

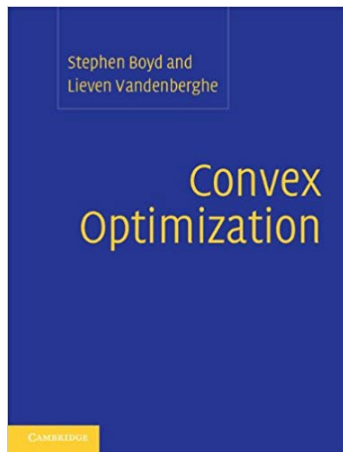
**CS495**  
**Optimiztaion**

Waleed A. Yousef, Ph.D.,  
Human Computer Interaction Lab.,  
Computer Science Department,  
Faculty of Computers and Information,  
Helwan University,  
Egypt.

February 28, 2019

Lectures follow:

Boyd and Vandenberghe (2004)



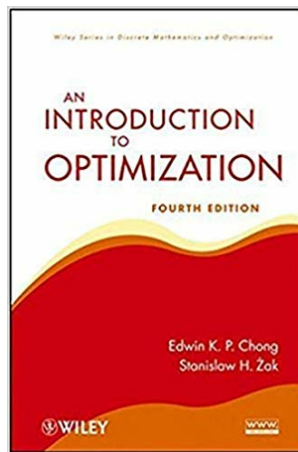
Boyd, S., & Vandenberghe, L. (2004). Convex Optimization. Cambridge: Cambridge University Press.

Book and Stanford course:

<http://web.stanford.edu/~boyd/cvxbook/>

Some examples from:

Chong and Zak (2001)



Chong, E. K., & Zak, S. (2001). An introduction to optimization: Wiley-Interscience.

# Course Objectives

- Developing rigorous mathematical treatment for mathematical optimization.
- Building intuition, in particular to practical problems.
- Developing computer practice to using optimization SW.

## Prerequisites

Calculus (both single and multivariable) and Linear Algebra.

# **Chapter 1:**

## **Introduction**

### **Snapshot on Optimization**

# Contents

<b>Contents</b>		<b>iv</b>
<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Mathematical Optimization	2
1.1.1	Motivation and Applications	4
1.1.2	Solving Optimization Problems	6
1.2	Least-Squares and Linear Programming	7
1.2.1	Least-Squares Problems	7
1.2.2	Linear Programming	8
1.3	Convex Optimization	9
1.4	Nonlinear Optimization	10

<b>I</b>	<b>Theory</b>	<b>12</b>
<b>2</b>	<b>Convex sets</b>	<b>13</b>
2.1	Affine and convex sets	14
2.1.1	Lines and line segments	14
2.1.2	Affine sets	15
2.1.3	Affine dimension and relative interior	17
2.1.4	Convex sets	18
2.1.5	Cones	19

<b>3</b>	<b>Convex functions</b>	<b>46</b>
3.1	Basic properties and examples	47
3.2	Operations that preserve convexity	48
3.3	The conjugate function	49
3.4	Quasiconvex functions	50
3.5	Log-concave and log-convex functions	51
3.6	Convexity with respect to generalized inequalities	52
<b>4</b>	<b>Convex optimization problems</b>	<b>67</b>
4.1	Optimization problems	68
4.2	Convex optimization	69
4.3	Linear optimization problems	70
4.4	Quadratic optimization problems	71
4.5	Geometric programming	72
4.6	Generalized inequality constraints	73
4.7	Vector optimization	74
<b>5</b>	<b>Duality</b>	<b>89</b>
5.1	The Lagrange dual function	90
5.2	The Lagrange dual problem	91
5.3	Geometric interpretation	92
5.4	Saddle-point interpretation	93
5.5	Optimality conditions	94
5.6	Perturbation and sensitivity analysis	95
5.7	Examples	96
5.8	Theorems of alternatives	97
5.9	Generalized inequalities	98
<b>II</b>	<b>Applications</b>	<b>109</b>
<b>6</b>	<b>Approximation and fitting</b>	<b>110</b>
6.1	Norm approximation	111
6.2	Least-norm problems	112
6.3	Regularized approximation	113
6.4	Robust approximation	114
6.5	Function fitting and interpolation	115

<b>7</b>	<b>Statistical estimation</b>	<b>142</b>
7.1	Parametric distribution estimation	143
7.2	Nonparametric distribution estimation	144
7.3	Optimal detector design and hypothesis testing	145
7.4	Chebyshev and Chernoff bounds	146
7.5	Experiment design	147
<b>8</b>	<b>Geometric problems</b>	<b>160</b>
8.1	Projection on a set	161
8.2	Distance between sets	162
8.3	Euclidean distance and angle problems	163
8.4	Extremal volume ellipsoids	164
8.5	Centering	165
8.6	Classification	166
8.7	Placement and location	167
8.8	Floor planning	168
<b>III Algorithms</b>		<b>191</b>
<b>9</b>	<b>Unconstrained minimization</b>	<b>192</b>
9.1	Unconstrained minimization	193
9.2	Descent methods	194
9.3	Gradient descent method	195
9.4	Steepest descent method	196
9.5	Newton's method	197
9.6	Self-concordance	198
9.7	Implementation	199
<b>10</b>	<b>Equality constrained minimization</b>	<b>226</b>
10.1	Equality constrained minimization problems	227
10.2	Newton's method with equality constraints	228
10.3	Infeasible start Newton method	229
10.4	Implementation	230
<b>11</b>	<b>Interior-point methods</b>	<b>239</b>
11.1	Inequality constrained minimization problems	240
11.2	Logarithmic barrier function and central path	241

11.3	The barrier method . . . . .	242
11.4	Feasibility and phase I methods . . . . .	243
11.5	Complexity analysis via self-concordance . . . . .	244
11.6	Problems with generalized inequalities . . . . .	245
11.7	Primal-dual interior-point methods . . . . .	246
11.8	Implementation . . . . .	247

## **Bibliography**



# **Chapter 1**

## **Introduction**

# 1.1 Mathematical Optimization

**Definition 1** A mathematical optimization problem or just optimization problem, has the form (Boyd and Vandenberghe, 2004):

$$\begin{aligned} &\underset{x}{\text{minimize}} && f_0(x) \\ &\text{subject to:} && f_i(x) \leq 0, && i = 1, \dots, m \\ &&& h_i(x) = 0, && i = 1, \dots, p, \end{aligned}$$

$x = (x_1, \dots, x_n) \in \mathbf{R}^n$ , (optimization variable)

$f_0: \mathbf{R}^n \mapsto \mathbf{R}$ , (objective (cost/utility) function)

$f_i: \mathbf{R}^n \mapsto \mathbf{R}$ , (inequality constraints (functions))

$h_i: \mathbf{R}^n \mapsto \mathbf{R}$ , (equality constraints (functions))

$$\mathcal{D}: \bigcap_{i=1}^m \text{dom } f_i \cap \bigcap_{i=1}^p \text{dom } h_i \quad (\text{feasible set})$$

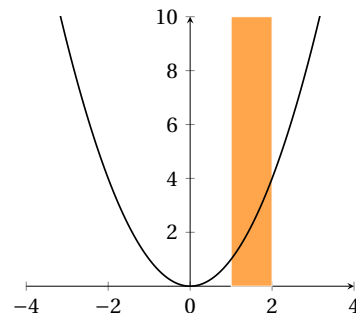
$$= \{x \mid x \in \mathbf{R}^n \wedge f_i(x) \leq 0 \wedge h_i(x) = 0\}$$

$$x^*: \{x \mid x \in \mathcal{D} \wedge f_0(x) \leq f_0(z) \forall z \in \mathcal{D}\} \quad (\text{solution})$$

- minimize  $f_0 \equiv \text{maximize } -f_0$ .
- $f_i \leq 0 \equiv -f_i \geq 0$ .
- 0s can be replaced of course by constants  $b_i, c_i$
- unconstrained problem when  $m = p = 0$ .

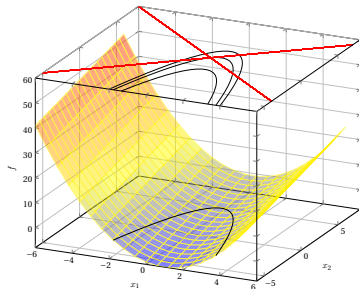
**Example 2 :**

$$\begin{aligned} &\underset{x}{\text{minimize}} && x^2 \\ &\text{subject to:} && x \leq 2 \wedge x \geq 1. \end{aligned}$$



$$x^* = 1.$$

If the constraints are relaxed, then  $x^* = 0$ .



$$\begin{aligned} &\underset{x}{\text{minimize}} && f_0(x) \\ &\text{subject to:} && f_i(x) \leq 0, && i = 1, \dots, m \\ &&& h_i(x) = 0, && i = 1, \dots, p, \end{aligned}$$

$x = (x_1, \dots, x_n) \in \mathbf{R}^n$ , (optimization variable)

$f_0: \mathbf{R}^n \mapsto \mathbf{R}$ , (objective (cost/utility) function)

$f_i: \mathbf{R}^n \mapsto \mathbf{R}$ , (inequality constraints (functions))

$h_i: \mathbf{R}^n \mapsto \mathbf{R}$ , (equality constraints (functions))

$$\mathcal{D}: \bigcap_{i=1}^m \text{dom } f_i \cap \bigcap_{i=1}^p \text{dom } h_i \quad (\text{feasible set})$$

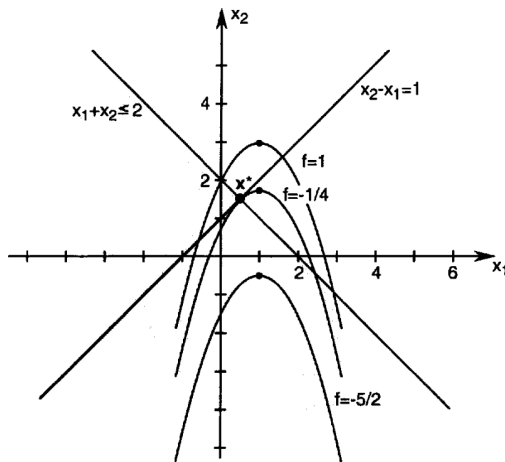
$$= \{x \mid x \in \mathbf{R}^n \wedge f_i(x) \leq 0 \wedge h_i(x) = 0\}$$

$$x^*: \{x \mid x \in \mathcal{D} \wedge f_0(x) \leq f_0(z) \forall z \in \mathcal{D}\} \quad (\text{solution})$$

**Example 3** (*Chong and Zak, 2001, Ex. 20.1, P. 454*):

$$\begin{aligned} &\underset{x}{\text{minimize}} && (x_1 - 1)^2 + x_2 - 2 \\ &\text{subject to:} && x_2 - x_1 = 1 \\ &&& x_1 + x_2 \leq 2. \end{aligned}$$

No global minimizer:  $\partial z / \partial x_2 = 1 \neq 0$ . However,  $z|_{(x_2 - x_1 = 1)} = (x_1 - 1)^2 + (x_1 - 1)$ , which attains a minimum at  $x_1 = 1/2$ .



$x^* = (1/2, 3/2)'$ . (Let's see animation)

### 1.1.1 Motivation and Applications

- *optimization problem* is an abstraction of how to make “best” possible choice of  $x \in \mathbf{R}^n$ .
- *constrains* represent trim requirements or specifications that limit the possible choices.
- *objective function* represents the *cost* to minimize or the *utility* to maximize for each  $x$ .

#### Examples:

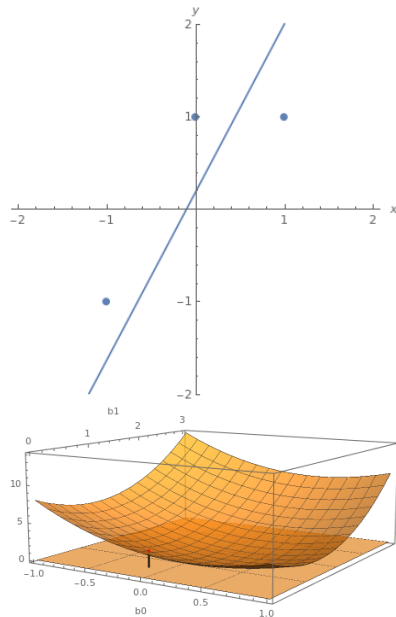
	<i>Any problem</i>	<i>Portfolio Optimization</i>	<i>Device Sizing</i>	<i>Data Science</i>
$x \in \mathbf{R}^n$	choice made	investment in capitals	dimensions	parameters
$f_i, h_i$	firm requirements /conditions	overall budget	engineering constraints	regularizer
$f_0$	cost (or utility)	overall risk	power consumption	error

- Amazing variety of practical problems. In particular, data science: two sub-fields: construction and assessment.
- The construction of: Least Mean Square (LMS), Logistic Regression (LR), Support Vector Machines (SVM), Neural Networks(NN), Deep Neural Networks (DNN), etc.
- Many techniques are for solving the optimization problem:
  - Closed form solutions: convex optimization problems
  - Numerical solutions: Newton’s methods, Gradient methods, Gradient descent, etc.
  - “Intelligent” methods: particle swarm optimization, genetic algorithms, etc.

#### Example 4 (Machine Learning: construction) :

Let's suppose that the best regression function is  $Y = \beta_0 + \beta_1 X$ , then for the training dataset  $(x_i, y_i)$  we need to minimize the MSE.

$$\underset{\beta_0, \beta_1}{\text{minimize}} \sum_i (\beta_0 + \beta_1 x_i - y_i)^2$$



- Half of ML field is construction: NN, SVM, etc.
- In DNN it is an optimization problem of millions of parameters.
- Let's see animation.
- Where are Probability, Statistics, and Linear Algebra here? Let's re-visit the chart.
- Is the optimization problem solvable:
  - closed form? (LSM)
  - numerically and guaranteed? (convex and linear)
  - numerically but not guaranteed? (non-convex):
    - \* numerical algorithms, e.g., GD,
    - \* local optimization,
    - \* heuristics, swarm, and genetics,
    - \* brute-force with exhaustive search

### 1.1.2 Solving Optimization Problems

- A *solution method* for a class of optimization problems is an algorithm that computes a solution.
- Even when the *objective function* and constraints are smooth, e.g., polynomials, the solution is very difficult.
- There are three classes where solutions exist, theory is very well developed, and amazingly found in many practical problems:

Linear  $\subset$  Quadratic  $\subset$  Convex  $\subset$  Non-linear (not linear and not known to be convex!)

- For the first three classes, the problem can be solved very reliably in hundreds or thousands of variables!

## 1.2 Least-Squares and Linear Programming

### 1.2.1 Least-Squares Problems

A *least-squares* problem is an optimization problem with no constraints (i.e.,  $m = p = 0$ ), and an objective in the form:

$$\underset{x}{\text{minimize}} f_0(x) = \sum_{i=1}^k (a'_i x - b_i)^2 = \|A_{k \times n} x_{n \times 1} - b_{k \times 1}\|^2.$$

The solution is given in **closed form** by:

$$x = (A' A)^{-1} A' b$$

- Good algorithms in many SC SW exist; it is a very mature technology.
- Solution time is  $O(n^2 k)$ .
- Easily solvable even for hundreds or thousands of variables.
- More on that in the Linear Algebra course.
- Many other problems reduce to typical LS problem:
  - Weighted LS (to emphasize some observations)

$$\underset{x}{\text{minimize}} f_0(x) = \sum_{i=1}^k w_i (a'_i x - b_i)^2.$$

- Regularization (to penalize for over-fitting)

$$\underset{x}{\text{minimize}} f_0(x) = \sum_{i=1}^k (a'_i x - b_i)^2 + \rho \sum_{j=1}^n x_j^2.$$

## 1.2.2 Linear Programming

A *linear programming* problem is an optimization problem with objective and all constraint functions are linear:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) = C'x \\ \text{subject to:} & a'_i x \leq b_i, \quad i = 1, \dots, m \\ & h'_i x = g_i, \quad i = 1, \dots, p, \end{array}$$

- **No** closed form solution as opposed to LS.
- Very robust, reliable, and effective set of methods for numerical solution; e.g., Dantzig's simplex, and interior point.
- Complexity is  $\simeq O(n^2 m)$ .
- Similar to LS, we can solve a problem of thousands of variables.
- Example is *Chebyshev minimization* problem:

$$\underset{x}{\text{minimize}} f_0(x) = \max_{i=1, \dots, k} |a'_i x - b_i|,$$

- The objective is different from the LS: minimize the maximum error. **Ex:**
- After some tricks, requiring familiarity with optimization, it is equivalent to a LP:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & t \\ \text{subject to:} & a'_i x - t \leq b_i, \quad i = 1, \dots, k \\ & -a'_i x - t \leq -b_i, \quad i = 1, \dots, k \end{array}$$



## 1.3 Convex Optimization

A *convex optimization* problem is an optimization problem with objective and all constraint function are convex:

$$\begin{array}{llll} \underset{x}{\text{minimize}} & f_0(x) & & \\ \text{subject to:} & f_i(x) \leq 0, & i = 1, \dots, m & \\ & h_i(x) = 0, & i = 1, \dots, p, & \\ & f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), & \alpha + \beta = 1, & 0 \leq \alpha, 0 \leq \beta, \quad 0 \leq i \leq m \\ & h_i(x) = a'_i x + b_i & & 0 \leq i \leq p \end{array}$$

- The LP and LS are special cases; however, only LS has closed-form solution.
- Very robust, reliable, and effective set of methods, including *interior point methods*.
- Complexity is almost:  $O(\max(n^3, n^2 m, F))$ , where  $F$  is the cost of evaluating 1st and 2nd derivatives of  $f_i$  and  $h_i$ .
- Similar to LS and LP, we can solve a problem of thousands of variables.
- However, it is not as very mature technology as the LP and LS yet.
- There are many practical problems that can be re-formulated as convex problem **BUT** requires mathematical skills; but once done the problem is solved. **Hint:** realizing that the problem is convex requires more mathematical maturity than those required for LP and LS.

## 1.4 Nonlinear Optimization

A *non-linear optimization* problem is an optimization problem with objective and constraint functions are non-linear **BUT** not known to be convex (**so far**). Even simple-looking problems in 10 variables can be extremely challenging. Several approaches for solutions:

**Local Optimization** : starting at initial point in space, using differentiability, then navigate

- does not guarantee global optimal.
- affected heavily by initial point.
- depends heavily on numerical algorithm and their parameters.
- More art than technology.
- In contrast to convex optimization, where a lot of art and mathematical skills are required to formulate the problem as convex; then numerical solution is straightforward.

**Global Optimization** : the true global solution is found; the compromise is complexity.

- The complexity goes exponential with dimensions.
- Sometimes it is worth it when: the cost is huge, not in real time, and dimensionality is low.

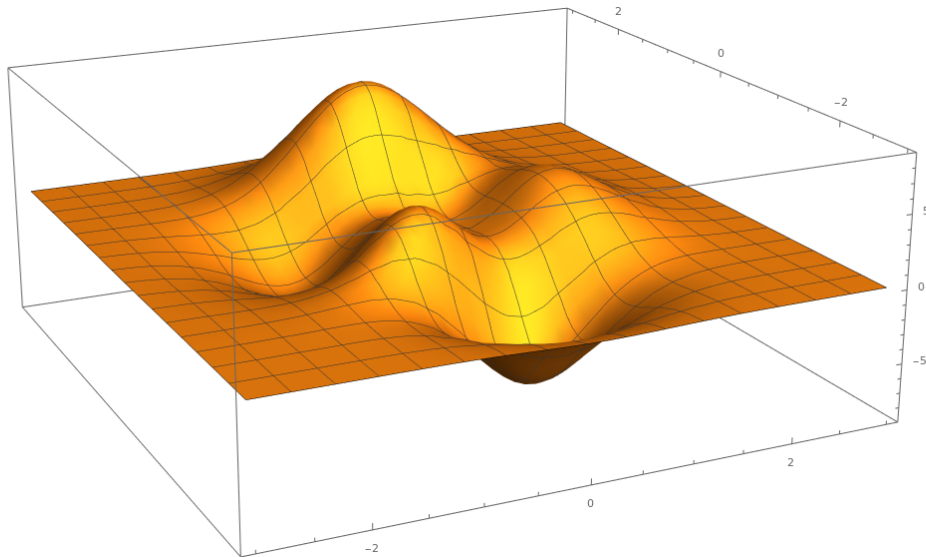
**Role of Convex Optimization** :

- Approximate the non-linear function to a convex one, finding the exact solution, then using it as a starting point for the original problem. (Also does not guarantee optimality)
- Setting bounds on the global solution.

**Evolutionary Computations** : Genetic Algorithm (GA), Simulated Annealing (SA), Particle Swarm Optimization (PSO), etc.

**Example 5 (Nonlinear Objective Function)** : (*Chong and Zak, 2001, Ex. 14.3*)

$$f(x, y) = 3(1 - x)^2 e^{-x^2 - (y+1)^2} - 10e^{-x^2 - y^2} \left( -x^3 + \frac{x}{5} - y^5 \right) - \frac{1}{3} e^{-(x+1)^2 - y^2}$$



# **Part I**

# **Theory**

## **Chapter 2**

# **Convex sets**

## 2.1 Affine and convex sets

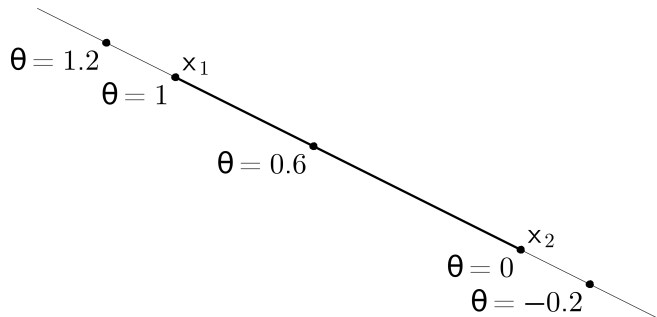
### 2.1.1 Lines and line segments

**Definition 6 (line and line segment)** Suppose  $x_1 \neq x_2 \in \mathbf{R}^n$ . Points of the form

$$\begin{aligned}y &= \theta x_1 + (1 - \theta)x_2 \\ &= x_2 + \theta(x_1 - x_2),\end{aligned}$$

where  $\theta \in \mathbf{R}$ , form the line passing through  $x_1$  and  $x_2$ .

- As usual, this is a definition for high dimensions taken from a proof for  $n \leq 3$ .
- We have done it many times: angle, norm, cardinality of sets, etc.
- if  $0 \leq \theta \leq 1$  this forms a line segment.



### 2.1.2 Affine sets

**Definition 7 (Affine sets)** A set  $C \subset \mathbf{R}^n$  is affine if the line through any two distinct points in  $C$  lies in  $C$ . I.e.,  $\forall x_1, x_2 \in C$  and  $\theta \in \mathbf{R}$ , we have  $\theta x_1 + (1 - \theta)x_2 \in \mathbf{R}^n$ . In other words,  $C$  contains any linear combination of any two points in  $C$  provided the coefficients sum to one.

**Examples:** what about line, line segment, circle, disk, strip?

**Corollary 8** Suppose  $C$  is an affine set, and  $x_1, \dots, x_k \in C$ , then  $C$  contains every general affine combination of the form  $\theta_1 x_1 + \dots + \theta_k x_k$ , where  $\theta_1 + \dots + \theta_k = 1$ .

**Proof.** trivial by induction or by summation (same) ■

**Definition 9 (Subspace from Linear Algebra)** a set  $V \subset \mathbf{R}^n$  of vector (here points) is a subspace if it is closed under sums and scalar multiplication. I.e.,  $\forall v_1, v_2 \in V$  and  $\forall \alpha, \beta \in \mathbf{R}$  we have  $\alpha v_1 + \beta v_2 \in V$ . Hint: Vimp this implies that  $\mathbf{0} \in V$ .

**Corollary 10** If  $C$  is affine set and  $x_0 \in C$ , then the set  $V = C - x_0 = \{x - x_0 | x \in C\}$  is a subspace; and then the dimension (rank) of  $C$  is defined to be the same as the dimension (rank) of  $V$ .

**Proof.** Suppose  $v_1, v_2 \in V$ ; then  $v_1 + x_0, v_2 + x_0 \in C$ .

$$\begin{aligned} c &= \alpha v_1 + \beta v_2 + x_0 \\ &= \alpha(v_1 + x_0) + \beta(v_2 + x_0) + (1 - \alpha - \beta)x_0 \in C \end{aligned}$$

Then  $c - x_0 = \alpha v_1 + \beta v_2 \in V$ ; and hence  $V$  is a subspace. ■

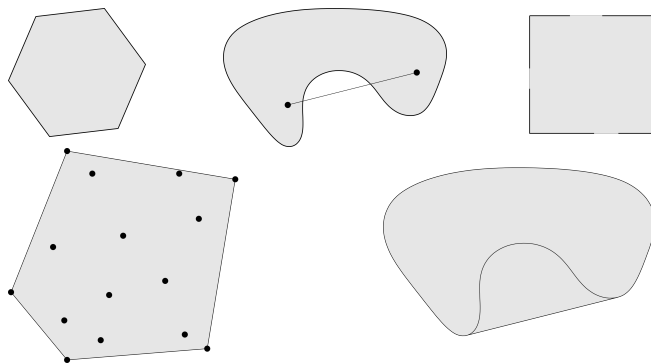
- This is true for any  $x_0$ .
- *affine* is a *subspace* plus offset.
- every subspace is affine but not the vice versa.
- subspace is a special case of affine.

### **Example 11**



### **2.1.3 Affine dimension and relative interior**

### 2.1.4 Convex sets



**2.1.5 Cones**

# **Part II**

# **Applications**

# **Part III**

# **Algorithms**

# Bibliography

Boyd, S. and Vandenberghe, L. (2004), *Convex Optimization*, Cambridge: Cambridge University Press.

Chong, E. K. and Zak, Stanislaw, H. (2001), *An Introduction to Optimization*, Wiley-Interscience, 4th ed.