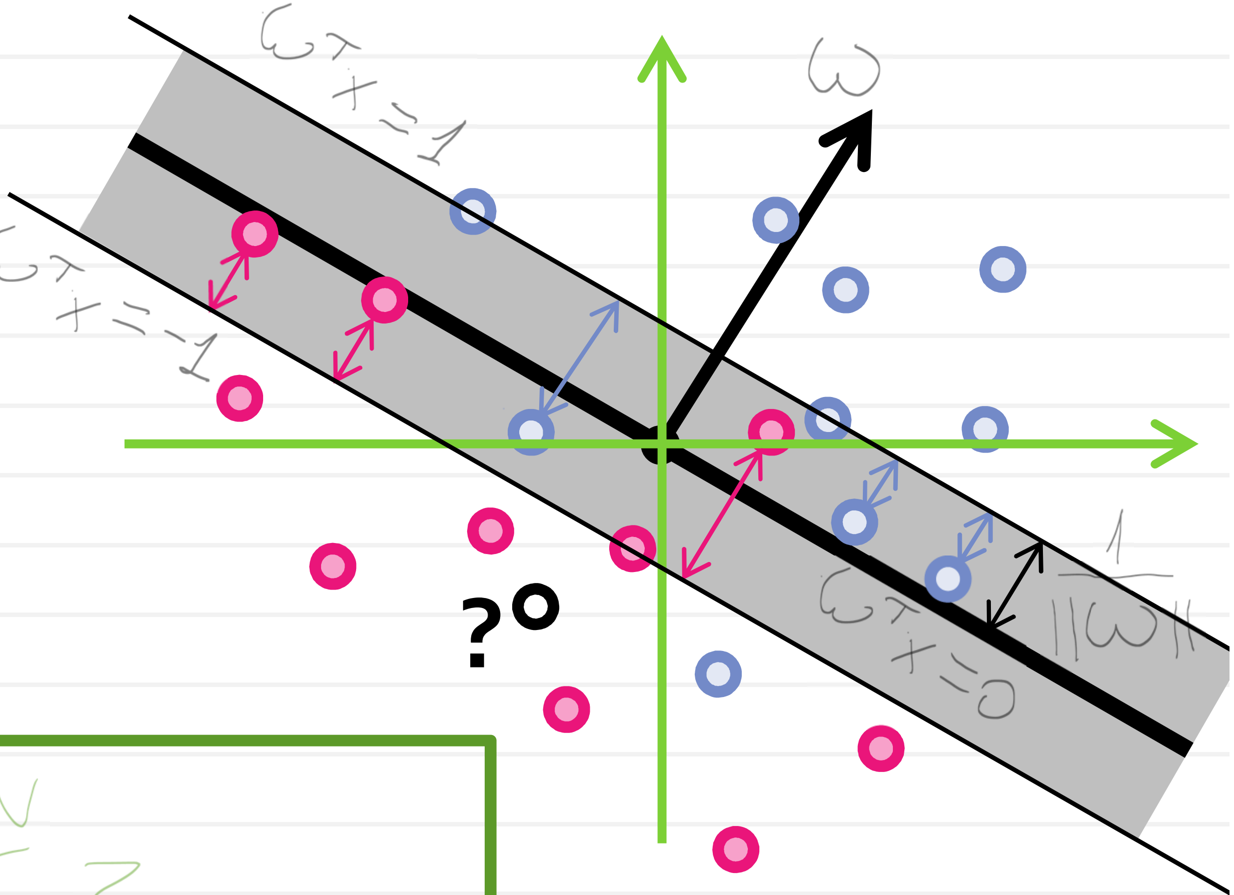
SVM QP:

$$\boxed{\begin{aligned} &\min_{\omega,\xi} \frac{1}{2}\|\omega\|^2 + \frac{C}{N}\sum_{i=1}^{N} \xi_i \\ &\text{s.t.:} \quad \xi_i \geq 1 - y_i \omega^T x_i \\ &\qquad\quad \xi_i \geq 0 \end{aligned}}$$

Geometric interpretation for hinge loss

SVM:

- Maximizes 1/||w||
- Minimizes slacks



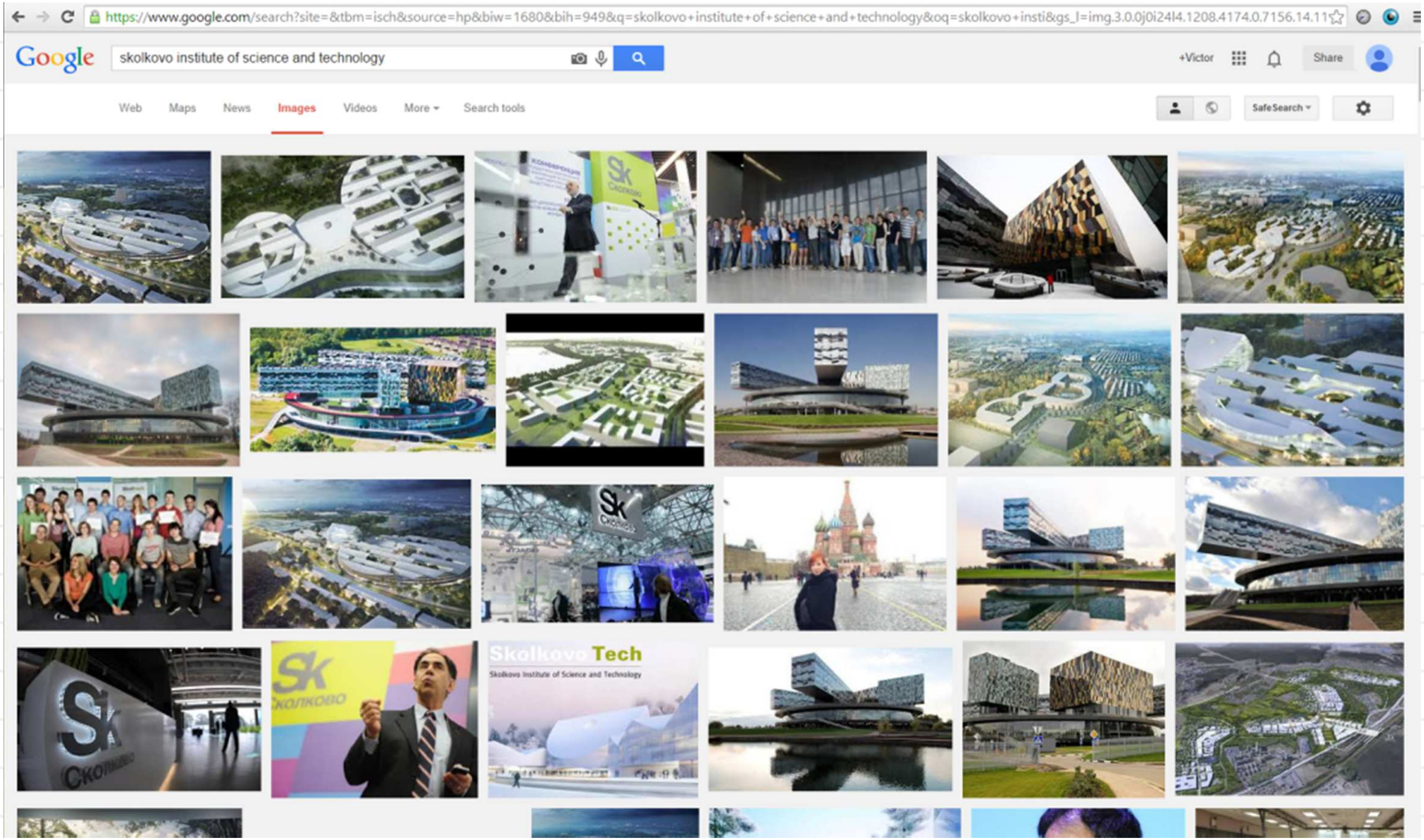
SVM QP:

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N} \xi_i$$

$$\text{s.t.:} \quad \xi_i \geq 1 - y_i w^T x_i$$

$$\xi_i \geq 0$$

Learning to rank



Learning to Rank

(q_i, x_{ij}, y_{ij}) training sample

$y_{ij} \in \{0, 1, 2, 3, 4\}$

$\Phi(q, x)$ joint feature mapping

ω parameters we learn

$\omega^T \Phi(q, x)$ our prediction for the relevance

Main idea: learn based on pairs

Learning to Rank

(q_i, x_{ij}, y_{ij}) training sample

$\Phi(q, x)$ joint feature mapping

ω parameters we learn

$\omega^T \Phi(q, x)$ our prediction for the relevance

$$\min_{\omega} \frac{1}{2}\|\omega\|^2 + C \sum_{i=1}^{N} \sum_{\substack{(j,k) \\ y_{ij} < y_{ik}}} L\left(\omega^T \Phi(q_i, x_{ik}) - \omega^T \Phi(q_i, x_{ij})\right)$$

Rank-SVM

$$\min_{\omega} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\sum_{\substack{(j,k)\\ y_{ij}\le y_{ik}}} L\Big(\underbrace{\omega^T\varphi(q_i, x_{ik}) - \omega^T\varphi(q_i, x_{ij})}_{\Delta}\Big)$$

$$\max(0, 1-\Delta)$$

$$\boxed{\begin{aligned} &\min_{\omega,\xi} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\sum_{(j,k)} \xi_{i,j,k} \\ &\text{s.t.: } \xi_{i,j,k} \ge 1 - \omega^T\varphi(q_i, x_{ik}) + \omega^T\varphi(q_i, x_{ij}) \\ &\qquad \xi_{i,j,k} \ge 0 \end{aligned}}$$

Learning to rank

$$\min_{\omega,\xi} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\sum_{(j,k)} \xi_{i,j,k}$$

$$\text{s.t.:} \quad \xi_{i,j,k} \geq 1 - \omega^T\varphi(q_i, x_{ik}) + \omega^T\varphi(q_i, x_{ij})$$

$$\xi_{i,j,k} \geq 0$$

$\approx NM^2$ constraints

- Constraint-generation (cutting plane) methods
- Stochastic subgradient…

Markowitz portfolio optimization

Task: invest a certain amount of money into multiple bonds/commodities

Objective: achieve a certain expected return, minimize risk

Markowitz portfolio optimization

$p \in \mathbb{R}^n$ change in prices over the next years

$p \sim \mathcal{N}(\bar{p}, \Sigma)$ assuming Gaussian distribution

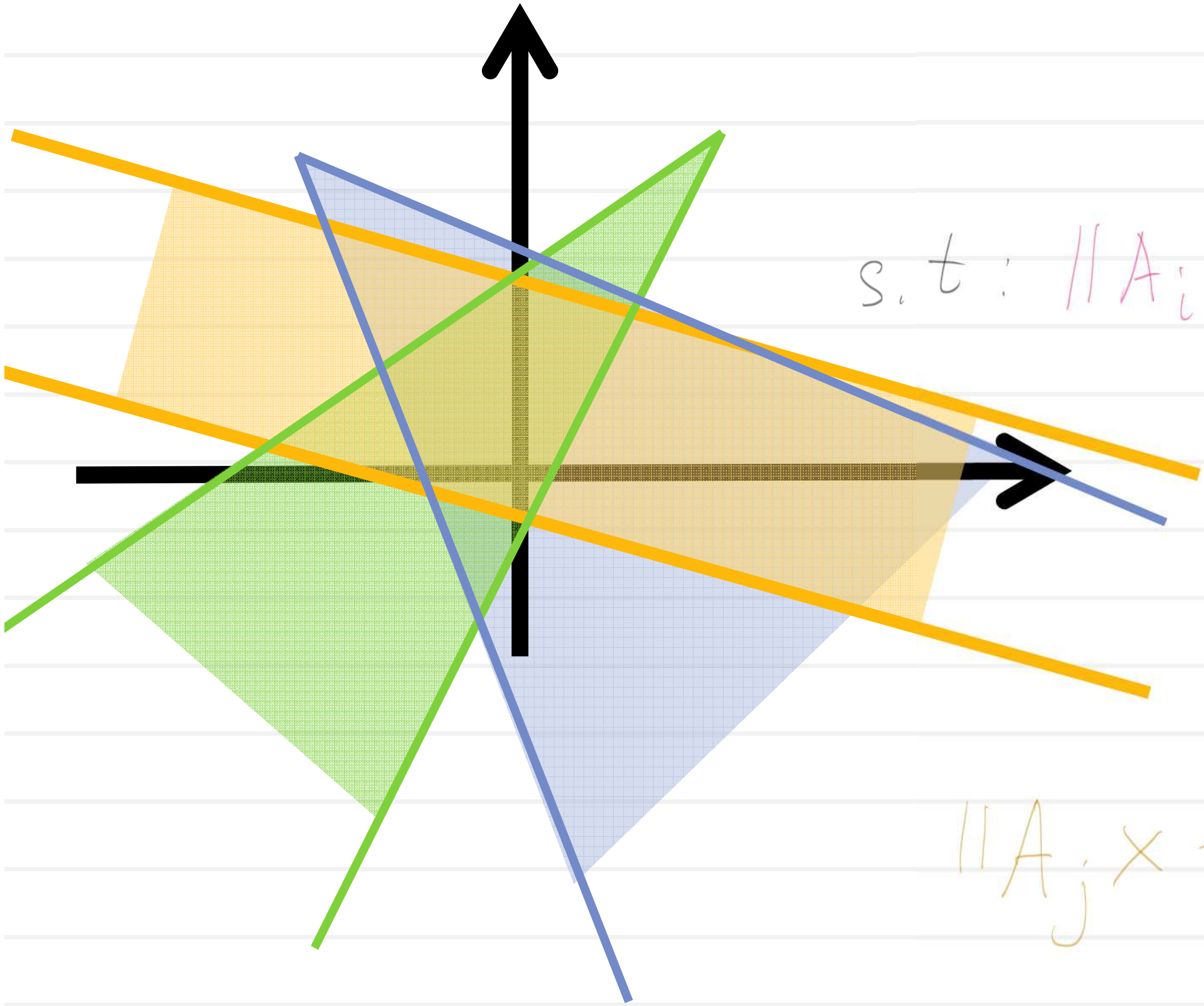
x my portfolio

$p^T x \sim \mathcal{N}(\bar{p}^T x, x^T \Sigma x)$ my return

portfolio QP:

$$\boxed{\begin{aligned} \min_x \quad & x^T \Sigma x \\ \text{s.t.:} \quad & \bar{p}^T x \geq r \\ x \geq 0 \quad & \mathbb{1}^T x = 1 \end{aligned}}$$

Second-order convex cone programming



$$\min \ p^T x$$

$$\text{s.t.:} \ \|A_i x + b_i\| \leq c_i^T x + d_i$$

$$Ex = f$$

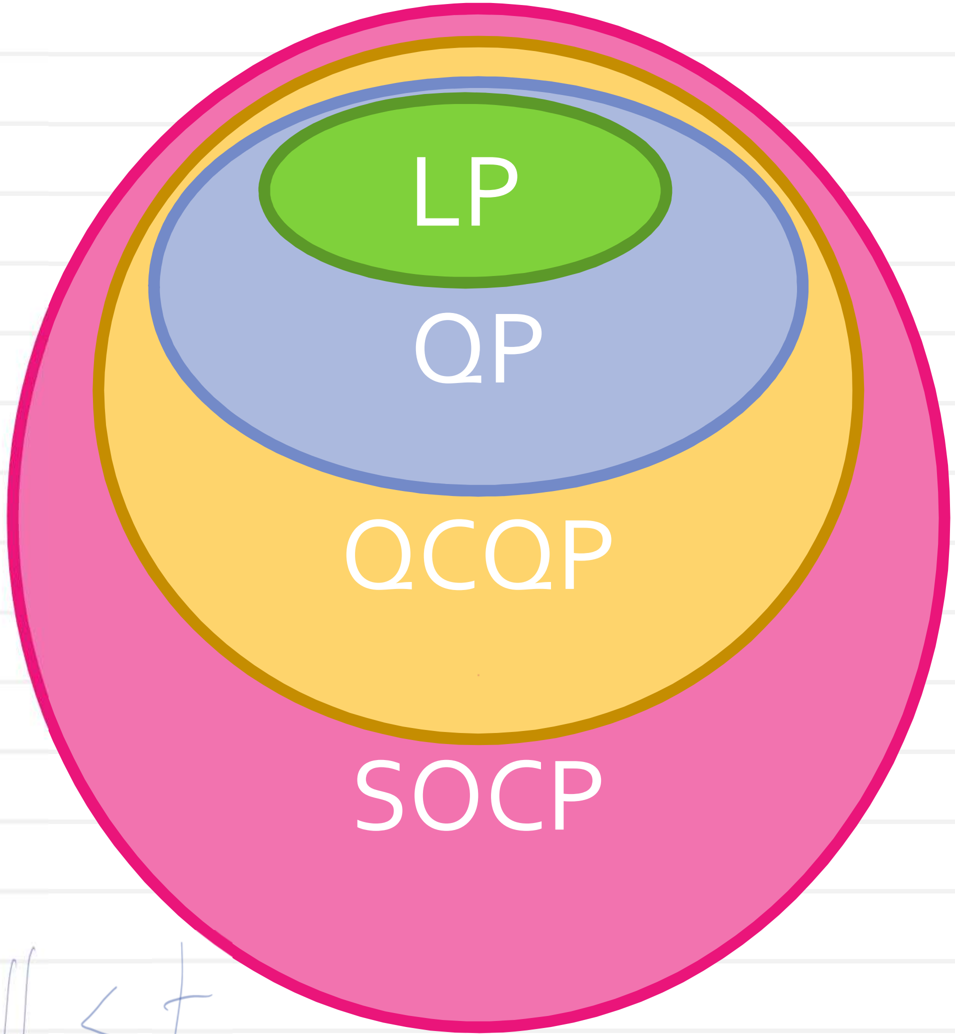
$$\|A_j x + b_j\| \leq d_j$$

- Generalizing LP by adding a special cone constraint
- Very efficient solvers, e.g. *SeDuMi* exist

From QP to SOCP

$$\min_x \frac{1}{2} x^T Q x + p^T x$$
$$\text{s.t. } Ax \le b$$
$$Cx = d$$

$$\min_x t$$
$$\text{s.t. } \|Q^{1/2} x + Q^{-\frac{1}{2}} p\| \le t$$
$$Ax \le b$$
$$Cx = d$$



Robust linear programs

$$\min \ c^T x$$

$$\text{s.t.: } a_i^T x \le b_i$$

$$a_i \in \mathcal{E}_i = \{\bar{a}_i + P_i u \mid \|u\|_2 \le 1\}$$

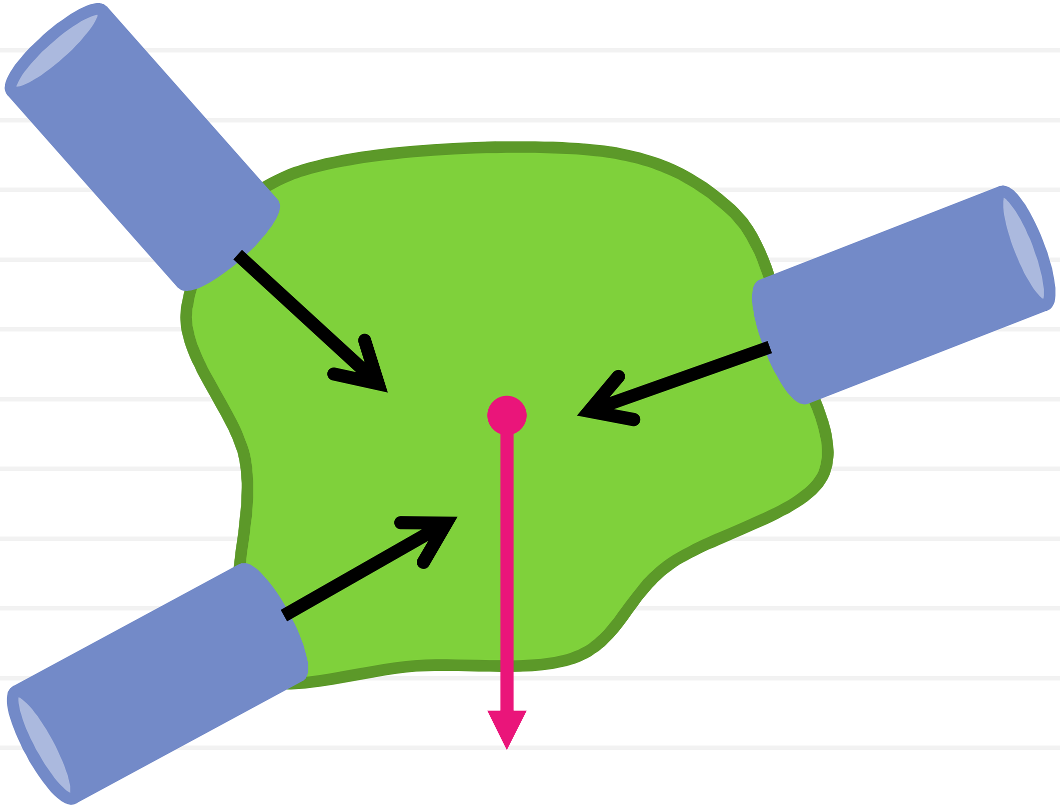
$$\sup\{a_i^T x \mid a_i \in \mathcal{E}_i\} = \bar{a}_i^T x +$$

$$+ \sup\{u^T P_i^T x \mid \|u\|_2 \le 1\} =$$

$$= \bar{a}_i^T x + \|P_i^T x\|_2$$

$$\text{s.t.: } \bar{a}_i^T x + \|P_i^T x\|_2 \le b_i$$

Grasping robot



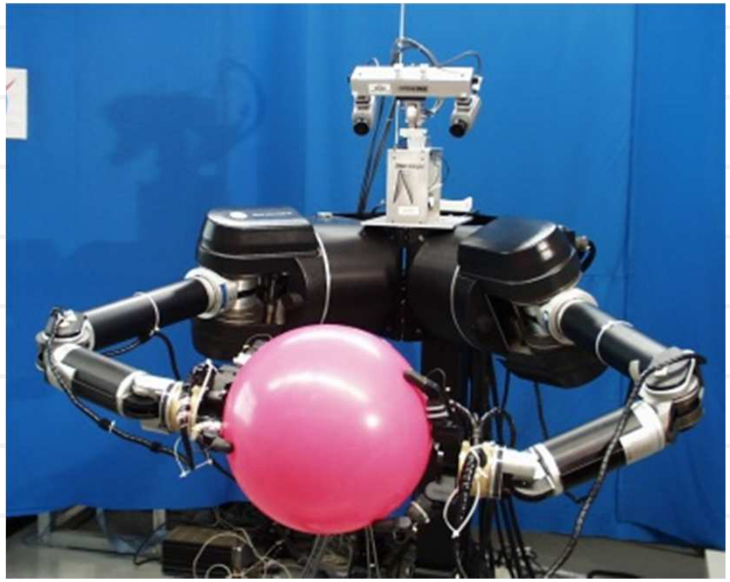
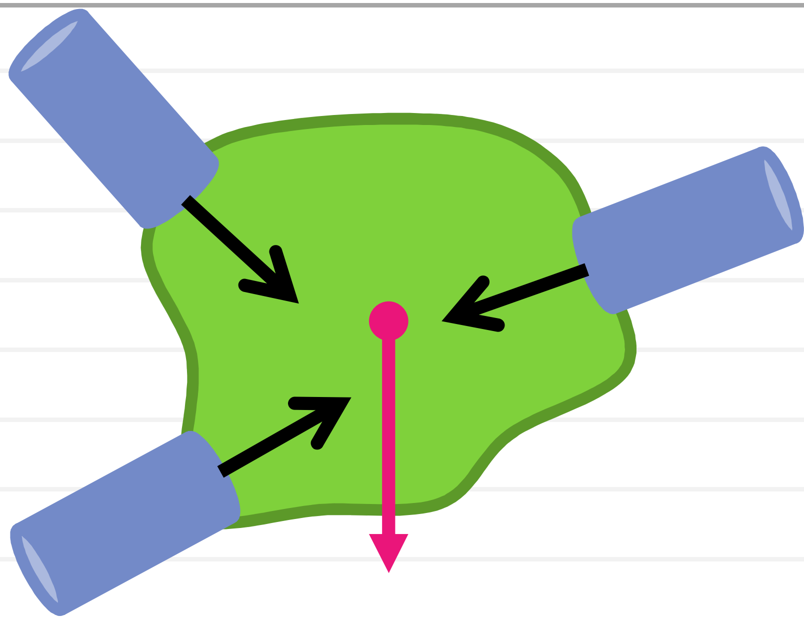


image from Lab for Perceptual Robotics@UMass

- External force (e.g. gravity)
- Force from each finger (orthogonal to surface)
- Static friction force at each finger (tangential to surface)

Optimize finger forces to held object

Grasping robot



$$F_i = v_i^T F_i v_i + (I - v_i v_i^T) F_i$$

$$\|(I - v_i v_i^T) F_i\| \le \mu v_i^T F_i$$

external force

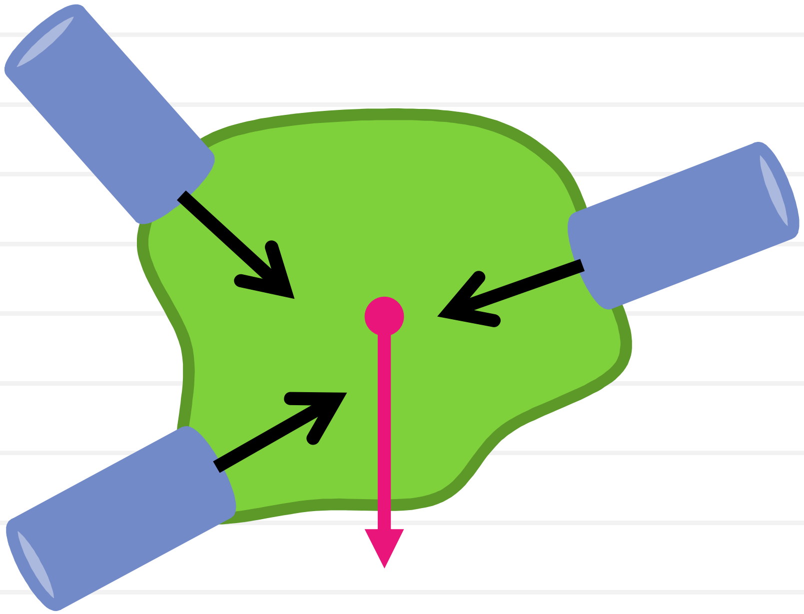
$$\sum_{i=1}^N F_i + F_{ext} = 0$$

external torque

$$\sum_{i=1}^N p_i \times F_i + T_{ext} = 0$$

Grasping robot

Grasping robot SOCP



$$\min_{t, F_i} \ t$$

$$v_i^T F_i \leq t$$

$$\|(I - v_i v_i^T) F_i\| \leq \mu v_i^T F$$

$$\sum_{i=1}^{N} F_i + F_{ext} = 0$$

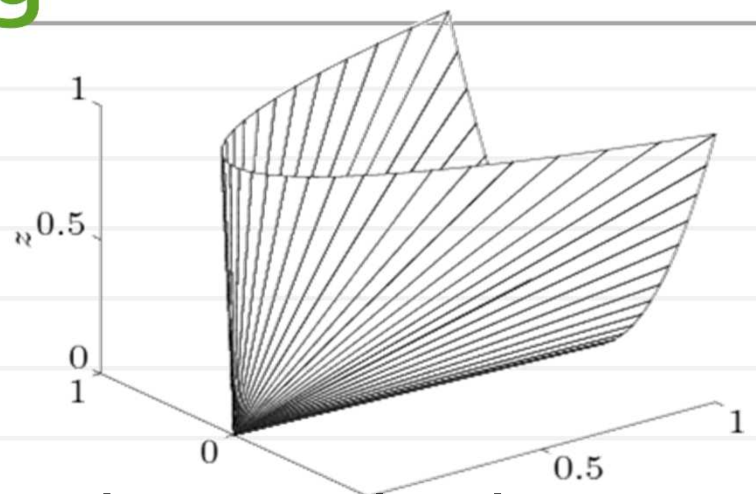
$$\sum_{i=1}^{N} p_i \times F_i + T_{ext} = 0$$

Semidefinite programming

$$\min c^T x$$

s.t.:

$$x_1 F_1 + \ldots + x_n F_n + G \succeq 0$$



- Multiple SDP constraints can be stacked in a block-diagonal way
- E.g. expressing the linear constraints in SDP:

$$Ax + b \geq 0 \qquad \begin{bmatrix} a_1^T x + b_1 & & & 0 \\ & a_2^T x + b_2 & & \\ & & \ddots & \\ 0 & & & a_n^T x + b_n \end{bmatrix} \succeq 0$$

$$x_1 \operatorname{diag}(A^{(1)}) + x_2 \operatorname{diag}(A^{(2)}) + \ldots + x_n \operatorname{diag}(A^{(n)}) + \operatorname{diag}(b) \succeq 0$$

Example: non-linear SDP

$$\min \frac{(c^T x)^2}{d^T x}$$

$$\text{s.t.: } Ax + b \geq 0$$

$$\min t$$

$$\text{s.t.: } Ax + b \geq 0$$

$$\frac{(c^T x)^2}{d^T x} \leq t$$

$$\min t$$

s.t.:

$$\begin{bmatrix} \operatorname{diag}(Ax+b) & 0 & 0 \\ 0 & t & c^T x \\ 0 & c^T x & d^T x \end{bmatrix} \succeq 0$$

Example: expressing SOCP as SDP

$$\|Ax+b\|_2 \leq c^T x + d$$

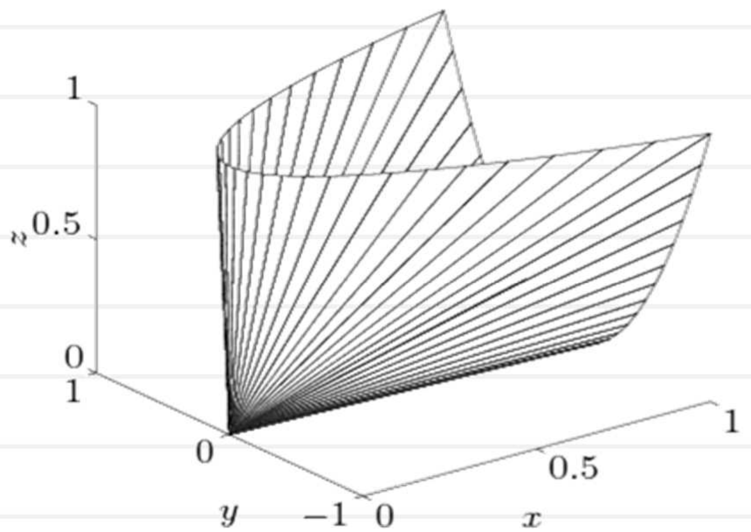
$$\begin{pmatrix} c^Tx+d & a_1^Tx+b_1 & \cdots & a_n^Tx+b_n \\ a_1^Tx+b_1 & c^Tx+d & & 0 \\ \vdots & & \ddots & \\ a_n^Tx+b_n & 0 & & c^Tx+d \end{pmatrix} \succeq 0$$

Semidefinite programming

$\min \ c^T x$

$\text{s.t.:} \quad Ax = b$

$x_1 F_1 + \dots + x_n F_n + G \succeq 0$



Applications:

- structural design
- geometric problems (e.g. finding optimal ellipsoids)
- relaxation of binary problems (coming tomorrow)

Nested doll of Convex Optimization