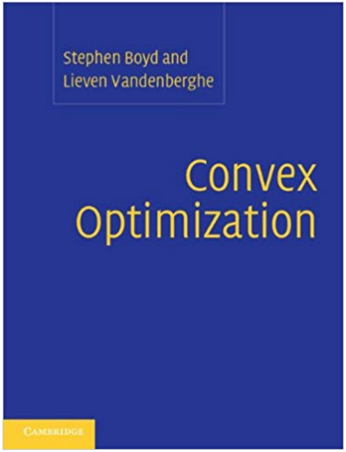
# CS495
# Optimiztaion

Waleed A. Yousef, Ph.D.,

Human Computer Interaction Lab.,
Computer Science Department,
Faculty of Computers and Information,
Helwan University,
Egypt.

March 3, 2019

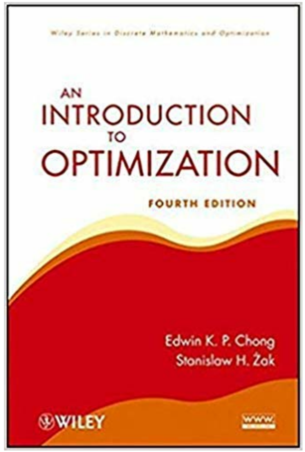
Lectures follow:
Boyd and Vandenberghe (2004)




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

# Acknowledgment

Figures from Boyd and Vandenberghe (2004) are extracted from the PDF version of the book[1], with permission from the authors.

[1] http://web.stanford.edu/~boyd/cvxbook/

# Contents

# Chapter 1

# Introduction

## 1.1 Mathematical Optimization

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

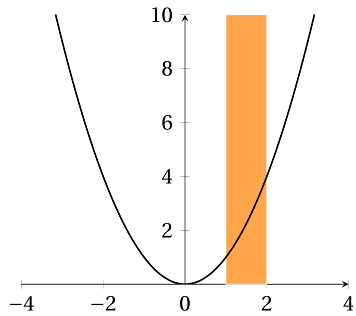
$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) & \\ \text{subject to:} \quad & f_i(x) \le 0, & i = 1, \ldots, m \\ & h_i(x) = 0, & i = 1, \ldots, p, \end{aligned}$$

$$\begin{aligned} x = (x_1, \ldots, x_n) \in \mathbf{R}^n, & \quad (\textit{optimization variable}) \\ f_0 : \mathbf{R}^n \mapsto \mathbf{R}, & \quad (\textit{objective (cost/utility) function}) \\ f_i : \mathbf{R}^n \mapsto \mathbf{R}, & \quad (\textit{inequality constraints (functions)}) \\ h_i : \mathbf{R}^n \mapsto \mathbf{R}, & \quad (\textit{equality constraints (functions)}) \\ \mathcal{D} : \bigcap_{i=1}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i & \quad (\textit{feasible set}) \\ = \{x \mid x \in \mathbf{R}^n \wedge f_i(x) \le 0 \wedge h_i(x) = 0\} & \\ x^* : \{x \mid x \in \mathcal{D} \wedge f_0(x) \le f_0(z)\ \forall z \in \mathcal{D}\} & \quad (\textit{solution}) \end{aligned}$$

- minimize $f_0 \equiv$ maximize $-f_0$.
- $f_i \le 0 \equiv -f_i \ge 0$.
- 0s can be replaced of course by constants $b_i$, $c_i$
- unconstrained problem when $m = p = 0$.

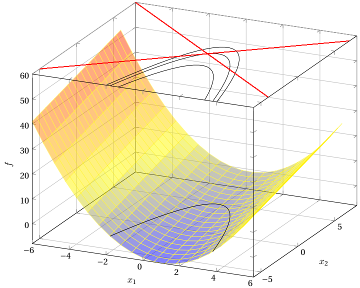
**Example 2** *:*

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




$x^* = 1$.

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



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

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

$$\mathcal{D} : \bigcap_{i=1}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i \qquad (\textit{feasible set})$$

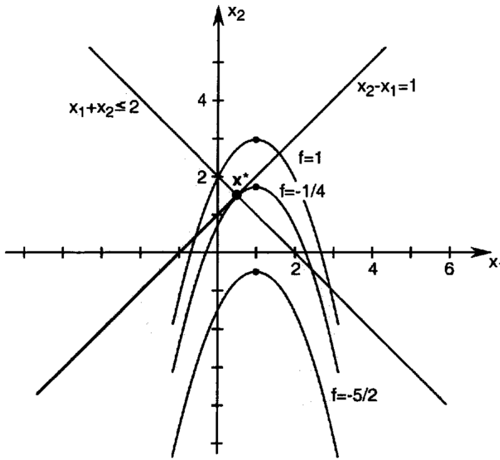
$$= \{x \mid x \in \mathbf{R}^n \wedge f_i(x) \le 0 \wedge h_i(x) = 0\}$$

$$x^* : \{x \mid x \in \mathcal{D} \wedge f_0(x) \le f_0(z)\ \forall z \in \mathcal{D}\} \qquad (\textit{solution})$$

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

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

*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 minima 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:**

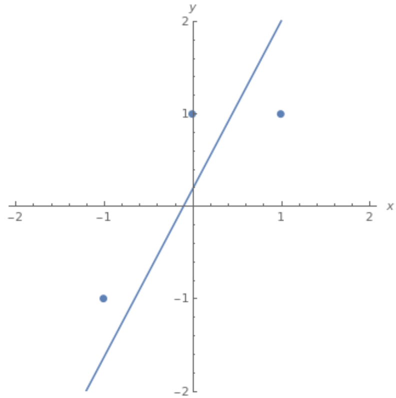
| | *Any problem* | *Portfolio Optimization* | *Device Sizing* | *Data Science* |
|---|---|---|---|---|
| $x \in \mathbf{R}^n$ | choice made | investment in capitals | dimensions | parameters |
| $f_i,\ h_i$ | firm requirements /conditions | overall budget | engineering constraints | regularizer |
|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

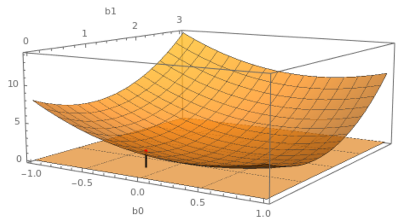
- Amazing variety of practical problems. In particular, data science: two sub-fields: construction and assessment.
- The construction of: Least Mean Square (LMS), Logistic Regression (LR), Support Vector Machines (SVM), Neural Networks(NN), Deep Neural Networks (DNN), etc.
- Many techniques are for solving 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.*

- 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

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



### 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:

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

- 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:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) = C'x \\ \text{subject to:} \quad & a_i'x \le b_i, \qquad i = 1, \dots, m \\ & h_i'x = g_i, \qquad i = 1, \dots, p, \end{aligned}$$

- **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 $\simeq \mathcal{O}(n^2 m)$.
- 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:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & t \\ \text{subject to:} \quad & a_i'x - t \le b_i, \qquad i = 1, \dots, k \\ & -a_i'x - t \le -b_i, \qquad i = 1, \dots, k \end{aligned}$$

## 1.3 Convex Optimization

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

$$
\begin{aligned}
\underset{x}{\text{minimize}} \quad & f_0(x) \\
\text{subject to:} \quad & f_i(x) \le 0, && && i = 1, \ldots, m \\
& h_i(x) = 0, && && i = 1, \ldots, p, \\
& f_i(\alpha x + \beta y) \le \alpha f_i(x) + \beta f_i(y), && \alpha + \beta = 1, && 0 \le \alpha,\ 0 \le \beta, && 0 \le i \le m \\
& h_i(x) = a_i' x + b_i && && 0 \le i \le p
\end{aligned}
$$

- 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: $O\big(\max(n^3, n^2 m, F)\big)$, 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 differentiablity, 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.

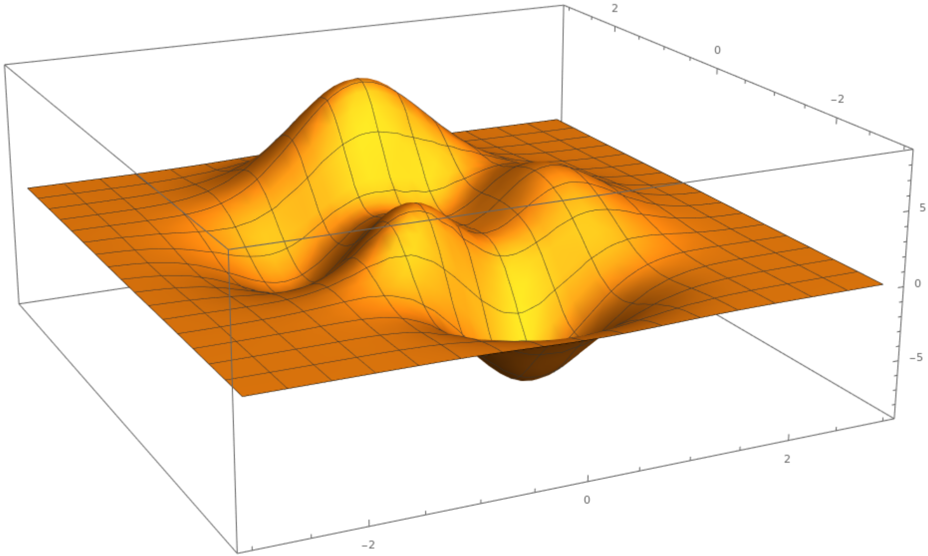
**Role of Convex Optimization** :

- 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

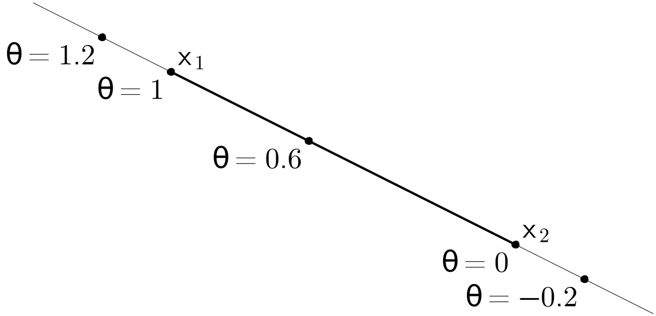
## 2.1.1 Lines and line segments

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

$$y = \theta x_1 + (1-\theta)x_2$$
$$= x_2 + \theta(x_1 - x_2),$$

*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 (summing to one) of any two points in $C$.*

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

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

**Wrong Proof.** Suppose $y_1, y_2 \in C$, then

$$x = \sum_{i=1}^{k} \theta_i x_i = \sum_{i=1}^{k} \theta_i (\alpha_i y_1 + (1-\alpha_i) y_2);$$

and the summation of the coefficients will be

$$\sum_{i=1}^{k} \theta_i \alpha_i + \sum_{i=1}^{k} \theta_i (1-\alpha_i) = \sum_{i=1}^{k} \theta_i (\alpha_i + 1 - \alpha_i) = \sum_{i=1}^{k} \theta_i = 1.$$

Where is the bug? ■

**Correct Proof.** **base:** $k = 3$.

$$\begin{aligned} x &= \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \\ &= (1-\theta_3)\left(\frac{\theta_1}{1-\theta_3} x_1 + \frac{\theta_2}{1-\theta_3} x_2\right) + \theta_3 x_3. \\ &= (1-\theta_3)(\cdot \in C) + \theta_3 (\cdot \in C). \end{aligned}$$

**induction:** suppose it is true for some $k \geq 3$; i.e., $\sum_{i=1}^{k} \theta_i x_i \in C$ when $\sum_{i=1}^{k} \theta_i = 1$ . Then

$$\begin{aligned} x &= \sum_{i=1}^{k+1} \theta_i x_i \\ &= \sum_{i=1}^{k} \theta_i x_i + \theta_{k+1} x_{k+1} \\ &= (1-\theta_{k+1}) \sum_{i=1}^{k} \theta_i / (1-\theta_{k+1}) x_i + \theta_{k+1} x_{k+1} \\ &= (1-\theta_{k+1})(\cdot \in C) + \theta_{k+1} (\cdot \in C), \\ &\qquad \text{(from the induction hypothesis)} \end{aligned}$$

which completes the proof. ■

**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 $\alpha v_1 + \beta v_2 \in V$.*

**Remember:**

- $\alpha + \beta$ not necessarily equals 1
- $\alpha = 0, \beta = 0 \to \mathbf{0} \in V$.
- Any subspace $V$ is the solution set of $A_{m\times n} x_{n\times 1} = 0$, which is $\mathcal{N}(A)$ (the null space of $A$). Geometry?
- $\dim(V)$ is min. number of vectors to express any $v \in V$.
- $\mathbf{rank}(A) = \dim(V) + m$. (for $m < n$)

**Corollary 10** .

1. *If $C$ is affine, then $V = C - x_0 = \{x - x_0 | x, x_0 \in C\}$ is a subspace.*
2. *If $V$ is a subspace, then $C = V + x_0 = \{x + x_0 | x \in V\}$ is affine.*
3. *An affine set $C$ can be represented as the solution set of a nonhomogeneous linear system $Ax = b$, where $V = C - x_0$ is $\mathcal{N}(A)$.*
4. *The solution set of any nonhomogeneous system is an affine set. (Ex. 2.1)*

**Proof.**

1. Suppose $x_1, x_2, x_0 \in C$, an affine set. Both $x_1 - x_0$ and $x_2 - x_0$, by construction, $\in V$; then

$$\begin{aligned} x &= \alpha(x_1 - x_0) + \beta(x_2 - x_0) + x_0 \\ &= \alpha x_1 + \beta x_2 + (1 - \alpha - \beta) x_0 \in C \end{aligned}$$

   Then $x - x_0 = \alpha(x_1 - x_0) + \beta(x_2 - x_0) \in V$; hence $V$ is a subspace.

2. Suppose $x_1, x_2 \in V$, a subspace. Both $x_1 + x_0$ and $x_2 + x_0$, by construction, $\in C$; then

$$\begin{aligned} x &= \theta(x_1 + x_0) + (1 - \theta)(x_2 + x_0) \\ &= \theta x_1 + (1 - \theta) x_2 + x_0 = (\cdot \in V) + x_0 \in C \end{aligned}$$

3. If $C$ is affine and $x_0 \in C$, then

$$\begin{aligned} C - x_0 &= \{x | Ax = 0\} && \text{(a subspace)} \\ C &= \{x + x_0 | A(x + x_0) = Ax_0\} \\ C &= \{c | Ac = b\}. \end{aligned}$$

4. $C = \{x | Ax = b\}$; if $x_0 \in C$ then $Ax_0 = b$ and

$$C - x_0 = \{x - x_0 | A(x - x_0) = b - Ax_0 = 0\}.$$

   Hence, $C - x_0$ is a subspace and $C$ is affine.

■

**Proof of the book:** Suppose $x_1, x_2 \in C$, where $C = \{x | Ax = b\}$. Then

$$A(\theta x_1 + (1-\theta)x_2) = \theta Ax_1 + (1-\theta)Ax_2 = \theta b + (1-\theta)b = b.$$

which means $\theta x_1 + (1-\theta)x_2 \in C$ as well.

- The dimension of *affine* is defined to be the dimension of the associate *subspace.*
- *affine* is a *subspace* plus offset.
- every subspace is affine but not the vice versa.
- subspace is a special case of affine.

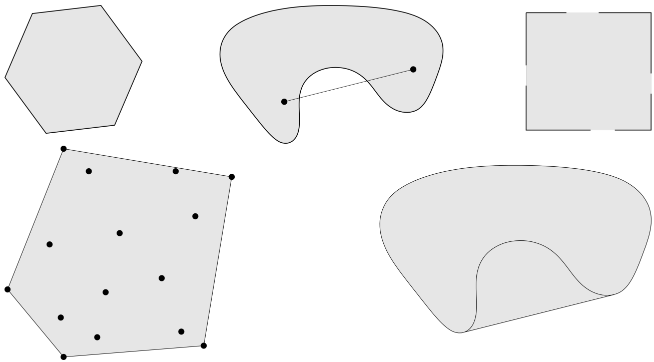
**Definition 11 (affine hull)** *The set of all affine combinations of point in some set $C$ (not necessarily affine) is called the affine hull aff$C$.*

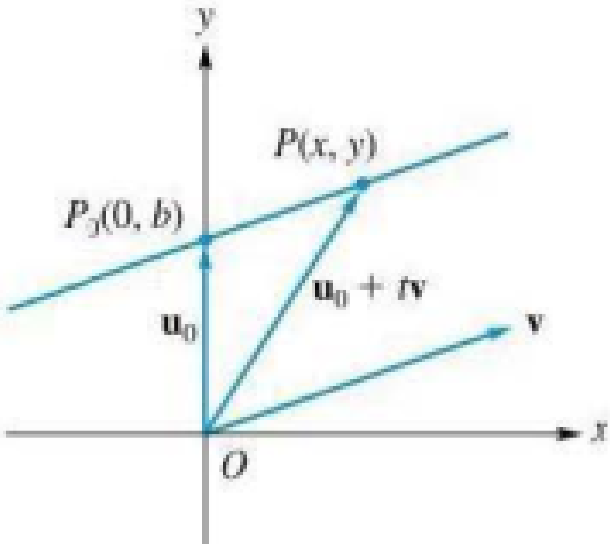
$$affdomC = \{\sum_{i=1}^{k} \theta_i x_i | x_i \in C, \sum_i \theta_i = 1\}.$$

**Example 12** *Construct*

### 2.1.3 Affine dimension and relative interior

### 2.1.4 Convex sets





y
P(x, y)
P₁(0, b)
u₀ + tv
u₀
v
x
O

### 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.