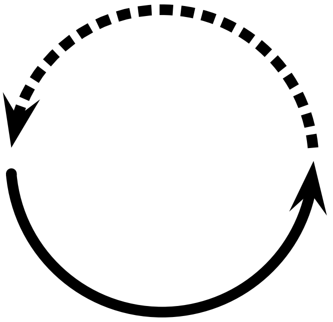
Milan Hladík

Discrete and Continuous Optimization

textbook

May 17, 2022



This is a textbook to course *Discrete and continuous optimization* for the Computer Science program at Charles University, Faculty of Mathematics and Physics. In particular, it serves for the continuous optimization part.

I build mostly on books Bazaraa et al. [2006]; Boyd and Vandenberghe [2004]; Luenberger and Ye [2008].

You can report bugs or send your comments to: hladik@kam.mff.cuni.cz.

Contents

Chapter 1

Introduction

An optimization problem (or a mathematical programming problem) reads

$$\min\ f(x)\ \text{ subject to }\ x \in M,$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is *the objective function* and $M \subseteq \mathbb{R}^n$ is *the feasible set.* In general, this problem is undecidable, that is, there is provably no algorithm that can solve it [Zhu, 2006]; another well-known undecidable problem is *the halting problem.* On the other hand, there are effectively solvable sub-classes of problems.

Depending on the character of the feasible set M, we distinguish two types:

- **Discrete optimization.**

  The set M is (typically) finite, but usually large enough to inspect and process all feasible solutions. Usually $|M| \geq 2^n$.

  Examples include the shortest path problem, the minimum spanning tree problem or the minimum matching problem in a graph. These problems are effectively solvable. In contrast, some problems in discrete optimization are NP-hard: integer linear programming, the travelling salesman problem, the knapsack problem or finding the max cut in a graph.

- **Continuous optimization.**

  Here, the feasible set M is uncountably infinite. Surprisingly, this may pay off: linear programming is polynomially solvable, but the additional integrality requirement makes it NP-hard.

  The typical problems are linear programming (LP) and diverse kinds of nonlinear programming (such as convex programming, quadratic programming or semidefinite programming).

Relation discrete vs. continuous optimization

Discrete and continuous optimization are not disjoint. In fact, they are closely related and techniques from one area are used in the second one. To see it, consider *integer programming*: most of the methods are based on a relaxation to a continuous problem and an iterative improvement.

Conversely, an integer condition can easily be reduced to a continuous one. For example, the condition $x \in \{0, 1\}$ is equivalent to $x = x^2$ (in reality, however, this is not used).

We will illustrate a relation of both areas on the example of flows in networks.

Example 1.1 (Flows in networks)**.** Consider a directed graph $G = (V, E)$, where $s \in V$ is the source vertex and $t \in V$ the terminal vertex. Each edge has a capacity, which is represented by a function $u\colon E \mapsto \mathbb{R}^+$. The objective is to find a maximum flow from s to t. The flow coming into any intermediate vertex needs to equal the flow going out of it (flow in = flow out, called the conservation law).

Including an artificial edge (t, s) in the graph, the maximum flow problem is then equivalently formulated as finding the maximum flow through the additional edge

$$\begin{aligned}\max\ x_{ts}\ \text{ subject to }\ \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} &= 0,\ \forall i \in V\\ 0 \leq x_{ij} \leq u_{ij},&\ \forall (i,j) \in E,\end{aligned}$$

which is an integer linear programming problem. Denoting A the incidence matrix of graph G, the problem has a compact form

$$\max\ x_{t,s} \ \text{ subject to } \ Ax = 0,\ 0 \le x \le u.$$

The best known algorithms utilize the discrete nature of the problem. On the other hand, the LP formulation is beneficial, too. Since matrix A is totally unimodular, the resulting optimal solution is automatically integral, provided the capacities are integral. Hence the problem is efficiently solvable by means of linear programming, despite integer conditions.

Another advantage of the LP formulation is that we can easily modify it to different variants of the problem. Consider for example the problem of finding a *minimum-cost flow*. Denote by c_{ij} the cost of sending a unit of flow along the edge $(i, j) \in E$ and by $d > 0$ the minimum required flow. Then the problem reads as an LP problem

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij} \ \text{ subject to } \ Ax = 0,\ 0 \le x \le u,\ x_{ts} \ge d. \qquad \square$$

1.1 Motivation examples

Example 1.2 (Theoretical: eigenvalues)**.** Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix and $\lambda_1 \ge \cdots \ge \lambda_n$ its (real) eigenvalues. Consider the unit ball B in space $\mathbb{R}^n$; the ball is defined as $B := \{x \in \mathbb{R}^n;\ \|x\|_2 \le 1\}$. The maximal eigenvalue λ_1 is attained as the maximal value of the quadratic form x^TAx on ball B, and similarly the minimal eigenvalue λ_n is attained as the minimal value of the quadratic form x^TAx on B. Formally:

$$\lambda_1 = \max_{x:\|x\|_2\le 1} x^TAx, \quad \lambda_n = \min_{x:\|x\|_2\le 1} x^TAx.$$

This is a statement of the *Rayleigh–Ritz theorem.* Let us prove it for λ_1:

Inequality "$\le$": Let x_1 be an eigenvector corresponding to λ_1 and normalized such that $\|x_1\|_2 = 1$. Then $Ax_1 = \lambda_1 x_1$. Multiplying by x_1^T from the left yields

$$\lambda_1 = \lambda_1 x_1^T x_1 = x_1^T A x_1 \le \max_{x:\|x\|_2=1} x^TAx.$$

Inequality "$\ge$": Let $x \in \mathbb{R}^n$ be an arbitrary vector such that $\|x\|_2 = 1$. Let $A = Q\Lambda Q^T$ be a spectral decomposition of matrix A. Denoting $y := Q^Tx$, we have $\|y\|_2 = 1$ and

$$x^TAx = x^TQ\Lambda Q^Tx = y^T\Lambda y = \sum_{i=1}^n \lambda_i y_i^2 \le \sum_{i=1}^n \lambda_1 y_i^2 = \lambda_1\|y\|_2^2 = \lambda_1. \qquad \square$$

Example 1.3 (Functional optimization)**.** In principle, the number of variables need not be finite. For example, in a functional problem, we want to find a function satisfying certain constraints and minimizing a specified criterion. For illustration, imagine a problem of computing the best trajectory for a spacecraft traveling from Earth to Mercury; the variable here is the curve of the trajectory described by a function, and the objective is to minimize travel time. Certain simple functional problems can be solved analytically, but in general they are solved by discretization of the unknown function and then application of classical optimization methods.

Isoperimetric problems belong to this area, too. It is well-known that the ball has the smallest surface area of all surfaces that enclose a given volume. But how it is when two volumes are given and we wish to minimize the surface area (including the separating surface)? This problem is known as *the double bubble problem* and it had not been solved until Hutchings et al. [2002]. The minimum area shape consists of two spherical surfaces meeting at angles of $120° = \frac{2}{3}\pi$. The separating area is also a spherical surface; it is a disc in case of two equally sized volumes. See illustration on Figure 1.1. □

Example 1.4 (When the nature optimizes)**.** Snell's law quantifies the bending of light as it passes through a boundary between two media. The less dense the medium, the faster light travels. The trajectory of light is such that it is traversed in the least time (the so called Fermat's principle of least time). See illustration on Figure 1.2. □

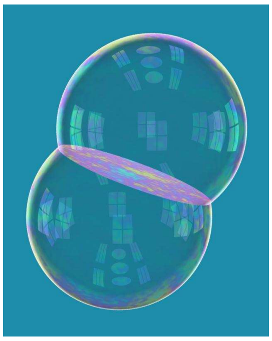


Figure 1.1: (Example 1.3) The double bubble problem.

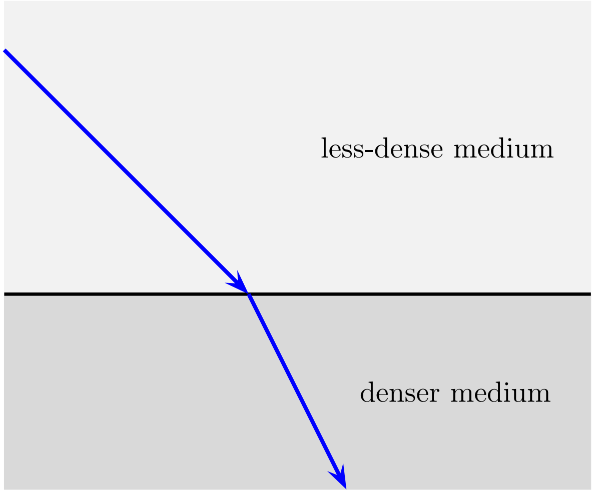


Figure 1.2: (Example 1.4) Snell's law and Fermat's principle.

1.2 Continuous optimization: First steps

Local and global minima

A solution can be categorized in several types; see Figure 1.3. A point $x^* \in M$ is called

- *a (global) minimum* if $f(x^*) \leq f(x)$ for every $x \in M$,
- *a strict (global) minimum* if $f(x^*) < f(x)$ for every $x^* \neq x \in M$,
- *local minimum* if $f(x^*) \leq f(x)$ for every $x \in M \cap \mathcal{O}_\varepsilon(x^*)$,
- *a strict local minimum* if $f(x^*) < f(x)$ for every $x^* \neq x \in M \cap \mathcal{O}_\varepsilon(x^*)$.

Naturally, to solve a problem $\min_{x \in M} f(x)$ means to find its minimum, called the optimal solution. However, sometimes the problem is so hard that we are contented with an approximate solution instead.

Be ware that the minimal value of function $f(x)$ on set M need not be attained. Consider for example the problem $\min_{x \in \mathbb{R}} x$, which is unbounded from below, or the problem $\min_{x \in \mathbb{R}} e^x$, which is bounded from below. A sufficient condition for existence of a minimum is given by the Weierstrass theorem.

Theorem 1.5 (Weierstrass)**.** *If $f(x)$ is continuous and M compact, then $f(x)$ attains a minimum on M.*

Another problem appears when local minima exist. The basic methods for solving optimization problems are iterative. They start at an initial point and move in the decreasing direction of the objective function. When they approach a local minimum, they get stuck and they have to overcome this situation problem.

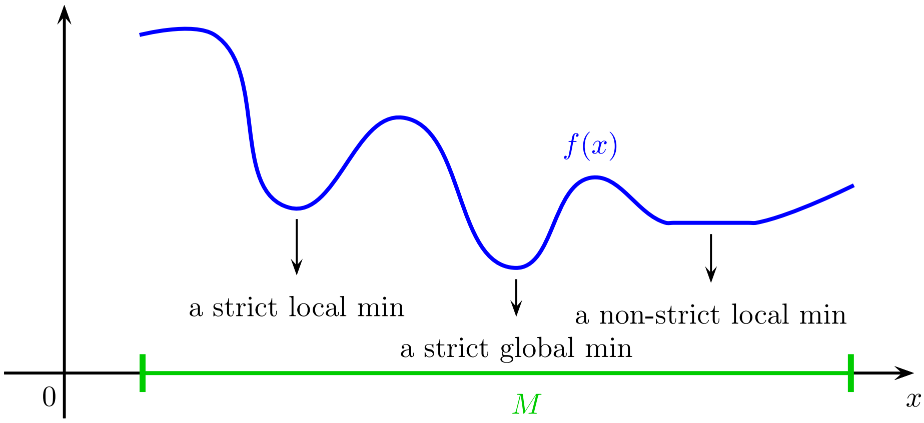


Figure 1.3: Local and global minima.

This phenomenon does occur in linear programming, or more generally in convex optimization, since each local minimum is a global one (see Theorem 3.15).

Notice that the concept of a local minimum can be used in discrete optimization, too. For instance, in the minimum spanning tree problem we can define a local neighbourhood as the set of all spanning trees obtained by replacing just one edge.

Classification

The feasible set M is often defined by a system of equations and inequalities

$$\begin{aligned} g_j(x) &\le 0, \quad j = 1, \dots, J, \\ h_\ell(x) &= 0, \quad \ell = 1, \dots, L, \end{aligned}$$

where $g_j(x), h_\ell(x)\colon \mathbb{R}^n \to \mathbb{R}$. We will employ a short form

$$g(x) \le 0, \quad h(x) = 0,$$

where $g\colon \mathbb{R}^n \to \mathbb{R}^J$ and $h\colon \mathbb{R}^n \to \mathbb{R}^L$. Depending on the type of the objective function and the feasible set, we classify the optimization problems as follows:

- *Linear programming.* Functions $f(x)$, $g_j(x)$, $h_\ell(x)$ are linear. We assume that the reader has a basic background in linear programming

- *Unconstrained optimization.* Here $M = \mathbb{R}^n$.

- *Convex optimization.* Functions $f(x)$, $g_j(x)$ are convex and $h_\ell(x)$ are linear.

Basic transformations

If one wants to find a maximum of $f(x)$ on set M, then the problem is easily reduced to the minimization problem

$$\max_{x \in M} f(x) = -\min_{x \in M} -f(x).$$

An equation constraint can be reduced to inequalities since $h(x) = 0$ is equivalent to $h(x) \le 0$, $h(x) \ge 0$, but this is not recommended in view of numerical issues.

Transformations of functions. The optimization problem

$$\min\ f(x) \ \text{ subject to } \ g(x) \le 0,\ h(x) = 0$$

can be transformed to

$$\min\ \varphi(f(x))\ \text{ subject to }\ \psi(g(x)) \le 0,\ \eta(h(x)) = 0,$$

provided

- $\varphi(z)$ is increasing on its domain, e.g., z^k, $z^{1/k}$, $\log(z)$;
- $\psi(z)$ preserves nonnegativity, i.e., $z \le 0 \ \Leftrightarrow\ \psi(z) \le 0$, e.g., z^3;
- $\eta(z)$ preserves roots, i.e., $z = 0 \ \Leftrightarrow\ \eta(z) = 0$, e.g., z^2.

Both optimization problems then possess the same minima. The optimal values are different, but they can be easily computed from the optimal solutions.

Example 1.6 (Geometric programming)**.** The transformation turns out to be very convenient in geometric programming, for instance. To illustrate it, consider the particular example

$$\min\ x^2y\ \text{ subject to }\ 5xy^3 \le 1,\ 7x^{-3}y \le 1,\ x, y > 0.$$

The logarithm of both sides yields

$$\min\ 2\log(x) + \log(y)\ \text{ subject to }\ \log(5) + \log(x) + 3\log(y) \le 0,\ \log(7) - 3\log(x) + \log(y) \le 0.$$

The substitution $x' := \log(x)$, $y' := \log(y)$ then leads to an LP problem

$$\min\ 2x' + y'\ \text{ subject to }\ \log(5) + x' + 3y' \le 0,\ \log(7) - 3x' + y' \le 0. \qquad \square$$

Moving the objective function to the constraints. The frequently used transformation is to move the objective function to the constraints, that is, the problem $\min_{x\in M} f(x)$ is transformed to

$$\min\ z\ \text{ subject to }\ f(x) \le z,\ x \in M.$$

The objective function now is linear, and all possible obstacles are hidden in the constraints.

Example 1.7 (a finite minimax)**.** Consider the problem

$$\min_{x\in M}\ \max_{i=1,\dots,s}\ f_i(x).$$

The problems of type min–max are very hard in general. However, in our situation, the outer objective function is the maximum on a finite set. The problem thus can be written as

$$\min\ z\ \ \text{ subject to }\ f_i(x) \le z,\ i = 1, \dots, s,\ x \in M.$$

In the original formulation, the outer objective function $\max_{i=1,\dots,s} f_i(x)$ is nonsmooth. After the transformation, the objective function is linear. □

Surprisingly, the converse transformation can be sometimes convenient as well. Moving the constraints into the objective function is addressed in Section 6.3.3.

Elimination of equations and variables. Consider the problem

$$\min\ f(x)\ \text{ subject to }\ g(x) \le 0,\ Ax = b,$$

where $A \in \mathbb{R}^{m\times n}$ has full row rank. First, we solve the system of equations $Ax = b$. Suppose that the solution set is not empty, so it has the form of $x^0 + \operatorname{Ker}(A)$, where x^0 is one (arbitrarily chosen) solution and $\operatorname{Ker}(A)$ is the kernel of A. Construct matrix $B \in \mathbb{R}^{n\times(n-m)}$ such that its columns form a basis of $\operatorname{Ker}(A)$. Then any solution of $Ax = b$ can be expressed as $x = x^0 + Bz$, where $z \in \mathbb{R}^{n-m}$. Substitution for x results in optimization problem

$$\min\ f(x^0 + Bz)\ \text{ subject to }\ g(x^0 + Bz) \le 0.$$

This approach eliminates the equations and reduces the dimension of the problem (i.e., the number of variables) by m.

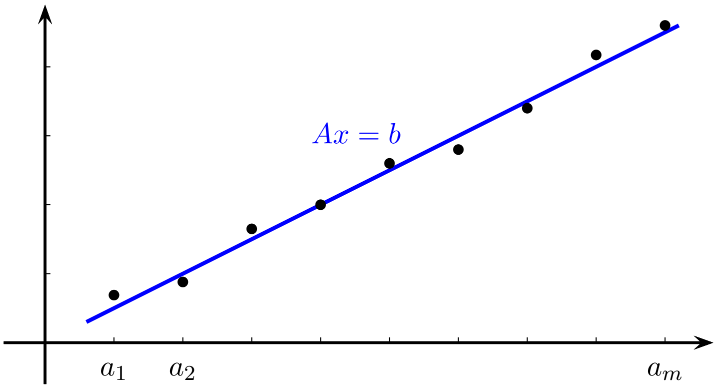


Figure 1.4: Linear regression.

1.3 Linear regression

The problem of linear regression is to find a linear dependence in data $(a_1, b_1), \dots, (a_m, b_m) \in \mathbb{R}^{n+1}$; see Figure 1.4. Linear regression is widely used in many disciplines, including economy, biology and computer science. In pattern recognition, for example, one wants a computer system to predict and make decisions autonomously (e.g., spam filtering, books and movie recommendations, face recognition, credit card fraud detection). Of course, the true dependence need not be linear and there exist models for nonlinear regression; we focus to the linear case only.

Let the matrix $A \in \mathbb{R}^{m\times n}$ consist of rows $a_1, \dots, a_m$. Then the goal is to find a vector $x \in \mathbb{R}^n$ such that $Ax \approx b$. Since usually $m \gg n$, the system of linear equations $Ax = b$ is overdetermined and has no solution. Therefore we will seek for an approximate solution.

Mathematically, we can model the problem as an optimization problem to find $x \in \mathbb{R}^n$ such that the difference between the left and the right hand side is minimal in a certain norm:

$$\min_{x\in\mathbb{R}^n} \ \|Ax - b\|.$$

The geometric interpretation of this problem is to find the projection of vector $b \in \mathbb{R}^m$ to the column space $\mathcal{S}(A)$ of matrix A. The typical choices are the following norms:

- *Euclidean norm.* The problem then reads $\min_{x\in\mathbb{R}^n} \ \|Ax - b\|_2^2 = \min_{x\in\mathbb{R}^n} \ \sum_{i=1}^m (A_{i*}x - b_i)^2$, that is, it is the ordinary least squares problem. If matrix A has full column rank, then the solution is unique and has the form $x^* = (A^TA)^{-1}A^Tb$. This approach is also justified by statistics: Suppose that the dependence is really linear and the entries of the right-hand side vector b are affected by independent and normally distributed errors. Then $x^*$is the best linear unbiased estimator and also the the maximum likelihood estimator.

- *Manhattan norm.* The problem $\min_{x\in\mathbb{R}^n} \ \|Ax - b\|_1$ can be expressed as the linear program

  $$\min \ e^Tz \ \text{ subject to } \ -z \leq Ax - b \leq z, \ z \in \mathbb{R}^m, \ x \in \mathbb{R}^n.$$

  This case has also a statistical interpretation. The optimal solution produces the maximum likelihood estimator as long as the noise follows the Laplace distribution.

- *Maximum norm.* The problem $\min_{x\in\mathbb{R}^n} \ \|Ax - b\|_\infty$ is also equivalent to an LP problem

  $$\min \ z \ \text{ subject to } \ -ze \leq Ax - b \leq ze, \ z \in \mathbb{R}, \ x \in \mathbb{R}^n.$$

Outliers. An outlier is an observation that differs significantly from the others; see Figure 1.5. Usually, it is caused by some experimental error. An outlier spoils the linear tendency in data and the resulting estimator can be distorted. The Manhattan norm is less sensitive to outliers than the other norms, but still outliers can cause problems.

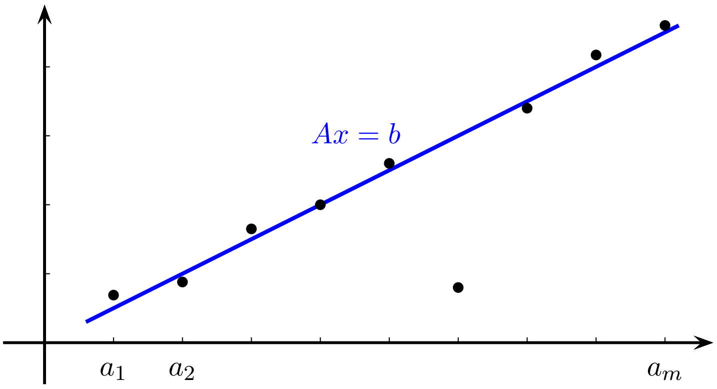


Figure 1.5: Linear regression with an outlier.

If we expect or estimate that there are $k \ll m$ outliers in data, then we can solve the linear regression problem as follows

$$\min \ \|A_I x - b_I\| \ \text{ subject to } \ x \in \mathbb{R}^n, \ I \subseteq \{1, \dots, m\}, \ |I| \geq m - k,$$

where A_I, b_I denotes submatrices indexed by I. Nevertheless, this is a hard combinatorial optimization problem.

Cardinality. The cardinality of a vector $x \in \mathbb{R}^n$ is the number of nonzero entries and it is denoted by

$$\|x\|_0 = |\{i;\, x_i \neq 0\}|.$$

This notation resembles the vector ℓ_p-norm $\|x\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}$. Indeed, the cardinality is obtained by the limit transition, neglecting the p-th roots

$$\|x\|_0 = \lim_{p \to 0^+} \textstyle\sum_{i=1}^n |x_i|^p.$$

However, $\|x\|_0$ is not a vector norm.

In regression, we usually aim to explain b by using a small number of variables (or regressors), that is, we also want to minimize $\|x\|_0$. We join both criteria by a weighted sum, resulting to a formulation

$$\min \ \|Ax - b\|_2 + \gamma \|x\|_0,$$

where $\gamma > 0$ is a suitably chosen constant. Again, this is a hard combinatorial problem. That is why $\|x\|_0$ is approximated by the Manhattan norm (in some sense, it is the best approximation). As a consequence, we get an effectively solvable optimization problem

$$\min \ \|Ax - b\|_2 + \gamma \|x\|_1.$$

Example 1.8 (Signal reconstruction)**.** Consider the problem of a signal reconstruction. Let a vector $\tilde{x} \in \mathbb{R}^n$ represent the unknown signal, and let $y = \tilde{x} + \mathrm{err}$ represent the observed noisy signal. We want to smooth the noisy signal and find a good approximation of $\tilde{x}$. To this end, we will seek for a vector $x \in \mathbb{R}^n$ that is close to y and that is also smoothed, i.e., there are not big oscillations.

This idea leads to multi-objective optimization problem

$$\min_{x \in \mathbb{R}^n} \ \|x - y\|_2, \ |x_{i+1} - x_i| \ \forall i.$$

A single-objective scalarization is obtained by a weighted sum of the objectives

$$\min_{x \in \mathbb{R}^n} \ \|x - y\|_2 + \gamma \textstyle\sum_{i=1}^{n-1} |x_{i+1} - x_i|,$$

where $\gamma > 0$ is a parameter. A smaller value of γ prioritizes the first objective and so the resulting signal is closer to the observed signal, while a larger γ penalizes oscillations and produces more smoothed signals.

Denote by $D \in \mathbb{R}^{(n-1)\times n}$ the difference matrix with entries $D_{ii} = 1$, $D_{i,i+1} = -1$ and zeros elsewhere. Then the problem reads

$$\min_{x\in\mathbb{R}^n} \|x - y\|_2 + \gamma\|Dx\|_1.$$

This can again be viewed as an approximation of a cardinality problem

$$\min_{x\in\mathbb{R}^n} \|x - y\|_2 + \gamma\|Dx\|_0,$$

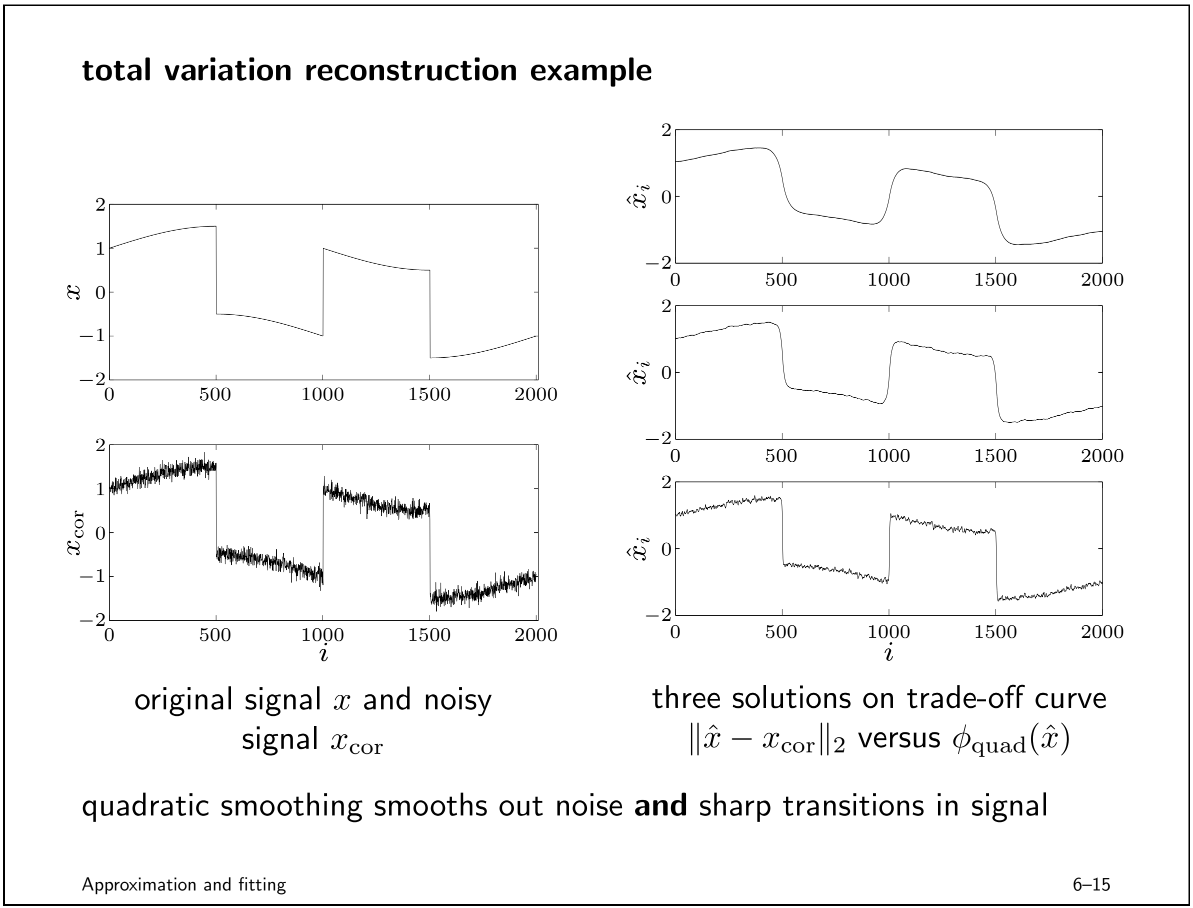
in which we aim to find a signal approximation in the form of a piecewise constant function. This approach is called *total variation reconstruction*, and it is used when processing digital signals.

A comparison by pictures is presented in Figure 1.6, originating from the website

`http://stanford.edu/class/ee364a/lectures/approx.pdf`

A similar approach is used for image analysis and processing, e.g., for deblurring of blurred images, reconstruction of damaged images, etc. See website

`http://www.imm.dtu.dk/~pcha/mxTV/` □



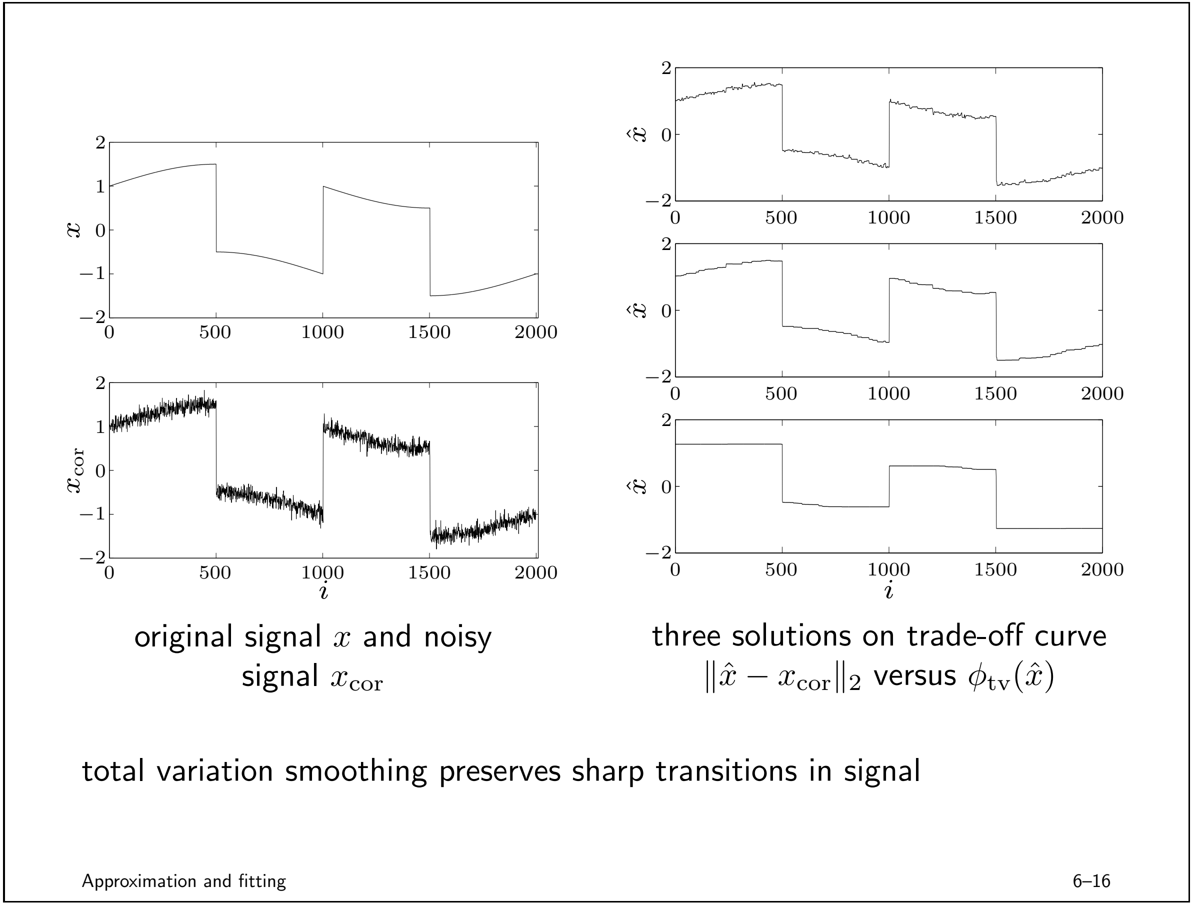
total variation reconstruction example

original signal x and noisy signal x_{cor}

three solutions on trade-off curve $\|\hat{x} - x_{\mathrm{cor}}\|_2$ versus $\phi_{\mathrm{quad}}(\hat{x})$

quadratic smoothing smooths out noise **and** sharp transitions in signal

Approximation and fitting 6–15



original signal x and noisy signal x_{cor}

three solutions on trade-off curve $\|\hat{x} - x_{\mathrm{cor}}\|_2$ versus $\phi_{\mathrm{tv}}(\hat{x})$

total variation smoothing preserves sharp transitions in signal

Approximation and fitting 6–16

Figure 1.6: Example 1.8: In both pictures on the left-hand side, there is the original signal and beneath it is the noisy signal. On the right-hand side, there are reconstructed signals with decreasing values of γ. The top picture employs quadratic smoothing (i.e., $\|Dx\|_2$ instead of $\|Dx\|_1$), while the bottom picture uses the total variation reconstruction, which better approximates the digital signal.

Chapter 2

Unconstrained optimization

An unconstrained optimization problem reads

$$\min\ f(x) \ \text{ subject to } \ x \in \mathbb{R}^n.$$

The objective function $f\colon \mathbb{R}^n \to \mathbb{R}$ is either general or we impose some differentiability assumptions later on. First we present the well-known first order necessary optimality condition.

Theorem 2.1 (First order necessary optimality condition). *Let $f(x)$ be differentiable and let $x^* \in \mathbb{R}^n$ be a local extremal point. Then $\nabla f(x^*) = o$.*

Proof. Without loss of generality assume that x^* is a local minimum. Recall that for any $i = 1, \ldots, n$

$$\nabla_i f(x) = \frac{\partial f(x^*)}{\partial x_i} = \lim_{h\to 0} \frac{f(x_1^*, \ldots, x_{i-1}^*, x_i^* + h, x_{i+1}^*, \ldots, x_n^*) - f(x^*)}{h}.$$

The limit must be the same if we consider the limit from the left or from the right. In the first case,

$$\nabla_i f(x) = \lim_{h\to 0^+} \frac{f(x_1^*, \ldots, x_{i-1}^*, x_i^* + h, x_{i+1}^*, \ldots, x_n^*) - f(x^*)}{h} \geq 0,$$

and in the second case analogously $\nabla_i f(x) \leq 0$. Therefore $\nabla_i f(x) = 0$. □

Obviously, the above condition is only a necessary condition for optimality since it cannot distinguish between minima, maxima and inflection points; see Figure 2.1. The point with zero gradient is called *a stationary point.*

We mention two second order optimality conditions, one is a necessary condition and one is a sufficient condition.

Theorem 2.2 (Second order necessary optimality condition). *Let $f(x)$ be twice continuously differentiable and let $x^* \in M$ be a local minimum. Then the Hessian matrix $\nabla^2 f(x^*)$ is positive semidefinite.*

Proof. The continuity of second partial derivatives implies that for every $\lambda \in \mathbb{R}$ and $y \in \mathbb{R}^n$ there is $\theta \in (0,1)$ such that

$$f(x^* + \lambda y) = f(x^*) + \lambda \nabla f(x^*)^T y + \frac{1}{2}\lambda^2 y^T \nabla^2 f(x^* + \theta\lambda y) y. \tag{2.1}$$

In other words, this is Taylor's expansion with Lagrange remainder. Due to minimality of x^* we have $f(x^* + \lambda y) \geq f(x^*)$, and from Theorem 2.1 we have $\nabla f(x^*) = o$. Hence

$$\lambda^2 y^T \nabla^2 f(x^* + \theta\lambda y) y \geq 0.$$

By the limit transition $\lambda \to 0$ we get $y^T \nabla^2 f(x^*) y \geq 0$. □

Theorem 2.3 (Second order sufficient optimality condition). *Let $f(x)$ be twice continuously differentiable. If $\nabla f(x^*) = o$ and $\nabla^2 f(x^*)$ is positive definite for a certain $x^* \in M$, then x^* is a strict local minimum.*

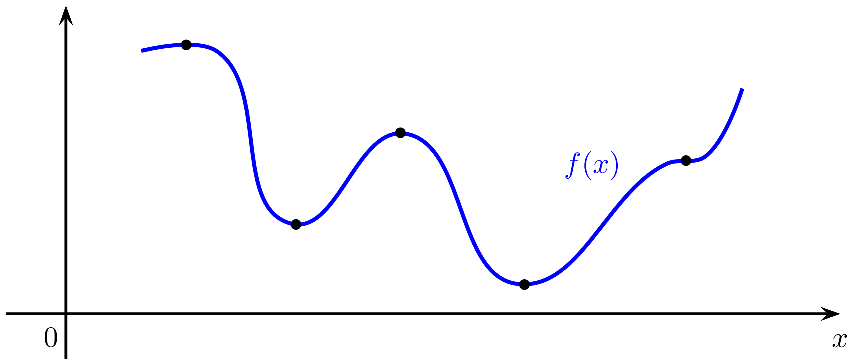


Figure 2.1: Stationary points of function $f(x)$ include local minima, local minima and inflection points.

Proof. We proceed similarly as in the proof of Theorem 2.2. In equation (2.1) we have for $\lambda \neq 0$, $y \neq o$ and sufficiently small θ

$$\lambda \nabla f(x^*)^T y = 0, \quad \frac{1}{2}\lambda^2 y^T \nabla^2 f(x^* + \theta\lambda y) y > 0.$$

Therefore $f(x^* + \lambda y) > f(x^*)$. □

We see that there is a quite tight gap between the necessary and the sufficient conditions. However, the example $f(x) = -x^4$ shows that the gap is not zero: The point $x = 0$ is a strict local maximum (and not a minimum), the sufficient condition is not satisfied (as expected), but the necessary condition is satisfied.

Example 2.4 (The least squares method)**.** Consider a system of linear equations $Ax = b$, where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and matrix A has rank n (cf. Section 1.3). Usually, m is much greater than n. Since this system has practically never an exact solution, we seek for an approximate solution by means of an optimization problem

$$\min_{x\in\mathbb{R}^n} \|Ax - b\|_2.$$

Here we aim to find such a vector x that minimizes the Euclidean norm of the difference between the left and right hand sides of system $Ax = b$. Since the square is an increasing function, the minimum is attained at the same point as for the problem

$$\min_{x\in\mathbb{R}^n} \|Ax - b\|_2^2 = (Ax - b)^T(Ax - b) = x^T A^T Ax - 2b^T Ax + b^T b.$$

We now check for the assumptions of Theorem 2.3. The gradient of the objective function is $2A^T Ax - 2A^T b$ (see the appendix, page. 63). Since it should be zero, we get the condition $A^T Ax = A^T b$, whence $x = (A^T A)^{-1} A^T b$. The Hessian of the objective function is $2A^T A$, which is a positive definite matrix. Therefore the point $x = (A^T A)^{-1} A^T b$ is a strict local minimum. Moreover, since the objective function is convex, this solution is indeed the global minimum (we will see later from Theorem 4.4).

If matrix A has not full column rank, then any solution of the system of linear equations $A^T Ax = A^T b$ is a candidate for an optimum. In fact, one can show that all these infinitely many solutions of the system of equations are optimal solutions of our problem. □

Chapter 3

Convexity

Convex sets and convex function appeared more than 100 years ago and the topic was pioneered by Hölder (1889), Jensen (1906), Minkowski (1910) and other famous mathematicians.

3.1 Convex sets

Definition 3.1. A set $M \subseteq \mathbb{R}^n$ is *convex* if for every $x_1, x_2 \in M$ and every $\lambda_1, \lambda_2 \geq 0$, $\lambda_1 + \lambda_2 = 1$, the convex combination satisfies $\lambda_1 x_1 + \lambda_2 x_2 \in M$.

Example: The empty set $\emptyset$ or a singleton $\{x\}$ are convex sets.

From the geometric point of view, the convexity of a set M means that for any two points in M the set also includes the whole line segment connecting these two points. The line segment connecting points x_1 and x_2 will be denoted

$$u(x_1, x_2) := \{x \in \mathbb{R}^n;\ x = \lambda_1 x_1 + \lambda_2 x_2,\ \lambda_1, \lambda_2 \geq 0,\ \lambda_1 + \lambda_2 = 1\}.$$

Convexity of a set can be equivalently characterized by using convex combinations of all k-tuples of its points.

Theorem 3.2. *Let $k \geq 2$. Then a set $M \subseteq \mathbb{R}^n$ is convex if and only if for any $x_1, \ldots, x_k \in M$ and any $\lambda_1, \ldots, \lambda_k \geq 0$, $\sum_{i=1}^k \lambda_i = 1$ one has $\sum_{i=1}^k \lambda_i x_i \in M$.*

Proof. An exercise. □

Obviously, the union of convex sets need not be convex. On the other hand, the intersection of convex sets is always convex.

Theorem 3.3. *If $M_i \subseteq \mathbb{R}^n$, $i \in I$, are convex, then $\cap_{i \in I} M_i$ is convex.*

Proof. Let $x_1, x_2 \in \cap_{i \in I} M_i$. Then for every $i \in I$ we have $x_1, x_2 \in M_i$, and hence also their convex combination $\lambda_1 x_1 + \lambda_2 x_2 \in M_i$. □

This property justifies introduction of the concept of the convex hull of a set M as the minimal (with respect to inclusion) convex set containing M.

Definition 3.4. *The convex hull* of a set $M \subseteq \mathbb{R}^n$ is the intersection of all sets in $\mathbb{R}^n$ including M. We denote it by $\operatorname{conv}(M)$.

Now, convexity of a set can be characterized by yet another mean.

Theorem 3.5. *A set $M \subseteq \mathbb{R}^n$ is convex if and only if $M = \operatorname{conv}(M)$.*

Proof. "⇒" Since M is convex, it is one the those convex sets that are intersected to $\operatorname{conv}(M)$.

"⇐" Due to Theorem 3.3, the set $\operatorname{conv}(M)$ is convex, so M is also convex. □

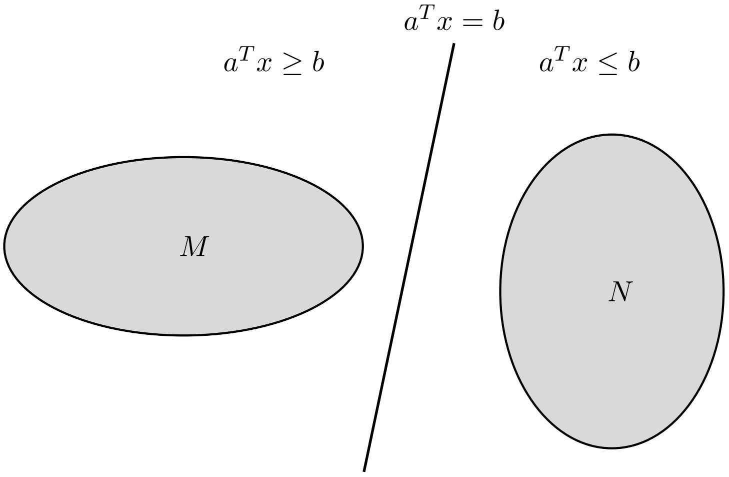


Figure 3.1: Separation of sets M by N by a hyperplane $a^Tx = b$.

Recall that *the relative interior* of a set $M \subseteq \mathbb{R}^n$ is the interior of M when restricted to the smallest affine subspace containing M. We denote it by $\mathrm{ri}(M)$.

Theorem 3.6. *If* $M \subseteq \mathbb{R}^n$ *is convex, then* $\mathrm{ri}(M)$ *is convex.*

Proof. Let $x_1, x_2 \in \mathrm{ri}(M)$. Then there exist their relative ε-neighbourhoods $\mathcal{O}_\varepsilon(x_1), \mathcal{O}_\varepsilon(x_2) \subseteq M$. Consider a convex combination $x := \lambda_1 x_1 + \lambda_2 x_2$ and the point $y \in \mathcal{O}_\varepsilon(o)$. Then an arbitrary point in $\mathcal{O}_\varepsilon(x)$ has the form of $x + y = \lambda_1 x_1 + \lambda_2 x_2 + y = \lambda_1(x_1 + y) + \lambda_2(x_2 + y)$, which belongs to M thanks to the fact that $x_1 + y, x_2 + y \in M$. □

An important property of disjoint convex sets is their linear separability; see Figure 3.1.

Definition 3.7. Two nonempty sets $M, N \subseteq \mathbb{R}^n$ are *separable* if there exists a vector $o \neq a \in \mathbb{R}^n$ and a number $b \in \mathbb{R}$ such that

$$a^Tx \leq b \quad \forall x \in M,$$
$$a^Tx \geq b \quad \forall x \in N,$$

but not

$$a^Tx = b \quad \forall x \in M \cup N.$$

We state one version of the separation theorem below. We omit the proof as it is included in another course.

Theorem 3.8 (Separation theorem)**.** *Let* $M, N \subseteq \mathbb{R}^n$ *be nonempty and convex. Then they are separable if and only if* $\mathrm{ri}(M) \cap \mathrm{ri}(N) = \emptyset$.

Let $M \subseteq \mathbb{R}^n$ be convex and closed. Using separation property we can separate a boundary point $x^* \in M$ and the set M by a hyperplane $a^Tx = b$; we call this hyperplane as *a supporting hyperplane* of M. We then have $a^Tx^* = b$ (i.e., the hyperplane contains the point x^*) and set M lies in the positive halfspace defined by the hyperplane, that is, $a^Tx \leq b$ for every $x \in M$.

Proposition 3.9. *Let* $M \subseteq \mathbb{R}^n$ *be convex and closed. Then* M *is equal to the intersection of the positive halfspaces determined by all supporting hyperplanes of* M.

Proof. From property $a^Tx \leq b \; \forall x \in M$ we get that M lies in the intersection of the halfspaces. We prove the converse inclusion by contradiction: If there is $x^* \notin M$ lying in the intersection of the halfspaces, then we can separate it (or more precisely, its neighbourhood) from M by a supporting hyperplane. Thus we found a halfspace not containing x^*; a contradiction. □

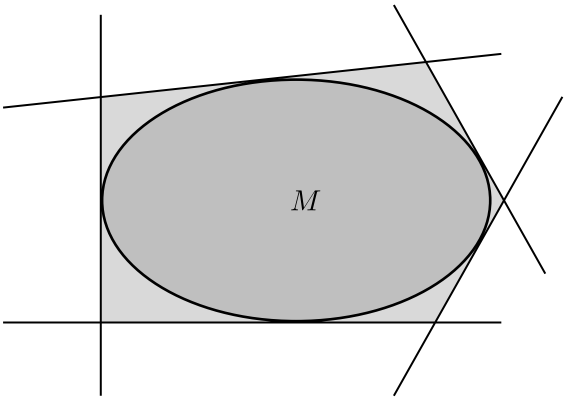


Figure 3.2: Outer approximation of a set M by supporting hyperplanes.

The above statement is not only of a theoretical importance. Using supporting hyperplanes, we can enclose set M to a convex polyhedron with an arbitrary precision; see Figure 3.2. This property is used in certain algorithms, too; they start with an initial selection of supporting hyperplanes and then they iteratively include other ones when needed, in particular when one has to separate some points from set M.

3.2 Convex functions

Convexity regards not only sets, but also functions.

Definition 3.10. Let $M \subseteq \mathbb{R}^n$ be a convex set. Then a function $f\colon \mathbb{R}^n \to \mathbb{R}$ is *convex* on M if for every $x_1, x_2 \in M$ and every $\lambda_1, \lambda_2 \geq 0$, $\lambda_1 + \lambda_2 = 1$, one has

$$f(\lambda_1 x_1 + \lambda_2 x_2) \leq \lambda_1 f(x_1) + \lambda_2 f(x_2).$$

If we have

$$f(\lambda_1 x_1 + \lambda_2 x_2) < \lambda_1 f(x_1) + \lambda_2 f(x_2)$$

for every convex combination with $x_1 \neq x_2$ and $\lambda_1, \lambda_2 > 0$, then f is *strictly convex* on M.

Analogously we define *a concave* function: $f(x)$ is concave if $-f(x)$ is convex. Obviously, a function is linear (or, more precisely, affine) if and only if it is both convex and concave.

Example 3.11. Any vector norm is a convex function because by definition for any $x_1, x_2 \in \mathbb{R}^n$ and $\lambda_1, \lambda_2 \geq 0$, $\lambda_1 + \lambda_2 = 1$,

$$\|\lambda_1 x_1 + \lambda_2 x_2\| \leq \|\lambda_1 x_1\| + \|\lambda_2 x_2\| = \lambda_1 \|x_1\| + \lambda_2 \|x_2\|.$$

In particular, the smooth Euclidean norm $\|x\|_2$ is convex as well as the non-smooth norms $\|x\|_1$ and $\|x\|_\infty$, or any matrix norm.

Analogously as in Theorem 3.2 we can characterize convex functions by means of convex combinations of k-tuples of points.

Theorem 3.12 (Jensen's inequality)**.** *Let $k \geq 2$ and let $M \subseteq \mathbb{R}^n$ be convex. Then a function $f\colon \mathbb{R}^n \to \mathbb{R}$ is convex on M if and only if for any $x_1, \ldots, x_k \in M$ and $\lambda_1, \ldots, \lambda_k \geq 0$, $\sum_{i=1}^k \lambda_i = 1$, one has*

$$f\big(\textstyle\sum_{i=1}^k \lambda_i x_i\big) \leq \sum_{i=1}^k \lambda_i f(x_i).$$

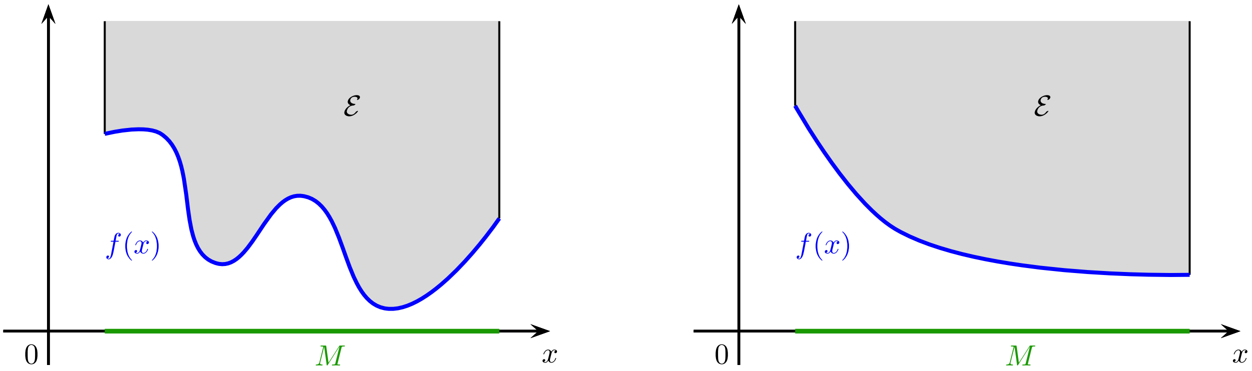


Figure 3.3: The epigraph $\mathcal{E}$ of a nonconvex function (on the left) and a convex function (on the right).

Proof. We will proceed by mathematical induction on k. The statement is obvious for $k = 2$, so we turn our attention to the induction step. Define $\alpha := \sum_{i=1}^{k-1} \lambda_i$. Since $\alpha + \lambda_k = 1$ and $\sum_{i=1}^{k-1} \alpha^{-1}\lambda_i = 1$, we get using the induction hypothesis

$$\begin{aligned} f\big(\textstyle\sum_{i=1}^{k} \lambda_i x_i\big) &= f\big(\alpha \textstyle\sum_{i=1}^{k-1} \alpha^{-1}\lambda_i x_i + \lambda_k x_k\big) \le \alpha f\big(\textstyle\sum_{i=1}^{k-1} \alpha^{-1}\lambda_i x_i\big) + \lambda_k f(x_k) \\ &\le \alpha \textstyle\sum_{i=1}^{k-1} \alpha^{-1}\lambda_i f(x_i) + \lambda_k f(x_k) = \textstyle\sum_{i=1}^{k} \lambda_i f(x_i). \end{aligned}$$ □

The following observation is useful for practical verification of convexity of a function.

Theorem 3.13. *A function $f(x)$ is convex on M if and only if it is convex on each segment in M. That is, the function $g(t) = f(x + ty)$ is convex on the corresponding compact interval domain of variable t for every $x \in M$ and every y of norm 1.*

Another characterization of convex functions is by means of epigraphs; see Figure 3.3.

Definition 3.14. *The epigraph* of a function $f\colon \mathbb{R}^n \to \mathbb{R}$ on a set $M \subseteq \mathbb{R}^n$ is the set

$$\{(x, z) \in \mathbb{R}^{n+1};\ x \in M,\ z \ge f(x)\}.$$

Theorem 3.15 (Fenchel, 1951)**.** *Let $M \subseteq \mathbb{R}^n$ be a convex set. Then a function $f\colon \mathbb{R}^n \to \mathbb{R}$ is convex if and only if its epigraph is a convex set.*

Proof. "⇒" Denote by $\mathcal{E}$ the epigraph of $f(x)$ on M, and let $(x_1, z_1), (x_2, z_2) \in \mathcal{E}$ be arbitrarily chosen. Consider their convex combination

$$\lambda_1(x_1, z_1) + \lambda_2(x_2, z_2) = (\lambda_1 x_1 + \lambda_2 x_2, \lambda_1 z_1 + \lambda_2 z_2).$$

Due to convexity of M we have $\lambda_1 x_1 + \lambda_2 x_2 \in M$, and convexity of $f(x)$ then implies

$$f(\lambda_1 x_1 + \lambda_2 x_2) \le \lambda_1 f(x_1) + \lambda_2 f(x_2) \le \lambda_1 z_1 + \lambda_2 z_2.$$

"⇐" Let $\mathcal{E}$ be convex. For any $x_1, x_2 \in M$ we have $(x_1, f(x_1)), (x_2, f(x_2)) \in \mathcal{E}$. Consider a convex combination $\lambda_1 x_1 + \lambda_2 x_2 \in M$. Due to convexity of $\mathcal{E}$ we have

$$\lambda_1(x_1, f(x_1)) + \lambda_2(x_2, f(x_2)) = (\lambda_1 x_1 + \lambda_2 x_2, \lambda_1 f(x_1) + \lambda_2 f(x_2)) \in \mathcal{E},$$

whence $f(\lambda_1 x_1 + \lambda_2 x_2) \le \lambda_1 f(x_1) + \lambda_2 f(x_2)$. □

The following property, illustrated on Figure 3.4, is frequently used in optimization. We will see later in Chapter 4 that the feasible set M of an optimization problem $\min_{x\in M} f(x)$ is usually described by a system of inequalities $g_j(x) \le 0$, $j = 1, \dots, J$. If functions g_j are convex, then the set M is convex, too.

Theorem 3.16. *Let $M \subseteq \mathbb{R}^n$ be a convex set and $f\colon \mathbb{R}^n \to \mathbb{R}$ a convex function. For any $b \in \mathbb{R}$ the set $\{x \in M;\ f(x) \le b\}$ is convex.*

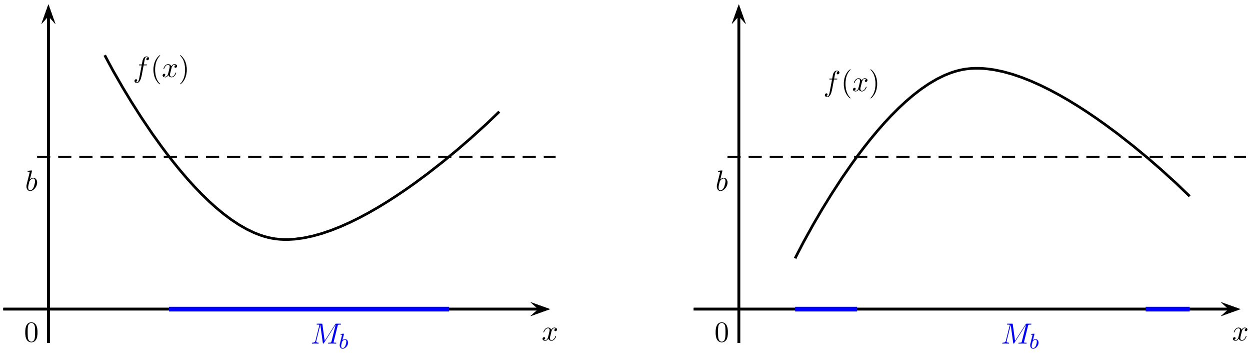


Figure 3.4: The set $M_b := \{x \in M;\, f(x) \le b\}$ illustrated for a convex function (on the left) and a nonconvex function (on the right); see Theorem 3.16.

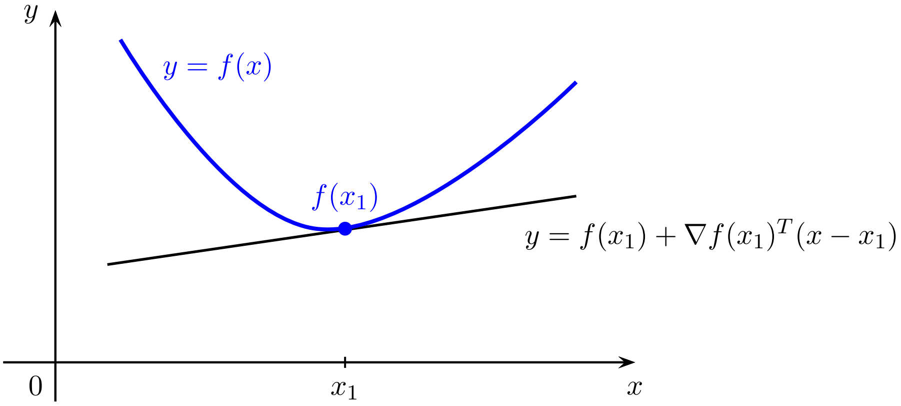


Figure 3.5: The tangent line to the graph of a convex function $f(x)$ at point $(x_1, f(x_1))$.

Proof. For arbitrary $x_1, x_2 \in \{x \in M;\, f(x) \le b\}$ consider a convex combination $\lambda_1 x_1 + \lambda_2 x_2 \in M$. From convexity of function $f(x)$ we get

$$f(\lambda_1 x_1 + \lambda_2 x_2) \le \lambda_1 f(x_1) + \lambda_2 f(x_2) \le \lambda_1 b + \lambda_2 b = b. \qquad \square$$

Another nice property of convex functions is their continuity. We state this result without a proof, which can be found e.g. in Lange [2016].

Theorem 3.17. *Let $M \subseteq \mathbb{R}^n$ be a nonempty convex set of dimension n, and let $f\colon \mathbb{R}^n \to \mathbb{R}$ be a convex function. Then $f(x)$ is continuous and locally Lipschitz on* $\operatorname{int} M$.

3.3 The first and second order characterization of convex functions

The first order characterization of a convex function $f(x)$ can be viewed visually; see Figure 3.5. The tangent line to the graph of $f(x)$ at any point $(x_1, f(x_1))$ must lie below the graph, that is, $f(x) \ge f(x_1) + \nabla f(x_1)^T(x - x_1)$.

Theorem 3.18 (The first order characterization of a convex function, Avriel 1976, Mangasarian 1969)**.** *Let $\emptyset \ne M \subseteq \mathbb{R}^n$ be a convex set and let $f(x)$ be a function differentiable on an open superset of M. Then $f(x)$ is convex on M if and only if for every $x_1, x_2 \in M$*

$$f(x_2) - f(x_1) \ge \nabla f(x_1)^T(x_2 - x_1). \tag{3.1}$$

Proof. "⇒" Let $x_1, x_2 \in M$ and $\lambda \in (0,1)$ be arbitrary. Then

$$\begin{aligned} f((1-\lambda)x_1 + \lambda x_2) &\leq (1-\lambda)f(x_1) + \lambda f(x_2), \\ f((1-\lambda)x_1 + \lambda x_2) - f(x_1) &\leq \lambda(f(x_2) - f(x_1)), \\ \frac{f(x_1 + \lambda(x_2 - x_1)) - f(x_1)}{\lambda} &\leq f(x_2) - f(x_1). \end{aligned}$$

By the limit transition $\lambda \to 0$ we get (3.1) utilizing the chain rule for the derivative of a composite function $g(\lambda) = f(x_1 + \lambda(x_2 - x_1))$ with respect to λ.

"⇐" Let $x_1, x_2 \in M$ and consider a convex combination $x = \lambda_1 x_1 + \lambda_2 x_2$. By (3.1) we have

$$\begin{aligned} f(x_1) - f(x) &\geq \nabla f(x)^T(x_1 - x) = \nabla f(x)^T(x_1 - (\lambda_1 x_1 + \lambda_2 x_2)) = \lambda_2 \nabla f(x)^T(x_1 - x_2), \\ f(x_2) - f(x) &\geq \nabla f(x)^T(x_2 - x) = \nabla f(x)^T(x_2 - (\lambda_1 x_1 + \lambda_2 x_2)) = \lambda_1 \nabla f(x)^T(x_2 - x_1). \end{aligned}$$

Multiply the first inequality by λ_1, the second one by λ_2, and summing up we get

$$\lambda_1(f(x_1) - f(x)) + \lambda_2(f(x_2) - f(x)) \geq 0,$$

or $\lambda_1 f(x_1) + \lambda_2 f(x_2) \geq f(x)$. □

Remark 3.19. For strict convexity we have an analogous characterization

$$\forall x_1, x_2 \in M, x_1 \neq x_2 : f(x_2) - f(x_1) > \nabla f(x_1)^T(x_2 - x_1). \tag{3.2}$$

Theorem 3.20 (The second order characterization of a convex function, Fenchel, 1951)**.** *Let $\emptyset \neq M \subseteq \mathbb{R}^n$ be an open convex set of dimension n, and suppose that a function $f\colon M \to \mathbb{R}$ is twice continuously differentiable on M. Then $f(x)$ is convex on M if and only if the Hessian $\nabla^2 f(x)$ is positive semidefinite for every $x \in M$.*

Proof. Let $x^* \in M$ be arbitrary. Due to continuity of the second partial derivatives we have that for every $\lambda \in \mathbb{R}$ and $y \in \mathbb{R}^n$, $x^* + \lambda y \in M$, there is $\theta \in (0,1)$ such that

$$f(x^* + \lambda y) = f(x^*) + \lambda \nabla f(x^*)^T y + \frac{1}{2}\lambda^2 y^T \nabla^2 f(x^* + \theta\lambda y) y. \tag{3.3}$$

"⇒" From Theorem 3.18 we get

$$f(x^* + \lambda y) \geq f(x^*) + \lambda \nabla f(x^*)^T y,$$

so that (3.3) implies

$$y^T \nabla^2 f(x^* + \theta\lambda y) y \geq 0.$$

By the limit transition $\lambda \to 0$ we have $y^T \nabla^2 f(x^*) y \geq 0$.

"⇐" Due to positive semidefiniteness of the Hessian we have $y^T \nabla^2 f(x^* + \theta\lambda y) y \geq 0$ in the expression (3.3). Hence

$$f(x^* + \lambda y) \geq f(x^*) + \lambda \nabla f(x^*)^T y,$$

which shows convexity of $f(x)$ in view of Theorem 3.18. □

Remark 3.21. For strict convexity, we can state the following conditions:

(1) If f is strictly convex, then the Hessian $\nabla^2 f(x)$ is positive definite almost everywhere on M; in the remaining cases it is positive semidefinite there.

(2) If the Hessian $\nabla^2 f(x)$ is positive definite on M, then f is strictly convex.

In the first item, we cannot claim positive definiteness everywhere on M. Using an analogous reasoning as in the proof of Theorem 3.20 the limit transition $\lambda \to 0$ can turn the strict inequality to a non-strict one.

Example 3.22.

1. Function $f(x) = x^4$ is strictly convex on $\mathbb{R}$, but its Hessian $f(x)'' = 12x^2$ vanishes at $x = 0$.
2. Function $f(x) = x^{-2}$ has the second derivatives positive everywhere on $\mathbb{R} \setminus \{0\}$, but it is not convex there. The reason is that $\mathbb{R} \setminus \{0\}$ is not a convex set, and also the definition of a convex function is not satisfied even when zero avoids the convex combinations. Therefore it is necessary that the domain is a convex set. Hence $f(x)$ is convex separately on $(0, \infty)$ and on $(-\infty, 0)$. □

Example 3.23. Consider a quadratic function $f\colon \mathbb{R}^n \to \mathbb{R}$ given by formula $f(x) = x^T Ax + b^T x + c$, where $A \in \mathbb{R}^{n\times n}$ is symmetric, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$ (see the appendix, page. 63). Then

- $f(x)$ is convex if and only if A is positive semidefinite,
- $f(x)$ is strictly convex if and only if A is positive definite. □

3.4 Other rules for detecting convexity of a function

In this section we discuss if or how is convexity preserved under addition, product, composition and other operations.

Theorem 3.24. *Let $f, g\colon \mathbb{R} \to \mathbb{R}$.*

(1) If $f(x), g(x)$ are both convex, nonnegative and nondecreasing (or both nonincreasing), then $f(x) \cdot g(x)$ is convex.

(2) If $f(x)$ is convex, nonnegative and nondecreasing, and $g(x)$ is concave, positive and nonincreasing, then $f(x)/g(x)$ is convex.

Proof. We prove the first property, the second one is analogous.

We have $(fg)'' = f''g + 2f'g' + fg'' \geq 0$ since each term is nonnegative. Therefore fg is convex in view of Theorem 3.20. □

Example 3.25. Both functions $f(x) = x$ and $g(y) = y$ are convex, but their product $h(x, y) = xy$ is not convex even when restricted to domain $(x, y) \in [0, \infty)^2$; it is strictly concave on the segment between points $(1, 0)$ and $(0, 1)$. Therefore, in order to apply Theorem 3.24, we need that both functions are of the same variable. Notice that function $f(x) \cdot g(x) = x^2$ is convex now. □

Theorem 3.26. *Let $f\colon \mathbb{R}^n \to \mathbb{R}^k$ and $g\colon \mathbb{R}^k \to \mathbb{R}$.*

(1) If $f_i(x)$ is convex for each $i = 1, \dots, k$ and $g(y)$ is convex and nondecreasing in each coordinate, then $(g \circ f)(x) = g(f(x))$ is convex.

(2) If $f(x)$ is concave for each $i = 1, \dots, k$ and $g(y)$ is convex and nonincreasing in each coordinate, then $(g \circ f)(x) = g(f(x))$ is convex.

Proof. We will show the first property only; the second property is analogous.

Let $x_1, x_2 \in \mathbb{R}^n$. For a convex combination $\lambda_1 x_1 + \lambda_2 x_2$ we have

$$g(f(\lambda_1 x_1 + \lambda_2 x_2)) \leq g(\lambda_1 f(x_1) + \lambda_2 f(x_2)) \leq \lambda_1 g(f(x_1)) + \lambda_2 g(f(x_2)),$$

where the first inequality follows from convexity of f and monotonicity of g, and the second inequality is due to convexity of g. □

Example 3.27.

1. If $f\colon \mathbb{R}^n \to \mathbb{R}$ is convex, then $e^{f(x)}$ is convex. For example, e^{e^x}, e^{x^2-x}, $e^{x_1-x_2}$, ...
2. If $f(x) \geq 0$ and convex, then $f(x)^p$ is convex for every $p \geq 1$.
3. If $f(x) \geq 0$ and concave, then $-\log(f(x))$ is convex. □

Example 3.28. The monotonicity assumption in Theorem 3.26 is necessary, indeed. For example, functions $f(x) = x^2 - 1$ and $g(y) = y^2$ are convex, but $g(f(x)) = (x^2 - 1)^2$ is not convex. □

Remark 3.29. Checking convexity of a function is a hard problem in general. Ahmadi et al. [2013] proved that it is an NP-hard problem for a class of multivariate polynomials of degree at most 4 (i.e., the sum of degrees in each term is at most 4). For a “general” function, it is still an open problem whether convexity testing is decidable.

Chapter 4

Convex optimization

The problem of *convex optimization* reads

$$\min\ f(x)\ \text{ subject to }\ x \in M,$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a convex function and $M \subseteq \mathbb{R}^n$ is a convex set. Often the feasible set M is described in the form as follows

$$M = \{x \in \mathbb{R}^n;\ g_j(x) \le 0,\ j = 1, \dots, J\},$$

where $g_j(x)\colon \mathbb{R}^n \to \mathbb{R}$, $j = 1, \dots, J$, are convex functions. By Theorem 3.16 the set M is convex then. In this chapter, however, we will deal with a general convex set M.

Example 4.1. An example of a convex optimization problem:

$$\min\ x_1 + x_2\ \text{ subject to }\ x_1^2 + x_2^2 \le 2.$$

Another example:

$$\min\ x_1^2 + x_2^2 + 2x_2\ \text{ subject to }\ x_1^2 + x_2^2 \le 2. \qquad \square$$

4.1 Basic properties

Theorem 4.2 (Fenchel, 1951)**.** *For a convex optimization problem we have:*

(1) Each local minimum is a global minimum.

(2) The optimal solution set is convex.

(3) If $f(x)$ is a strictly convex function, then the minimum is either unique or none.

Proof.

(1) Let $x^0 \in M$ be a local minimum and suppose to the contrary that there is $x^* \in M$ such that $f(x^*) < f(x^0)$. Consider the convex combination $x = \lambda x^* + (1-\lambda)x^0 \in M$, $\lambda \in (0, 1)$. Then

$$f(x) \le \lambda f(x^*) + (1-\lambda)f(x^0) < \lambda f(x^0) + (1-\lambda)f(x^0) = f(x^0).$$

This is in contradiction with local minimality of x^0 since for arbitrarily small $\lambda > 0$ we have $f(x) < f(x^0)$.

(2) Let $x_1, x_2 \in M$ be two optimal solutions and denote by $z = f(x_1) = f(x_2)$ the optimal value. The convex combination $x = \lambda_1 x_1 + \lambda_2 x_2 \in M$ then satisfies

$$f(x) \le \lambda_1 f(x_1) + \lambda_2 f(x_2) = \lambda_1 z + \lambda_2 z = z,$$

that is, x is also an optimal solution.

(3) Suppose to the contrary that $x_1, x_2 \in M$, $x_1 \neq x_2$, are two optimal solutions. Denote by $z = f(x_1) = f(x_2)$ the optimal value. The convex combination $x = \lambda_1 x_1 + \lambda_2 x_2 \in M$, $\lambda_1, \lambda_2 > 0$, then satisfies

$$f(x) < \lambda_1 f(x_1) + \lambda_2 f(x_2) = \lambda_1 z + \lambda_2 z = z,$$

that is, x is better that the optimal solution; a contradiction. □

Notice that a convex optimization problem need not possess an optimal solution. Consider, for example, $\min_{x \in \mathbb{R}} e^x$. This situation may happen even if the feasible set is compact:

Example 4.3. Consider the function $f\colon [1,2] \to \mathbb{R}$ defined by

$$f(x) = \begin{cases} x & \text{if } 1 < x \leq 2, \\ 2 & \text{if } x = 1. \end{cases}$$

This function is convex, but not continuous, and the minimum on $[1,2]$ is not attained. □

Theorem 4.4. *Let $\emptyset \neq M \subseteq \mathbb{R}^n$ be an open convex set and $f\colon M \to \mathbb{R}$ a convex differentiable function on M. Then $x^* \in M$ is an optimal solution if and only if $\nabla f(x^*) = o$.*

Proof. "⇒" Let $x^* \in M$ be an optimal solution. Then it is a local minimum, too, and according to Theorem 2.1 we have $\nabla f(x^*) = o$.

"⇐" Let $\nabla f(x^*) = o$. By Theorem 3.18 we have $f(x) - f(x^*) \geq \nabla f(x^*)^T (x - x^*) = 0$ for any $x \in M$. Therefore $f(x) \geq f(x^*)$ and x^* is an optimal solution. □

We cannot remove the assumption that M is open. For instance, for the problem $\min_{x \in [1,2]} x$ we have $M = [1,2]$ convex and the objective function $f(x) = x$ is differentiable on $\mathbb{R}$, but its derivative at the optimal point $x^* = 1$ is $f'(1) = 1$.

We can generalize the theorem as follows.

Theorem 4.5. *Let $\emptyset \neq M \subseteq \mathbb{R}^n$ be a convex set and $f\colon M' \to \mathbb{R}$ a convex function differentiable on an open set $M' \supseteq M$. Then $x^* \in M$ is an optimal solution if and only if $\nabla f(x^*)^T (y - x^*) \geq 0$ for every $y \in M$.*

Proof. "⇒" Suppose to the contrary that there is $y \in M$ such that $\nabla f(x^*)^T (y - x^*) < 0$. Consider the convex combination $x_\lambda = \lambda y + (1 - \lambda) x^* = x^* + \lambda (y - x^*) \in M$. Then

$$0 > \nabla f(x^*)^T (y - x^*) = \lim_{\lambda \to 0^+} \frac{f(x^* + \lambda(y - x^*)) - f(x^*)}{\lambda} = \lim_{\lambda \to 0^+} \frac{f(x_\lambda) - f(x^*)}{\lambda}.$$

Hence $f(x_\lambda) < f(x^*)$ for a sufficiently small $\lambda > 0$; a contradiction.

"⇐" By Theorem 3.18, for every $y \in M$ we have $f(y) - f(x^*) \geq \nabla f(x^*)^T (y - x^*) \geq 0$. Therefore $f(y) \geq f(x^*)$, and x^* is an optimal solution. □

The condition from Theorem 4.5 is particularly satisfied if $\nabla f(x^*) = o$. This means that each stationary point is a global minimum.

Example 4.6. The first problem in Example 4.1 reads

$$\min\ x_1 + x_2 \ \text{ subject to } \ x_1^2 + x_2^2 \leq 2.$$

Obviously, the optimum is $x^* = (-1, -1)^T$. We can verify it by means of Theorem 4.5. First, compute $\nabla f(x^*) = (1, 1)^T$. Now, we have to show that for each feasible y we have

$$\nabla f(x^*)^T (y - x^*) = (1, 1) \begin{pmatrix} y_1 + 1 \\ y_2 + 1 \end{pmatrix} \geq 0,$$

or $y_1 + y_2 \geq -2$. This is clearly true.

The second problem in Example 4.1 reads

$$\min\ x_1^2 + x_2^2 + 2x_2 \ \text{ subject to } \ x_1^2 + x_2^2 \leq 2.$$

We compute $\nabla f(x^*) = (2x_1, 2x_2 + 2)^T$, and this gradient is zero at point $x^\star = (0, -1)$. Since this point satisfies the constrait, it is the optimum. □

Example 4.7 (Rating system)**.** Many methods have been developed to provide ratings of sport teams or other entities. Here we present the following method [Langville and Meyer, 2012]. Consider n teams that we want to rate by numbers $r_1, \dots, r_n \in \mathbb{R}^n$. Let $A \in \mathbb{R}^{n\times n}$ be a known scoring matrix, where a_{ij} gives the scoring of team i against team j. This matrix is skew symmetric, that is $A = -A^T$, since $a_{ii} = 0$ and $a_{ij} = -a_{ji}$. The rating vector $r = (r_1, \dots, r_n)^T$ should reflect the scorings, so ideally we have $a_{ij} = r_i - r_j$, or in matrix form $A = re^T - er^T$. This is hardly satisfied in practice, but we aim to find the best approximation, which leads to an optimization formulation

$$\min_{x\in\mathbb{R}^n} \ f(x) = \|A - (xe^T - ex^T)\|^2.$$

We choose the Frobenius matrix norm, defined for $M \in \mathbb{R}^{n\times n}$ as $\|M\| = \sqrt{\sum_{i,j} m_{ij}^2} = \sqrt{\operatorname{tr}(M^T M)}$; that is why we minimize the square of the norm. The objective function then reads

$$\begin{aligned} f(x) &= \operatorname{tr}\big(\big(A - (xe^T - ex^T)\big)^T \big(A - (xe^T - ex^T)\big)\big) \\ &= \operatorname{tr}(A^T A) - \operatorname{tr}\big(A^T(xe^T - ex^T) + (xe^T - ex^T)^T A\big) + \operatorname{tr}\big((xe^T - ex^T)^T(xe^T - ex^T)\big) \\ &= \operatorname{tr}(A^T A) - 4x^T Ae + 2n(x^T x) - 2(e^T x)^2. \end{aligned}$$

The gradient and the Hessian read

$$\nabla f(x) = -4Ae + 4nx - 4ee^T x, \quad \nabla^2 f(x) = 4(nI_n - ee^T).$$

Since the Hessian is positive semidefinite, function $f(x)$ is convex. The optimality condition $\nabla f(x) = 0$ yields the system of linear equations

$$(nI_n - ee^T)x = Ae.$$

The matrix has rank $n - 1$ and so the solution set is the line $x = \frac{1}{n}Ae + \alpha e$, $\alpha \in \mathbb{R}$. Function $f(x)$ is constant on this line, so the whole line is the optimal solution set. In practice, we usually normalize the rating vector such that $e^T r = 0$. Since $e^T Ae = 0$, we obtain the resulting formula for the rating vector $r = \frac{1}{n}Ae$. □

Naturally, for special problems in convex optimization we can derive special properties. In the following sections we will discuss several particular classes of convex optimization problems.

4.2 Quadratic programming

A quadratic programming problem reads

$$\min \ x^T Cx + d^T x \ \text{ subject to } \ x \in M,$$

where $C \in \mathbb{R}^{n\times n}$ is symmetric, $d \in \mathbb{R}^n$ and $M \subseteq \mathbb{R}^n$ is a convex polyhedral set. If matrix C is positive semidefinite, then it is a convex problem, called *a convex quadratic program.*

Convex quadratic programs are effectively solvable in polynomial time [Floudas and Pardalos, 2009, Section “Complexity Theory: Quadratic Programming”]. If C is not positive semidefinite, then the problem is NP-hard, even finding a local minimum is NP-hard. It is interesting that NP-hardness remains valid even for the subclass of problems defined by matrix C having exactly one eigenvalue negative [Pardalos and Vavasis, 1991; Vavasis, 1991]. NP-hardness of the subclass having C negative definite is proved below; we formulate the problem equivalently as maximization of a convex quadratic function as $\max_{x\in M} x^T Cx = -\min_{x\in M} -x^T Cx.$.

Theorem 4.8. *The problem* $\max_{x\in M} x^T Cx$ *is NP-hard even when C is positive definite.*

Proof. We will construct a reduction from the NP-complete problem Set-Partitioning: Given a set of numbers $\alpha_1, \dots, \alpha_n \in \mathbb{N}$, can we group them into two subsets such that the sums of the numbers in both subsets are the same? Equivalently, is there $x \in \{\pm 1\}^n$ such that $\sum_{i=1}^n \alpha_i x_i = 0$? This problem can be formulated as follows

$$\max \sum_{i=1}^n x_i^2 \text{ subject to } \sum_{i=1}^n \alpha_i x_i = 0,\ x \in [-1,1]^n.$$

The optimal value of this problem is n if and only if Set-Partitioning is solvable. This optimization problem follows the template since the constraints are linear and the objective function has the form of $x^T C x + d^T x$ for $C = I_n$ and $d = o$. □

Example 4.9 (Portfolio selection problem)**.** This is a textbook example of an application of convex quadratic programming. The pioneer in this area was Harry Markowitz, a Nobel Prize winner in Economics in 1990, awarding his results from 1952.

The problem is formulated as follows: capital K is to be invested in n investments. The return of investment i is c_i. The mathematical formulation of the portfolio selection problem is as a linear program

$$\max\ c^T x \text{ subject to } e^T x = K,\ x \geq o.$$

The returns of investments are usually not known exactly and they are modelled as random quantities. Suppose that the vector c is random, its expected value is $\tilde{c} := \mathrm{E}\, c$ and the covariance matrix is $\Sigma := \operatorname{cov} c = \mathrm{E}\,(c-\tilde{c})(c-\tilde{c})^T$, which is positive semidefinite (*Proof:* for every $x \in \mathbb{R}^n$ we have $x^T \Sigma x = x^T(\mathrm{E}\,(c-\tilde{c})(c-\tilde{c})^T)x = \mathrm{E}\, x^T(c-\tilde{c})(c-\tilde{c})^T x = \mathrm{E}\,((c-\tilde{c})^T x)^2 \geq 0$). For a real vector $x \in \mathbb{R}^n$, the expected value of the objective function value $c^T x$ is $\mathrm{E}\,(c^T x) = \tilde{c}^T x$, and the variance of $c^T x$ is $\operatorname{var}(c^T x) = x^T \Sigma x$.

Maximizing the expected value of the reward leads to the linear programming problem

$$\max\ \tilde{c}^T x \text{ subject to } e^T x = K,\ x \geq o.$$

Taking into account the risks of investments, we model the problem as a convex quadratic program

$$\max\ \tilde{c}^T x - \gamma x^T \Sigma x \text{ subject to } e^T x = K,\ x \geq o,$$

where $\gamma > 0$ is the so called risk aversion coefficient. □

Example 4.10 (Quadrocopter trajectory planning)**.** We need to plan a trajectory for a quadrocopter fleet such that a collision is avoided and the the fleet is transferred from an initial state to a terminal state with minimum effort. In our model, time is discretized into time slots of length h. The variables are the position $p_i(k)$, velocity $v_i(k)$ and acceleration $a_i(k)$ for quadrocopter i in time step k. The constraints are:

- physical constraints: the relations between velocity and acceleration, position and velocity, ... (e.g., $v_i(k) = v_i(k-1) + h \cdot a_i(k-1)$, $p_i(k) = p_i(k-1) + h \cdot v_i(k-1)$, ...)
- restrictions on the ts maximum velocity, acceleration and jerk (i.e., the derivative of acceleration),
- the initial and terminal state (positions etc.),
- the collision avoidance constraint is nonlinear ($\|p_i(k) - p_j(k)\|_2 \geq r\ \forall i \neq j$), so we have to linearize it.

The objective function is given by the sum of norms of accelerations in particular time steps ($\sum_{i,k} \|a_i(k) + g\|_2^2$). For more details see:

- `https://www.youtube.com/watch?v=wwK7WvvUvlI`
- F. Augugliaro, A.P. Schoellig, and R. D'Andrea, *Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach*, EEE/RSJ International Conference on Intelligent Robots and Systems, 2012: pp. 1917–1922.

The practical importance of this problem is underlined by the fact that collision free planning is a very topical research problem in air traffic control of airports.

4.3 Convex cone programming

This section comes maily from book Ben-Tal and Nemirovski [2001]. The motivation to cone programming is as follows. The linear programming problem

$$\min\ c^T x \ \text{ subject to }\ Ax \geq b$$

can be generalized in several ways. In Section 4.2 we replaced linear functions with quadratic ones. Another way of a generalization is to generalize the relation "$\geq$".

Definition 4.11. A set $\emptyset \neq \mathcal{K} \subseteq \mathbb{R}^n$ is a *convex cone* if two conditions are satisfied:

(1) for every $\alpha \geq 0$ and $x \in \mathcal{K}$ we have $\alpha x \in \mathcal{K}$,

(2) for every $x, y \in \mathcal{K}$ we have $x + y \in \mathcal{K}$.

A cone is called *pointed* if it contains no complete line.

Proposition 4.12. *If $\mathcal{K}$ is a pointed convex cone, then it induces*

(1) a partial order by definition $x \geq_{\mathcal{K}} y \ \Leftrightarrow\ x - y \in \mathcal{K}$,

(2) a strict partial order by definition $x >_{\mathcal{K}} y \ \Leftrightarrow\ x - y \in \operatorname{int}\mathcal{K}$.

From now on we consider only a pointed convex closed cone $\mathcal{K}$ with nonempty interior.

Example 4.13 (Examples of cones)**.** The frequently used cones are:

- *The nonnegative orthant* $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n;\ x \geq 0\}$. The corresponding partial order is the standard entrywise inequality $\geq$ for vectors.
- *Lorentz cone (ice cream cone)* $\mathcal{L} = \{x \in \mathbb{R}^n;\ x_n \geq \sqrt{\sum_{i=1}^{n-1} x_i^2}\} = \{x \in \mathbb{R}^n;\ x_n \geq \|(x_1, \ldots, x_{n-1})\|_2\}$.
- *Generalized Lorentz cone* $\mathcal{L} = \{x \in \mathbb{R}^n;\ x_n \geq \|(x_1, \ldots, x_{n-1})\|\}$, where $\|\cdot\|$ is an arbitrary norm.
- *Convex polyhedral cone* is characterized by the system $Ax \leq 0$. This cathegory involves, for example, the nonnegative orthant or the generalized Lorentz cone with the Manhattan or maximum norm.
- *The cone of positive semidefinite matrices.*

Now we are ready to introduce cone programming. The *cone programming problem* reads

$$\min\ c^T x \ \text{ subject to }\ Ax \geq_{\mathcal{K}} b. \tag{4.1}$$

Example 4.14 (Examples of cone programs)**.**

- For $\mathcal{K} = \mathbb{R}^n_+$ we get the standard linear programming.
- Employing the Lorentz cone, we have a more interesting example

  $$\min\ c^T x \ \text{ subject to }\ \|Bx - a\|_2 \leq d^T x + f. \tag{4.2}$$

  This problem is not easily transformable to a problem with convex quadratic constraints since the squaring of both sides yields quadratic constraints which need not be convex (convexity of the constraint function was destroyed by the squaring).
- The cone constraints can be combined, so we can consider also problems such as

  $$\min\ c^T x \ \text{ subject to }\ Ax \geq b,\ \|Bx - a\|_2 \leq d^T x + f.$$

  The reason is that the Cartesian product of cones is again a cone. In our case we used the cone $\mathcal{K} := \mathbb{R}^n_+ \times \mathcal{L}$.

  This problem belongs to a called second order cone programming; see Section 4.3.2.
- The cone of positive semidefinite matrices leads to the problems of type

  $$\min\ c^T x \ \text{ subject to }\ \sum_{k=1}^{n} x_k A^{(k)} - B \text{ is positive semidefinite},$$

  where $A_1, \ldots, A_n, B$ are symmetric matrices. Such problems are called semidefinite programs; see Section 4.3.3.

4.3.1 Duality in convex cone programming

Motivation. Recall the derivation of duality of a linear program $\min\{c^Tx;\ Ax \geq b\}$: Let x be a feasible solution. Then for every $y \geq 0$ we have $y^TAx \geq y^Tb$. If y in addition satisfies $y^TA = c^T$, then we get $c^Tx = y^TAx \geq y^Tb$. In other words, y^Tb is a lower bound on the optimal value for every $y \geq 0$ such that $A^Ty = c$. This leads to the dual problem formulation and weak duality

$$\min\{c^Tx;\ Ax \geq b\} \geq \max\{b^Ty;\ A^Ty = c,\ y \geq 0\}.$$

Now the question is, in the case of convex cone programming (4.1), which relation should replace $y \geq 0$? It is not hard to see that neither $y \geq 0$ nor $y \geq_{\mathcal{K}} 0$ works well. In fact, we are interested in such y, for which we have $y^Ta \geq 0$ for every $a \geq_{\mathcal{K}} 0$. Obviously, the set of such ys forms a cone – this cone is called the dual cone of $\mathcal{K}$.

Definition 4.15. Let $\mathcal{K} \subseteq \mathbb{R}^n$ be a cone. Then its *dual cone* is the cone

$$\mathcal{K}^* = \{y \in \mathbb{R}^n;\ y^Ta \geq 0\ \forall a \in \mathcal{K}\}.$$

By using the dual cone, we formulate the dual problem to (4.1) as follows

$$\max\ b^Ty \ \text{ subject to }\ A^Ty = c,\ y \geq_{\mathcal{K}^*} 0. \tag{4.3}$$

Weak duality is then a direct adaptation of weak duality in linear programming.

Theorem 4.16 (Weak duality). *We have:* $\min\{c^Tx;\ Ax \geq_{\mathcal{K}} b\} \geq \max\{b^Ty;\ A^Ty = c,\ y \geq_{\mathcal{K}^*} 0\}$.

Proof. For every $y \geq_{\mathcal{K}^*} 0$ such that $A^Ty = c$ and for every x such that $Ax \geq_{\mathcal{K}} b$ we have

$$c^Tx = y^TAx \geq y^Tb.$$

In other words, the objective value of each feasible solution is an upper bound on every objective value of the dual problem. Therefore the inequality holds true even for the extremal values. □

Now we state some basic properties of dual cones. Some of them are illustrated on Figure 4.1. For instance, Proposition 4.18(4) is illustrated by Figures 4.1a and 4.1c.

Example 4.17.

- Nonnegative orthant is self-dual, that is, $(\mathbb{R}^n_+)^* = \mathbb{R}^n_+$ (see Figure 4.1a).
- The Lorentz cone is self-dual as well, $\mathcal{L}^* = \mathcal{L}$ (see Figure 4.1b).
- The cone of positive semidefinite matrices is also self-dual; herein, the scalar product of positive semidefinite matrices A, B is defined by $\langle A, B\rangle := \operatorname{tr}(AB) = \sum_{i,j} a_{ij}b_{ij}$. □

Proposition 4.18. *We have:*

(1) $\mathcal{K}^*$ is a closed convex cone.

(2) If $\mathcal{K}$ is a closed convex cone, then $(\mathcal{K}^*)^* = \mathcal{K}$.

(3) If $\mathcal{K}_1, \mathcal{K}_2$ are cones, then $\mathcal{K}_1 \times \mathcal{K}_2$ is a cone and $(\mathcal{K}_1 \times \mathcal{K}_2)^* = \mathcal{K}_1^* \times \mathcal{K}_2^*$.

(4) If $\mathcal{K}_1 \subseteq \mathcal{K}_2$ are cones, then $\mathcal{K}_1^* \supseteq \mathcal{K}_2^*$.

Based on the above properties, we can see that a convex cone program can also have the form of

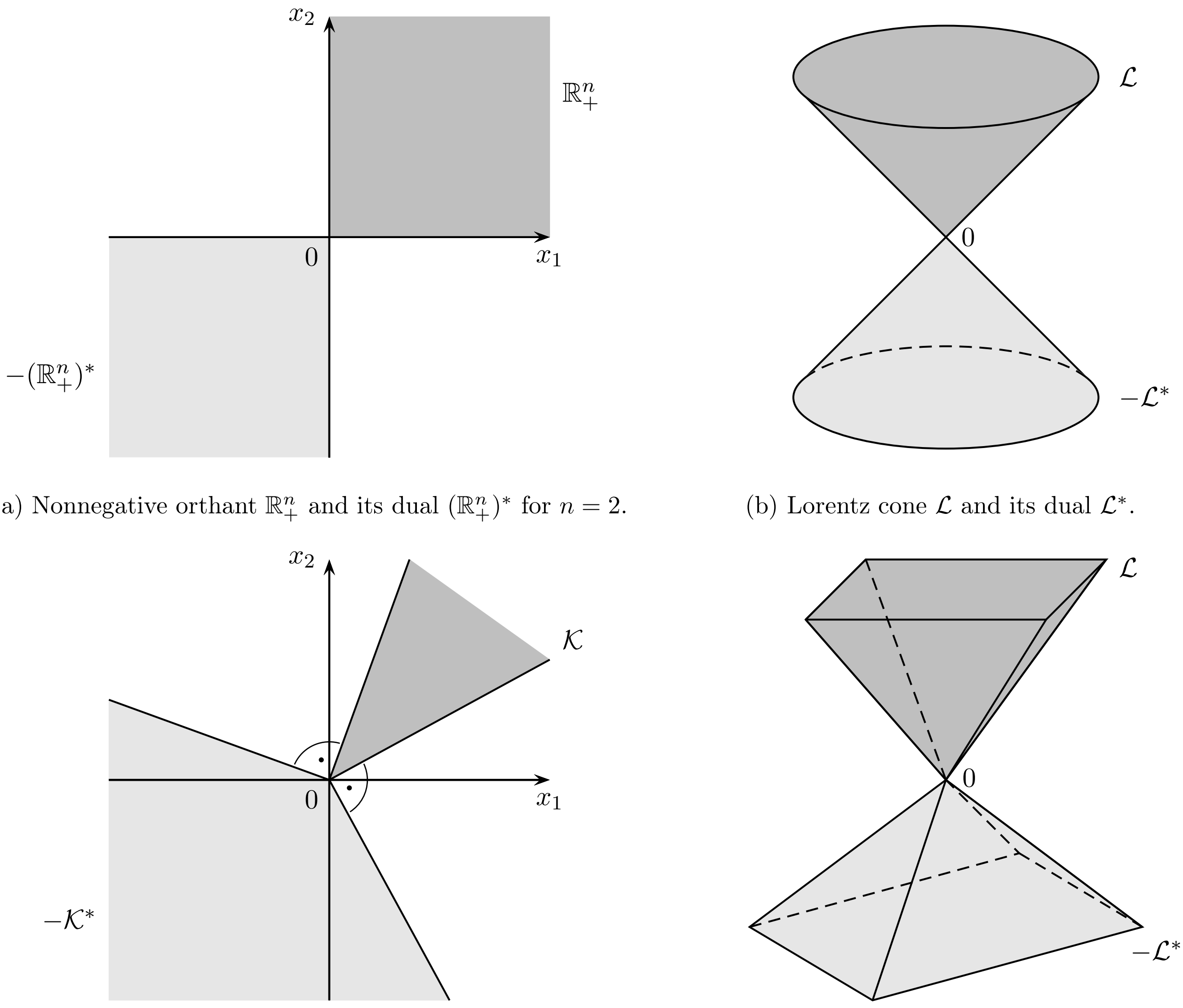
$$\min\ c^Tx \ \text{ subject to }\ Ax \geq b,\ Bx \geq_{\mathcal{K}} d,$$

the dual problem of which is

$$\max\ b^Ty + d^Tz \ \text{ subject to }\ A^Ty + B^Tz = c,\ y \geq 0,\ z \geq_{\mathcal{K}^*} 0.$$

Similarly we can consider combinations of other cone constraints.

Can we state strong duality in convex cone programming? In general not, but under mild assumptions the strong duality holds.



(a) Nonnegative orthant $\mathbb{R}^n_+$ and its dual $(\mathbb{R}^n_+)^*$ for $n = 2$.

(b) Lorentz cone $\mathcal{L}$ and its dual $\mathcal{L}^*$.

(c) A cone and its dual in $\mathbb{R}^2$.

(d) Generalized Lorentz cone $\mathcal{L}$ and its dual $\mathcal{L}^*$.

Figure 4.1: Cones and their duals (for the sake of better visibility the dual cones are multiplied by -1, i.e., rotated around the origin).

Theorem 4.19 (Strong duality)**.** *The primal and dual optimal values are the same provided at least one of the following conditions holds*

(1) the primal problem is strictly feasible, that is, there is x such that $Ax >_{\mathcal{K}} b$,

(2) the dual problem is strictly feasible, that is, there is $y >_{\mathcal{K}^*} 0$ such that $A^T y = c$.

Proof. We present the basic idea of the proof of (1) without technical details; in view of duality the point (2) is analogous.

Let f^* be the optimal value and assume that $c \neq 0$ (otherwise $f^* = 0$ and we have strong duality with $y = 0$). Define the set

$$\mathcal{M} := \{y = Ax - b;\ x \in \mathbb{R}^n,\ c^T x \leq f^*\}.$$

It is easy to see that $\mathcal{M} \cap \operatorname{int}(\mathcal{K}) = \emptyset$. Otherwise there is x such that $Ax >_{\mathcal{K}} b$ and $c^T x \leq f^*$, so by a small change of x in the direction of $-c$ we obtain a super-optimal value.

Since both $\mathcal{M}$ and $\mathcal{K}$ are convex sets, we can separate them by a hyperplane $\lambda^T y = 0$ (the zero right-hand side follows from the fact that $\mathcal{K}$ is a cone). Since $\mathcal{K}$ lies in the positive halfspace, we have $\lambda^T y \geq 0$ for every $y \in \mathcal{K}$, whence $\lambda \in \mathcal{K}^*$. Since $\mathcal{M}$ lies in the negative halfspace, we have $\lambda^T y \leq 0$ for every $y \in \mathcal{M}$; so $\lambda^T Ax \leq \lambda^T b$ for every x such that $c^T x \leq f^*$. This can happen only if the normal vectors $A^T\lambda$ and c are linearly dependent, that is, $A^T\lambda = \mu c$ for $\mu \geq 0$.

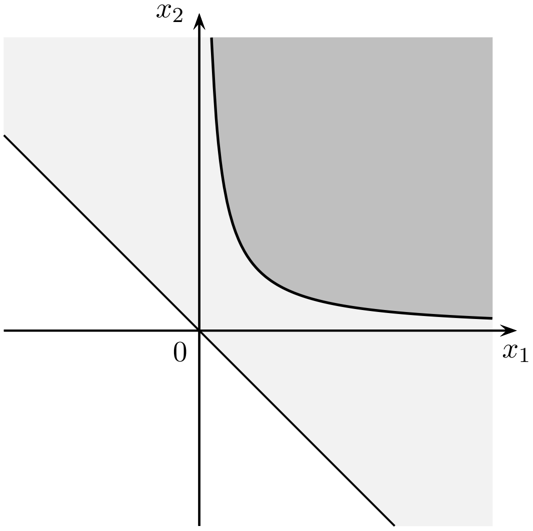


Figure 4.2: (Example 4.20) A second order cone program, for which the optimal value is not attained.

Observe that $\mu > 0$. Otherwise, if $\mu = 0$, then $A^T\lambda = 0$ and also $\lambda^T b \geq 0$. Due to strict feasibility of the primal problem there is $\tilde{x}$ such that $A\tilde{x} >_{\mathcal{K}} b$. Since $\lambda \geq_{\mathcal{K}^*} 0$, $\lambda \neq 0$, we get by premultiplication that $\lambda^T(A\tilde{x} - b) > 0$, or $\lambda^T b < 0$; a contradiction.

By normalizing $\mu \equiv 1$ we obtain $A^T\lambda = c$. This yields a dual feasible solution λ since it satisfies $A^T\lambda = c$, $\lambda \geq_{\mathcal{K}^*} 0$. Moreover, we know that $\lambda^T b \geq A\lambda^T x = c^T x$ for every x such that $c^T x \leq f^*$ (including x such that $c^T x = f^*$), whence $\lambda^T b \geq f^*$. In conjunction with weak duality we obtain equality. □

Notice that even when strong duality holds and both primal and dual optimal values are (the same and) finite, it may happen that the optimal value is not attained (formally, we should write "inf" instead of "min"). The following example illustrates this situation.

Example 4.20. Consider the convex cone program of form (4.2)

$$\min\ x_1 \ \text{ subject to } \ \sqrt{(x_1 - x_2)^2 + 1} \leq x_1 + x_2.$$

By squaring both sides of the inequality we get

$$\min\ x_1 \ \text{ subject to } \ 4x_1x_2 \leq 1,\ x_1 + x_2 > 0.$$

Even though the problem is strictly feasible, the optimal value is not attained; see Figure 4.2. □

The next example illustrates the situation when the assumption of Theorem 4.19 as well as strong duality are not satisfied.

Example 4.21. Consider the second order cone program

$$\min\ x_2 \ \text{ subject to } \ \sqrt{x_1^2 + x_2^2} \leq x_1.$$

We express it equivalently as

$$\min\ x_2 \ \text{ subject to } \ x_2 = 0,\ x_1 \geq 0.$$

We can see that the optimal value is 0 and each feasible solution is optimal, that is, the optimal solution set consists of the nonnegative part of the first axis.

To construct the dual problem, we rewrite the primal program into the canonical form

$$\min\ x_2 \ \text{ subject to } \ (x_1, x_2, x_1)^T \geq_{\mathcal{L}} 0.$$

The dual problem then reads

$$\max\ 0\ \text{ subject to }\ y_1 + y_3 = 0,\ y_2 = 1,\ y \geq_{\mathcal{L}} 0.$$

The inequality $y \geq_{\mathcal{L}} 0$ takes the form of $y_3 \geq \sqrt{y_1^2 + y_2^2}$, which together with $y_1 + y_3 = 0$ leads to $y_2 = 1$; a contradiction. Hence the dual problem is infeasible, even though the primal problem has a finite optimal value. □

4.3.2 Second order cone programming

Second order cone programming deals with convex cone programs with linear constraints and constraints corresponding to the Lorentz cone. For the sake of simplicity we employ just one Lorentz cone, and so the problem reads

$$\min\ c^T x\ \text{ subject to }\ Ax \geq b,\ Bx \geq_{\mathcal{L}} d. \tag{4.4}$$

We express

$$(B \mid d) = \left(\begin{array}{c|c} D & f \\ p^T & q \end{array} \right)$$

so the condition $Bx \geq_{\mathcal{L}} d$ takes the form of $\|Dx - f\|_2 \leq p^T x - q$. Thus we have an explicit description of problem (4.4)

$$\min\ c^T x\ \text{ subject to }\ Ax \geq b,\ \|Dx - f\|_2 \leq p^T x - q. \tag{4.5}$$

Recall that the problem is not easily transformable to a convex quadratic problem, even when allowing convex quadratic constraints. Thus we have a new class of optimization problems, which are efficiently solvable and contain many interesting problems. Actually, a lot of functions and nonlinear conditions can be expressed in the form of (4.5).

Example 4.22 (Examples of second order cone programs)**.**

- *Quadratic constraints.* For example, the condition $x^T x \leq z$ can be expressed as $x^T x + \frac{1}{4}(z-1)^2 \leq \frac{1}{4}(z+1)^2$, the square root of which gives $\|(x^T, \frac{1}{2}(z-1))\|_2 \leq \frac{1}{2}(z+1)$.
- *Hyperbola.* The condition $x \cdot y \geq 1$ on $y \geq 0$ can be expressed as $\frac{1}{4}(x+y)^2 \geq 1 + \frac{1}{4}(x-y)^2$, the square root of which (notice $y \geq 0$) gives $\|(1, \frac{1}{2}(x-y))\|_2 \leq \frac{1}{2}(x+y)$.

On the other hand, condition $e^x \leq z$ is not a second order cone constraint.

The dual problem is

$$\max\ b^T y + d^T z\ \text{ subject to }\ A^T y + B^T z = c,\ y \geq 0,\ z \geq_{\mathcal{L}^*} 0.$$

Letting $z = (u^T, v)^T$ we get

$$\max\ b^T y + d^T z\ \text{ subject to }\ A^T y + B^T z = c,\ y \geq 0,\ v \geq \|u\|_2.$$

The dual problem is thus also a second order cone program.

4.3.3 Semidefinite programming

Employing the cone of positive semidefinite matrices in the convex cone programming problem (4.1), we obtain the class of *semidefinite programming* problems

$$\min\ c^T x\ \text{ subject to }\ \sum_{k=1}^{n} x_k A^{(k)} \succeq B, \tag{4.6}$$

where $c \in \mathbb{R}^n$, matrices $A^{(1)}, \dots, A^{(n)}, B \in \mathbb{R}^{m\times m}$ are symmetric and the relation $A \succeq B$ means that $A - B$ is positive semidefinite.[1)]

How to construct the dual problem? According to (4.3), the dual problem has m^2 variables, so that they constitute a matrix of variables $Y \in \mathbb{R}^{m\times m}$. The dual objective function is $\sum_{i,j} b_{ij} y_{ij}$, the equations have the form of $\sum_{i,j} a_{ij}^{(k)} y_{ij} = c_k$, and the condition $Y \geq_{\mathcal{K}^*} 0$ takes the form $Y \succeq 0$. In total, the dual problem reads

$$\max\ \operatorname{tr}(BY)\ \text{ subject to }\ \operatorname{tr}(A^{(k)}Y) = c_k,\ k = 1,\dots,n,\ Y \succeq 0. \tag{4.7}$$

Example 4.23 (Examples of semidefinite programs)**.**

- *Linear constraints.* Linear inequalities $Ax \leq b$ are expressed as semidefinite conditions as follows:

$$\begin{pmatrix} b_1 - A_{1*}x & 0 & \dots & 0 \\ 0 & b_2 - A_{2*}x & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & b_m - A_{m*}x \end{pmatrix} \succeq 0.$$

- *Second order cone constraints.* They can be expressed as semidefinite constraints. Basically, it is sufficient to show it for the condition $\|x\|_2 \leq z$; the others can be handled by a linear transformation. We have

$$\|x\|_2 \leq z \quad \Leftrightarrow \quad \begin{pmatrix} z \cdot I_n & x \\ x^T & z \end{pmatrix} \succeq 0. \tag{4.8}$$

  Proof. For $z = 0$ the equivalence holds, so we assume $z > 0$. We consider the matrix as the matrix of a quadratic form and we transform it to a block diagonal matrix by using row & column elementary transformations. Subtracting $\frac{1}{z}x^T$-multiple of the first block row from the second one, and applying the same for the columns, we get

$$\begin{pmatrix} z \cdot I_n & x \\ x^T & z \end{pmatrix} \sim \begin{pmatrix} z \cdot I_n & 0 \\ 0 & z - \frac{1}{z}x^T x \end{pmatrix}.$$

  This matrix is positive semidefinite if and only if $z > 0$ and $x^T x \leq z^2$, or after taking the square root, $\|x\|_2 \leq z$.

- *Eigenvalues.* Many conditions on eigenvalues can be expressed as semidefinite programs. For instance, the largest eigenvalue $\lambda_{\max}$ of a symmetric matrix $A \in \mathbb{R}^{n\times n}$:

$$\lambda_{\max} = \min\ z\ \text{ subject to }\ z \cdot I_n \succeq A.$$

Example 4.24. Consider the portfolio selection problem (Example 4.9)

$$\max\ c^T x\ \text{ subject to }\ e^T x = K,\ x \geq o,$$

where c is a random vector with the expected value $\tilde{c} := \operatorname{E} c$ and the covariance matrix $\Sigma := \operatorname{cov} c = \operatorname{E}(c - \tilde{c})(c - \tilde{c})^T$. Assume that a portfolio $\tilde{x}$ is chosen, but for the covariance matrix we know only an interval estimation $\Sigma_1 \leq \Sigma \leq \Sigma_2$. What is the risk of portfolio $\tilde{x}$? The risk is given by the variance of the reward $c^T\tilde{x}$, which is equal to $\tilde{x}^T\Sigma\tilde{x}$. Thus the largest variance is computed by a semidefinite program

$$\max\ \tilde{x}^T\Sigma\tilde{x}\ \text{ subject to }\ \Sigma_1 \leq \Sigma \leq \Sigma_2,\ \Sigma \succeq 0.$$

The objective function is linear in variable Σ, and the constraints are easily transformed to the basic form (4.7) by means of Example 4.23. □

[1)] Relation $\succeq$ defines a partial order, known also as the Löwner order. Karel Löwner was an American mathematician of Czech origin (born near Prague).

4.4 Computational complexity

In general, convex optimization problems are considered to be tractable. Indeed, under some assumptions (see Section 4.4.1), they are solvable in polynomial time. On the other hand, there are some intractable convex optimization problems, too. In Section 4.4.2 we present one such class of optimization problems.

4.4.1 Good news – the ellipsoid method

A convex optimization problem $\min_{x\in M} f(x)$ is solvable in polynomial time by the ellipsoid method under general assumptions. This result is, however, rather theoretical. To solve the problem practically, other methods, such as interior point methods, are usually more convenient.

The ellipsoid method is designed to find a feasible solution, but the same idea works to find an optimal solution as well. Thus we focus on the problem of finding a point $x \in M$ or determining that M is empty. First, we construct a sufficiently large ellipsoid $\mathcal{E}$ covering the whole set M. Then we check if the center c of ellipsoid $\mathcal{E}$ lies in M. If yes, we are done. If not, then we construct a hyperplane containing point c and being disjoint to M (e.g., it can be a hyperplane tangent to M and shifted to contain c) such that M lies in halfspace $a^T x \le b$. Then we construct a smaller (minimum volume) ellipsoid covering the intersection $\mathcal{E} \cap \{x;\ a^T x \le b\}$. We repeat this process until we find a feasible point or prove $M = \emptyset$. The convergence is guaranteed by the fact that the size of the ellipsoids exponentially decreases.

In order that the above algorithm is correct and runs in polynomial time, we need to ensure certain conditions:

- The feasible set M shouldn't be too flat or too large. There must exist "reasonably" large numbers $r, R > 0$ such that M contains a ball of radius r and also M lies in the ball $\{x;\ \|x\|_2 \le R\}$.
- *Separation oracle.* For every $x^* \in \mathbb{R}^n$ we need to check for $x^* \in M$ in polynomial time. If $x^* \notin M$, then we need to find a vector $a \neq o$ such that $a^T x^* \ge \sup_{x\in M} a^T x$. This gives us a hyperplane $a^T x = a^T x^*$ satisfying $a^T x \le a^T x^*$ for every $x \in M$.

  The separation oracle is often implemented as follows. If the description of M contains a violated constraint $g(x) \le 0$, then we can take $a := \nabla g(x^*)$ since from $g(x^*) > 0$ and convexity of g we have $a^T(x - x^*) = \nabla g(x^*)^T (x - x^*) \le g(x) - g(x^*) < g(x) \le 0$ for every $x \in M$.

The ellipsoid method solves a problem up to certain accuracy. Nevertheless, the optimal solution can be an irrational number such as $\sqrt{2}$, which is not exactly representable in the standard computational model. To quantify the accuracy we need a measure of infeasibility. For linear constraints, we can use the value

$$\min\ z \quad \text{subject to} \quad Ax - ze \le b,\ z \ge 0,$$

and for a semidefinite constraint $\sum_{k=1}^n x_k A^{(k)} \succeq B$ we can use the measure

$$\min\ z \quad \text{subject to} \quad \textstyle\sum_{k=1}^n x_k A^{(k)} + zI_m \succeq B,\ z \ge 0.$$

Example 4.25. In some cases, the ellipsoid method provides a polynomial algorithm for problems with exponentially many or even infinitely many constraints. For example, let M be a unit ball described by the tangent hyperplanes, that is,

$$M = \{x \in \mathbb{R}^n;\ a^T x \le 1,\ \forall a : \|a\|_2 = 1\}.$$

To check if a given point $x^* \in \mathbb{R}^n$ belongs to the set M, we do not need to process all the infinitely many inequalities. It is sufficient to check the possibly violated constraint, which is the case of $a = \frac{1}{\|x^*\|_2} x^*$. □

4.4.2 Bad news – copositive programming

Not every convex optimization problem is tractable. Here we present a convex problem that is NP-hard. Denote by

$$\mathcal{C} := \{A \in \mathbb{R}^{n\times n};\ A = A^T,\ x^T Ax \ge 0\ \forall x \ge 0\}$$

the convex cone of *copositive matrices* and by

$$\mathcal{C}^* := \operatorname{conv}\{xx^T;\ x \geq 0\}$$

its dual cone of *completely positive matrices.* Obviously, the set $\mathcal{C}$ covers both nonnegative symmetric matrices and positive semidefinite matrices, but it contains other matrices, too. Similarly the matrices in $\mathcal{C}^*$ are nonnegative positive semidefinite, but not each such matrix belongs to $\mathcal{C}^*$. Notice that even to decide if a given matrix is copositive is a co-NP-complete problem [Murty and Kabadi, 1987]. Checking complete positivity of a matrix is NP-hard [Dickinson and Gijben, 2014], but if the problem is in NP is not known yet.

Consider a *copositive program* [Dür, 2010]

$$\min\ \operatorname{tr}(CX)\ \text{ subject to }\ \operatorname{tr}(A_iX) = b_i,\ i = 1, \dots, m,\ X \in \mathcal{C}, \tag{4.9}$$

where $C, A_1, \dots, A_k \in \mathbb{R}^{n\times n}$ and $b_1, \dots, b_k \in \mathbb{R}$. The objective function

$$\operatorname{tr}(CX) = \sum_{i,j} c_{ij}x_{ij}$$

is a linear function in variables X and the equations are linear, too. The only nonlinear constraint is $X \in \mathcal{C}$, which makes the problem to be a convex conic program. Consider also a convex program with a complete positivity condition of matrix X:

$$\min\ \operatorname{tr}(CX)\ \text{ subject to }\ \operatorname{tr}(A_iX) = b_i,\ i = 1, \dots, m,\ X \in \mathcal{C}^*. \tag{4.10}$$

Both problems are convex, but NP-hard. We prove it for the latter.

Theorem 4.26. *Problem* (4.10) *is NP-hard.*

The proof is based on a reduction from the maximum independent set problem. Let $G = (V, E)$ be a graph with n vertices and let α denote the size of a maximum independent set in graph G, that is, the size of a maximum set $I \subseteq V$ such that $i, j \in I \ \Rightarrow\ \{i, j\} \notin E$.

Theorem 4.27. *We have*

$$\alpha = \max\ \operatorname{tr}(ee^TX)\ \ \textit{subject to}\ \ x_{ij} = 0\ \forall\{i,j\} \in E,\ \operatorname{tr}(X) = 1,\ X \in \mathcal{C}^*. \tag{4.11}$$

Proof. Consider the convex cone

$$\{X \in \mathcal{C}^*;\ x_{ij} = 0\ \forall\{i,j\} \in E\}.$$

Its extreme directions are matrices of the form xx^T, where $x \geq 0$ and the support of vector x (i.e, the indices of positive entries) corresponds to an independent set in graph G. The constraint $\operatorname{tr}(X) = 1$ in problem (4.11) then normalizes the vectors in this cone. Since the objective function of problem (4.11) is linear, the optimal solution is attained in an extreme point. Therefore we can assume that the optimal solution X^* takes the form of

$$X^* = x^*x^{*T},\quad x^* \geq 0,\quad \|x^*\| = 1$$

since the condition $\operatorname{tr}(X^*) = 1$ is equivalent to $1 = \sqrt{\operatorname{tr}(X^*)} = \sqrt{\operatorname{tr}(x^*x^{*T})} = \sqrt{\operatorname{tr}(x^{*T}x^*)} = \sqrt{x^{*T}x^*} = \|x^*\|$. The support of vector x^* then corresponds to an independent set of size $\alpha(x^*)$. Denote by $\tilde{x} \in \mathbb{R}^{\alpha(x^*)}$ the restriction of vector x to its positive entries, so the zero entries are removed. Then the optimal value h of problem (4.11) can be expressed as

$$h = \max_{\|\tilde{x}\|=1} (e^T\tilde{x})^2,\quad \tilde{x} \geq 0$$

since $\operatorname{tr}(ee^TX) = \operatorname{tr}(ee^Txx^T) = \operatorname{tr}(e^Txx^Te) = (e^Tx)^2$. It is not hard to see that the optimal solution of this problem is a vector of identical entries, that is, $\tilde{x}^* = \alpha(x^*)^{-1/2}e$. Hence $h = (e^T\tilde{x}^*)^2 = (\alpha(x^*)^{-1/2}\alpha(x^*))^2 = \alpha(x^*)$. That is why h equals the size of the maximum independent set in graph G. □

4.5 Applications

4.5.1 Robust PCA

Let $A \in \mathbb{R}^{m\times n}$ be a matrix representing certain data. The problem is to determine some essential information hidden in the data. For example, if the matrix represents a picture, then we may want to recognize some pattern (e.g., a face) or to perform some operations such as reconstruction of a damaged picture.

To this end the SVD decomposition of A may serve well, however, for some purposes it is not sufficient. We will formulate the problem as the so called robust PCA (principal component analysis):

$\rightarrow$ Decompose $A = L + S$ such that L has low rank and S is sparse.

Then L represents the fundamental information in the data and S can be interpreted as a noise. This problem is rather vaguely defined and that is why we consider the (approximate) optimization problem formulation

$$\min \ \|L\|_* + \|S\|_{\ell_1} \ \text{ subject to } \ A = L + S, \tag{4.12}$$

where $\|S\|_{\ell_1}$ is the entrywise sum norm defined as

$$\|S\|_{\ell_1} := \sum_{i,j} |s_{ij}|,$$

and $\|L\|_*$ the nuclear norm defined as the sum of the singular values, that is,

$$\|L\|_* := \sum_i \sigma_i(L).$$

Notice that the nuclear norm is a good approximation of the matrix rank since it is the best convex underestimator of the rank on a unit ball. Similarly, the entrywise sum norm is a good approximation of matrix sparsity.

Problem (4.12) is a convex optimization problem since a norm is always convex. Hence the problem is effectively solvable even though the best algorithms used are not so easy to describe by simple means.

Foreground and background detection in a video

The Robust PCA technique can effectively be used to recognize foreground and background in a video or a sequence of pictures. The columns of matrix A represent the particular video frames. Then we can expect that matrix L corresponds to the background since it is static and the matrix has low rank. In contrast, matrix S captures the foreground then.

For more details see:

- `http://sites.google.com/site/rpcaforegrounddetection/`
- E. Candes, X. Li, Y. Ma, J. Wright, *Robust Principal Component Analysis?*, J. ACM 58(3), 2011

4.5.2 Minimum volume enclosing ellipsoid

The aim is to find an ellipsoid with minimum volume and covering a given convex polyhedron [Todd, 2016]. Let $x_1, \dots, x_m \in \mathbb{R}^n$ be vertices of a convex polyhedron that we want to enclose by an ellipsoid. For simplicity we restrict to a full-dimensional ellipsoid centered in the origin. Such an ellipsoid is described by $x^T H x \le 1$, where $H \in \mathbb{R}^{n\times n}$ is a positive definite matrix (in short we write $H \succ 0$). The volume of the ellipsoid is inversely proportional to $\det(H)$. Therefore the problem can be formulated as

$$\min \ -\det(H) \ \text{ subject to } \ H \succ 0, \ x_i^T H x_i \le 1, \ i = 1, \dots, m,$$

where the unknown matrix H is variable. This problem is not convex, so take the logarithm of the objective function

$$\min \ -\log\det(H) \ \text{ subject to } \ H \succ 0, \ x_i^T H x_i \le 1, \ i = 1, \dots, m.$$

Function $-\log\det(H)$ is strictly convex on the set of positive definite matrices, so we have a convex optimization problem, which is efficiently solvable.

Chapter 5

Karush–Kuhn–Tucker optimality conditions

In this chapter we consider the following optimization problem

$$\min\ f(x)\ \text{ subject to }\ x \in M,$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a differentiable function and the feasible set $M \subseteq \mathbb{R}^n$ is described by the system

$$\begin{aligned} g_j(x) &\le 0, \quad j = 1, \dots, J,\\ h_\ell(x) &= 0, \quad \ell = 1, \dots, L, \end{aligned}$$

where $g_j(x), h_\ell(x)\colon \mathbb{R}^n \to \mathbb{R}$.

By Theorem 2.1 we known that in case $M = \mathbb{R}^n$ the necessary condition for optimality of $x \in \mathbb{R}^n$ is $\nabla f(x) = 0$. If $f(x)$ is convex, then the condition is also sufficient (Theorem 4.4).

This chapter generalizes the above condition to a constrained optimization problem, which results to the so called Karush–Kuhn–Tucker conditions. First we consider only equality constraints, and then we extent the results to the general form.

Equality constraints

Consider for a while an equality constrained problem

$$\min\ f(x)\ \text{ subject to }\ h(x) = 0. \tag{5.1}$$

Let x^* be a feasible point. When x^* is optimal? First we discuss the case when the constraints are linear.

Proposition 5.1. *If $x^* \in \mathbb{R}^n$ is a local optimum of*

$$\min\ f(x)\ \textit{ subject to }\ Ax = b,$$

then $\nabla f(x^*) \in \mathcal{R}(A)$.

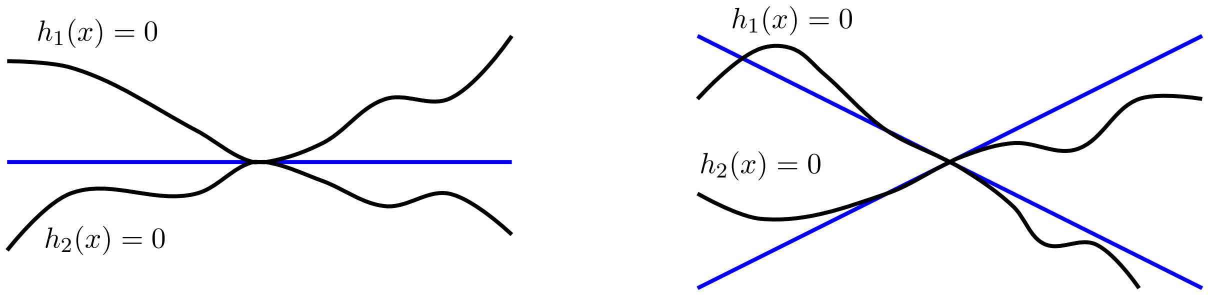
Proof. The feasible set is the solution set of the system $Ax = b$, so it is an affine subspace $x^* + \operatorname{Ker}(A)$. Let B be a matrix such that its columns form a basis of $\operatorname{Ker}(A)$. Then the feasible set can be expressed as $x = x^* + Bv$, $v \in \mathbb{R}^k$. Substituting for x we obtain an unconstrained optimization problem

$$\min\ f(x^* + Bv)\ \text{ subject to }\ v \in \mathbb{R}^k.$$

By Theorem 2.1, the necessary condition for local optimality of $v = 0$ is zero gradient, that is, $\nabla f(x^*)^T B = 0^T$. In other words, $\nabla f(x^*) \in \operatorname{Ker}(A)^\perp = \mathcal{R}(A)$. □

Now, the idea is based on linearization of possibly nonlinear functions h_ℓ. The equation $h_\ell(x) = 0$ will be replaced by the tangent hyperplane of the corresponding manifold at point x^*:

$$\nabla h_\ell(x^*)^T (x - x^*) = 0,$$



(a) Degenerate case: the intersection of the curves is a point, but the intersection of the tangent lines is a line.

(b) Regular case: the intersection of the curves is a point as well as the intersection of the tangent lines.

Figure 5.1: Linearization of nonlinear constraints – approximate curves by tangent lines.

so that the linearized constraints can be expressed as $A(x - x^*) = 0$. In order that x^* is optimal, the objective function gradient $\nabla f(x^*)$ must be perpendicular to the intersection of the tangent hyperplanes; in other words, $\nabla f(x^*)$ must be a linear combination of the gradients $\nabla h_\ell(x^*)$ of the tangent hyperplanes. According to Proposition 5.1 we have $\nabla f(x^*) \in \mathcal{R}(A)$. This leads to the condition

$$\nabla f(x^*) + \sum_{\ell=1}^{L} \nabla h_\ell(x^*)\mu_\ell = 0.$$

As illustrated by Figure 5.1, this idea can be *wrong* since a degenerate situation may appear as depicted on the figure. Thus we need to avoid such a degenerate case. This can be achieved by the assumption on linear independence of gradients $\nabla h_\ell(x^*)$.

Theorem 5.2. *Let $\nabla h_\ell(x^*)$, $\ell = 1, \dots, L$, be linearly independent. If x^* is a local optimum, then there is $\mu \in \mathbb{R}^L$ such that*

$$\nabla f(x^*) + \nabla h(x^*)\mu = 0.$$

Proof. See the basic calculus course. □

Coefficients $\mu_1, \dots, \mu_L$ are called *Lagrange multipliers.* The condition stated in the theorem is a necessary condition. This is convenient for us since we can restrict the feasible set to a much smaller set of candidates for optima – ideally the candidate is unique.

Equality and inequality constraints

Now we consider the general case with both equality and inequality constraints. The active set of a feasible point x is the set of those inequalities that are satisfied as equations:

$$I(x) = \{j;\ g_j(x) = 0\}.$$

Theorem 5.3 (KKT conditions)**.** *Let $\nabla h_\ell(x^*)$, $\ell = 1, \dots, L$, $\nabla g_j(x^*)$, $j \in I(x^*)$, be linearly independent. If x^* is a local optimum, then there exist $\lambda \in \mathbb{R}^J$, $\lambda \geq 0$, and $\mu \in \mathbb{R}^L$ such that*

$$\nabla f(x^*) + \nabla h(x^*)\mu + \nabla g(x^*)\lambda = 0, \tag{5.2}$$

$$\lambda^T g(x^*) = 0. \tag{5.3}$$

Remark. Condition (5.3) is called *complementarity condition* since it says that for every $j = 1, \dots, J$ we have $\lambda_j = 0$ or $g_j(x^*) = 0$. If $g_j(x^*) < 0$, then $\lambda_j = 0$ and hence variable λ_j does not act in the KKT conditions; this corresponds to the situation that x^* does not lie on the border of the set described by this constraint. Conversely, if $g_j(x^*) = 0$, then the complementarity makes no restriction on λ_j. In summary, we can say that the complementarity condition enforces to consider the Lagrange multipliers λ_j for the active constraints only.

Proof. (Main idea.) We linearize the problem such that the objective function and the constraint functions are replaced by their tangent hyperplanes at point x^*. This results in a linear programming problem

$$\begin{aligned} \min\ \nabla f(x^*)^T x \ \text{ subject to }\ &\nabla g_j(x^*)^T(x - x^*) \le 0, \quad j \in I(x^*),\\ &\nabla h_\ell(x^*)^T(x - x^*) = 0, \quad \ell = 1, \dots, L. \end{aligned}$$

Due to the linear independence assumption, the solution x^* remains optimal (this is a small step for a reader, but a giant leap in the proof). The dual problem to the linear program is

$$\begin{aligned} \max\ \sum_{\ell=1}^{L} \big(\nabla h_\ell(x^*)^T x^*\big)\mu_\ell + \sum_{j\in I(x^*)} \big(\nabla g_j(x^*)^T x^*\big)\lambda_j \ \text{ subject to}\\ \nabla f(x^*) + \sum_{\ell=1}^{L} \nabla h_\ell(x^*)\mu_\ell + \sum_{j\in I(x^*)} \nabla g_j(x^*)\lambda_j = 0,\\ \lambda_j \ge 0, \quad j \in I(x^*). \end{aligned}$$

Since the primal problem has an optimum, the dual problem must be feasible. For $j \notin I(x^*)$ define $\lambda_j := 0$ and we have that also the problem

$$\begin{aligned} \max\ (x^*)^T \nabla g(x^*)\lambda + (x^*)^T \nabla h(x^*)\mu \ \text{ subject to}\\ \nabla f(x^*) + \nabla h(x^*)\mu + \nabla g(x^*)\lambda = 0,\\ \lambda \ge 0 \end{aligned}$$

is feasible. Hence there exist $\lambda \ge 0, \mu$ satisfying (5.2). Condition (5.3) is fulfilled since for $j \in I(x^*)$ we have $g_j(x^*) = 0$ by definition, and for $j \notin I(x^*)$ we can put $\lambda_j = 0$. □

Conditions (5.2)–(5.3) are called *Karush–Kuhn–Tucker* conditions [Karush, 1939; Kuhn and Tucker, 1951], or KKT conditions in short.

Since the linear independence assumption is hard to check in general (notice that x^* is unknown), alternative assumptions were derived, too. Usually, they are more easy to verify but on account of stronger assumptions. One commonly used assumption is *Slater's condition*

$$\exists x^0 \in M : g(x^0) < 0.$$

Theorem 5.4. *Consider the optimization problem*

$$\min\ f(x) \ \textit{ subject to }\ g(x) \le 0,\ x \in M,$$

where $f(x), g_j(x)$ are convex functions and M is a convex set. Suppose that Slater's condition is satisfied. If x^* is an optimum of the above problem, then there exists $\lambda \ge 0$ such that x^* is an optimum of the problem

$$\min\ f(x) + \lambda^T g(x) \ \textit{ subject to }\ x \in M, \tag{5.4}$$

and, moreover, $\lambda^T g(x^*) = 0$.

Proof. Define the sets

$$\begin{aligned} \mathcal{A} &:= \{(r, s) \in \mathbb{R}^J \times \mathbb{R};\ r \ge g(x),\ s \ge f(x),\ x \in M\},\\ \mathcal{B} &:= \{(r, s) \in \mathbb{R}^J \times \mathbb{R};\ r \le 0,\ s \le f(x^*)\}; \end{aligned}$$

see Figure 5.2. Both sets are convex, and their interiors are disjoint since otherwise there is a point $x \in M$ such that $g(x) < 0$ and $f(x) < f(x^*)$. Therefore a separating hyperplane exists having the form of $\lambda^T r + \lambda_0 s = c$, where $(\lambda, \lambda_0) \ne 0$. The separability implies:

$$\begin{aligned} \forall (r, s) \in \mathcal{A} : \lambda^T r + \lambda_0 s \ge c,\\ \forall (r, s) \in \mathcal{B} : \lambda^T r + \lambda_0 s \le c. \end{aligned}$$

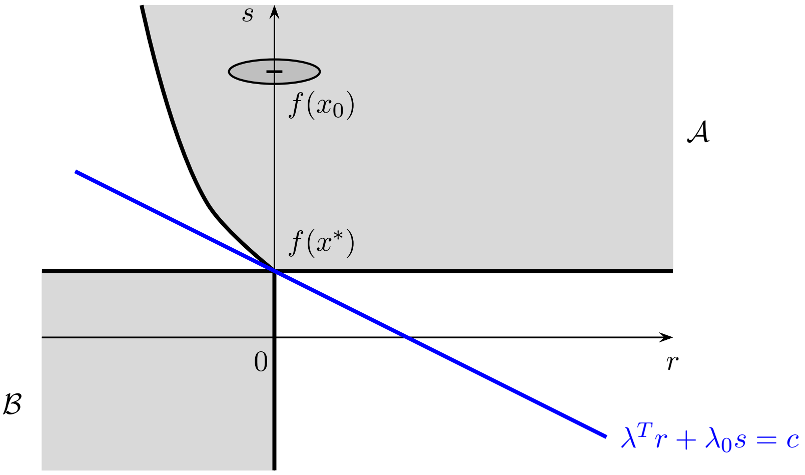


Figure 5.2: Illustration to the proof of Theorem 5.4.

Since $(0, f(x^*)) \in \mathcal{A}\cap\mathcal{B}$, this point lies on the hyperplane, and thus $c = \lambda_0 f(x^*)$. Analogously $(g(x^*), f(x^*)) \in \mathcal{A}\cap\mathcal{B}$, so this point also lies on the hyperplane, yielding

$$\lambda^T g(x^*) + \lambda_0 f(x^*) = c = \lambda_0 f(x^*),$$

which gives the complementarity constraint $\lambda^T g(x^*) = 0$.

For every i we have $(-e_i, f(x^*)) \in \mathcal{B}$, so this point lies in the negative halfspace. This means that $-\lambda^T e_i + \lambda_0 f(x^*) \leq c$, from which $\lambda_i \geq 0$. Therefore $\lambda \geq 0$. Analogously we deduce $\lambda_0 \geq 0$: Since $(o, f(x^*) - 1) \in \mathcal{B}$, so $\lambda^T o + \lambda_0(f(x^*) - 1) \leq c$, and hence $\lambda_0 \geq 0$.

Since $g(x^0) < 0$, we have $(r, f(x^0)) \in \mathcal{A}$ for every r in the neighbourhood of 0. Hence the separating hyperplane cannot be vertical, which means $\lambda_0 \neq 0$. Without loss of generality we normalize it such that $\lambda_0 = 1$. Let us prove it formally: Suppose to the contrary that $\lambda_0 = 0$. Now, $c = 0$ and in view of $\lambda \neq 0$ there is i such that $\lambda_i > 0$. Substituting the point $(-\varepsilon e_i, f(x^0)) \in \mathcal{A}$, where $\varepsilon > 0$ is small enough, into the inequality, we get $-\varepsilon\lambda_i \geq 0$; a contradiction.

For every $x \in M$ we have $(g(x), f(x)) \in \mathcal{A}$, which fulfills

$$\lambda^T g(x) + f(x) \geq c = \lambda^T g(x^*) + f(x^*).$$

This proves that x^* is the optimum of (5.4). □

Applying the optimality conditions from Theorem 2.1 to problem (5.4), we obtain the KKT conditions as a corollary:

Corollary 5.5. *Suppose that Slater's condition is satisfied for the convex optimization problem*

$$\min\ f(x) \quad \textit{subject to} \quad g(x) \leq 0.$$

If x^* is an optimum, then there exists $\lambda \geq 0$ such that the KKT conditions are satisfied, i.e.,

$$\nabla f(x^*) + \nabla g(x^*)\lambda = 0, \tag{5.5a}$$
$$\lambda^T g(x^*) = 0. \tag{5.5b}$$

We obtain also a general form involving equality constraints.

Corollary 5.6. *Suppose that Slater's condition is satisfied for the convex optimization problem*

$$\min\ f(x) \quad \textit{subject to} \quad g(x) \leq 0,\ \ Ax = b.$$

If x^* is an optimum, then there exist $\lambda \geq 0$ and μ such that the KKT conditions are satisfied, i.e.,

$$\nabla f(x^*) + \nabla g(x^*)\lambda + A^T\mu = 0,$$
$$\lambda^T g(x^*) = 0.$$

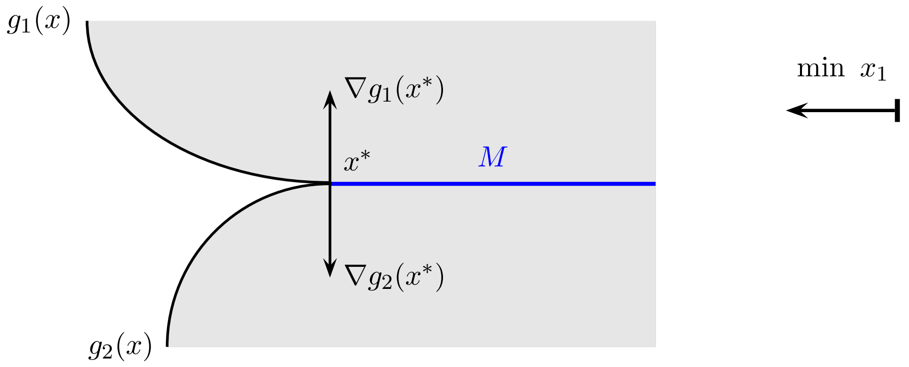


Figure 5.3: (Example 5.7) Slater's condition is not satisfied and KKT conditions property (Corollary 5.5) fails.

Example 5.7. If Slater's condition is not satisfied, then the KKT conditions property (Corollary 5.5) can fail. Consider an optimization problem $\min_{x \in M} x_1$ illustrated in Figure 5.3. Two constraints describe the feasible set having the form of a half-line starting from point x^*. Point x^* is optimal. The KKT conditions read $-\nabla f(x^*) = \nabla g(x^*)\lambda$, but the point x^* does not fulfill them since the gradients $\nabla g_1(x^*) = (0, -1)^T$ and $\nabla g_2(x^*) = (0, 1)^T$ span a vertical line, not containing the opposite of the objective function gradient $-\nabla f(x^*) = (-1, 0)^T$. □

In optimization, necessary optimality conditions are usually preferred to sufficient optimality conditions since they often help to restrict the feasible set to a smaller set of candidate optimal solutions. Anyway, sufficient optimality conditions are also of interest, and below we show that the KKT conditions do this job under general assumptions.

Theorem 5.8 (Sufficient KKT conditions)**.** *Let $x^* \in \mathbb{R}^n$ be a feasible solution of*

$$\min \ f(x) \quad \textit{subject to} \quad g(x) \leq 0,$$

let $f(x)$ be a convex function, and let $g_j(x)$, $j \in I(x^*)$, be convex functions, too. If KKT conditions (5.5) *are satisfied for x^* with certain $\lambda \geq 0$, then x^* is an optimal solution.*

Proof. Convexity of function $f(x)$ implies $f(x) - f(x^*) \geq \nabla f(x^*)^T(x - x^*)$ due to Theorem 3.18. Analogously, for functions $g_j(x)$, $j \in I(x^*)$, we have $g_j(x) - g_j(x^*) \geq \nabla g_j(x^*)^T(x - x^*)$. KKT conditions give $\nabla f(x^*) = -\nabla g(x^*)\lambda$, from which premultiplying by $(x - x^*)$ we get

$$\begin{aligned} f(x) - f(x^*) &\geq \nabla f(x^*)^T(x - x^*) = -\lambda^T \nabla g(x^*)^T (x - x^*) \\ &= -\sum_{j \in I(x^*)} \lambda_j \nabla g_j(x^*)^T (x - x^*) \geq -\sum_{j \in I(x^*)} \lambda_j (g_j(x) - g_j(x^*)) \\ &= -\sum_{j \in I(x^*)} \lambda_j g_j(x) \geq 0. \end{aligned}$$

Therefore $f(x^*)$ is the optimal value and x^* is an optimal solution. □

Chapter 6

Methods

To solve an optimization problem is a very difficult task in general; indeed, it is undecidable (provably there cannot exist an algorithm)! Thus we can hardly hope to solve optimally every problem. Many algorithms thus produce approximate solutions only – KKT solutions, local optima etc. If the problem is large and hard, then we often use heuristic methods (genetic and evolutionary algorithms, simulated annealing, tabu search,...). On the other hand, many hard optimization problems can be solved by using global optimization techniques. However, they work in small dimensions only since their computational complexity is high. The choice of a suitable method thus depends not only on the type of the problem, but also on the dimensions, time restrictions etc.

In the following sections, we present selected methods for basic types of optimization problems.

6.1 Line search

By a line search we mean minimization of a univariate function $f(x)\colon \mathbb{R} \to \mathbb{R}$, that is, we have $n = 1$. Even this particular case is important since it often serves as an auxiliary sub-procedure in the general case.

Our goal is to find a local minimum (or its approximation) in the neighbourhood of the current point. We present two approaches: Armijo rule, which aims to move to a point of a significant decrease of the objective function, and the Newton method, which converges to a local minimum under certain conditions.

Armijo rule

We assume that $f(x)$ is differentiable and $f'(0) < 0$, so it locally decreases at $x = 0$. We want to decrease the objective function by moving to the right from point $x = 0$. We wish to decrease it significantly, that is, not to get stuck locally close to $x = 0$, but to move away from this current point if possible.

Consider the condition

$$f(x) \leq f(0) + \varepsilon \cdot f'(0) \cdot x, \tag{6.1}$$

where $0 < \varepsilon < 1$ is a given parameter; usually we take $\varepsilon \approx 0.2$.

The condition is used as follows: Choose a value of parameter $\beta > 0$ (e.g., $\beta = 2$ or $\beta = 10$) and an arbitrary $x > 0$. Now

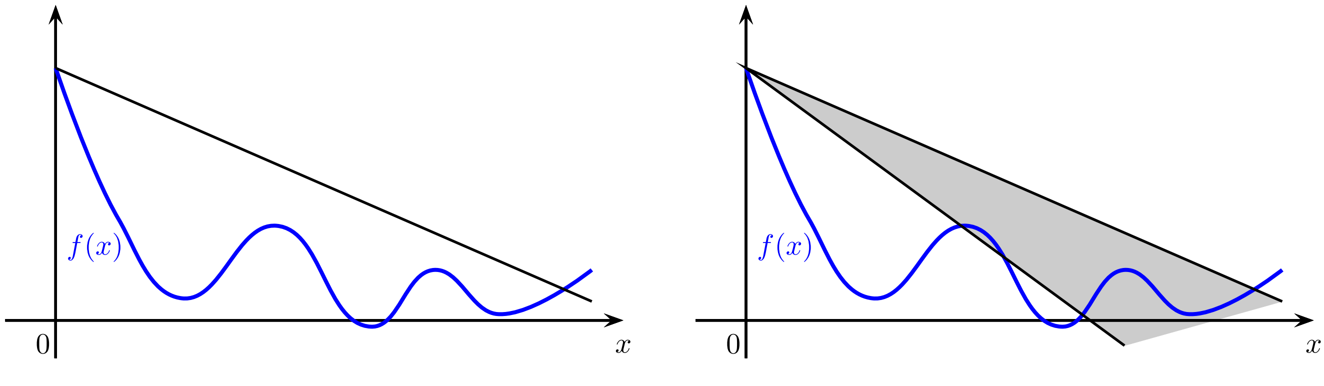
- if condition (6.1) is satisfied, then set $x := \beta x$ and while the condition holds, repeat this process;
- if condition (6.1) is not satisfied, then set $x := x/\beta$ and repeat until the condition holds.

This procedure ensures that we move to a point with smaller objective value and simultaneously we move far from the initial point; see Figure 6.1a.

Armijo rule is also used as the termination condition within other line search methods: Condition (6.1) cannot be violated (which ensures that x is not too large) and simultaneously the converse inequality

$$f(x) \geq f(0) + \varepsilon' \cdot f'(0) \cdot x,$$

must be satisfied for certain parameter $\varepsilon' > \varepsilon$ (which ensures that x is not too large small); see Figure 6.1b.



(a) Armijo rule – seeking for the intersection point.

(b) Armijo rule as the termination condition.

Figure 6.1: Armijo rule

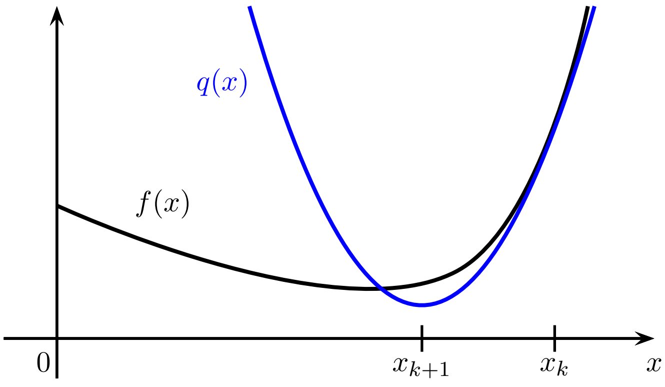


Figure 6.2: Newton method: approximation $f(x)$ at point x_k by a quadratic function $q(x)$, and move to its minimum x_{k+1}.

Newton method

It is the classical Newton method for finding a root of $f'(x) = 0$. Here we need $f(x)$ to be twice differentiable.

This method is iterative and we construct a sequence of points $x_0 = 0, x_1, x_2, \ldots$ that, under some assumptions, converge to a local minimum. The basic idea is to approximate function $f(x)$ by a function $q(x)$ such that they both have the same value and the first and second derivatives and the current point x_k (in the kth iteration). Thus we want $q(x_k) = f(x_k)$, $q'(x_k) = f'(x_k)$ and $q''(x_k) = f''(x_k)$; see Figure 6.2. This suggests that it is suitable for $q(x_k)$ to be a quadratic polynomial. Such a quadratic function is unique and it is described

$$q(x) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2.$$

(Proof: put $x := x_k$.) The minimum of quadratic function $q(x_k)$ is at the stationary point (where the derivative is zero), so

$$0 = f'(x_k) + f''(x_k)(x - x_k).$$

From this we get

$$x = x_k - \frac{f'(x_k)}{f''(x_k)},$$

which is the current point x_{k+1} of the subsequent iteration.

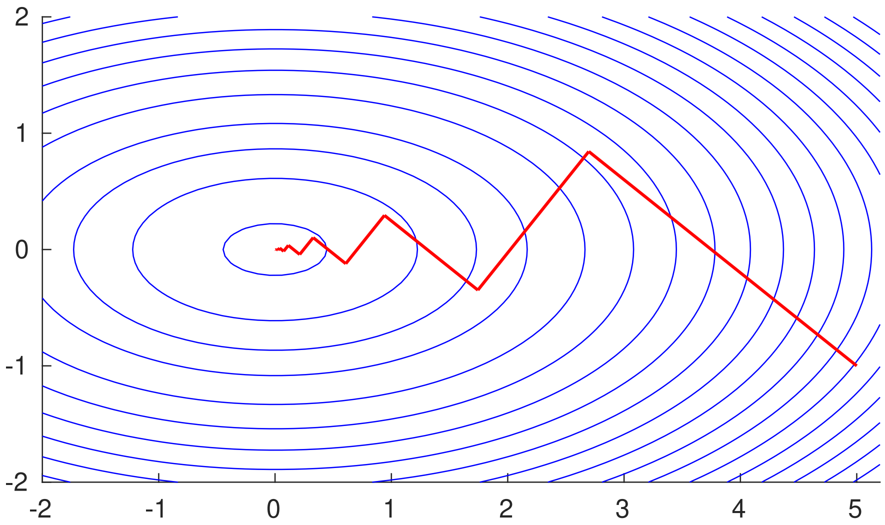


Figure 6.3: In blue: the contours of the convex quadratic function $f(x) = x_1^2 + 4x_2^2$. In red: the iterations of the steepest descent method with initial point $(5, -1)^T$.

6.2 Unconstrained problems

Consider the optimization problem

$$\min \ f(x) \ \text{ subject to } \ x \in \mathbb{R}^n,$$

where $f(x)$ is a differentiable function.

A basic approach is an iterative method, generating a sequence of points $x_0, x_1, x_2, \ldots$, which, under certain assumptions, converge to a local minimum. The initial point x_0 can be chosen arbitrarily, unless we have some additional information that we can utilize. The iterations terminate when the objective function values at points x_k get stabilized.

Gradient methods[1)]

In kth iteration, the current point is x_k. We determine a direction d_k in which the objective function locally decreases, that is, $\nabla f(x_k)^T d_k < 0$. Now we call a line search method applied to the function $\varphi(\alpha) := f(x_k + \alpha d_k)$. Denote by α_k the output value. Then the next point is set as $x_{k+1} := x_k + \alpha_k d_k$.

How to choose d_k? The simplest way is *the steepest descent method*, which takes $d_k := -\nabla f(x_k)$, that is, the direction in which the objective function locally decreases the most rapidly. This choice need no be the best one; see Figure 6.3, which illustrates the slow convergence even for the simple convex quadratic function $f(x) = x_1^2 + 4x_2^2$. There are advanced methods that take into account also the Hessian $\nabla^2 f(x_k)$ or its approximation and they combine the steepest descent direction and the directions of the previous iteration(s); see also the conjugate gradient methods in Section 6.4.

Example 6.1 (Learning of neural networks)**.** Basically, the steepest descent method is used in learning of artificial neural networks (for an introduction see Higham and Higham [2019]). The goal of the learning is to set up weights of inputs of particular neurons such that the neural network performs best on the training data. Mathematically speaking, the variables are the weights of inputs of the neurons. The objective function that we minimize is the distance between the actual output vector and the ideal output vector. It is hard ho find the optimal solution since this optimization problem is nonlinear, nonconvex and high-dimensional. That is why the problem is solved iteratively and at each step the weights are refined by means of the steepest descent. To compute the gradient of the objective function is also computationally

[1)] The history of gradient methods dates back to 1847, when L.A. Cauchy introduced a gradient-like method to solve the astronomical problem of calculating the orbit of a celestial body.

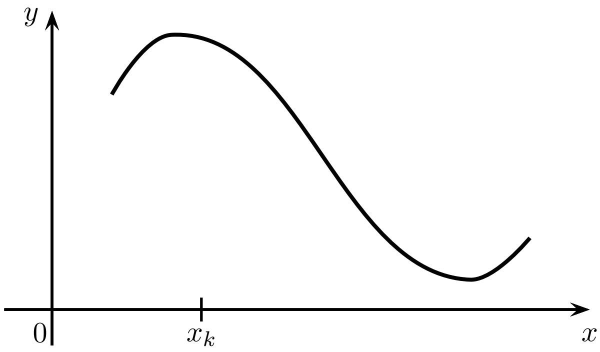


Figure 6.4: The Hessian $\nabla^2 f(x_k)$ is not positive definite.

demanding since there are usually large training data, so we simplify further and we approximate the gradient by its partial value based on the gradient of a randomly chosen training sample point. This approach is called *stochastic gradient descent.* □

Example 6.2. Optimization techniques are also used to solve problems that are not optimization problems in the essence. Consider for example the problem of solving a system of linear equations $Ax = b$, where A is a positive definite matrix. Then the optimal solution of the convex quadratic program

$$\min_{x \in \mathbb{R}^n} \ \frac{1}{2} x^T A x - b^T x$$

is the point $A^{-1}b$, the same as the solution of the equations, since at this point the gradient $\nabla f(x) = Ax - b$ of the objective function $f(x) = \frac{1}{2}x^T Ax - b^T x$ vanishes. Thus we can solve linear equations by using optimization techniques. This is really used in practice, in particular for large and sparse systems. There exist several ways how to choose the vector d_k in this context. For instance, the conjugate gradient method combines the gradient and the previous direction, so it takes a linear combination of $\nabla f(x_k)$ and d_{k-1}; see Section 6.4. □

Newton method

This works in a similar fashion as in the univariate case. We approximate the objective function by a quadratic function, whose minimum is the current point of the subsequent iteration.

In step k, the current point is x_k and at this point we approximate $f(x)$ by using Taylor expansion

$$f(x) \approx f(x_k) + \nabla f(x_k)^T (x - x_k) + \frac{1}{2}(x - x_k)^T \nabla^2 f(x_k)(x - x_k).$$

This gives us a quadratic function. If its Hessian matrix $\nabla^2 f(x_k)$ is positive definite, then its minimum is unique and it is the point with zero gradient. This leads us to the system

$$\nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) = 0,$$

from which we express the solution

$$x = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k).$$

This point is set as the current point x_{k+1} of the next iteration.

Comment. The expression $y := (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$ is evaluated by solving the system of linear equations $\nabla^2 f(x_k) y = \nabla f(x_k)$, not by inverting the matrix.

The advantage of this method is a rapid convergence (if we are close to the minimum). The drawback is that the Hessian $\nabla^2 f(x_k)$ need not be positive definite; see example on Figure 6.4. Another drawback is that the evaluation of the Hessian might be computationally demanding. Therefore, diverse variants of this method exist (quasi-Newton methods) that approximate the Hessian matrix or regularize it.

6.3 Constrained problems

Consider the optimization problem

$$\min\ f(x)\ \text{ subject to }\ x \in M,$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a differentiable function and the feasible set $M \subseteq \mathbb{R}^n$ is characterized by the system

$$\begin{aligned} g_j(x) &\le 0, \quad j = 1, \dots, J,\\ h_\ell(x) &= 0, \quad \ell = 1, \dots, L, \end{aligned}$$

where $g_j(x), h_\ell(x)\colon \mathbb{R}^n \to \mathbb{R}$.

The solution methods are again iterative, where we construct a sequence of points $x_0, x_1, x_2, \dots$ The initial point x_0 is chosen randomly, unless we have some additional knowledge about the problem. The iterations terminate when the objective function values at points x_k get stabilized.

6.3.1 Methods of feasible directions

These methods naturally the generalize gradient methods from unconstrained optimization. The basic idea is the same and the only difference is in the line search, when we must stay within the feasible set M. The equality constraints $h(x) = 0$ are hard to deal with in this case.

These methods are particularly convenient when M is a convex polyhedron. So in this section we assume that $M = \{x \in \mathbb{R}^n;\ Ax \le b\}$.

Method by Frank and Wolfe [1956]

Let x_k be the current feasible point in kth iteration. A feasible descent direction d_k is computed by an auxiliary linear program

$$\min\ \nabla f(x_k)^T x\ \text{ subject to }\ Ax \le b.$$

Denote by x_k^* its optimal solution. Then we take $d_k := x_k^* - x_k$. This direction is feasible since $x_k^* \in M$. Moreover, d_k corresponds to a steep descent since the objective function $\nabla f(x_k)^T(x - x_k)$ yields the derivative of function f at point x_k in the direction of $x - x_k$ (the term $\nabla f(x_k)^T x_k$ is negligible since it is constant).

Method by Zoutendijk [1960]

This method is similar to the previous one, but the auxiliary problem has the form

$$\min\ \nabla f(x_k)^T x\ \text{ subject to }\ Ax \le b,\ \|x - x_k\| \le 1.$$

If we use the Euclidean norm, then we are seeking for the steepest descent direction that is feasible. In order that the auxiliary problem is easy to solve, we usually employ the maximum or the Manhattan norm. For the latter, for example, the problem takes the form of a linear program, in which $\|x - x_k\| \le 1$ is replaced by

$$e^T z \le 1, \quad x - x_k \le z, \quad -x + x_k \le z.$$

6.3.2 Active-set methods

These methods reduce the problem to a sequence of optimization problems with equality constraints only.

Let x_k be a current feasible solution and let

$$W := \{j;\ g_j(x_k) = 0\}$$

be the active set. Then we solve an auxiliary problem

$$\min\ f(x)\ \text{ subject to }\ h(x) = 0, \quad g_j(x) = 0,\ j \in W.$$

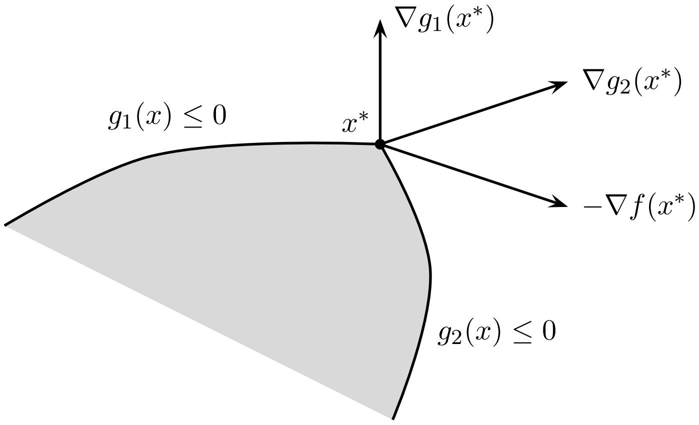


Figure 6.5: At point x^* we have $\nabla f(x^*) - \nabla g_1(x^*) + \nabla g_2(x^*) = 0$, so index 1 is removed from the active set and index 2 remains there.

If we move to the boundary of M during the computation and another constraint becomes active, then we include it to W. If we achieve a local minimum x^* during the computation of this auxiliary problem, then we assume that KKT conditions are satisfied. That is, there exists λ such that

$$\nabla f(x^*) + \nabla h(x^*)\mu + \sum_{j\in W} \lambda_j \nabla g_j(x^*) = 0.$$

Now, if $\lambda_j \geq 0$, then j remains in W; otherwise the index j is removed from W. This treatment is based on the interpretation of Lagrange multipliers as the negative derivatives of the objective function with respect to the right-hand side of the constraints. Hence, $\lambda_j < 0$ implies that locally a decrease of $g_j(x)$ makes a decrease of $f(x)$; see Figure 6.5.

The schema of this method resembles the simplex method in linear programming, in which we move from one feasible basis to another and dynamically change the active set. Therefore, the active-set method is primarily used in optimization problems with linear constraints.

6.3.3 Penalty and barrier methods

These methods transform the problem in such a way that the constraint functions are added to the objective function and the problem is reduced to an unconstrained problem (in fact, to a series of unconstrained problems). This transformation works such that we pay a penalization in the form of higher objective values for infeasible points (penalty methods), or we force the computed points to stay in the interior of the feasible set by increasing the objective function values to infinity on its boundary (barrier methods).

Penalty methods

Consider the problem

$$\min\ f(x)\ \text{ subject to }\ x \in M,$$

where $f(x)$ is a continuous function and $M \neq \emptyset$ is a closed set. *A penalty function* is any continuous nonnegative function $q\colon \mathbb{R}^n \to \mathbb{R}$ satisfying the conditions:

- $q(x) = 0$ for every $x \in M$,
- $q(x) > 0$ for every $x \notin M$.

Penalty methods are based on a transformation of the problem to an unconstrained problem

$$\min\ f(x) + c \cdot q(x)\ \text{ subject to }\ x \in \mathbb{R}^n,$$

where $c > 0$ is a parameter.

Penalty methods are implemented such that c is not constant, but it is increased during the iterations. Too high value of c at the beginning leads to a numerically ill-conditioned problem. That is why in practice the values from a suitable sequence $c_k > 0$, where $c_k \to_{k\to\infty} \infty$, are used.

Theorem 6.3. *Let x_k be an optimal solution of problem*

$$\min\ f(x) + c_k \cdot q(x) \ \ \textit{subject to} \ \ x \in \mathbb{R}^n.$$

If $x_k \to_{k\to\infty} x^*$, then x^* is an optimal solution of the original problem $\min_{x\in M} f(x)$.

Proof. If $x^* \not\in M$, then for k^* large enough we have $x_k \not\in M\ \forall k \geq k^*$, and thus the objective function grows without bound. Hence $f(x^*) + c_k \cdot q(x^*) \to_{k\to\infty} \infty$ and also $f(x_k) + c_k \cdot q(x_k) \to_{k\to\infty} \infty$, which contradicts optimality of x_k.

Consider now the case of $x^* \in M$ and suppose to the contrary that x^* is not optimal. Then there is a point $x' \in M$ such that $f(x') < f(x^*)$. Since the penalization is zero within the feasible set M, we get

$$f(x') + c_k \cdot q(x') < f(x^*) + c_k \cdot q(x^*)$$

for every $k \in \mathbb{N}$. Due to continuity we have for sufficiently large k

$$f(x') + c_k \cdot q(x') < f(x_k) + c_k \cdot q(x_k),$$

which is a contradiction to the optimality of x_k in iteration k. □

Example 6.4. For constraints of type $g(x) \leq 0$ we often use the penalty function

$$q(x) := \sum_{j=1}^{J} (g_j(x)^+)^2 = \sum_{j=1}^{J} \max(0, g_j(x))^2,$$

which preserves smoothness of the objective function, and for constraints of type $h(x) = 0$ we can use the penalty function

$$q(x) := \sum_{\ell=1}^{L} h_\ell(x)^2.$$

Barrier methods

Consider again the problem

$$\min\ f(x) \ \ \text{subject to} \ \ x \in M,$$

where $f(x)$ is a continuous function. Suppose that M is a connected set satisfying $M = \operatorname{cl}(\operatorname{int} M)$, that is, it is equal to the closure if its interior. *A barrier function* is any continuous nonnegative function $q\colon \operatorname{int} M \to \mathbb{R}$ such that $q(x) \to \infty$ for every $x \to \partial M$. This means that when x approaches to the boundary of M, then the barrier function grows to infinity.

The original problem is then transformed to an unconstrained problem

$$\min\ f(x) + \frac{1}{c}\, q(x) \ \ \text{subject to} \ \ x \in \mathbb{R}^n, \tag{6.2}$$

where $c > 0$ is a parameter.

The algorithm is similar to penalty methods, that is, we iteratively seek for optimal solutions of auxiliary problems when $c \to \infty$. A drawback of this method if that we have to know an initial feasible solution at the beginning. The advantage is its simplicity.

The pioneers of these methods are Fiacco and McCormick [1968].

Example 6.5. For constraints of type $g(x) \leq 0$ we often use the barrier function in the form

$$q(x) := -\sum_{j=1}^{J} \frac{1}{g_j(x)}$$

or in the form

$$q(x) := -\sum_{j=1}^{J} \log(-g_j(x)).$$

The latter is utilized in the popular interior point methods, which implementations can solve linear programs and certain convex optimization problems (such as quadratic programs) in polynomial time. For example, the linear program

$$\min\ c^T x \ \text{ subject to } \ Ax \leq b$$

is transformed to the problem

$$\min\ c^T x - \frac{1}{c} \sum_{i=1}^{m} \log(b_i - A_{i*}x).$$

For semidefinite condition $X \succeq 0$ we can use the barrier function

$$q(X) := -\log(\det(X)).$$

Under certain assumptions the optimal solutions of the auxiliary problems converge to the optimum of the original problem.

Theorem 6.6. *Let $c_k > 0$ be a sequence of numbers such that $c_k \to_{k\to\infty} \infty$. Let x_k be an optimal solution of problem*

$$\min\ f(x) + \frac{1}{c_k} q(x) \ \textit{ subject to } \ x \in \mathbb{R}^n.$$

If $x_k \to_{k\to\infty} x^*$, then x^* is an optimal solution of the original problem $\min_{x\in M} f(x)$.

Proof. Suppose to the contrary that x^* is not optimal, that is, there is $x' \in M$ such that $f(x') < f(x^*)$. Due to continuity of $f(x)$ there is $x'' \in \operatorname{int} M$ such that $f(x'') < f(x^*)$. Then for k large enough we have

$$f(x'') + \frac{1}{c_k} q(x'') < f(x^*) + \frac{1}{c_k} q(x^*).$$

For k large enough we also have

$$f(x'') + \frac{1}{c_k} q(x'') < f(x_k) + \frac{1}{c_k} q(x_k),$$

which is a contradiction to the optimality of x_k in step k. □

For convex optimization problems under general assumptions (e.g., strictly convex barrier function and M bounded) the optimal solution $x(c)$ of (6.2) is unique and the points $x(c)$, $c > 0$, draw a smooth curve, called *the central path*, whose limit as $c \to \infty$ is the optimal solution of the original problem.

Certain algorithms use the same principle: For the increasing values of c they find (approximation of) the optimal solutions $x(c)$. With a small change of c the point $x(c)$ moves continuously, so it is easy and fast to reoptimize and find the new optimum. For theoretical analysis of polynomiality of certain convex optimization problems short steps are used, but in practice larger steps are convenient. Typically, we increase c with a factor of 1.1.

A natural question is, why not to choose a large value of c at the beginning? The numerical issues cause troubles then. Next, such a choice makes not the algorithm faster. The Newton method (or other methods used to solve (6.2)) is slow if we start far from the optimum. Therefore tracing the central path using fast steps is the most convenient way. Notice that we have some difficulties at the beginning, but this issue can be overcome.

6.4 Conjugate gradient method

This method was derived to solve a system of linear equations $Ax = b$, where matrix $A \in \mathbb{R}^{n\times n}$ is positive definite. Its authors are Hestenes and Stiefel [1952], and it belongs to both optimization textbooks [Luenberger and Ye, 2008] and textbooks on numerical mathematics [Liesen and Strakoš, 2013]. Even though the methods is iterative, it converges to the solution in at most n steps. Since it does not transform matrix A and has low space complexity, its is convenient for very large systems in particular.

The basic idea is to consider the quadratic function

$$f(x) = \frac{1}{2}x^TAx - b^Tx.$$

Since A is positive definite, the function is strictly convex and attains the unique minimum. The minimum is the point, in which the gradient $\nabla f(x) = Ax - b$ is zero. Hence the minimum of function $f(x)$ is the same as the solution of $Ax = b$. In this way we reduced the problem of solving linear equations to an optimization problem.

We will describe the method is a simplified way. First, instead of the standard basis of space $\mathbb{R}^n$ we consider an orthonormal basis $d_1, \dots, d_n$ and the inner product $\langle x, y\rangle := x^TAy$ instead of the standard one; to avoid confusion, the corresponding orthogonality is called A-orthogonality and the orthonormal basis is called A-orthonormal. We will show later on how to choose the basis $d_1, \dots, d_n$. Denote by $x^* := A^{-1}b$ the solution we are seeking for, and denote by x_k an approximate solution obtained in kth iteration. At the beginning, the initial points x_1 is chosen arbitrarily.

Basic scheme. We express vector $x^* - x_1$ as a linear combination of the basis vectors

$$x^* - x_1 = \sum_{k=1}^{n} \alpha_k d_k.$$

The basic scheme of the method is simple – imagine we move from a vertex of a box to the opposite vertex by using the (mutually perpendicular) edges:

Iterate $x_{k+1} := x_k + \alpha_k d_k$, $k = 1, \dots$

To implement the method we need to determine the basis $d_1, \dots, d_n$ and show how to compute coefficients α_k effectively. Denote $g_k := \nabla f(x_k) = Ax_k - b$, which represents not only the gradient at point x_k in kth iteration, but also the residual, that is, the difference between the left and right-hand sides of the system (when the residual is 0, then we get the solution). Notice that for any $j \in \{1, \dots, k\}$,

$$x_{k+1} = x_k + \alpha_k d_k = x_{k-1} + \alpha_k d_k + \alpha_{k-1}d_{k-1} = \ldots = x_j + \sum_{i=j}^{k} \alpha_i d_i.$$

Computation of α_k. Since $d_1, \dots, d_n$ is an A-orthonormal basis, the coordinates α_k are the Fourier coefficients and we compute them easily as $\alpha_k = \langle d_k, x^* - x_1\rangle$. The problem is that x^* is unknown. Since $x_k - x_1 = \sum_{i=1}^{k-1} \alpha_i d_i$, vector $x_k - x_1$ is A-orthogonal to d_k, that is, $\langle d_k, x_k - x_1\rangle = 0$. We derive

$$\begin{aligned}\alpha_k &= \langle d_k, x^* - x_1\rangle = \langle d_k, x^* - x_k + x_k - x_1\rangle = \langle d_k, x^* - x_k\rangle + \langle d_k, x_k - x_1\rangle \\ &= \langle d_k, x^* - x_k\rangle = d_k^TA(x^* - x_k) = d_k^T(b - Ax_k) = -d_k^Tg_k.\end{aligned}$$

Proposition 6.7. *Vector x_{k+1} is the minimum of $f(x)$ on the affine subspace $x_1 + \operatorname{span}\{d_1, \dots, d_k\}$, that is, $g_{k+1}^Td_j = 0$ for $j = 1, \dots, k$ (i.e., $g_{k+1} \perp d_j$ meaning the standard orthogonality).*

Proof. It is sufficient to show that vector $\nabla f(x_{k+1}) = g_{k+1}$ is perpendicular to subspace $x_1 + \operatorname{span}\{d_1, \dots, d_k\}$, that is, it is perpendicular to every vector $d_1, \dots, d_k$. Write

$$g_{k+1} = Ax_{k+1} - b = A\left(x_j + \textstyle\sum_{i=j}^{k} \alpha_i d_i\right) - b$$

For any $j \in \{1, \dots, k\}$ we have

$$\begin{aligned}d_j^Tg_{k+1} &= d_j^T\left(A\left(x_j + \textstyle\sum_{i=j+1}^{k} \alpha_i d_i\right) - b\right) = d_j^T(Ax_j - b) + d_j^TA\left(\textstyle\sum_{i=j}^{k} \alpha_i d_i\right) \\ &= d_j^Tg_j + \alpha_j = 0.\end{aligned}$$

□

The choice of basis $d_1, \dots, d_n$. We choose the basis such that $\operatorname{span}\{d_1, \dots, d_k\} = \operatorname{span}\{g_1, \dots, g_k\}$ for every $k = 1, \dots, n$. At the beginning we naturally put $d_1 := -g_1/\sqrt{\langle g_1, g_1\rangle}$. In $(k+1)$st iteration we construct vector d_{k+1} from vector $-g_{k+1}$ by making it orthogonal to subspace $\operatorname{span}\{d_1, \dots, d_k\}$.

Proposition 6.8. *We have* $\operatorname{span}\{g_1, \dots, g_k\} = \operatorname{span}\{g_1, Ag_1, \dots, A^{k-1}g_1\}$.

Proof. We prove it by mathematical induction on k. By definition and from the induction hypothesis we have $g_k = Ax_k - b$, where

$$x_k \in x_1 + \operatorname{span}\{g_1, \dots, g_{k-1}\} = x_1 + \operatorname{span}\{g_1, Ag_1, \dots, A^{k-2}g_1\}.$$

Hence

$$g_k \in Ax_1 - b + \operatorname{span}\{Ag_1, A^2g_1, \dots, A^{k-1}g_1\} \subseteq \operatorname{span}\{g_1, Ag_1, A^2g_1, \dots, A^{k-1}g_1\}.$$

In fact, we have equality since g_k does not belong to $\operatorname{span}\{g_1, \dots, g_{k-1}\}$. Otherwise, according to Proposition 6.7, vector g_k is orthogonal to this subspace and $g_k = Ax_k - b$ must be zero, meaning that x_k is the solution x^*. □

Since g_{k+1} is orthogonal (in the standard sense) to vectors $d_1, \dots, d_k$, it is also orthogonal to $g_1, \dots, g_k$, and by Proposition 6.8 it is A-orthogonal to vectors $g_1, \dots, g_{k-1}$, too. Thus, in order to compute d_{k+1}, it is sufficient to make $-g_{k+1}$ orthogonal to vector d_k. This is performed by the following statement. Notice that the resulting value of d_{k+1} is not normalized, so we have to normalize it afterwards.

Proposition 6.9. *We have* $d_{k+1} = -g_{k+1} + \beta_{k+1}d_k$*, where* $\beta_{k+1} = \langle d_k, g_{k+1}\rangle$.

Proof. We already know that $\langle g_{k+1}, d_i\rangle = 0$ for $i = 1, \dots, k-1$. Hence d_{k+1} has the form of $d_{k+1} = -g_{k+1} + \beta_{k+1}d_k$ for certain β_{k+1}. From the equality $0 = \langle d_k, d_{k+1}\rangle = d_k^T A(-g_{k+1} + \beta_{k+1}d_k)$ we derive the value of $\beta_{k+1} = \frac{d_k^T Ag_{k+1}}{d_k^T Ad_k} = \frac{\langle d_k, g_{k+1}\rangle}{\langle g_k, g_k\rangle} = \langle d_k, g_{k+1}\rangle$. □

Summary. Now we have all the ingredients to explicitly write the algorithm:

```
1: choose $x_1 \in \mathbb{R}^n$ and put $d_0 := 0$,
2: for $k = 1, \dots, n$ do
       $g_k := Ax_k - b,$
       $\beta_k := d_{k-1}^T Ag_k,$
       $d_k := -g_k + \beta_k d_{k-1}, \quad d_k := d_k/\sqrt{d_k^T Ad_k},$
       $\alpha_k := -d_k^T g_k,$
       $x_{k+1} := x_k + \alpha_k d_k,$
```

Remark 6.10. A few of comments to the conjugate gradient method:

(1) The method has low memory requirement and makes no operations on matrix A. The method is beneficial particularly when matrix A is large and sparse. The running time of one iteration is relatively low. Moreover, not all n iterations are needed to perform in general – we can achieve the solution or its tight approximation much sooner.

(2) Often the method is presented without the normalization of vector d_k. Then the expressions with d_k must be adjusted accordingly.

(3) If we choose $x_1 = 0$, then $\operatorname{span}\{d_1, \dots, d_k\} = \operatorname{span}\{b, Ab, A^2b, \dots, A^{k-1}b\}$ is called the Krylov subspace and the theory behind is very interesting [Liesen and Strakoš, 2013].

The basic idea of the conjugate gradient method can be used to minimize a general nonlinear function $f(x)$ over space $\mathbb{R}^n$. Herein, the key idea is to construct the improving direction d_k as a linear combination of gradient g_k and the previous direction d_{k-1}. Vector g_k is then the gradient of function $f(x)$ at point x_k, and the coefficients are computed analogously. The resulting method is called the method of Fletcher–Reeves (1964). There exist several variants, which differ in the values of coefficients β_k.

There are also methods employing Krylov subspaces for solving systems $Ax = b$, where matrix A is not necessarily symmetric positive definite. For example, let us mention GMRES (Generalized minimal residual method, Saad & Schultz, 1986), which in kth iteration computes vector x_k that minimizes the Euclidean norm of the residual (i.e., $\|Ax - b\|$) over subspace $\operatorname{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\}$.

Chapter 7

Selected topics

7.1 Robust optimization

In practice, data are often inexact or subject to various uncertainties. This motivates us to seek for solutions that are *robust*. There is no precise definition, but basically it means that a robust solution is feasible and optimal even for specific data perturbations; see Ben-Tal et al. [2009]. We present two approaches to robustness, the interval one and the ellipsoidal one.

Interval uncertainty (I)

Consider first a linear program in the form

$$\min\ c^T x \ \text{ subject to }\ Ax \le b,\ x \ge 0.$$

Suppose that A and b are not known exactly and the only information that we have are interval estimations of the values. That is, we know a matrix of intervals $[\underline{A}, \overline{A}]$ and the vector of interval right-hand sides $[\underline{b}, \overline{b}]$. We say that a vector x is a robust feasible solution if it fulfills inequality $Ax \le b$ for each $A \in [\underline{A}, \overline{A}]$ and $b \in [\underline{b}, \overline{b}]$. Due to nonnegativity of variables we have that x is robust feasible if and only if $\overline{A}x \le \underline{b}$. Hence the robust counterpart of the linear programu reads

$$\min\ c^T x \ \text{ subject to }\ \overline{A}x \le \underline{b},\ x \ge 0.$$

Example 7.1 (Catfish diet problem)**.** This example comes from `http://www.fao.org/3/x5738e/x5738e0h.htm`. It is a simplified example of an optimization model of finding a minimum cost catfish diet in Thailand. The mathematical formulation reads

$$\min\ c^T x \ \text{ subject to }\ Ax \ge b,\ x \ge 0, \tag{7.1}$$

where variable x_j stands for the number of units of food j to be consumed by the catfish, b_i is the required minimal amount of nutrient i, c_j is the price per unit of food j, and a_{ij} is the amount of nutrient i contained in one unit of food j. The data are recorded in Table 7.1. Thus we have

$$A = \begin{pmatrix} 9 & 65 & 44 & 12 & 0 \\ 1.10 & 3.90 & 2.57 & 1.99 & 0 \\ 0.02 & 3.7 & 0.3 & 0.1 & 38.0 \end{pmatrix}, \quad b = \begin{pmatrix} 30 \\ 250 \\ 0.5 \end{pmatrix}, \quad c = \begin{pmatrix} 2.15 \\ 8.0 \\ 6.0 \\ 2.0 \\ 0.4 \end{pmatrix}.$$

Since the nutritive values are not known exactly, we assume that their accuracy is 5%. Hence the exact value of each entry of matrix A lies in interval $[0.95 \cdot a_{ij}, 1.05 \cdot a_{ij}]$. According to the lines described above, the robust counterpart is obtained by setting the constraint matrix to be $\underline{A}$, that is,

$$\underline{A} = \begin{pmatrix} 8.550 & 61.75 & 41.800 & 11.400 & 0.00 \\ 1.045 & 3.705 & 2.4415 & 1.8905 & 0.00 \\ 0.019 & 3.515 & 0.2850 & 0.0950 & 36.1 \end{pmatrix}. \qquad \square$$

	Cost (THB/kg)	Protein (%)	Energy (Mcal/kg)	Calcium (%)
Maize	2.15	9	1.10	0.02
Fishmeal	8.0	65	3.90	3.7
Soymeal	6.0	44	2.57	0.3
Ricebran	2.0	12	1.99	0.1
Limestone	0.4	0	0	38.0
Demand	min	30	250	0.5

Table 7.1: (Example 7.1) Catfish diet problem: Nutritive value of foods and the nutritional demands

Interval uncertainty (II)

Consider now a linear program in the form with variables unrestricted in sign

$$\min\ c^T x \ \text{ subject to } \ Ax \le b.$$

Let $a^T x \le d$ be a selected inequality. Let intervals $[\underline{a}, \overline{a}] = ([\underline{a}_1, \overline{a}_1], \dots, [\underline{a}_n, \overline{a}_n])^T$ and $[\underline{d}, \overline{d}]$ be given. A solution x is a robust solution of the selected inequality if it satisfies

$$a^T x \le d \quad \forall a \in [\underline{a}, \overline{a}],\ \forall d \in [\underline{d}, \overline{d}],$$

or,

$$\max_{a \in [\underline{a}, \overline{a}]} a^T x \le \underline{d}.$$

Lemma 7.2. *Denote by $a_\Delta = \frac{1}{2}(\overline{a} - \underline{a})$ the vector of interval radii and by $a_c = \frac{1}{2}(\underline{a} + \overline{a})$ the vector of interval midpoints. Then*

$$\max_{a \in [\underline{a}, \overline{a}]} a^T x = a_c^T x + a_\Delta^T |x|.$$

Proof. For every $a \in [\underline{a}, \overline{a}]$ we have

$$a^T x = a_c^T x + (a - a_c)^T x \le a_c^T x + |a - a_c|^T |x| \le a_c^T x + a_\Delta^T |x|.$$

The inequality is attained as equation for certain $a \in [\underline{a}, \overline{a}]$. If $x \ge 0$, then $a_c^T x + a_\Delta^T |x| = a_c^T x + a_\Delta^T x = \overline{a}^T x$. If $x \le 0$, then $a_c^T x + a_\Delta^T |x| = a_c^T x - a_\Delta^T x = \underline{a}^T x$. Otherwise we apply this idea entrywise, so that inequality is attained as equation for a each entry of which is the interval left or right endpoint. □

We use this lemma to express the robust solution constraint as

$$a_c^T x + a_\Delta^T |x| \le \underline{d}.$$

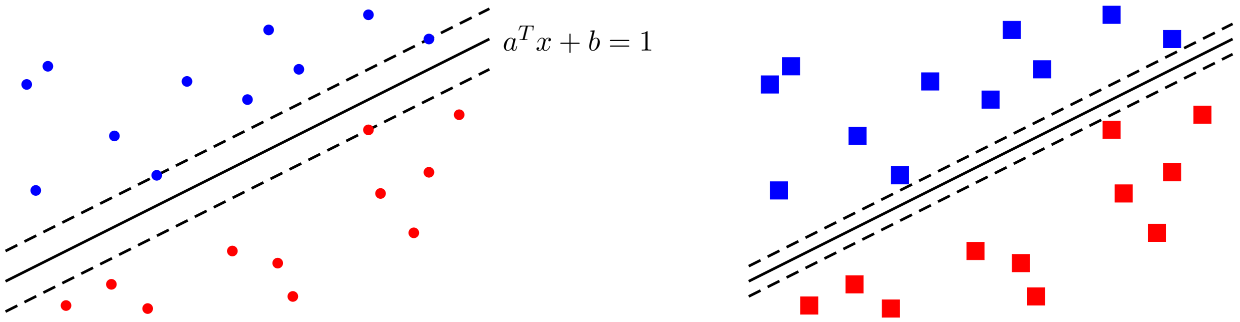
The left-hand side function is convex, but not smooth. Nevertheless, we can rewrite the constraint as a linear constraint by introducing an auxiliary variable $y \in \mathbb{R}^n$

$$a_c^T x + a_\Delta^T y \le \underline{d}, \quad x \le y, \quad -x \le y.$$

Therefore linearity is preserved – the robust solutions of interval linear programs are also described by linear constraints.

Example 7.3 (Robust classification)**.** Consider two classes of data, the first one comprises given points $x_1, \dots, x_p \in \mathbb{R}^n$, and the second one contains given points $y_1, \dots, y_q \in \mathbb{R}^n$. We wish to construct a classifier that is able to predict to which class a new input belongs to. A basic linear classifier is based on data separation by a widest separating band. Mathematically, we seek for a hyperplane $a^T x + b = 1$ such that the first set of points belongs to the positive halfspace, the second set of points belongs to the negative halfspace, and the separating band is maximal. This leads to a convex quadratic program (see Figure 7.1a)

$$\min\ \|a\|_2 \ \text{ subject to } \ a^T x_i + b \ge 1\ \forall i,\ a^T y_j + b \le -1\ \forall j.$$



(a) The widest separating band for real data.

(b) The widest separating band for interval data.

Figure 7.1: (Example 7.3) A linear classifier for real data and the robust linear classifier for interval data.

Suppose now that data are not measured exactly and one knows them with a specified accuracy only. Hence we are given vectors of intervals $[\underline{x}_i, \overline{x}_i] = [(x_c)_i - (x_\Delta)_i, (x_c)_i + (x_\Delta)_i]$, $i = 1, \dots, p$, and $[\underline{y}_j, \overline{y}_j] = [(y_c)_j - (y_\Delta)_j, (y_c)_j + (y_\Delta)_j]$, $j = 1, \dots, q$, comprising the true data. Using the approach described above, the robust counterpart model reads (see Figure 7.1b)

$$\min\ \|a\|_2 \ \text{ subject to } \ (x_c)_i^T a + (x_\Delta)_i^T a' + b \leq 1\ \forall i,\ (y_c)_j^T a + (y_\Delta)_j^T a' + b \leq -1\ \forall j,\ \pm a \leq a'.$$

Again, it is a convex quadratic program (in variables $a, a' \in \mathbb{R}^n$ and $b \in \mathbb{R}$). □

Ellipsoidal uncertainty

Consider again the linear program in the form with variables unrestricted in sign

$$\min\ c^T x \ \text{ subject to } \ Ax \leq b.$$

Let $a^T x \leq d$ be a selected inequality. Consider an ellipsoid

$$\mathcal{E} = \{a \in \mathbb{R}^n;\ a = p + Pu,\ \|u\|_2 \leq 1\},$$

which is expressed as the image of a unit ball under a linear (or more precisely affine) mapping. A point x is a robust solution of the selected inequality if it satisfies

$$a^T x \leq d \quad \forall a \in \mathcal{E}$$

or,

$$\max_{a \in \mathcal{E}}\ a^T x \leq d.$$

Lemma 7.4. *We have*

$$\max_{a \in \mathcal{E}}\ a^T x = p^T x + \|P^T x\|_2.$$

Proof. Write

$$\begin{aligned} \max_{a \in \mathcal{E}}\ a^T x &= \max_{\|u\|_2 \leq 1}\ (p + Pu)^T x = p^T x + \max_{\|u\|_2 \leq 1}\ (P^T x)^T u \\ &= p^T x + (P^T x)^T \tfrac{1}{\|P^T x\|_2} P^T x = p^T x + \|P^T x\|_2. \end{aligned}$$ □

Using the lemma, we can express the robust solution constraint as

$$p^T x + \|P^T x\|_2 \leq d.$$

The left-hand side function is smooth and convex – indeed it is a second order cone constraint.

Example 7.5. Consider again the portfolio selection problem (example 4.9)

$$\max\ c^T x \ \text{ subject to } \ e^T x = K,\ x \geq o, \tag{7.2}$$

where c is a random Gaussian vector, its expected value is $\tilde{c} := \mathrm{E}\, c$ and the covariance matrix is $\Sigma := \operatorname{cov} c = \mathrm{E}\,(c-\tilde{c})(c-\tilde{c})^T$. The level sets of the density function represent ellipsoids, so it is natural to work with them. For a random vector c we have that the probability $P(c-\tilde{c} \in \mathcal{E}_\eta) = \eta$, where $\mathcal{E}_\eta$ is a certain ellipsoid (concretely, $\mathcal{E}_\eta = \{d \in \mathbb{R}^n;\ d = F^{-1}(\eta)\sqrt{\Sigma}u,\ \|u\|_2 \leq 1\}$, where $F^{-1}(\eta)$ is the quantile function of the normal distribution and $\sqrt{\Sigma}$ is the positive semidefinite square root of matrix Σ, i.e., $(\sqrt{\Sigma})^2 = \Sigma$).

One of the possible ways to solve (7.2) is to consider the deterministic counterpart

$$\max\ z \ \text{ subject to } \ P(c^T x \geq z) \geq \eta,\ e^T x = K,\ x \geq o,$$

where $\eta \in [\frac{1}{2}, 1]$ is a fixed value, e.g., $\eta = 0.95$. Obviously, condition $P(c^T x \geq z) \geq \eta$ is fulfilled if $d^T x \geq z$ holds for every $d \in \mathcal{E}_\eta + \tilde{c}$. Hence we can approximate the problem as

$$\max\ z \ \text{ subject to } \ d^T x \geq z\ \forall d \in \mathcal{E}_\eta + \tilde{c},\ e^T x = K,\ x \geq o.$$

This optimization problem involves ellipsoidal uncertainty, so we can equivalently write it as

$$\max\ z \ \text{ subject to } \ \tilde{c}^T x - F^{-1}(\eta)\|\sqrt{\Sigma}x\|_2 \geq z,\ e^T x = K,\ x \geq o.$$

Since $F^{-1}(\eta) \geq 0$ for any $\eta \geq \frac{1}{2}$, it is a second order cone programming problem.

7.2 Concave programming

Concave programming means minimizing a concave function on a convex set, or equivalently maximizing a convex function

$$\max\ f(x) \ \text{ subject to } \ x \in M,$$

where $M \subseteq \mathbb{R}^n$ is a convex set and $f\colon \mathbb{R}^n \to \mathbb{R}$ is a convex function.

Theorem 7.6. *Let M be a bounded convex polyhedron. Then the optimal solution exists and it is attained in at least one of the vertices of M.*

Remark. The theorem can be extended as follows: Any continuous convex function on a compact set M attains its maximum in an extreme point of M; see Avriel et al. [1988]. This property holds even more generally, when function $f(x)$ is so called quasiconvex.

Proof. Let $v_1, \dots, v_m$ be vertices of M, and without loss of generality assume that $f(v_1) = \max_{i=1,\dots,m} f(v_i)$. Then every point $x \in M$ can be expressed as a convex combination $x = \sum_{i=1}^m \alpha_i v_i$, where $\alpha_i \geq 0$ and $\sum_{i=1}^m \alpha_i = 1$. Now

$$f(x) = f(\textstyle\sum_{i=1}^m \alpha_i v_i) \leq \sum_{i=1}^m \alpha_i f(v_i) \leq \sum_{i=1}^m \alpha_i f(v_1) = f(v_1).$$

Therefore v_1 is an optimum. □

This property holds in linear programming, too. For computing an optimal solution, however, it is not very convenient since polyhedron M may contain many vertices, and we do not know which one is optimal. By Theorem 4.8, concave programming is NP-hard.

Typical problems resulting in concave programming comprise

- *Fixed charged problems.* The objective function has the form $f(x) = \sum_{i=1}^k f_i(x_i)$, where $f_i(x_i) = 0$ for $x_i = 0$ and $f_i(x_i) = c_i + g_i(x_i)$ for $x_i > 0$. Herein, $f_i(x_i)$ represents a price (e.g., the price for the transport of goods of size x_i). Hence the price is naturally zero when $x_i = 0$. When $x_i > 0$, we pay a fixed charge c_i plus the price $g_i(x_i)$ depending on the size of x_i. We can assume that $g_i(x_i)$ is concave since the larger x_i, the smaller relative price for the unit of goods (e.g., due to discounts).

- *Multiplicative programming.* The objective function has the form $f(x) = \prod_{i=1}^k x_i$. This is not a concave function in general, but its logarithm gives a concave function $\log(f(x)) = \sum_{i=1}^k \log(x_i)$. Such problems appear in geometry, where, for example, we minimize the volume of a body (e.g., a cuboid) subject to some constraints (e.g., the cuboid contains specified points).

Appendix

Derivative of matrix expressions. Let $A \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ and $c \in \mathbb{R}$. Consider the quadratic function $f\colon \mathbb{R}^n \to \mathbb{R}$ defined as

$$f(x) = x^T A x + b^T x + c.$$

Its gradient reads

$$\nabla f(x) = (A + A^T)x + b,$$

and the Hessian matrix takes the form

$$\nabla^2 f(x) = A + A^T.$$

In particular, if matrix A is symmetric, then

$$\nabla f(x) = 2Ax + b, \quad \nabla^2 f(x) = 2A.$$

Proof. First we consider the linear term

$$\frac{\partial}{\partial x_k} b^T x = \frac{\partial}{\partial x_k} \sum_{i=1}^{n} b_i x_i = b_k,$$

whence $\nabla b^T x = b$.

For the quadratic term, we get

$$\begin{aligned}\frac{\partial}{\partial x_k} x^T A x &= \frac{\partial}{\partial x_k} \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j = \frac{\partial}{\partial x_k}\left(a_k x_k^2 + \sum_{i\neq k}(a_{ik} + a_{ki}) x_i x_k + \sum_{i,j\neq k} a_{ij} x_i x_j \right)\\ &= 2a_k x_k + \sum_{i\neq k}(a_{ik} + a_{ki}) x_i = \sum_{i=1}^{n}(a_{ik} + a_{ki}) x_i = \big((A + A^T)x\big)_i.\end{aligned}$$

Hence the gradient reads $\nabla x^T A x = (A + A^T)x$. Since this is a linear function, the particular coordinates are differentiated in the same way as for the linear term. Therefore $\nabla^2 x^T A x = A + A^T$. □

Notation

Sets and numbers

$\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$	the set of natural numbers, integers, rational numbers, and reals, respectively
$U + V$	sumset (Minkowski sum), $U + V = \{u + v;\ u \in U, v \in V\}$
$U + v$	particular sumset, $U + v = U + \{v\} = \{u + v;\ u \in U\}$
$\operatorname{conv}(M)$	the convex hull of a set M
$\operatorname{int}(M)$	the topological interior of a set M
$\mathcal{K}^*$	the dual cone to cone $\mathcal{K}$, see Definition 4.15
r^+	the real part of a real number, $r^+ = \max(r, 0)$

Matrices and vectors

$\operatorname{tr}(A)$	the trace of a matrix A, $\operatorname{tr}(A) = \sum_i a_{ii}$
$\mathcal{S}(A)$	the column space of a matrix A
A^T	the transposition of a matrix A
$A \geq B$	componentwise inequality, i.e., $a_{ij} \geq b_{ij}$
$A > B$	componentwise strict inequality, i.e., $a_{ij} > b_{ij}$
$A \succeq B$	matrix $A - B$ is positive semidefinite
$A \succ B$	matrix $A - B$ is positive definite
A_{i*}	the ith row of a matrix A
A_{*j}	the jth column of a matrix A
$\operatorname{diag}(v)$	the diagonal matrix with entries $v_1, \ldots, v_n$
0_n, 0	zero matrix (all entries are zero)
I_n, I	the identity matrix (the diagonal matrix with ones on the diagonal)
e_i	the ith canonical unit vector, $e_i = I_{*i} = (0, \ldots, 0, 1, \ldots, 0)^T$
e	the vector of ones, $e = (1, \ldots, 1)^T$

Functions

$\lVert x\rVert_p$	ℓ_p-norm of a vector $x \in \mathbb{R}^n$, $\lVert x\rVert_p = \big(\sum_{i=1}^n \lvert x\rvert_i^p\big)^{\frac{1}{p}}$
$\lVert x\rVert_1$	Manhattan norm of a vector $x \in \mathbb{R}^n$, $\lVert x\rVert_1 = \sum_{i=1}^n \lvert x\rvert_i$
$\lVert x\rVert_2$	Euclidean norm of a vector $x \in \mathbb{R}^n$, $\lVert x\rVert_2 = \sqrt{\sum_{i=1}^n x_i^2}$
$\lVert x\rVert_\infty$	maximum norm of a vector $x \in \mathbb{R}^n$, $\lVert x\rVert_\infty = \max_{i=1,\ldots,n} \lvert x\rvert_i$
$P(c)$	probability of a random event c

Bibliography

A. A. Ahmadi, A. Olshevsky, P. A. Parrilo, and J. N. Tsitsiklis. NP-hardness of deciding convexity of quartic polynomials and related problems. *Math. Program.*, 137(1-2):453–476, 2013.
`https://arxiv.org/pdf/1012.1908`. 25

M. Avriel, W. E. Diewert, S. Schaible, and I. Zang. *Generalized concavity*, volume 36 of *Mathematical Concepts and Methods in Science and Engineering.* Plenum Press, New York, 1988. 62

M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming. Theory and Algorithms. 3rd ed.* John Wiley & Sons., NJ, 2006. 3

A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization. Analysis, algorithms, and engineering applications.* SIAM, Philadelphia, PA, 2001.
`https://www2.isye.gatech.edu/~nemirovs/Lect_ModConvOpt.pdf`. 31

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization.* Princeton University Press, 2009.
`https://www2.isye.gatech.edu/~nemirovs/FullBookDec11.pdf`. 59

S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.
`http://web.stanford.edu/~boyd/cvxbook/`. 3

S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi. A tutorial on geometric programming. *Optim. Eng.*, 8(1):67, 2007.

P. J. Dickinson and L. Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Comput. Optim. Appl.*, 57(2):403–415, 2014.
`http://dx.doi.org/10.1007/s10589-013-9594-z`. 38

M. Dür. Copositive programming – a survey. In M. Diehl, F. Glineur, E. Jarlebring, and W. Michiels, editors, *Recent Advances in Optimization and its Applications in Engineering: The 14th Belgian-French-German Conference on Optimization*, pages 3–20. Springer, Berlin, Heidelberg, 2010.
`http://www.optimization-online.org/DB_FILE/2009/11/2464.pdf`. 38

A. Fiacco and G. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* Wiley, New York, 1968. 53

C. A. Floudas and P. M. Pardalos, editors. *Encyclopedia of Optimization. 2nd ed.* Springer, New York, 2009. 29

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Res. Logist. Quart.*, 3(1-2): 95–110, 1956. 51

M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.*, 49(6):409–436, 1952.
`https://doi.org/10.6028/jres.049.044`. 55

C. F. Higham and D. J. Higham. Deep learning: An introduction for applied mathematicians. *SIAM Rev.*, 61(4):860–891, 2019. 49

M. Hutchings, F. Morgan, M. Ritoré, and A. Ros. Proof of the double bubble conjecture. *Ann. Math.*, 155(2):459–489, 2002.
`https://arxiv.org/pdf/math/0406017`. 8

W. Karush. Minima of functions of several variables with inequalities as side constraints. M.Sc. dissertation, Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939. 43

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*, pages 481–492, Berkeley, 1951. University of California Press. 43

K. Lange. *MM Optimization Algorithms*, volume 147 of *Other Titles Appl. Math.* SIAM, Philadelphia, PA, 2016. 23

A. N. Langville and C. D. Meyer. *Who's #1? The science of rating and ranking.* Princeton University Press, Princeton, NJ, 2012. 29

J. Liesen and Z. Strakoš. *Krylov Subspace Methods, Principles and Analysis.* Oxford University Press, Oxford, 2013. 55, 56

D. Luenberger and Y. Ye. *Linear and Nonlinear Programming.* Springer, New York, third edition, 2008. 3, 55

K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Math. Program.*, 39(2):117–129, 1987.
`https://doi.org/10.1007/BF02592948`. 38

Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming.* SIAM, Philadelphia, 1994.

P. M. Pardalos and S. A. Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *J. Glob. Optim.*, 1(1):15–22, 1991.
`https://doi.org/10.1007/BF00120662`. 29

M. J. Todd. *Minimum-Volume Ellipsoids: Theory and Algorithms*, volume 23 of *MOS-SIAM Series on Optimization.* SIAM & Mathematical Optimization Society, Philadelphia, PA, 2016. 39

S. A. Vavasis. *Nonlinear Optimization: Complexity Issues.* Oxford University Press, New York, 1991. 29

W. Zhu. Unsolvability of some optimization problems. *Appl. Math. Comput.*, 174(2):921–926, 2006.
`https://doi.org/10.1016/j.amc.2005.05.025`. 7

G. Zoutendijk. *Methods of feasible directions: A study in linear and nonlinear programming.* PhD thesis, University of Amsterdam, Amsterdam, Netherlands, 1960. 51