הטכניון – מכון טכנולוגי לישראל

Technion – Israel Institute of Technology

Faculty of Industrial Engineering and Management
Technion-City, Haifa 32000, Israel

הפקולטה להנדסת תעשיה ונהול
קרית-הטכניון, חיפה 32000

CONVEX OPTIMIZATION IN ENGINEERING:
Modeling, Analysis, Algorithms

Aharon Ben-Tal and Arkadi Nemirovski

Preface

The goals. To make decisions optimally is one of the most basic desires of a human being. Whenever the situation and the targets admit tractable formalization, this desire can, to some extent, be satisfied by mathematical tools, specifically, by those offered by the Optimization Theory and Algorithms. For our purposes, a general enough mathematical setting of an optimization problem will be the *Mathematical Programming* one:

$$\begin{array}{lrcl} \text{minimize} & f_0(x) & & \\ \text{subject to} & & & \\ & f_i(x) & \leq & b_i,\ i=1,...,m, \\ & x & \in & X \subset \mathbf{R}^n. \end{array} \tag{P}$$

In this problem, we are given an *objective* $f_0(x)$ and finitely many *functional constraints* $f_i(x)$, $i = 1, ..., m$, which are real-valued functions of n-dimensional *design vector* x varying in a given domain X. Our goal is to minimize the objective over the *feasible set* of the problem – the set which is cut off the domain X by the system of inequalities $f_i(x) \leq b_i$, $i = 1, ..., m$.

> In typical engineering applications, the design vector specifies a decision to be made (e.g., the parameters of certain construction), the domain X is the set of “meaningful” values of the design vector, and the functional constraints represent design specifications – restrictions on certain characteristics of the decision.

The last decade has witnessed a tremendous progress in Optimization, especially in the area of Convex Programming, progress which has improved dramatically the abilities to build and to analyze theoretically optimization models of complex real-world problems, same as the abilities to process the resulting models computationally. However, high applied potential of modern Optimization not always is properly utilized in practice, mainly because potential end-users – engineers, managers, etc., – typically have a regrettably poor impression of the state-of-the-art in Optimization. The goal of the our course is to contribute to shrinking the gap between those developing optimization models and techniques and those who could use these models and techniques in practice, to demonstrate what is modern Optimization, what it can do well and what is problematic. This final goal determines the approach we are taking, same as several specific targets which we are trying to achieve:

- Theoretically speaking, “what modern Optimization can solve well” are *convex optimization problems*. The “applied essence” of the two-decade-long investigation of complexity issues in Optimization can be summarized in the following conclusion:

  (!) *As far as numerical processing of problems* (P) *is concerned, there exists a “solvable case” – the one of* <u>convex</u> *optimization problems, those where*

  – The domain X is a closed <u>convex</u> subset of $\mathbf{R}^n$ [1)]

[1)] Recall that a set $X \subset \mathbf{R}^n$ is called convex, if along with every pair x, x' of its points it contains the entire segment linking the points:

$$x, x' \in X \Rightarrow x + \lambda(x' - x) \in X \quad \forall \lambda \in [0,1]$$

– The objective $f_0(x)$ and the functional constraints $f_i(x)$, $i = 1, ..., m$, are convex functions on X [2)]

Under minimal additional "computability assumptions" (which are satisfied in basically all applications), a convex optimization problem is "computationally tractable" – the computational effort required to solve the problem to a given accuracy "grows moderately" with the dimensions of the problem and the required number of accuracy digits.

In contrast to this, a general-type non-convex problems are too difficult for numerical solution – the computational effort required to solve such a problem by the best known so far numerical methods grows prohibitively fast with the dimensions of the problem and the number of accuracy digits, and there are serious theoretical reasons to guess that this is the intrinsic feature of non-convex problems rather than a drawback of the existing optimization techniques.

Just to give an example, consider a pair of optimization problems. The first is

$$\begin{array}{lrcl} \text{minimize} & -\sum_{i=1}^{n} x_i & & \\ \text{subject to} & & & \\ & x_i^2 - x_i & = & 0,\ i = 1, ..., n; \\ & x_i x_j & = & 0 \quad \forall (i,j) \in \Gamma, \end{array} \tag{A}$$

Γ being a given set of pairs (i, j) of indices i, j. This is a fundamental combinatorial problem of computing the *stability number* of a graph, and the corresponding "engineering story" is as follows:

> Assume that we are given n letters which can be sent through a telecommunication channel, say, $n = 256$ usual bytes. When passing trough the channel, an input letter can be corrupted by errors and as a result can be converted into another letter. Assume that the errors are "symmetric" – if a letter i can be converted into letter j, then j can be converted to i as well, and let Γ be the set of "dangerous pairs of letters" – pairs (i, j) of distinct letters i, j such that sending through the channel the letter i we can get on output the letter j. If we are interested in error-free transmission, we should restrict the set S of letters we actually use to be *independent* – to be such that no two distinct letters from S can be converted by the channel one into another. And in order to utilize best of all the capacity of the channel, we are interested to use a maximal – with maximum possible number of letters – independent sub-alphabet. It turns out that the minus optimal value in (A) is exactly the cardinality of such a maximal independent sub-alphabet.

[2)] Recall that a real-valued function $f(x)$ defined on a convex set $X \subset \mathbf{R}^n$ is called convex, if its epigraph

$$\{(t, x) \mid x \in X, t \geq f(x)\}$$

is a convex set in $\mathbf{R}^{n+1}$, or, equivalently, if

$$x, x' \in X \Rightarrow f(x + \lambda(x' - x)) \leq f(x) + \lambda(f(x') - f(x)) \quad \forall \lambda \in [0, 1].$$

Our second problem is

$$\begin{array}{lcl}
\text{minimize} & -2\sum_{i=1}^{k}\sum_{j=1}^{m} c_{ij}x_{ij} + x_{00} & \\
\text{subject to} & & \\
& \lambda_{\min}\left(\begin{pmatrix} x_1 & & & \sum_{j=1}^{m} b_{pj}x_{1j} \\ & \ddots & & \cdots \\ & & x_k & \sum_{j=1}^{m} b_{pj}x_{kj} \\ \sum_{j=1}^{m} b_{pj}x_{1j} & \cdots & \sum_{j=1}^{m} b_{pj}x_{kj} & x_{00} \end{pmatrix}\right) \geq 0, & \\
& p = 1, ..., N, & \\
& \sum_{i=1}^{k} x_i = 1, & \\
& & \text{(B)}
\end{array}$$

where $\lambda_{\min}(A)$ denotes the minimum eigenvalue of a symmetric matrix A. This problem is responsible for the design of a *truss* (a mechanical construction comprised of linked with each other thin elastic bars, like an electric mast, a bridge or the Eifel Tower) capable to withstand best of all to k given loads.

When looking at the analytical forms of (A) and (B), it seems that the first problem is easier than the second: the constraints in (A) are simple explicit quadratic equations, while the constraints in (B) involve much more complicated functions of the design variables – the eigenvalues of certain matrices depending on the design vector. The truth, however, is that the first problem is, in a sense, "as difficult as an optimization problem can be", and the worst-case computational effort to solve this problem within absolute inaccuracy 0.5 by all known optimization methods is about 2^n operations; for $n = 256$ (just 256 design variables corresponding to the "alphabet of bytes"), the quantity $2^n \approx 10^{77}$, for all practical purposes, is the same as $+\infty$. In contrast to this, the second problem is quite "computationally tractable". E.g., for $k = 6$ (6 loads of interest) and $m = 100$ (100 degrees of freedom of the construction) the problem has about 600 variables (twice the one of the "byte" version of (A)); however, it can be reliably solved within 6 accuracy digits in a couple of minutes. The dramatic difference in computational effort required to solve (A) and (B) finally comes from the fact that (A) is a non-convex optimization problem, while (B) is convex.

By the outlined reasons, in our course we restrict ourselves with Convex Programming only. Moreover, our emphasis will be on *well-structured convex problems* like those of Linear, Conic Quadratic and Semidefinite Programming. These are exactly the areas which are most of all affected by recent progress, areas where we possess well-developed techniques for building large-scale models and their instructive theoretical ("on paper") and numerical processing. And these areas, except Linear Programming, simply did not exist 10 years ago, so that most of the users who could benefit from recent developments just do not suspect that something like this exists and can be utilized.

- Restricting ourselves with "well-structured" convex optimization problems, it becomes logical to skip a significant part of what is traditionally thought of as the theory and algorithms of Mathematical Programming. Those interested in the Gradient Descent, Quasi-Newton methods, and even Sequential Quadratic Programming, are kindly asked to use excellent existing textbooks on these important subjects; our course should be thought of as a self-contained complement to, and not as an extended version of, these textbooks. We even have dared to omit the Karush-Kuhn-Tucker optimality conditions in their standard form, since they are too general to be "algorithmic"; the role of the KKT conditions in our course is played by their particular – and much more "algorithmic" –

form of the Conic Duality Theorem.

- The course is primarily addressed to potential users (mainly, engineers), not to the newcomers of the professional optimization community. Consequently, the emphasis is done on building and processing instructive engineering models rather than on details of optimization algorithms. The underlying motivation is twofold. First, we are trying to convince an engineer that Optimization indeed has something to do with his professional interests. Second, and mainly, we believe that the crucial necessary condition for successful practical applications of optimization is understanding of *what are optimization models which can be efficiently processed* and *how one can convert a "seemingly bad" initial description of the situation to an "efficiently processable" model*, what is desirable and what should be avoided at the *modeling* stage. And we believe that the best way to provide a reader with relevant insight is to present, along with general concepts and techniques, "non-toy" applications of these concepts and techniques. As about optimization algorithms, we believe that their presentation in a user-oriented course should be as non-technical as possible (to drive a car, a person should not be an expert in engines). More specifically, the part of the course devoted to algorithms focuses on the Ellipsoid method (due to its simplicity combined with capability to answer affirmatively on the fundamental question of whether Convex Programming is "computationally tractable"). Initially we intended to outline the ideas and to present a summary of the polynomial time Interior Point methods as well; however, teaching practice has demonstrated that this will be too much for a semester course[3)]

- In spite of the fact that the course is user-oriented, it is a mathematical course, and we did not try to achieve the (unreachable for us) goal to imitate engineers. In this respect, the goal we did try to achieve is to demonstrate that when processing "meaningful" *mathematical* models by *rigorous* mathematical methods (not by engineering surrogates of these methods), one can obtain results which admit meaningful and instructive engineering interpretation. Whether we have reached this goal – this is another story, and here the judgment is upon the reader.

Audience and prerequisites. Readers are supposed to know basic facts from Linear Algebra and Analysis – those presented in all standard undergraduate mathematical courses for engineers. As far as optimization-related areas are concerned, readers are assumed to know the definitions of a convex set and a convex function and to have heard (no more than that!) what is a Mathematical Programming problem. What is highly desirable, are basic elements of mathematical culture.

The exercises accompanying the course form its significant part. They are organized in small groups, each group being devoted to a specific topic somehow related to the corresponding lecture. Typically, the task in an exercise is to prove something. Most of the exercises are not too easy; those we find relatively more difficult, are marked with *. The results stated by the exercises are used in the sequel in the same way as the statements presented and proved in the main body of the course; consequently, a reader is kindly asked if not to do, then at least to

[3)] For a brief overview of polynomial time IP methods in Convex Programming, see [5]; as about detailed presentation of the subject, there are several books/lecture notes available, the closest in style to the present course being [6].

read carefully all exercises. The order of exercises is of primary importance: in many cases the preceding exercises contain information/hints necessary to succeed in the subsequent ones.

Some exercises have to do with modeling issues and are not formulated as "rigorously posed" mathematical questions. Well, this is the type of situations our potential readers are supposed to survive, and the only thing we apologize for is that the number of exercises of this type is less than it should be.

Grade policy. An exercise in the Lecture Notes starts like "**Exercise X.X.X** ^{Y}", and the superscript ^{Y} is the weight of the exercise (the # of points earned by a complete solution of the exercise). The grade policy will be based on the total weight of exercises solved by a participant of the course. The preliminary version of the grade policy is as follows: in order to get full grade, a participant should meet the following two requirements:

1. The total number of points for the exercises solved by him/her should be at least 30;
2. He/she should solve at least one exercise from every one of the assignments.

The policy may be modified during the first four weeks of the course.

The contents of the course is as follows:

Part I. Generic convex optimization models: theory and applications in Engineering

- Linear Programming, with applications to design of filters, antennae and trusses
- Conic Programming
  - General theory
  - Conic Quadratic Programming, with applications to Robust Linear Programming, Robotics,...
  - Semidefinite Programming, with applications to Stability Analysis/Synthesis, Structural Design, synthesis of chips,...

Part II. Algorithms for solving convex optimization models

- The Ellipsoid method and computational tractability of Convex Programming

Acknowledgements. The "applied" part of the course is strongly influenced by the activity of Professor Stephen Boyd from Stanford University and his colleagues, systematically working in the area of engineering applications of Convex Optimization. A quite significant part of applications discussed in our course is taken from the papers of this research group, and we are greatly indebted to Prof. Stephen Boyd for providing us with the most recent results of his group. Applications related to Structural Design were developed in tight collaboration with our German colleagues Prof. Jochem Zowe and Dr. Michael Kočvara.

This course for the first time was given at the Faculty of Technical Mathematics of the Technical University of Delft (Spring Semester 1998), and the second author is very grateful to his Delft hosts Prof. Cornelis Roos and Dr. Tamas Terlaky for stimulating discussions and hospitality.

Contents

Lecture 1

Linear Programming

Our first topic is Linear Programming. Our primary goal to present the basic results on the LP duality in a form which makes its easy to extend in the mean time these results to the nonlinear case.

1.1 Linear programming: basic notions

A *Linear Programming* (LP) program is an optimization program of the form

$$c^T x \to \min \mid Ax - b \geq 0, \tag{LP}$$

where

- $x \in \mathbf{R}^n$ is the design vector
- $c \in \mathbf{R}^n$ is a given *objective*
- A is a given $m \times n$ *constraint matrix*, and $b \in \mathbf{R}^m$ is a given *right hand side of the constraints.*

As any other optimization problem, (LP) is called
– *feasible*, if its *feasible set*

$$\{x \mid Ax - b \geq 0\}$$

is nonempty; a point from the latter set is called a *feasible solution* to (LP);
– *below bounded*, if it is either infeasible, or its objective $c^T x$ is below bounded on the feasible set.

> For a feasible below bounded problem, the lower bound of the objective on the feasible set – the quantity
>
> $$c^* \equiv \inf_{x:Ax-b\geq 0} c^T x$$
>
> – is called the *optimal value* of the problem. For an infeasible problem, the optimal value, by definition, is $+\infty$, while for feasible below *un*bounded problem the optimal value, by definition, is $-\infty$.

– *solvable*, if it is feasible, below bounded and the optimal value is attained: there exists feasible x with $c^Tx = c^*$. An x of this type is called an *optimal solution* to (LP).

A priori it is unclear whether a feasible and below bounded LP program is solvable: why should the infimum be achieved? It turns out, however, that a feasible and below bounded LP program *always* is solvable. This nice fact (we shall establish it later) is specific for LP. E.g., a very simple *nonlinear* optimization program

$$\frac{1}{x} \to \min \mid x \geq 1$$

is feasible and below bounded, but is not solvable.

1.2 An example: Tschebyshev approximation and its applications

In the majority of textbooks known to us, examples of LP programs have to do with economics, production planning, etc., and indeed the major applications of LP are in these areas. In our course, however, we would prefer to use, as a basic example, a problem related to engineering. Let us start with the mathematical formulation.

1.2.1 The best uniform approximation

Problem 1.2.1 [Tschebyshev approximation] *Given an $M \times N$ matrix $A = \begin{bmatrix} a_1^T \\ a_2^T \\ \dots \\ a_M^T \end{bmatrix}$ and a vector $b \in \mathbf{R}^M$, solve the problem*

$$\min_{x \in \mathbf{R}^n} \| Ax - b \|_\infty, \quad \| Ax - b \|_\infty \equiv \max_{i=1,\dots,M} |a_i^T x - b_i|. \tag{1.2.1}$$

As stated, problem (1.2.1) is *not* an LP program – its objective is nonlinear. We can, however, immediately convert (1.2.1) to an equivalent LP program

$$t \to \min \mid -t \leq a_i^T x - b_i \leq t, \ i = 1, \dots, M, \tag{1.2.2}$$

t being an additional design variable. Thus, (1.2.1) is equivalent to an LP program.

A typical origin of the Tschebyshev problem is as follows: we are interested to approximate best of all a given "target" function $\beta(t)$ on, say, the unit segment $[0,1]$ of values of t by a linear combination $\sum_{j=1}^N x_j \alpha_i(t)$ of N given functions $\alpha_j(t)$; the quality of approximation is measured by its uniform distance from β, i.e., by the quantity

$$\| \beta - \sum_{j=1}^N x_j \alpha_j \|_\infty \equiv \sup_{0 \leq t \leq 1} |\beta(t) - \sum_{j=1}^N x_j(t) \alpha_j(t)| \tag{1.2.3}$$

As we shall see in a while, the problem

$$\min_{x \in \mathbf{R}^N} \| \beta - \sum_{j=1}^N x_j \alpha_j \|_\infty \tag{1.2.4}$$

is important for several engineering applications. From the computational viewpoint, the drawback of the problem is that its objective is "implicit" – it involves maximization with respect to a continuously varying variable. As a result, already the related analysis problem – given a vector of coefficients x, to evaluate the quality of the corresponding approximation – can be quite difficult numerically. The simplest way to overcome this drawback is to approximate in (1.2.3) the maximum over t running through $[0,1]$ by the maximum over t running through a "fine finite grid", e.g., through the finite set

$$T_M = \{t_i = \frac{i}{M} \mid i = 1, ..., M\}.$$

With this approximation, the objective in the problem (1.2.4) becomes

$$\max_{i=1,\dots,M} |\beta(t_i) - \sum_{j=1}^{N} x_j \alpha_j(t_i)| \equiv \| Ax - b \|_\infty,$$

where the columns of A are the restrictions of the functions $\alpha_j(\cdot)$ on the grid T_M, and b is the restriction of $\beta(\cdot)$ on the grid. Consequently, the optimization problem (1.2.1) can be viewed as a discrete version of the problem (1.2.4).

1.2.2 Application example: synthesis of filters

As it was already mentioned, problem (1.2.4) arises in a number of engineering applications. Consider, e.g., the problem of synthesis a linear time-invariant (LTI) dynamic system (a "filter") with a given impulse response. [1]

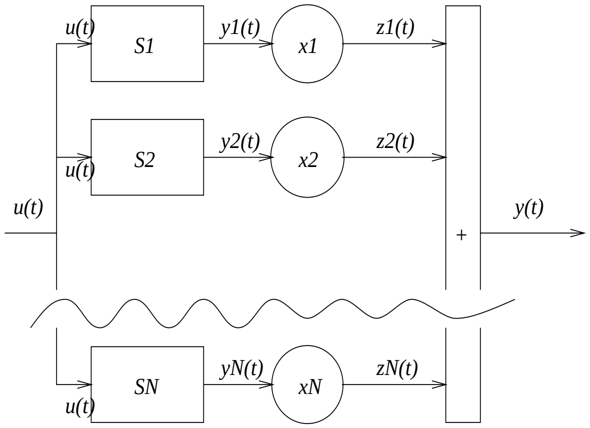
> A (continuous time) time-invariant linear dynamic system S is, mathematically, a transformation from the space of "signals" – functions on the axis – to the same space given by the convolution with certain fixed function:
>
> $$u(t) \mapsto y(t) = \int_{-\infty}^{\infty} u(s)h(t-s)ds,$$
>
> $u(\cdot)$ being an *input*, and $y(\cdot)$ being the corresponding *output* of S. The convolution kernel $h(\cdot)$ is a characteristic function of the system called the *impulse response* of S.

Consider the simplest synthesis problem

Problem 1.2.2 [Filter Synthesis, I] *Given a desired impulse response $h_*(t)$ along with N "building blocks" – standard systems S_j with impulse responses $h_j(\cdot)$, $j = 1, ..., N$ – assemble these*

[1] The "filter synthesis" and the subsequent "antenna array" examples are taken from [2]

building blocks as shown on Fig. 1.1



$$\begin{array}{rcl} y_j(t) & = & \int_{-\infty}^{\infty} u(s)h_j(t-s)ds \\ z_j(t) & = & x_j y_j(t) \\ y(t) & = & z_1(t) + z_2(t) + ... + z_N(t). \end{array}$$

Figure 1.1. Parallel structure with amplifiers

into a system S in such a way that the impulse response of the latter system will be as close as possible to the desired impulse response $h_*(\cdot)$.

Note that the structure of S is given, and all we can play with are the amplification coefficients x_j, $j = 1, ..., N$.

The impulse response of the structure on Fig. 1.1 clearly is

$$h(t) = \sum_{j=1}^{N} x_j h_j(t).$$

Assuming further that h_* and all h_j vanish outside $[0,1]$ [2)] and that we are interested in the best possible uniform on $[0,1]$ approximation of the desired impulse response h_*, we can pose our synthesis problem as (1.2.4) and further approximate it by (1.2.1). As we remember, the latter problem is equivalent to the LP program (1.2.2) and can therefore be solved by linear programming tools.

1.2.3 Filter synthesis revisited

In the Filter Synthesis problem we were interested to combine given "building blocks" S_i to get a system with impulse response as close to the target one as possible. It makes sense, but a more typical design requirement is to get a system with a desired *transfer function*; the latter is nothing but the Fourier transform of the impulse response. The role of the transfer function becomes clear when we represent the action of an LTI system S in the "frequency

[2)] Assumptions of this type have a quite natural interpretation. Namely, the fact that impulse response vanishes to the left of the origin means that the corresponding system is *casual* – its output till any time instant t depends solely on the input till the same instant and is independent of what happens with the input after the instant t. The fact that impulse response vanishes after certain $T > 0$ means that the *memory* of the corresponding system is at most T: output at a time instant t depends on what is the input starting with the time instant $t - T$ and is independent of what was the input before the instant $t - T$.

domain" – in the space of the Fourier transforms of inputs/outputs. In the frequency domain the transformation carried out by the system becomes

$$U(\omega) \mapsto Y(\omega) = U(\omega)H(\omega), -\infty < \omega < \infty, \tag{1.2.5}$$

where upper case letters denote the Fourier transforms of their lower-case counterparts: $U(\omega)$ stands for the Fourier transform of the input $u(t)$, $Y(\omega)$ – for the Fourier transform of the output $y(t)$, and the characteristic for the system *transfer function* $H(\omega)$ is the Fourier transform of the impulse response. Relation (1.2.5) demonstrates that the action of an LTI system in the frequency domain is very simple – just multiplication by the transfer function; this is why the analysis of an LTI system usually is carried out in the frequency domain and, as a result, the reason why typical design requirements on LTI systems are formulated in terms of their transfer functions.

The "frequency domain" version of the Filter Synthesis problem is as follows:

Problem 1.2.3 [Filter Synthesis, II] *Given a target transfer function $H_*(t)$ along with N "building blocks" – standard systems S_j with transfer function $H_j(\cdot)$, $j = 1, ..., N$ – assemble these building blocks as shown on Fig. 1.1 into a system S in such a way that the transfer function of the latter system will be as close as possible to the target function $H_*(\cdot)$.*

Assume that we again measure the quality of approximating the target transfer function by the uniform, on a given segment $[\omega_{\min}, \omega_{\max}]$ in the frequency domain, norm of the approximation error. Thus, the problem of interest is

$$\sup_{\omega_{\min}\leq\omega\leq\omega_{\max}} |H_*(\omega) - \sum_{j=1}^{N} x_j H_j(\omega)| \to \min,$$

and its "computationally tractable approximation" is

$$\max_{i=1,...,M} |H_*(\omega_i) - \sum_{j=1}^{N} x_j H_j(\omega_i)| \to \min, \tag{1.2.6}$$

$\{\omega_1, \omega_2, ..., \omega_M\}$ being a grid in $[\omega_{\min}, \omega_{\max}]$. Mathematically, the latter problem looks exactly as (1.2.1), and one could think that it can be immediately converted to an LP program. We should, however, take into account an important consideration as follows:

> In contrast to impulse response, transfer function is, generally, complex-valued. Consequently, the absolute values in (1.2.6) are absolute values of complex numbers. As a result, the conversion of (1.2.1) to an LP program now fails – the crucial for it possibility to represent the nonlinear inequality $|a| \leq t$ by two linear inequalities $a \leq t$ and $a \geq -t$ exists in the case of real data only!

The difficulty we have met can be overcome in two ways:

– first, the inequality $|a| \leq t$ with *complex-valued* a can be represented as the inequality $\sqrt{\Re^2(a) + \Im^2(a)} \leq t$ with real data. As a result, problem (1.2.6) can be posed as a *conic quadratic* problem we will focus on in Lecture 3;

– second, the inequality $|a| \leq t$ with complex-valued a can be easily *approximated* by a number of linear inequalities on $\Re(a)$ and $\Im(a)$. Indeed, let us inscribe into the unit circle D on the complex plane $\mathbf{C} = \mathbf{R}^2$ a $(2k)$-vertex perfect polygon P_{2k}:

$$P_k = \{(u,v) \in \mathbf{R}^2 : |u\cos(l\phi) + v\sin(l\phi)| \leq \cos(\phi/2), l = 1, ..., k\} \quad \left[\phi = \frac{\pi}{k}\right]$$

We claim that for every $z = (u, v) \in \mathbf{R}^2$ one has

$$|z| \geq p_k(z) \equiv \max_{l=1,\dots,k}[u\cos(l\phi) + v\sin(l\phi)] \geq \cos(\phi/2)|z|. \tag{1.2.7}$$

Indeed, the left inequality in (1.2.7) follows from the Cauchy inequality:

$$|u\cos(l\phi) + v\sin(l\phi)| \leq |z|\sqrt{\cos^2(l\phi) + \sin^2(l\phi)} = |z| \Rightarrow p_k(z) \leq |z|.$$

The right inequality follows from the fact that P_k is inside D:

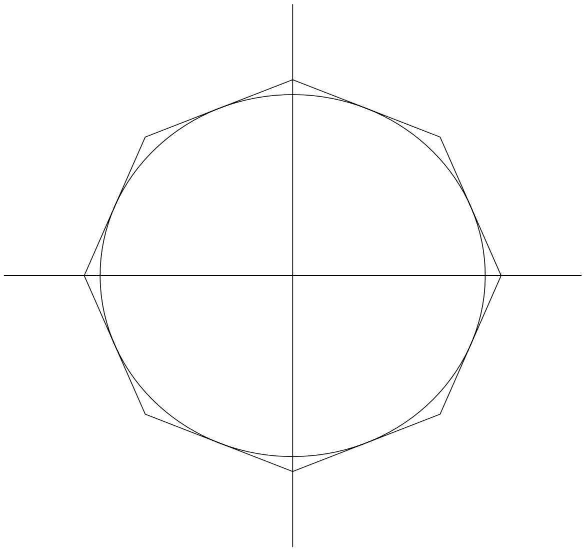
$$|z| = 1 \Rightarrow z \notin \text{int } P_k \Rightarrow p(z) \geq \cos(\phi/2),$$

whence $p_k(z) \geq \cos(\phi/2)|z|$ whenever $|z| = 1$. Since both $|z|$ and $p_k(z)$ are of the homogeneity degree 1:

$$p_k(\lambda z) = \lambda p_k(z), |\lambda z| = \lambda|z| \quad \forall \lambda \geq 0,$$

the validity of the inequality $p_k(z) \geq \cos(\phi/2)|z|$ in the case of $|z| = 1$ implies the validity of the inequality for all z.

We see that the absolute value $|z|$ of a complex number can be approximated, within relative accuracy $1 - \cos(\phi/2) = 1 - \cos(\pi/(2k))$, by the "polyhedral norm" $p_k(z)$ – the maximum of k linear forms of $\Re(z)$ and $\Im(z)$. E.g., taking $k = 4$, we approximate $|z|$ within the 7.7% margin:



The contours $|z| = 1$ (circle) and $p_4(z) = 1$ (polygone)

For most of the engineering applications, it is basically the same – to approximate H_* in the uniform norm on a grid $\Omega = \{\omega_1, ..., \omega_M\}$ or in the polyhedral norm

$$\max_{i=1,\dots,M} p_4(H_*(\omega_i) - \sum_{j=1}^{N} x_j H_j(\omega_i))$$

– the corresponding measures of quality of an approximation differ from each other by less than 8%. Consequently, one can pass from the problem (1.2.6) to its approximation

$$\max_{i=1,\dots,M} p_4(H_*(\omega_i) - \sum_{j=1}^{N} x_j H_j(\omega_i)) \to \min\,. \tag{1.2.8}$$

Problem (1.2.8) is equivalent to the program

$$t \to \min \mid \pi_4(H_*(\omega_i) - \sum_{j=1}^{N} x_j H_j(\omega_i)) \le t, \; i = 1, ..., M,$$

or, which is the same due to the structure of $p_4(\cdot)$, is equivalent to the LP program

$$t \to \min \mid \left| \cos(l\phi)\Re(H_*(\omega_i) - \sum_{j=1}^{N} x_j H_j(\omega_i)) + \sin(l\phi)\Im(H_*(\omega_i) - \sum_{j=1}^{N} x_j H_j(\omega_i)) \right| \le t, \quad \begin{array}{l} i = 1, ..., M \\ l = 1, ..., 4 \end{array} \tag{1.2.9}$$

1.2.4 Synthesis of arrays of antennae

Among the engineering applications of the Tschebyshev approximation problem an important one is the synthesis of arrays of antennae. An antenna is an electro-magnetic device capable to generate electro-magnetic waves. The main characteristic of an antenna is its *diagram* $Z(\delta)$. To understand what is the diagram, let us fix a direction $\delta \in \mathbf{R}^3$, $\| \delta \|_2 = 1$, and look what is the electro-magnetic field created by the antenna at a point $P = r\delta$ (we assume that the antenna is placed at the origin), i.e., when we move away from the antenna by a distance r in the direction δ. Physics says that the electric and the magnetic components of the field are, respectively,

$$\begin{array}{rcl} E(r\delta) & = & a(\delta)r^{-1}\cos(\phi(\delta) + t\omega - 2\pi r/\lambda), \\ F(r\delta) & = & a(\delta)r^{-1}\sin(\phi(\delta) + t\omega - 2\pi r/\lambda), \end{array} \tag{1.2.10}$$

where t stands for time, ω is the frequency of the wave, λ is the wavelength, and $a(\delta) > 0$ is responsible for the energy sent in the direction δ and $\phi(\delta)$ is the initial phase of the signal sent in this direction [3]. It is convenient to treat $E(r\delta)$ and $F(r\delta)$ as the real and the imaginary parts of the complex-valued function $W(r\delta)$:

$$\begin{array}{rcl} W(r\delta) & = & a(\delta)r^{-1}(\cos(\phi(\delta) + t\omega - 2\pi r/\lambda) + \mathrm{i}\sin(\phi(\delta) + t\omega - 2\pi r/\lambda)) \\ & = & r^{-1}\exp\{-2\pi r/\lambda\}Z(\delta)\exp\{\mathrm{i}\omega t\}, \end{array}$$

where

$$Z(\delta) = a(\delta)[\cos(\phi(\delta)) + \mathrm{i}\sin(\phi(\delta))].$$

Note that if our antenna is comprised of N components with diagrams $Z_1(\delta), ..., Z_N(\delta)$, the function W of the antenna is just the sum of similar functions of the components (electro-magnetic equations are linear!), so that the diagram of the antenna is the sum of those of the components:

$$Z(\delta) = \sum_{j=1}^{N} Z_j(\delta).$$

Now, when designing an array of antennae – a complex antenna comprised of a number of components – an engineer starts with N given "building blocks" with diagrams $Z_1, ..., Z_N$. When arranging the array, there is a possibility to amplify the signal sent by a block by an

[3] Relations (1.2.10) work when the distance r between P and the antenna is much larger than the linear sizes of the antenna. Mathematically, the difference between the left and the right hand sides in (1.2.10) is $o(r^{-1})$ as $r \to \infty$.

amplifying factor ρ_j and to shift the initial phase $\phi_j(\cdot)$ by a constant ψ_j. In other words, the engineer can modify the original diagrams of the blocks according to

$$Z_j(\delta) \equiv a_j(\delta)[\cos(\phi_j(\delta)) + \mathrm{i}\sin(\phi_j(\delta))] \mapsto Z_j^+(\delta) = \rho_j a_j(\delta)[\cos(\phi_j(\delta) + \psi_j) + \mathrm{i}\sin(\phi_j(\delta) + \psi_j)].$$

Thus, there is a possibility to multiply the initial diagram of every block by an arbitrary complex constant

$$z_j = \rho_j(\cos\psi_j + \mathrm{i}\sin\psi_j) \equiv u_j + \mathrm{i}v_j.$$

The diagram of the resulting complex antenna will be

$$Z(\delta) = \sum_{j=1}^{N} z_j Z_j(\delta). \tag{1.2.11}$$

A typical design problem in the area in question is to choose the design parameters z_j, $j = 1, ..., N$, in order to get a diagram (1.2.11) as close as possible to a target diagram $Z_*(\delta)$. In many cases a relevant way to measure "closeness" is to use the uniform norm; the corresponding synthesis problem becomes

$$\max_{\delta:\|\delta\|_2=1} |Z_*(\delta) - \sum_{j=1}^{N} z_j Z_j(\delta)| \to \min,$$

the design variables being N complex numbers $z_1, ..., z_N$, or, which is the same, $2N$ real numbers $\Re(z_j)$, $\Im(z_j)$. Mathematically, the resulting problem is completely similar to the one discussed in the previous section, up to the fact that now the outer maximum in the description of the objective is taken over the unit sphere in $\mathbf{R}^3$ rather than over a segment on the axis; even this difference disappears when we approximate the maximum over continuously varying direction by the maximum over a finite grid on the sphere (this in any case is necessary to get an efficiently solvable optimization program). Thus, the problem of synthesis of an array of antennae is, mathematically, identical to the problem of synthesis an LTI system with a desired transfer function; in particular, we have a possibility to approximate the problem by an LP program.

Example. Let a planar antenna array be comprised of a central circle and 9 concentric rings of the same area as the circle (Fig. 1.2.(a)). The array is placed in the XY-plane ("Earth's surface"), and the outer radius of the outer ring is 1m.

One can easily see that the diagram of a ring $\{a \le r \le b\}$ in the plane XY (r is the distance from a point to the origin) as a function of a 3-dimensional direction δ depends on the altitude (the angle θ between the direction and the plane) only. The resulting function of θ turns to be *real-valued*, and its analytic expression is

$$Z_{a,b}(\theta) = \frac{1}{2}\int_a^b \left[\int_0^{2\pi} r\cos\left(2\pi r\lambda^{-1}\cos(\theta)\cos(\phi)\right) d\phi\right] dr,$$

λ being the wavelength. Fig. 1.2.(b) represents the diagrams of our 10 rings, the wavelength being 50cm.

Now assume that our goal is to design an array with the real-valued diagram which should be axial symmetric with respect to the Z-axis and should be "concentrated" in the cone $\pi/2 \ge \theta \ge \pi/2 - \pi/12$. In other words, we are interested in real-valued Z_* depending on the altitude θ only; the resulting function $Z_*(\theta)$ is 0 when $0 \le \theta \le \pi/2 - \pi/12$ and somehow approaches 1 as

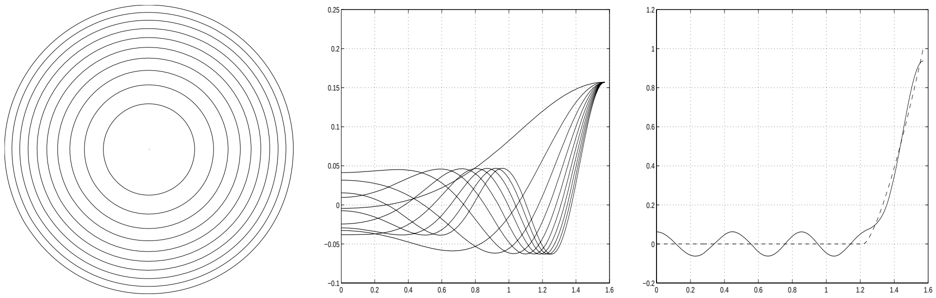
θ approaches $\pi/2$. The target diagram $Z_*(\theta)$ used in our design is represented on Fig. 1.2.(c) (the dashed curve).

In the case in question the outlined approach is simplified a lot by the fact that the diagrams of our "building blocks", same as the target diagram, are real-valued; as a result, we have no troubles with complex numbers, and the problem we should finally solve is

$$\max_{\theta\in T} |Z_*(\theta) - \sum_{j=1}^{10} x_j Z_{r_{j-1},r_j}(\theta)| \to \min,$$

where T is a finite grid on the segment $[0,\pi/2]$ of values of θ (in the design represented on Fig. 1.2.(c), the 120-point equidistant grid is used), $r_0 = 0$ and r_j, $j \geq 1$, is the outer radius of the j-th element of the array. Both the data and the design variables in the problem are real, so that we can immediately convert the problem into an equivalent LP program.

The solid line on Fig. 1.2.(c) represents the diagram of the array of antennae given by the synthesis. The uniform distance between the actual and the target diagrams is ≈ 0.0621. Table 1 displays the optimal amplification coefficients (i.e., the coordinates x_j of the optimal solution).



(a) (b) (c)

Figure 1.2. Synthesis of antennae array

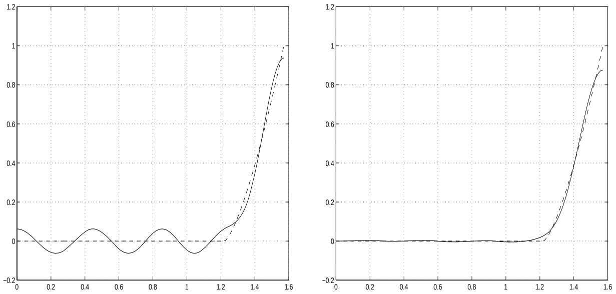
(a): 10 array elements of equal area in the XY-plane
the outer radius of the largest ring is 1m, the wavelength is 50cm
(b): "building blocks" – the diagrams of the rings as functions of the altitude angle θ
(c): the target diagram (dashed) and the synthesied diagram (solid)

Table 1. Optimal amplifying coefficients (rounded to 5 significant digits)

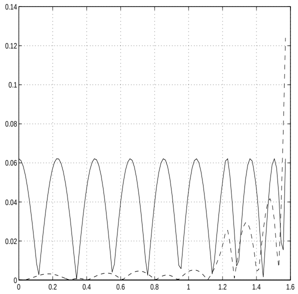
element #	1	2	3	4	5	6	7	8	9	10
coefficient	1624.4	-14700	55383	-107247	95468	19221	-138620	144870	-69303	13311

Why the uniform approximation? The Antenna Array example inspires a natural question: why the distance between the target diagram $Z_*(\cdot)$ and the synthesized one $Z(\cdot)$ is measured by the uniform norm of the residual $\| Z_* - Z \|_\infty = \max_\theta |Z_*(\theta) - Z(\theta)|$ and not by, say, the 2-norm $\| Z_* - Z \|_2 = \sqrt{\sum_\theta |Z_*(\theta) - Z(\theta)|^2}$? With this latter norm – i.e., with the standard *Least Squares* approximation – the (squared) objective to be minimized would be a sum of squares of affine forms of the design variables (recall that for every θ $Z(\theta)$ linearly depends on the amplifying coefficients), i.e., it would be a convex quadratic form $\frac{1}{2}x^T Ax + b^T x + c$ of x, and we could immediately write down the optimal solution: $x^* = -A^{-1}b$, thus avoiding any necessity in numerical optimization.

The answer to the outlined question is as follows: the *only* advantage of the $\|\cdot\|_2$-accuracy measure is that it leads to a computationally cheap approximation routine; from the modeling viewpoint, Least Squares in many cases are not attractive at all. Indeed, consider the case when the target function is "nearly singular" – it is close to one constant, say, to 0, in the "major part" A of its domain and is close to another constant, say, to 1, in another, relatively small, part B of the domain. This is the case in our Antenna Synthesis example: we are trying to concentrate the signal in a small cone, and what is of actual interest is exactly the “nearly singular” behaviour of the signal. Now, with an “integral-type” norm of residual, like the $\|\cdot\|_2$-norm, the squared “typical” deviation between an approximation and the target in A is taken with relatively large weight (proportional to the cardinality of A), while the squared typical deviation between the functions in B is taken with relatively small weight. It follows that in order to get a better $\|\cdot\|_2$-approximation, it makes sense to use the “flexibility resource” of our approximation in order to reproduce first of all the “background” behaviour of the target – the one in A, even at the price of poor reproduction of the “singular behaviour” of the target – the one in B. As a result, the Least Squares usually result in “oversmoothened” approximations badly reproducing the “nearly singularities” of the target – i.e., the features of the target we indeed are interested in. In contrast to this, the $\|\cdot\|_\infty$-norm of the residual “pays the same attention” to how well we reproduce the “background” behaviour of the target and to how well we reproduce its singularities; this feature of the uniform norm makes it a more preferable quality measure in “approximation problems with singularities” than the $\|\cdot\|_2$-norm. To illustrate this point, let us look at what the Least Squares yield in our example:



Best uniform approximation (left) vs. the Least Squares one (right)



Errors of the Least Squares (solid) and the best uniform (dashed) approximations

We see that the Least Squares approximation indeed “pays more attention” to the “background” than to the “singularity”. The uniform distance between the target and the Least Squares approximation is 0.1240 – more than twice as much as the distance corresponding to the best uniform approximation!

1.3 Duality in linear programming

The most important and interesting feature of linear programming as a mathematical entity (i.e., aside of computations and applications) is the wonderful *LP duality theory* we are about to consider. The question we are interested in now is:

Given an LP program

$$c^T x \to \min,\ Ax - b \geq 0, \tag{LP}$$

find a systematic way to bound from below its optimal value.

Why this is an important question and how the answer to it helps to deal with LP programs, this will be seen in the sequel. For the time being, let us just believe that the question is worthy of effort.

A trivial answer to the posed question is: solve (LP) and look what is the optimal value. There is, however, a smarter and a much more instructive way to answer our question. Just to get an idea of this smart way, let us look at the following example:

$$x_1 + x_2 + ... + x_{1999} \to \min \mid \left\{ \begin{array}{rcl} x_1 + 2x_2 + ... + 1998x_{1998} + 1999x_{1999} - 1 & \geq & 0, \\ 1999x_1 + 1998x_2 + ... + 2x_{1998} + x_{1999} - 100 & \geq & 0, \\ ..... & ... & ... \end{array} \right.$$

We claim that the optimal value in the problem is $\geq \frac{101}{2000}$. If you would ask how one could certify this bound, the answer would be very simple: "add the first two constraints to get the inequality

$$2000(x_1 + x_2 + ... + x_{1998} + x_{1999}) - 101 \geq 0,$$

and divide the resulting inequality by 2000." The LP duality is nothing but a straightforward generalization of this simple trick.

1.3.1 Certificates for solvability and insolvability

Consider a (finite) system of scalar inequalities with n unknowns. To be as general as possible, for the time being we do not assume the inequalities to be linear and allow for both non-strict and strict inequalities in the system, as well as for equations. Since an equation can be represented by a pair of non-strict inequalities, our system always can be written down as

$$f_i(x)\,\Omega_i\,0, \ \ i = 1, ..., m, \qquad (\mathcal{S})$$

where Ω_i, for every i, is either the relation " $>$ " or the relation " $\geq$ ".

The basic question about $(\mathcal{S})$ is

(?) Whether $(\mathcal{S})$ has or has no solution.

Knowing how to answer the question (?), we are able to answer many other questions. E.g., to verify that a given real a is a lower bound on the optimal value of an LP program (LP) is the same as to verify whether the system

$$\left\{ \begin{array}{rcl} -c^T x + a & > 0 & \\ Ax - b & \geq & 0 \end{array} \right.$$

has no solutions.

The general question above is too difficult, and it makes sense to pass from it to a seemingly simpler one:

(??) How to certify that $(\mathcal{S})$ has, or does not have, a solution.

Imagine, e.g., that you are very smart and know the correct answer to (?); how could you convince somebody that your answer is correct? What could be an "evident for everybody" certificate of the validity of your answer?

If your claim is that $(\mathcal{S})$ is solvable, a certificate could be very simple: it suffices to point out a solution x^* to $(\mathcal{S})$. Given this certificate, one can substitute x^* into the system and check whether x^* indeed is a solution.

Assume now that your claim is that $(\mathcal{S})$ has no solutions. What could be a "simple certificate" of this claim? How one could certify a *negative* statement? This is a highly nontrivial problem which goes far beyond the bounds of mathematics. E.g., it is crucial in law: how somebody accused in a murder should prove his innocence? The "real life" answer to the question "how to certify a negative statement" is discouraging: such a statement normally *cannot* be certified (this is where the rule "a person is innocent until his/her guilt is certified" comes from). In mathematics, however, the situation is different: in some cases there exist "simple certificates" of negative statements. E.g., in order to certify that $(\mathcal{S})$ has no solutions, it suffices to demonstrate that one can obtain, as a consequence of the system $(\mathcal{S})$, the contradictory inequality

$$-1 \geq 0.$$

For example, assume that λ_i, $i = 1, ..., m$, are nonnegative weights. Combining inequalities from $(\mathcal{S})$ with these weights, we come to the inequality

$$\sum_{i=1}^{m} \lambda_i f_i(x)\,\Omega\,0 \qquad (\mathrm{Cons}(\lambda))$$

where Ω is either " $>$ " (this is the case when the weight of at least one strict inequality from $(\mathcal{S})$ is positive), or " $\geq$ " (otherwise). The resulting inequality, due to its origin, is a consequence of the system $(\mathcal{S})$ – it for sure is satisfied by every solution to $\mathcal{S}$. Thus, if $(\mathrm{Cons}(\lambda))$ happens to be contradictory – has no solutions at all – we may be sure that $(\mathcal{S})$ has no solutions; whenever it is the case, we may treat the corresponding vector λ as a "simple certificate" of the fact that $(\mathcal{S})$ is infeasible.

Let us look what does the outlined approach mean when $(\mathcal{S})$ is comprised of *linear* inequalities:

$$(\mathcal{S}): \quad \{\alpha_i^T x\,\Omega_i\,b_i,\ i = 1, ..., m\} \quad \left[\Omega_i = \left\{\begin{array}{l} " > " \\ " \geq " \end{array}\right.\right]$$

Here the "combined inequality" also is linear:

$$(\mathrm{Cons}(\lambda)): \qquad (\sum_{i=1}^{m} \lambda a_i)^T x\,\Omega\,\sum_{i=1}^{m} \lambda b_i$$

(Ω is " $>$ " whenever $\lambda_i > 0$ for at least one i with $\Omega_i =$ " $>$ ", and Ω is " $\geq$ " otherwise). Now, when a *linear* inequality

$$d^T x\,\Omega\,e$$

can be contradictory? Of course, it can happen only in the case when the left hand side in this inequality is trivial – identically zero, i.e, only if $d = 0$. Whether in this latter case our inequality is contradictory, it depends on what is the relation Ω: in the case of $\Omega =$ " $>$ " the inequality is contradictory if and only if $e \geq 0$, and in the case of $\Omega =$ " $\geq$ " it is contradictory if and only if $e > 0$. We have established the following simple

Proposition 1.3.1 *Consider a system of linear inequalities*

$$(\mathcal{S}): \qquad a_i^T x\,\Omega_i\,b_i,\ i = 1, ..., m,$$

with n-dimensional vector of unknowns x, where for every i the relation Ω_i is either " $>$ ", or " $\geq$ "; to simplify notation, assume that Ω_i is " $>$ " for $i = 1, ..., m_s$ and $\Omega_i =$ " $\geq$ " for

$i = m_s + 1, ..., m$. Let us associate with $(\mathcal{S})$ two systems of linear inequalities and equations with m-dimensional vector of unknowns λ:

$$\mathcal{T}_{\rm I}: \quad \left\{ \begin{array}{lrcl} (a) & \lambda & \geq & 0; \\ (b) & \sum_{i=1}^m \lambda_i a_i & = & 0; \\ (c_{\rm I}) & \sum_{i=1}^m \lambda_i b_i & \geq & 0; \\ \hline (d_{\rm I}) & \sum_{i=1}^{m_s} \lambda_i & > & 0. \end{array} \right.$$

$$\mathcal{T}_{\rm II}: \quad \left\{ \begin{array}{lrcl} (a) & \lambda & \geq & 0; \\ (b) & \sum_{i=1}^m \lambda_i a_i & = & 0; \\ \hline (c_{\rm II}) & \sum_{i=1}^m \lambda_i b_i & > & 0. \end{array} \right.$$

Assume that at least one of the systems $\mathcal{T}_{\rm I}$, $\mathcal{T}_{\rm II}$ is solvable. Then the system $(\mathcal{S})$ is infeasible.

Proposition 1.3.1 says that in some cases it is easy to certify infeasibility of a linear system of inequalities: a "simple certificate" is a solution to another system of linear inequalities. Note, however, that the existence of a certificate of this latter type to the moment is only *sufficient*, but not *necessary*, condition for the infeasibility of $(\mathcal{S})$. A fundamental result in the theory of linear inequalities is that the sufficient condition in question in fact is also necessary:

Theorem 1.3.1 [General Theorem on Alternative] *In the notation from Proposition 1.3.1, system $\mathcal{S}$ has no solution if and only if either $\mathcal{T}_{\rm I}$, or $\mathcal{T}_{\rm II}$, or both these systems, is/are solvable.*

The proof of the Theorem on Alternative, as well as a number of useful particular cases of it, form one of the topics of the assignment to the lecture. It makes sense to formulate explicitly two most useful principles following from the theorem:

A. *A system of linear inequalities*

$$a_i^T x \, \Omega_i \, b_i, \ \ i = 1, ..., m$$

is infeasible if and only if one can combine the inequalities of the system in a linear fashion (i.e., multiply the inequalities by nonnegative weights and add the results) to get a contradictory inequality, namely, either the inequality $0^T x \geq 1$, or the inequality $0^T x > 0$.

B. *A linear inequality*

$$a_0^T x \, \Omega_0 \, b_0$$

is a consequence of a solvable system of linear inequalities

$$a_i^T x \, \Omega_i \, b_i, \ \ i = 1, ..., m$$

if and only if it can be obtained by combining, in a linear fashion, the inequalities of the system and the trivial inequality $0 > -1$.

It should stressed that the above principles are highly nontrivial and very deep. Consider, e.g., the following system of 4 linear inequalities with two variables u, v:

$$\begin{array}{c} -1 \leq u \leq 1 \\ -1 \leq v \leq 1. \end{array}$$

From these inequalities it follows that

$$u^2 + v^2 \leq 2, \qquad (!)$$

which in turn implies, by the Cauchy inequality, the linear inequality $u + v \leq 2$:

$$u + v = 1 \times u + 1 \times v \leq \sqrt{1^2 + 1^2}\sqrt{u^2 + v^2} \leq (\sqrt{2})^2 = 2. \qquad (!!)$$

The concluding inequality is linear and is a consequence of the original system, but in the demonstration of this fact both steps (!) and (!!) are "highly nonlinear". It is absolutely unclear a priori why the same consequence can, as it is stated by Principle **A**, be derived from the system in a linear manner as well [of course it can – it suffices just to add two inequalities $u \leq 1$ and $v \leq 1$].

Note that the Theorem on Alternative and its corollaries **A** and **B** heavily exploit the fact that we are speaking about *linear* inequalities. E.g., consider the following 2 quadratic and 2 linear inequalities with two variables:

$$\begin{array}{lrcl} (a) & u^2 & \geq & 1; \\ (b) & v^2 & \geq & 1; \\ (c) & u & \geq & 0; \\ (d) & v & \geq & 0; \end{array}$$

along with the quadratic inequality

$$\begin{array}{lrcl} (e) & uv & \leq & 1. \end{array}$$

The inequality (e) clearly is a consequence of $(a) - -(d)$. However, if we extend the system of inequalities $(a) - -(b)$ by all "trivial" – identically true – linear and quadratic inequalities with 2 variables, like $0 > -1$, $u^2 + v^2 \geq 0$, $u^2 + 2uv + v^2 \geq 0$, $u^2 - uv + v^2 \geq 0$, etc., and ask whether (e) can be derived in a *linear* fashion from the inequalities of the extended system, the answer will be negative. Thus, Principle **A** fails to be true already for quadratic inequalities (which is a great sorrow – otherwise there were no difficult problems at all!)

We are about to use the Theorem on Alternative to obtain the basic results of the LP Duality Theory.

1.3.2 Dual to an LP program: the origin

As it was already mentioned, the motivation for constructing the problem dual to an LP program

$$c^T x \to \min \mid Ax - b \geq 0 \quad \left[A = \begin{bmatrix} a_1^T \\ a_2^T \\ \dots \\ a_m^T \end{bmatrix} \in \mathbf{R}^{m \times n} \right] \qquad \text{(LP)}$$

is the desire to get a systematic way to generate lower bounds on the optimal value in (LP). Now, a real a is a lower bound on the optimal value if and only if $c^T x \geq a$ whenever $Ax \geq b$, or, which is the same, if and only if the system of linear inequalities

$$(\mathcal{S}_a): \qquad -c^T x > -a, Ax \geq b$$

has no solutions. And we already know that the latter fact means that some other system of linear equalities (more exactly, at least one of certain pair of systems) does have a solution. Namely, in view of the Theorem on Alternative

(*) *($\mathcal{S}_a$) has no solutions if and only if at least one of the following two systems with $m+1$ unknowns:*

$$\mathcal{T}_{\rm I}: \left\{\begin{array}{lrcl} (a) & \lambda=(\lambda_0,\lambda_1,...,\lambda_m) & \geq & 0; \\ (b) & -\lambda_0 c+\sum_{i=1}^m \lambda_i a_i & = & 0; \\ \hline (c_{\rm I}) & -\lambda_0 a+\sum_{i=1}^m \lambda_i b_i & \geq & 0; \\ (d_{\rm I}) & \lambda_0 & > & 0, \end{array}\right.$$

or

$$\mathcal{T}_{\rm II}: \left\{\begin{array}{lrcl} (a) & \lambda=(\lambda_0,\lambda_1,...,\lambda_m) & \geq & 0; \\ (b) & -\lambda_0 c-\sum_{i=1}^m \lambda_i a_i & = & 0; \\ \hline (c_{\rm II}) & -\lambda_0 a-\sum_{i=1}^m \lambda_i b_i & > & 0 \end{array}\right.$$

– has a solution.

Now assume that (LP) *is feasible.* Our claim is that *under this assumption ($\mathcal{S}_a$) has no solutions if and only if $\mathcal{T}_{\rm I}$ has a solution.*

> Indeed, the implication "$\mathcal{T}_{\rm I}$ has a solution $\Rightarrow$ $\mathcal{S}_a$ has no solution" is readily given by the above remarks. All we should verify is the inverse implication. Thus, assume that ($\mathcal{S}_a$) has no solutions and the system $Ax \leq b$ has a solution, and let us prove that then $\mathcal{T}_{\rm I}$ has a solution. By (*), at least one of the systems $\mathcal{T}_{\rm I}$, $\mathcal{T}_{\rm II}$ has a solution; assuming that the solvable system is not $\mathcal{T}_{\rm I}$, we should conclude that $\mathcal{T}_{\rm II}$ is solvable, and $\lambda_0=0$ for (every) solution to $\mathcal{T}_{\rm II}$ (since a solution to the latter system with $\lambda_0>0$ solves $\mathcal{T}_{\rm I}$ as well). But the fact that $\mathcal{T}_{\rm II}$ has a solution λ with $\lambda_0=0$ is independent of the values of a and c; if this fact would take place, it would mean, by the same Theorem on Alternative, that, e.g., the following modified version of ($\mathcal{S}_a$):
>
> $$0^T x \geq -1, Ax \geq b$$
>
> has no solutions; i.e., a solution to $\mathcal{T}_{\rm II}$ with $\lambda_0=0$ would certify that already the system $Ax \geq b$ has no solutions, which by assumption is not the case. ■

Now, if $\mathcal{T}_{\rm I}$ has a solution, this system has a solution with $\lambda_0=1$ as well (to see this, pass from a solution λ to the one λ/λ_0; this construction is well-defined, since $\lambda_0>0$ for every solution to $\mathcal{T}_{\rm I}$). Now, an $m+1$-dimensional vector $\lambda=(1,y)$ is a solution to $\mathcal{T}_{\rm I}$ if and only if the m-dimensional vector y solves the system of linear inequalities and equations

$$\begin{array}{rcl} y & \geq & 0; \\ A^T y \equiv \sum_{i=1}^m y_i a_i & = & c; \\ b^T y & \geq & a \end{array} \tag{D}$$

Summarizing our observations, we come to the following result.

Proposition 1.3.2 *Assume that system* (D) *associated with the LP program* (LP) *has a solution* (y,a). *Then a is a lower bound on the optimal value in* (LP). *Vice versa, if a is a lower bound on the optimal value of* <u>*feasible*</u> *LP program* (LP)*, then a can be extended by a properly chosen m-dimensional vector y to a solution to* (D).

We see that the entity responsible for lower bounds on the optimal value of (LP) is the system (D): every solution to the latter system induces a bound of this type, and *in the case when* (LP) *is feasible*, all lower bounds can be obtained from solutions to (D). Now note that if (y,a) is a solution to (D), then the pair (y,b^Ty) also is a solution to the same system, and the lower bound on c^* yielded by the latter solution – i.e., b^Ty – is not worse than the lower bound a yielded by the former solution. Thus, as far as lower bounds on c^* are concerned, we lose nothing by restricting ourselves with the solutions (y, a) to (D) with $a = b^T y$; the best lower bound on c^* given by (D) is therefore the optimal value in the problem

$$b^T y \to \max \mid \; A^T y = c, y \geq 0 \qquad \text{(LP}^*\text{)}$$

The problem we end up with is called the problem *dual* to the *primal* problem (LP); note that this problem also is a Linear Programming program. All we know about the dual problem to the moment is the following:

Proposition 1.3.3 *Whenever y is a feasible solution to* (LP*), *the corresponding value of the dual objective $b^T y$ is a lower bound on the optimal value c^* in* (LP). *If* (LP) *is feasible, then, for every lower bound a on the optimal value of* (LP), *there exists a feasible solution y to* (LP*) *with $b^T y \geq a$ (i.e., a feasible solution y yielding, via the corresponding value of the dual objective $b^T y$, a lower bound not worse than a).*

1.3.3 The LP Duality Theorem

Proposition 1.3.3 is in fact equivalent to the following

Theorem 1.3.2 [Duality Theorem in Linear Programming] *Consider a linear programming program*

$$c^T x \to \min \mid Ax \geq b \qquad \text{(LP)}$$

along with its dual

$$b^T y \to \max \mid \; A^T y = c, y \geq 0 \qquad \text{(LP}^*\text{)}$$

Then

1) *The duality is symmetric: the problem dual to dual is equivalent to the primal;*

2) *The value of the dual objective at every dual feasible solution is $\leq$ the value of the primal objective at every primal feasible solution*

3) *The following 5 properties are equivalent to each other:*

(i) *The primal is feasible and below bounded.*

(ii) *The dual is feasible and above bounded.*

(iii) *The primal is solvable.*

(iv) *The dual is solvable.*

(v) *Both primal and dual are feasible.*

Whenever (i) $\equiv$ (ii) $\equiv$ (iii) $\equiv$ (iv) $\equiv$ (v) *is the case, the optimal values in the primal and the dual problems are equal to each other.*

Proof. 1) is quite straightforward: writing the dual problem (LP*) in our standard form, we get

$$-b^T y \to \min \mid \begin{bmatrix} I_m \\ A^T \\ -A^T \end{bmatrix} y - \begin{pmatrix} 0 \\ -c \\ c \end{pmatrix} \geq 0,$$

I_m being the m-dimensional unit matrix. Applying the duality transformation to the latter problem, we come to the problem

$$0^T\xi + c^T\eta + (-c)^T\zeta \to \max \mid \left\{ \begin{array}{rcl} \xi & \geq & 0 \\ \eta & \geq & 0 \\ \zeta & \geq & 0 \\ \xi - A\eta + A\zeta & = & -b \end{array} \right.$$

which is clearly equivalent to (LP) (set $x = \eta - \zeta$).

2) is readily given by Proposition 1.3.3.

3):

(i)⇒(iv): If the primal is feasible and below bounded, its optimal value c^* (which of course is a lower bound on itself) can, by Proposition 1.3.3, be (non-strictly) majorated by a lower bound on c^* of the type $b^T y^*$ given by a feasible solution y^* to (LP*). In the situation in question, of course, $b^T y^* = c^*$; on the other hand, in view of the same Proposition 1.3.3 the optimal value in the dual is $\leq c^*$. We conclude that the optimal value in the dual is attained and is equal to the optimal value in the primal.

(iv)⇒(ii): evident;

(ii)⇒(iii): This implication, in view of the primal-dual symmetry (see 1)), follows from the already proved implication (i)⇒(iv).

(iii)⇒(i): evident.

We have seen that (i)≡(ii)≡(iii)≡(iv) and that the first (and consequently each) of these 4 equivalent properties implies that the optimal value in the primal problem is equal to the optimal value in the dual one. All which remains to prove is the equivalence between (i)–(iv), on one hand, and (v), on the other hand. This is immediate: (i)–(iv), of course, imply (v); vice versa, in the case of (v) the primal is not only feasible, but also below bounded (this is an immediate consequence of the feasibility of the dual problem, see 2)), and (i) follows. ■

An immediate corollary of the LP Duality Theorem is the following *necessary and sufficient* optimality condition in LP:

Theorem 1.3.3 [Necessary and sufficient optimality conditions in linear programming] *Consider an LP program* (LP) *along with its dual* (LP*), *and let* (x, y) *be a pair of primal and dual feasible solutions. The pair is comprised of optimal solutions to the respective problems if and only if*

$$y_i[Ax - b]_i = 0, \;\; i = 1, ..., m, \qquad \text{[complementary slackness]}$$

as well as if and only if

$$c^T x - b^T y = 0 \qquad \text{[zero duality gap]}$$

Indeed, the "zero duality gap" optimality condition is an immediate consequence of the fact that the value of primal objective at every primal feasible solution is $\geq$ the value of the dual objective at every dual feasible solution, while the optimal values in the primal and the dual are equal to each other, see Theorem 1.3.2. The equivalence between the "zero duality gap" and the "complementary slackness" optimality conditions is given by the following computation: whenever x is primal feasible and y is dual feasible, the products $y_i[Ax-b]_i$, $i=1,...,m$, are nonnegative, while the sum of these products is nothing but the duality gap:

$$y^T[Ax-b] = (A^Ty)^Tx - b^Ty = c^Tx - b^Ty.$$

Thus, the duality gap can vanish at a primal-dual feasible pair (x,y) if and only if all products $y_i[Ax-b]_i$ for this pair are zeros.

1.3.4 Illustration: the problem dual to the Tschebyshev approximation problem

Let us look what is the program dual to the (LP form of) the Tschebyshev approximation problem. Our primal LP program is

$$t \to \min \mid t - [b_i - a_i^Tx] \geq 0, t - [-b_i + a_i^Tx] \geq 0, \ i=1,...,M. \tag{P}$$

Consequently, the dual problem is the LP program

$$\sum_{i=1}^M b_i[\eta_i - \zeta_i] \to \max \mid \left\{ \begin{array}{rcl} \eta_i, \zeta_i & \geq & 0, \ i=1,...,M; \\ \sum_{i=1}^M[\eta_i + \zeta_i] & = & 1; \\ \sum_{i=1}^N[\eta_i - \zeta_i]a_i & = & 0. \end{array} \right.$$

In order to simplify the dual problem, let us pass from the variables η_i, ζ_i to the variables $p_i = \eta_i + \zeta_i$, $q_i = \eta_i - \zeta_i$. With respect to the new variables the problem becomes

$$\sum_{i=1}^M b_iq_i \to \max \mid \left\{ \begin{array}{rcl} p_i \pm q_i & \geq & 0, \ i=1,...,M; \\ \sum_{i=1}^M p_i & = & 1; \\ \sum_{i=1}^M q_ia_i & = & 0. \end{array} \right.$$

In the resulting problem one can easily eliminate the p-variables, thus coming to the problem

$$\sum_{i=1}^M b_iq_i \to \max \mid \left\{ \begin{array}{rcl} \sum_{i=1}^M q_ia_i & = & 0, \\ \sum_{i=1}^M |q_i| & \leq & 1. \end{array} \right. \tag{D}$$

The primal-dual pair (P) – (D) admits a nice geometric interpretation. Geometrically, the primal problem (P) is:

> *Given a vector $b \in \mathbf{R}^M$ and a linear subspace L in $\mathbf{R}^M$ spanned by N vectors $a_1,...,a_N$, find a closest to b in the norm*
>
> $$\| z \|_\infty = \max_{i=1,...,M} |z|_i$$
>
> *on $\mathbf{R}^M$ element of L.*

The dual problem (D) is

> *Given the same data as in* (P)*, find a linear functional $z \mapsto q^Tz$ on $\mathbf{R}^M$ of the $\| \cdot \|_1$-norm*
>
> $$\| q \|_1 = \sum_{i=1}^M |q_i|$$
>
> *not exceeding 1 which separates best of all the point b and the linear subspace L, i.e., which is identically 0 on L and is as large as possible at b.*

The Duality Theorem says, in particular, that the optimal values in (P) and in (D) are equal to each other; in other words,

> *the $\|\cdot\|_\infty$-distance form a point $b \in \mathbf{R}^M$ to a linear subspace $L \subset \mathbf{R}^M$ always is equal to the maximum quantity by which b can be separated from L by a linear functional of $\|\cdot\|_1$-norm 1.*

This is the simplest case of a very general and useful statement (a version of the Hahn-Banach Theorem):

> *The distance from a point b in a linear normed space $(E, \|\cdot\|)$ to a linear subspace $L \subset E$ is equal to the supremum of quantities by which b can be separated from L by a linear functional of the conjugate to $\|\cdot\|$ norm 1.*

1.3.5 Application: Truss Topology Design

Surprisingly enough, Linear Programming in general, and the Tschebyshev approximation problem in particular, may serve to solve seemingly highly nonlinear optimization problems. One of the most interesting examples of this type is the Truss Topology Design (TTD) problem.

Truss Topology Design: what is it

A *truss* is a mechanical construction comprised of linked with each other thin elastic bars, like an electric mast, a railroad bridge or the Eifel Tower. The points where the bars are linked to each other are called *nodes*. A truss can be subjected to an external *load* – a collection of simultaneous forces acting at the nodes:

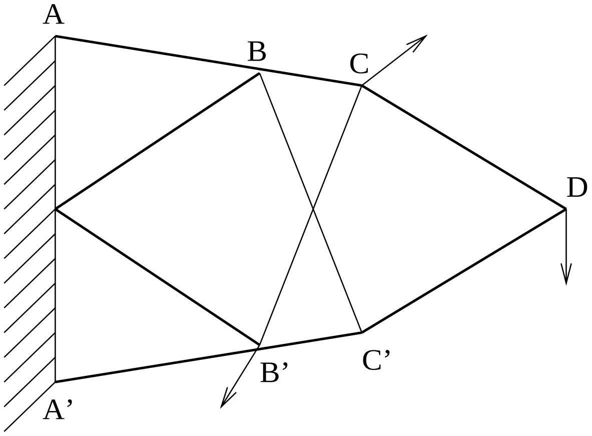


Figure 1.3. A simple planar truss and a load

nodes:	A,A',B,B',C,C',D
bars:	AB,BC,CD,A'B',B'C',C'D,BC',B'C
forces:	arrows

Under a load, the construction deformates a bit, until the tensions caused by the deformation compensate the external forces. When deformated, the truss stores certain potential energy; this energy is called the *compliance* of the truss with respect to the load. The less is the compliance, the more rigid is the truss with respect to the load in question.

In the simplest Truss Topology Design problem, we are given

- a *nodal set* – a finite set of points on the plane or in the space where the bars of the truss to be designed can be linked to each other,
- *boundary conditions* saying that some nodes are supported and cannot move (like nodes A,A' on the "wall" AA' on Fig. 1.3),
- a *load* – a collection of external forces acting at the nodes,

and the goal is *to design a truss of a given total weight capable to withstand best of all the given load*, i.e., to link some pairs of the nodes by bars of the given total weight in such a way that the compliance of the resulting truss with respect to the load of interest will be as small as possible.

An attractive feature of the TTD problem is that although it seemingly deals with the *sizing* (volumes of the bars) only, it in fact finds the structure of the truss as well. Indeed, we may start with a "dense" nodal grid and allow all pairs of nodes to be connected by bars. In the optimal truss yielded by the optimization some of the bars (typically – the majority of them) will get zero weights; in other words, the optimization problem will by itself decide which nodes to use and how to link them, i.e., it will find both the optimal pattern ("topology") of the construction and the optimal sizing.

Model's derivation

In order to pose the TTD problem as an optimization program, let us look in more details what happens with a truss under a load. Consider a particular bar AB (Fig. 1.4) in the unloaded truss; after the load is applied, the nodes A and B move a little bit, as shown on Fig. 1.4.

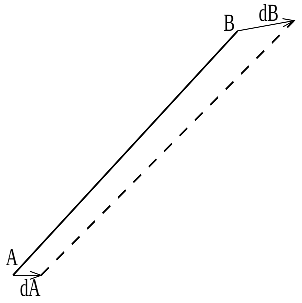


Figure 1.4. A bar before (solid) and after (dashed) load is applied

Assuming the nodal displacements dA and dB small and neglecting the second order terms, the elongation dl of the bar under load is the projection of the vector $dB - dA$ on the direction of the bar:

$$dl = (dB - dA)^T (B - A) / \parallel B - A \parallel .$$

The tension (the magnitude of the reaction force) caused by this elongation, by Hooke's law, is

$$\kappa \frac{dl \times S_{AB}}{\parallel B - A \parallel} = \kappa \frac{dl \times t_{AB}}{\parallel B - A \parallel^2},$$

where κ is a characteristic of the material (Young's module), S_{AB} is the cross-sectional area of the bar AB and t_{AB} is the volume of the bar. Thus, the tension is

$$\tau = \kappa t_{AB} (dB - dA)^T (B - A) \parallel B - A \parallel^{-3} .$$

The reaction force at the point B associated with the tension is the vector

$$\begin{array}{rcl} -\tau(B-A) \parallel B-A \parallel^{-1} & = & \kappa t_{AB}[(dB-dA)^T(B-A)](B-A) \parallel B-A \parallel^{-4} \\ & = & -t_{AB}[(dB-dA)^T\beta_{AB}]\beta_{AB}, \\ & & \beta_{AB} = \sqrt{\kappa}(B-A) \parallel B-A \parallel^{-2} . \end{array} \tag{1.3.1}$$

Note that the vector β_{AB} depends on the positions of the nodes linked by the bar and is independent of the load and of the design.

Now let us look what is the potential energy stored by our bar as a result of its elongation. Mechanics says that this energy is the half-product of the tension and the elongation, i.e., it is

$$\begin{array}{rcl} \frac{\text{tension}\times\text{elongation}}{2} & = & \frac{\tau dl}{2} = \frac{[(dB-dA)^T(B-A)\|B-A\|^{-1}][\kappa t_{AB}(dB-dA)^T(B-A)\|B-A\|^{-3}]}{2} \\ & = & \frac{1}{2}t_{AB}\left[(dB-dA)^T\beta_{AB}\right]^2 . \end{array} \tag{1.3.2}$$

Now it is easy to build the relevant model. Let m be the number of nodes in the nodal grid, and m_{f} be the number of the "free" nodes – those which are not fixed by the boundary conditions [4]. We define the space $\mathbf{R}^M$ of *virtual displacements* of the construction as the direct sum of the spaces of displacements of the free nodes; thus, M is either $2m_{\mathrm{f}}$ or $3m_{\mathrm{f}}$, depending on whether we are speaking about planar or spatial trusses. A vector v from $\mathbf{R}^M$ represents a displacement of the nodal grid: a free node ν corresponds to a pair (planar case) or a triple (spatial case) of (indices of) the coordinates of v, and the corresponding sub-vector $v[\nu]$ of v represents the "physical" 2D- or 3D-displacement of the node ν. It is convenient to define the sub-vectors $v[\nu]$ for fixed nodes ν as well; by definition, these sub-vectors are zeros.

Now, a load – a collection of external forces acting at the free nodes[5] – can be represented by a vector $f \in \mathbf{R}^M$; for every free node ν, the corresponding sub-vector $f[\nu]$ of f is the external force acting at ν.

Now let n be the number of tentative bars (i.e., pair connections between distinct nodes from the grid, at least one node in the pair being free). Let us order somehow all n our tentative bars and consider i-th of them. This bar links two nodes $\nu'(i)$, $\nu''(i)$, i.e., two points A_i and B_i from our "physical space" (i.e., from the 2D plane in the planar case and from the 3D space in the spatial case). Let us associate with tentative bar # i a vector $b_i \in \mathbf{R}^M$ as follows (cf. (1.3.1)):

$$b_i[\nu] = \begin{cases} \beta_{A_iB_i}, & \nu = \nu''(i) \text{ and } \nu \text{ is free} \\ -\beta_{A_iB_i}, & \nu = \nu'(i) \text{ and } \nu \text{ is free} \\ 0, & \text{in all remaining cases} \end{cases} \tag{1.3.3}$$

Now, a particular truss can be identified with a nonnegative vector $t = (t_1, ..., t_n)$, t_i being the volume the tentative bar # i is assigned in the truss. Consider a truss t, and let us look what are the reaction forces caused by a displacement v of the nodes of the truss. From (1.3.1) and (1.3.3) it follows that for every free node ν the component of the reaction force caused, under the displacement v, by i-th bar at the node ν is $-t_i(b_i^Tv)b_i[\nu]$. Consequently, the total reaction force at the node ν is

$$-\sum_{i=1}^{n} t_i(b_i^Tv)b_i[\nu],$$

[4] We assume for simplicity that a node is either completely fixed or is completely free. In some TTD problems it makes sense to speak also about *partially fixed* nodes which can move along a given line (or along a given 2D plane in the 3D space; this option makes sense for spatial trusses only). It turns out that the presence of partially fixed nodes does not change the mathematical structure of the resulting optimization problem

[5] It makes no sense to speak about external force acting at a fixed node – such a force will be compensated by the physical support which makes the node fixed

and the collection of the reaction forces at the nodes is

$$-\sum_{i=1}^{n} t_i(b_i^T v)b_i = -\left[\sum_{i=1}^{n} t_i b_i b_i^T\right] v.$$

We see that the M-dimensional vector representing the reaction forces caused by a displacement v depends on v linearly:

$$f_{\rm r} = -A(t)v,$$

where

$$A(t) = \sum_{i=1}^{n} t_i b_i b_i^T \qquad (1.3.4)$$

is the so called *bar-stiffness matrix* of the truss; this is an $M \times M$ symmetric matrix which depends linearly on the design variables – the volumes of tentative bars.

Now, at the equilibrium the reaction forces must compensate the external ones, which gives us a system of linear equations determining the displacement of the truss under an external load f:

$$A(t)v = f. \qquad (1.3.5)$$

To complete the model, we should also write down an expression for the compliance – the potential energy stored by the truss at equilibrium. According to (1.3.2) – (1.3.3), this energy is

$$\begin{array}{rcl}\frac{1}{2}\sum_{i=1}^{n} t_i\left[(v[\nu''(i)] - v[\nu'(i)])^T\beta_{A_iB_i}\right]^2 & = & \frac{1}{2}\sum_{i=1}^{n} t_i(v^T b_i)^2 \\ & = & \frac{1}{2}v^T\left[\sum_{i=1}^{n} t_i b_i b_i^T\right] v \\ & = & \frac{1}{2}v^T A(t)v \\ & = & \frac{1}{2}f^T v,\end{array}$$

the concluding equality being given by (1.3.5). Thus, the compliance of a truss t under a load f is

$$\mathrm{Compl}_f(t) = \frac{1}{2}f^T v, \qquad (1.3.6)$$

v being the corresponding displacement, see (1.3.5).

The expression for the compliance possesses a transparent mechanical meaning:

> *The compliance is just one half of the mechanical work performed by the external load on the displacement of the truss till the equilibrium.*

Remark 1.3.1 Mathematically speaking, there is a gap in the above considerations: the linear system (1.3.5) can have more than one solution v (same as can have no solutions at all); why do we know that in the former case the value of the right hand side of (1.3.6) is independent of the particular choice of the solution to the equilibrium equation? And what to do if (1.3.5) has no solutions?

The answers to these questions are as follows. If (1.3.5) has no solutions, that means that the truss t cannot carry the load f: it is crushed by this load. In this case it makes sense to define $\mathrm{Compl}_f(t)$ as $+\infty$. If (1.3.5) is solvable, then the quantity $f^T v$ does not depend on a particular choice of a solution to the equation. Indeed, if v solves (1.3.5), then

$$f = \sum_{i=1}^{n} t_i b_i(b_i^T v) \Rightarrow f^T v = \sum_{i=1}^{n} t_i(b_i^T v)^2;$$

the resulting quantity is independent of a particular choice of v due to the following observation:

Whenever $t \geq 0$, for every i such that $t_i > 0$, the quantity $b_i^T v$ does not depend on a particular choice of a solution to (1.3.5).

Indeed, if v, v' are solutions to (1.3.5), then

$$\begin{array}{rcl} \sum_{i=1}^n t_i b_i (b_i^T [v - v']) & = & 0 \Rightarrow \\ [v - v']^T \sum_{i=1}^n t_i b_i (b_i^T [v - v']) & = & 0 \Rightarrow \\ \sum_{i: t_i > 0} t_i \left(b_i^T [v - v']\right)^2 & = & 0. \end{array}$$

Now we can formulate the problem of designing the stiffest, with respect to a given load, truss of a given weight as the following optimization problem:

Problem 1.3.1 [The simplest TTD problem] *Given a "ground structure"*

$$M; n; \{b_i \in \mathbf{R}^M\}_{i=1}^n,^{6)}$$

a load $f \in \mathbf{R}^M$ and a total bar volume $w > 0$, find a truss $t = (t_1, ..., t_n)$ with nonnegative t_i satisfying the resource constraint

$$\sum_{i=1}^n t_i \leq w \tag{1.3.7}$$

with the minimum possible compliance $\mathrm{Compl}_f(t)$ *with respect to the load f.*

When speaking about the TTD problem, we always make the following

Assumption 1.3.1 *The vectors $\{b_i\}_{i=1}^n$ span the entire $\mathbf{R}^M$.*

This assumption has a very transparent mechanical interpretation. Let us look at a "full" truss – one where all tentative bars are of positive weights. Assumption 1.3.1 says that there should be no nonzero displacement v orthogonal to all b_i, or, which is the same, an arbitrary nonzero displacement should cause nonzero tensions in some bars of our full truss. In other words, the assumption just says that our boundary conditions forbid rigid body motions of the nodal set.

LP form of the TTD problem

As stated above, the Truss Topology Design problem 1.3.1 does not resemble an LP program at all: although the constraints

$$t \geq 0; \sum_{i=1}^n t_i \leq w$$

are linear, the objective $\mathrm{Compl}_f(t)$ given by (1.3.4) – (1.3.6) is highly nonlinear.

Surprisingly enough, the TTD problem can be converted into an LP program. The corresponding transformation is as follows.

For a loaded truss, a *stress* in a bar is the absolute value of the corresponding tension (i.e., the magnitude of the reaction force caused by bar's deformation) divided by the cross-sectional area of the bar; the larger is this quantity, the worse are the conditions the material is working in. According to (1.3.1), (1.3.3), the stress in bar # i is, up to the constant factor $\sqrt{\kappa}$ which is of no importance for us, a simple function of the displacement v of the truss:

$$s_i = |b_i^T v|. \tag{1.3.8}$$

Now let us formulate the following intermediate problem:

[6] From the engineering viewpoint, a ground structure is the data of a particular TTD problem, i.e., a particular nodal set along with its partition into fixed and free nodes, the Young modulus of the material, etc. The "engineering data" define, as explained above, the "mathematical data" of the TTD problem.

Problem 1.3.2 *Given a ground structure $M, n, \{b_i\}_{i=1}^n$ and a load f, find displacement v which maximizes the work $f^T v$ of the load under the constraint that all stresses are ≤ 1:*

$$f^T v \to \max \mid |b_i^T v| \leq 1,\ i = 1, ..., n. \tag{1.3.9}$$

A computation completely similar to the one in Section 1.3.4 demonstrates that the problem dual to (1.3.9) is (equivalent to) the program

$$\sum_{i=1}^n |q_i| \to \min \mid \sum_{i=1}^n q_i b_i = f. \tag{1.3.10}$$

Note that both the primal and the dual are feasible (for the primal it is evident, the feasibility of the dual follows from Assumption 1.3.1). By the LP Duality Theorem, both problems are solvable with common optimal value w_*. Let v^* be an optimal solution to (1.3.9) and q^* be an optimal solution to (1.3.10). It is easy to see that the complementary slackness optimality condition results in

$$|q_i^*| = q_i^*(b_i^T v^*),\ i = 1, ..., n. \tag{1.3.11}$$

Assuming $f \neq 0$ (this is the only case of interest in the TTD problem), we ensure that $w_* = \sum_{i=1}^n |q_i^*| > 0$, so that the vector

$$t^* : \quad t_i^* = \frac{w}{w_*}|q_i^*|,\ i = 1, ..., n$$

(w is the material resource in the TTD problem) is well-defined and is a feasible truss (i.e., is nonnegative and satisfies the resource constraint). We claim that

> (*) *The vector t^* is an optimal solution to the TTD problem, and $v^+ = \frac{w_*}{w}v^*$ is the corresponding displacement.*
>
> Indeed, we have
>
> $$\begin{array}{rcl} \sum_{i=1}^n t_i^* b_i b_i^T v^+ & = & \sum_{i=1}^n |q_i| b_i (b_i^T v^*) \\ & & \text{[by construction of } t^*, v^+\text{]} \\ & = & \sum_{i=1}^n q_i b_i \\ & & \text{[by (1.3.11)]} \\ & = & f \\ & & \text{[see (1.3.10)]} \end{array}$$
>
> so that v^+ indeed is the displacement of the truss t^* under the load f. The corresponding compliance is
>
> $$\begin{array}{rcl} \text{Compl}_f(t^*) & = & \frac{1}{2} f^T v^+ \\ & = & \frac{1}{2} \sum_{i=1}^n q_i^* b_i^T v^+ \\ & & \text{[since } f = \sum_i q_i^* b_i\text{]} \\ & = & \frac{1}{2} \frac{w_*}{w} \sum_{i=1}^n q_i^* b_i^T v^* \\ & = & \frac{w_*}{2w} \sum_{i=1}^n |q_i^*| \\ & & \text{[see (1.3.11)]} \\ & = & \frac{w_*^2}{2w}. \end{array} \tag{1.3.12}$$
>
> Thus, t^* is a feasible solution to the TTD problem with the value of the objective $\frac{w_*^2}{2w}$. To prove that the solution is optimal, it suffices to demonstrate that the latter quantity is a lower bound on the optimal value in the TTD problem. To see this, let t be a feasible solution to the TTD problem and v be the corresponding displacement. Let also
>
> $$q_i = t_i(b_i^T v).$$

We have

$$\sum_{i=1}^{n} q_i b_i = \sum_{i=1}^{n} t_i (b_i^T v) b_i = f \qquad (1.3.13)$$

(the equilibrium equation, see (1.3.4) – (1.3.5)). Thus, q is a feasible solution to (1.3.10), and

$$\begin{array}{rcl}
\mathrm{Compl}_f(t) & = & \frac{1}{2} f^T v \\
 & = & \frac{1}{2}\sum_{i=1}^{n} t_i (b_i^T v)^2 \\
 & & \text{[see (1.3.13)]} \\
 & = & \frac{1}{2}\sum_{i:q_i\neq 0} \frac{q_i^2}{t_i} \\
 & \geq & \frac{1}{2}\left[\sum_{i:q_i\neq 0} |q_i|\right]^2 \left[\sum_{i:q_i\neq 0} t_i\right]^{-1} \\
 & & \text{[since by the Cauchy inequality} \\
 & & \left(\sum_i |q_i|\right)^2 = \left(\sum_i [t_i^{-1/2}|q_i|] t_i^{1/2}\right)^2 \leq \left(\sum_i q_i^2/t_i\right)\left(\sum_i t_i\right)] \\
 & \geq & \frac{1}{2}\frac{w_*^2}{w},
\end{array}$$

Q.E.D.

Note that (*) not only provides us with a possibility to compute an optimal solution to the TTD problem via LP techniques, but also establishes a very important fact:

> (**) *When optimal truss t^* is loaded by f, the stresses in all bars which are actually present in the design (i.e., get there positive volumes) are equal to each other, so that the material in all bars is under the same working conditions.*

Indeed, as stated by (*), the displacement of t^* under the load f is v^+, i.e., it is proportional to v^*, and it remains to use (1.3.11).

Strictly speaking, the above reasoning is incomplete. First, v^+ is *a* solution to the equilibrium equation associated with t^*; why do we know that (**) holds true for other solutions to this equation? The answer: the stresses in those bars which are actually present in a truss are uniquely defined by the truss and the load, see Remark 1.3.1. Second, (**) is established for *an* optimal solution t^* to the TTD problem, one which can be obtained, in the aforementioned fashion, from an optimal solution to (1.3.13). A priori it may happen that the TTD problem has other optimal solutions; why (**) is true for *every* optimal solution to the TTD problem? It indeed is the case – it can be shown that every optimal solution to the TTD problem can be obtained from an optimal solution to (1.3.13).

Remark 1.3.2 Note that Problem (1.3.10) is, basically, the Tschebyshev approximation problem. Indeed, instead of asking what is the largest possible value of $f^T v$ under the constraints $|b_i^T v| \leq 1$, $i = 1, ..., n$, we might ask what is the minimum value of $\max_i |b_i^T v|$ under the constraint that $f^T v = 1$ – the optimal solutions to these two problems can be easily obtained from each other. Now, the second problem is nothing but the Tschebyshev approximation problem – we can use the equation $f^T v = 1$ to eliminate one of the design variables, thus coming to the Tschebyshev problem in the remaining variables.

1.4 Assignments to Lecture 1

1.4.1 Around uniform approximation

As we have already indicated, the Tschebyshev approximation problem normally arises as a "discrete version" of the best uniform approximation problem on a segment:

> *Given a segment $\Delta = [a, b]$, N basic functions $f_1, ..., f_N$ on the segment and a target function f_0 on it, find the best, in the uniform norm on the segment, approximation of f by a linear combination of f_i:*

$$\| f_0 - \sum_{j=1}^{N} x_j f_j \|_\infty = \sup_{t\in\Delta} |f_0(t) - \sum_{j=1}^{N} x_j f_j(t)| \to \min . \qquad (\mathrm{Appr}(\Delta))$$

The discrete version of the latter problem is obtained by replacing Δ with a finite set $T \subset \Delta$:

$$\| f_0 - \sum_{j=1}^{N} x_j f_j \|_{T,\infty} = \sup_{t\in T} |f_0(t) - \sum_{j=1}^{N} x_j f_j(t)| \to \min . \qquad (\mathrm{Appr}(T))$$

Whenever this indeed is the origin of the Tschebyshev approximation problem, the following questions are of primary interest:

A. What is the "quality of approximation" of (Appr(Δ)) by (Appr(T))? Specifically, may we write down an inequality

$$\| f_0 - \sum_{j=1}^{N} x_i f_i \|_\infty \leq \kappa \| f_0 - \sum_{j=1}^{N} x_i f_i \|_{T,\infty} \quad \forall x \qquad (1.4.1)$$

with appropriately chosen κ? If it is the case, then κ can be viewed as a natural measure of quality of approximating the original problem by its discrete version – the closer to 1 κ is, the better is the quality.

B. Given the total number M of points in the finite set T, how should we choose these points to get the best possible quality of approximation?

The goal of the subsequent series of problems is to provide some information on these two questions. The answers will be given in terms of properties of the functions from the linear space L spanned by $f_0, f_1, ..., f_N$:

$$L = \{f = \sum_{j=0}^{N} \xi_j f_j\}_{\xi\in\mathbf{R}^{N+1}}$$

Given a finite set $T \subset \Delta$, let us say that T is L-*dense*, if there exists $\kappa < \infty$ such that

$$\| f \|_\infty \leq \kappa \| f \|_{T,\infty} \quad \forall f \in L;$$

the minimum value of κ's with the latter property will be denoted by $\kappa_L(T)$. If T is not L-dense, we set $\kappa_L(T) = \infty$. Note that $\kappa_L(T)$ majorates the quality of approximating the problem (Appr(Δ)) by (Appr(T)), and this is the quantity we shall focus on.

Exercise 1.1 [5] *Let L be a finite-dimensional space comprised of continuous functions on a segment Δ, and let T be a finite subset in Δ. Prove that T is L-dense if and only if the only function from L which vanishes on T is $\equiv 0$.*

Exercise 1.2 [5] *Let $\alpha < \infty$, and assume L is $\underline{\alpha\text{-regular}}$, i.e., the functions from L are continuously differentiable and*

$$\| f' \|_\infty \leq \alpha \| f \|_\infty \quad \forall f \in L.$$

Assume that $T \subset \Delta$ is such that the distance from a point in Δ to the closest point of T does not exceed $\beta < \alpha^{-1}$. Prove that under these assumptions

$$\kappa_L(T) \leq \frac{1}{1 - \alpha\beta}.$$

Exercise 1.3 [5] *Let L be a k-dimensional linear space comprised of continuously differentiable functions on a segment Δ. Prove that L is α-regular for some α; consequently, choosing a fine enough finite grid $T \subset \Delta$, we can ensure a given quality of approximating* $\mathrm{Appr}(\Delta)$ *by* $\mathrm{Appr}(T)$.

To use the simple result stated in Exercise 1.2, we should know something about regular linear spaces L of functions. The most useful result of this type known to us is the following fundamental fact:

Theorem 1.4.1 [Theorem of Sergei N. Bernshtein on trigonometric polynomials] *Let $\Delta = [0, 2\pi]$, and let f be a trigonometric polynomial of degree $\leq k$ on Δ:*

$$f(t) = a_0 + \sum_{l=1}^{k} [a_0 \cos(lt) + b_0 \sin(lt)]$$

with real or complex coefficients. Then

$$\| f' \|_\infty \leq k \| f \|_\infty .$$

Note that the inequality stated in the Bernshtein Theorem is exact: for the trigonometric polynomial

$$f(t) = \cos(kt)$$

of degree k the inequality becomes equality.

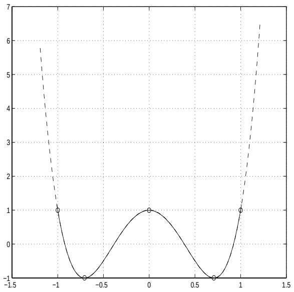
We see that the space of trigonometric polynomials of degree $\leq k$ on $[0, 2\pi]$ is k-regular. What about the space of *algebraic* polynomials of degree $\leq k$ on the segment, say, $[-1, 1]$?

Consider the *Tschebyshev polynomial* of degree k defined on the segment $\Delta = [-1, 1]$ by the relation

$$T_k(t) = \cos(k \operatorname{acos}(t))$$

(check that this indeed is a polynomial in t of degree k). This polynomial possesses the following property:

> (A) $\| T_k \|_\infty = 1$, and there are $k + 1$ *points of alternance of* T_k – the points $t_l = \cos\left(\frac{\pi(k-l)}{k}\right) \in \Delta$, $l = 0, ..., k$, where the absolute value of the polynomial is equal to $\| T_k \|_\infty = 1$, and the signs of the values alternate.



Tschebyshev polynomial T_4 and its 5 points of alternance

The derivative of T_k at the point $t = 1$ is k^2; thus, the factor α in an inequality

$$\| T'_k \|_\infty \leq \alpha \| T_k \|_\infty$$

is at least k^2. We conclude that the space L_k of real algebraic polynomials of degree $\leq k$ on the segment $[-1, 1]$ is *not* α-regular for $\alpha < k^2$. Is our space k^2-regular? We guess that the answer is positive, but we were too lazy to find out whether it indeed is the case. What we shall demonstrate is that L_k is $2k^2$-regular.

Exercise 1.4 [10] *Prove that if $f \in L_k$ and $\| f \|_\infty = 1$, then*

$$|f'(1)| \leq k^2 = T'_k(1). \qquad (*)$$

Hint: assuming that $f'(1) > T'_k(1)$, look at the polynomial $p(t) = T_k(t) - \frac{T'_k(1)}{f'(1)} f(t)$. Verify that the values of this polynomial at the points of alternance of T_k are of the same alternating signs that the values of T_k, so that p has at least k distinct zeros on $(-1, 1)$. Taking into account the latter fact and the equality $p'(1) = 0$, count the number of zeros of $p'(t)$.

Derive from $(*)$ that

$$|f'(t)| \leq 2k^2$$

for all $t \in [-1, 1]$ and conclude that L_k is $2k^2$-regular.

Now let us apply the information collected so far to investigating Questions A and B in the two simplest cases, where L is comprised of the trigonometric and the algebraic polynomials, respectively.

Exercise 1.5 [5] *Assume that $\Delta = [0, 2\pi]$, and let L be a linear space of functions on Δ comprised of all trigonometric polynomials of degree $\leq k$. Let also T be an equidistant M-point grid on Δ:*

$$T = \{\frac{(2l + 1)\pi}{M}\}_{l=0}^{M-1}.$$

1) Prove that if $M > k\pi$, then T is L-dense, with

$$\kappa_L(T) \leq \frac{M}{M - k\pi}.$$

2) Prove that the above inequality remains valid if we replace T with its arbitrary "shift modulo 2π", i.e., treat Δ as the unit circumference and rotate T by an angle.

3) Prove that if T is an arbitrary M-point subset of Δ with $M \leq k$, then $\kappa_L(T) = \infty$.

Exercise 1.6 [5] *Let $\Delta = [-1, 1]$, and let L be the space of all algebraic polynomials of degree $\leq k$.*

1) Assume that $2M > \pi k$ and T is the M-point set on Δ comprised of the M points

$$\{t_l = \cos\left(\frac{(2l+1)\pi}{2M}\right)\}_{l=0}^{M-1}.$$

Then T is L-dense, with

$$\kappa_L(T) \leq \frac{2M}{2M - \pi k}.$$

2) Let T be an M-point set on Δ with $M \leq k$. Then $\kappa_L(T) = \infty$.

Exercise 1.7 [7] The result stated in Exercise 1.6 says that when L_k is comprised of all real algebraic polynomials of degree not exceeding k on $[-1, 1]$ and we are interested to ensure $\kappa_L(T) = O(1)$, it suffices to take $M \equiv \mathrm{card}(T) = O(k)$; note, however, that the corresponding grid is non-uniform. Whether we may achieve similar results with a uniform grid? The answer is "no":

Prove that if $T = \{-1 + \frac{2l}{M}\}_{l=0}^{M}$ is the equidistant M-point grid on $\Delta = [-1, 1]$, then

$$\kappa_{L_k}(T) \geq c_1 M^{-1} \exp\{-c_2 k/\sqrt{M}\},$$

with absolute positive constants c_1, c_2. Thus, in order to get $\kappa_{L_k}(T) = O(1)$ for an equidistant grid T, the cardinality of the grid should be nearly quadratic in k.

Hint: let $t_1 = -1, ..., t_{M-1}, t_M = 1$ be the points of the M-point equidistant grid T. Reduce the question to the following one:

given a polynomial $f(t)$ of degree k which is ≤ 1 in absolute value on $[-1, t_{M-1}]$ and is equal to 0 at the point 1, how large may be the polynomial at the point $0.5(t_{M-1} + 1)$? To answer the latter question, look how the Tschebyshev polynomial T_k grows outside the segment $[-1, 1]$ (note that for $t \geq 1$ the polynomial is given by $T_k(t) = \cosh(k\,\mathrm{acosh}(t))$).

1.4.2 Around the Theorem on Alternative

The goal of the subsequent exercises is to prove the General Theorem on Alternative.

From the Homogeneous Farkas Lemma to the Theorem on Alternative

Consider the very particular case of the Theorem on Alternative – the one where we are interested when a specific system

$$\begin{array}{rcl} a^T x & < & 0 \\ a_i^T x & \geq & 0, \ i = 1, ..., m \end{array} \qquad (F)$$

of homogeneous linear inequalities in $\mathbf{R}^n$ has no solutions. The answer is given by the following

Lemma 1.4.1 [The Homogeneous Farkas Lemma] *System (F) has no solutions if and only if the vector a is a linear combination, with nonnegative coefficients, of the vectors $a_1, ..., a_m$:*

$$\{(F) \text{ is infeasible}\} \Leftrightarrow \{\exists \lambda \geq 0 : a = \sum_{i=1}^{m} \lambda_i a_i\}.$$

Exercise 1.8 [3] *Prove that Lemma 1.4.1 is exactly what is said by the Theorem on Alternative as applied to the particular system (F).*

Our plan is as follows: right now we shall demonstrate that the General Theorem on Alternative can be easily obtained from the Homogeneous Farkas Lemma, and in Section 1.4.3 we shall present a direct proof of the Lemma. Thus, for the time being you may use Lemma 1.4.1 as something granted.

Exercise 1.9 [5] *Consider the same system of linear inequalities as in the Theorem on Alternative:*

$$(\mathcal{S}): \quad \left\{\begin{array}{rcl} a_i^T x & > & b_i,\ i = 1, ..., m_{\rm s};\\ a_i^T x & \geq & b_i,\ i = m_{\rm s}+1, ..., m. \end{array}\right.$$

Prove that this system has no solutions if and only if this is the case for the homogeneous system of the type (F) as follows:

$$(\mathcal{S}^*): \quad \left\{\begin{array}{rcl} -s & < & 0;\\ t-s & \geq & 0;\\ a_i^T x - b_i t - s & \geq & 0,\ i = 1, ..., m_{\rm s};\\ a_i^T x - b_i t & \geq & 0,\ i = m_{\rm s}+1, ..., m, \end{array}\right.$$

the unknowns in $(\mathcal{S}^*)$ being x and two additional real variables s, t.

Derive from the above observation and the Homogeneous Farkas Lemma the General Theorem on Alternative.

The next exercise presents several useful consequences of the General Theorem on Alternative.

Exercise 1.10 [5] *Prove the following statements:*

1. [Gordan's Theorem on Alternative] *One of the inequality systems*

$$\text{(I)} \quad Ax < 0,\ x \in \mathbf{R}^n,$$

$$\text{(II)} \quad A^T y = 0,\ 0 \neq y \geq 0,\ y \in \mathbf{R}^m,$$

A being an $m \times n$ matrix, has a solution if and only if the other one has no solutions.

2. [Inhomogeneous Farkas Lemma] *A linear inequality*

$$a^T x \leq p \tag{1.4.2}$$

a consequence of a <u>solvable</u> *system of inequalities*

$$Ax \leq b$$

if and only if

$$a = A^T \nu$$

for some nonnegative vector ν such that

$$\nu^T b \leq p.$$

3. [Motzkin's Theorem on Alternative] *The system*

$$Sx < 0, \quad Nx \leq 0$$

has no solutions if and only if the system

$$S^T\sigma + N^T\nu = 0,\ \sigma \geq 0,\ \nu \geq 0,\ \sigma \neq 0$$

has a solution.

1.4.3 Proof of the Homogeneous Farkas Lemma

Here we present a series of exercises aimed at proving the Homogeneous Farkas Lemma. In fact we present two proofs: a "quick and dirty" one based on the Separation Theorem, and a more "intelligent" proof based on the Helley Theorem.

From the Separation Theorem to the Farkas Lemma

Exercise 1.11 [5] *Let K be a nonempty closed convex set in $\mathbf{R}^n$, and $x \in \mathbf{R}^n$ be a point not belonging to K.*

1) Prove that the distance from x to K is achieved: there exists $y^* \in K$ such that

$$\| x - x^* \|_2 = \min_{y \in K} \| x - y \|_2 .$$

2) Prove that $e = x - x^*$ strictly separates x and K, namely,

$$e^T(x-y) \geq \| e \|_2^2 > 0 \quad \forall y \in K.$$

3) In the situation in question, assume in addition that K is a cone, i.e.,

$$y \in K, \lambda \geq 0 \Rightarrow \lambda y \in K.$$

Prove that in this case

$$e^T x > 0 \quad \& \quad e^T y \leq 0 \quad \forall y \in K.$$

Exercise 1.12 [5] *Let $a_1, ..., a_m \in \mathbf{R}^n$. Consider the conic hull of these vectors, i.e., the set K of all their linear combinations with nonnegative coefficients:*

$$K = \{p = \sum_{i=1}^{m} \lambda_i a_i \mid \lambda \geq 0\}.$$

1) Prove that the set K is convex and, moreover, it is a cone.

2)* Prove that the set K is closed

Hint: Let $p_j = \sum_{i=1}^m \lambda_{ij} a_i$ with $\lambda_{ij} \geq 0$, and let the sequence $\{p_j\}$ converge to a point p; we should prove that p also can be represented as a linear combination, with nonnegative coefficients, of $a_1, ..., a_m$.

A. Assuming that $p \neq 0$ (this is the only nontrivial case), for every j with $p_j \neq 0$ consider the *minimal*in the number of nonzero λ_{ij}'s representation of p_j as a linear combination, with nonnegative coefficients, of a_i's, and prove that the a_i's participating in this combination are linearly independent.

B. Derive from A. that the weight vectors $(\lambda_{1j}, \lambda_{2j}, ..., \lambda_{mj})$ associated with minimal representations of p_j's form a bounded sequence.

C. Derive from B. that $p \in K$.

3) Given 1), 2) and the results of the previous exercise, demonstrate that for any vector $a \notin K$ there exists a vector x such that

$$a^T x < 0, a_i^T x \geq 0, \; i = 1, ..., m,$$

and derive from this fact the Homogeneous Farkas Lemma.

Hint: use as x the negation of the vector which separates a from K.

An "intelligent" proof

We start with the following basic fact:

Theorem 1.4.2 [The Helley Theorem] *Let $A_1, ..., A_M$ be a collection of convex sets in $\mathbf{R}^n$. Assume that the intersection of every $n+1$ sets from the collection is nonempty. Then the intersection of <u>all</u> M sets from the collection also is nonempty.*

Let us derive the Homogeneous Farkas Lemma from the Helley Theorem.

Exercise 1.13 [10] *Let $a, a_1, ..., a_m$ be vectors from $\mathbf{R}^n$ such that the system of inequalities*

$$\begin{array}{rcl} a^T x & < & 0 \\ a_i^T x & \geq & 0, \ i = 1, ..., m \end{array} \qquad (F)$$

has no solutions. We should prove that under this assumption a can be represented as a linear combination, with nonnegative coefficients, of $a_1, ..., a_m$; this is exactly what is said by the Homogeneous Farkas Lemma. The statement is evident in the case when $a = 0$, so that from now on we assume that $a \neq 0$.

Let us set

$$\begin{array}{rcl} \Pi & = & \{x \mid a^T x = -1\}, \\ A_i & = & \{x \in \Pi \mid a_i^T x \geq 0\} \\ & = & \{x \mid a^T x = -1, a_i^T x \geq 0\}. \end{array}$$

Let us call a sub-collection of the collection $\{a_1, ..., a_m\}$ a <u>contradiction</u>, if the sets A_i corresponding to the vectors from the sub-collection have no common point.

1) Prove that the entire collection $\{a_1, ..., a_m\}$ is a contradiction.

According to 1), contradictions exist. Consequently, there exists a minimal – with the minimum possible number of vectors – contradiction. By reordering the vectors a_i, we may assume that $\{a_1, ..., a_k\}$ is a minimal contradiction.

2) Prove that the vector a belongs to the linear span of the vectors $a_1, ..., a_k$.

Hint: Assuming that a does not belong to the linear span of $a_1, ..., a_k$, prove that there exists a vector x which is orthogonal to $a_1, ..., a_k$ and is not orthogonal to a, and conclude that $\{a_1, ..., a_k\}$ is not a contradiction.

3) Prove that the vectors $a_1, ..., a_k$ are linearly independent.*

Hint: Assuming that $a_1, ..., a_k$ are linearly dependent, consider the linear space L spanned by $a_1, ..., a_k$ along with its subsets

$$\begin{array}{rcl} \Pi' & = & \{x \in L : a^T x = -1\}, \\ A_i' & = & \{x \in \Pi' : a_i^T x \geq 0\}, \ i = 1, ..., k. \end{array}$$

3.1) Consider a sub-collection of the collection $A_1', ..., A_k'$. Prove that the sets of this sub-collection have a point in common if and only if the corresponding sets of the collection $A_1, ...A_k$ have a point in common.

3.2) Taking into account that $\{a_1, ..., a_k\}$ is a <u>minimal</u> contradiction, verify that every $k-1$ sets from the collection $A_1', ..., A_k'$ have a point in common.

Applying to the sets $A_1', ..., A_k'$ – they are convex subsets of Π, i.e., essentially, of a $(\dim(L)-1)$-dimensional linear space – the Helley theorem, prove that under the assumption $\dim(L) < k$ the sets $A_1', ..., A_k'$ have a point in common, which is impossible (since $\{a_1, ..., a_k\}$ is a contradiction).

4) Derive from 2), 3) and the fact that $\{a_1, ..., a_k\}$ is a contradiction that a is a linear combination of $a_1, ..., a_k$, and all coefficients in this combination are nonnegative, thus concluding the proof of the Homogeneous Farkas Lemma.

It is time to explain why the proof of the Homogeneous Farkas Lemma sketched in Exercise 1.13 is "more intelligent" than the proof coming from the Separation scheme (Exercises 1.11, 1.12). The reason is that the Helley Theorem itself, same as the reasoning outlined in Exercise 1.13, are *purely algebraic* facts: they do not use compactness arguments, as it does the reasoning from Exercise 1.11. As a result, *the proof sketched in Exercise 1.13 remains valid also in the case when we replace our "universe" $\mathbf{R}^n$, with, say, the linear space $\mathbf{Q}^n$ comprised of n-dimensional vectors with rational coefficients*[7]. From this observation we conclude that the Theorem on Alternative remains valid when we speak about linear inequalities with *rational* coefficients and are looking for *rational* solutions to these inequalities. This is a nontrivial and useful observation (it implies, e.g., that a solvable LP program with rational data has a rational solution).

Note that the proof via the Separation Theorem heavily exploits the compactness arguments and does not work at all in the case of "percolated" space $\mathbf{Q}^n$. Consider, e.g., the "rational plane" $\mathbf{Q}^2$ along with the convex cone

$$K = \{(u,v) \in \mathbf{Q}^2 \mid u + \sqrt{2}v \leq 0\}.$$

A point x from $\mathbf{Q}^2$ not belonging to K cannot be separated from K by a "legitimate" linear functional on $\mathbf{Q}^2$ – there does not exist a rational vector e such that $e^T x > e^T y$ for all $y \in K$. Consequently, in the case of the "rational universe" an attempt to prove the Farkas Lemma via a separation-type reasoning fails at the very first step.

1.4.4 The Helley Theorem

The goal of the subsequent exercises is to establish the Helley theorem and to illustrate some of its applications.

Exercise 1.14 ^{5} *Prove the following*

Theorem 1.4.3 [Radon] *Let $a_1, ..., a_m$ be a collection of $m \geq n+2$ vectors from $\mathbf{R}^n$. There exists a partitioning of the index set $\{1, ..., m\}$ into two nonempty and non-overlapping subsets I and J such that the convex hull of the points $\{a_i\}_{i \in I}$ intersects the convex hull of the points $\{a_i\}_{i \in J}$.*

Hint: note that the system of $m \geq n+2$ homogeneous linear equations

$$\begin{array}{rcl}\sum_{i=1}^m \lambda_i a_i & = & 0 \\ \sum_{i=1}^m \lambda_i & = & 0\end{array}$$

has a nontrivial solution λ^* and set $I = \{i : \lambda_i^* \geq 0\}, J = \{i : \lambda_i^* < 0\}$.

Exercise 1.15 ^{5} *Derive the Helley Theorem from the Radon Theorem.*

Hint: Apply induction on the number M of the sets $A_1, ..., A_M$. To justify the inductive step, it suffices to demonstrate that if the Helley theorem is valid for $M \geq n+1$ sets, it is

[7] $\mathbf{Q}^n$ should be treated as a linear space over the field $\mathbf{Q}$ of rationals, i.e., we allow to multiply the vectors from $\mathbf{Q}^n$ by rational scalars only, not by arbitrary reals.

valid for $M+1$ sets $A_1, ..., A_{M+1}$. To verify this implication, consider the $M+1$ nonempty (by the inductive hypothesis) sets

$$B_1 = A_2 \cap A_3 \cap ... \cap A_{M+1}; B_2 = A_1 \cap A_3 \cap A_4 \cap ... \cap A_{M+1}; ...; B_{M+1} = A_1 \cap A_2 \cap ... \cap A_m.$$

For every i, choose a point $a_i \in B_i$, apply to the resulting collection of $M+1 \geq n+2$ points the Radon Theorem and demonstrate that every point from the intersection of the corresponding convex hulls is a common point of the sets $A_1, ..., A_{M+1}$.

Exercise 1.16 [5] *Consider the Tschebyshev approximation problem:*

$$\max_{i=1,...,M} |a_i^T x - b_i| \to \min, \qquad (T)$$

and let k be the rank of the system $a_1, ..., a_M$. Prove that one can choose a subset $J \subset \{1, ..., M\}$ containing no more than $k+1$ indices in such a way that the optimal value in the "relaxed" problem

$$\max_{i \in J} |a_i^T x - b_i| \to \min \qquad (T_J)$$

is equal to the optimal value σ^* in (T).

Hint: Look at the convex sets $X_i = \{x \mid |a_i^T x - b_i| < \sigma^*\}$.

Prove that if every k of the vectors $a_1, ..., a_M$ are linearly independent and $\sigma^* > 0$, then for every optimal solution x^* to (T) there exist $k+1$ "indices of alternance" – there exists a subset $J \subset \{1, ..., M\}$ of the cardinality $k+1$ such that

$$|a_i^T x^* - b_i| = \sigma^* \quad \forall i \in J.$$

Cubature formulas and the Gauss points

A cubature formula is a formula of the type

$$\int_\Delta f(t)dt \approx \sum_{i=1}^N \alpha_i f(t_i)$$

with *nonnegative* weights α_i. Given a cubature formula (i.e., a finite set of cubature points $t_1, ..., t_N$ and nonnegative weights $\alpha_1, ..., \alpha_N$), one may ask how wide is the set of functions for which the formula is precise. E.g., the equidistant 2-point formula

$$\int_{-1}^1 f(t)dt \approx f(-1/2) + f(1/2)$$

is exact for linear functions, but is not exact for quadratic polynomials. In contrast to this, the Gauss formula

$$\int_{-1}^1 f(t)dt \approx f(-1/\sqrt{3}) + f(1/\sqrt{3})$$

is precise on all polynomials of degree ≤ 3.

It turns out that the Helley theorem and the Farkas Lemma allow to get the following very general result:

(*) *Let Δ be a subset of $\mathbf{R}^k$, L be a n-dimensional linear space comprised of continuous real-valued functions on Δ and $I(f) : L \to \mathbf{R}$ be "an integral" – a linear functional such that $I(f) \geq 0$ for every $f \in L$ such that $f(t) \geq 0$ everywhere on Δ. Assume also*[8)] *that if a function $f \in L$ is nonnegative on Δ and is not identically 0, then $I(f) > 0$. Then there exists a precise n-point cubature formula for I, i.e., exist n points $t_1, ..., t_n \in \Delta$ and n nonnegative weights $\alpha_1, ..., \alpha_n$ such that*

$$I(f) = \sum_{i=1}^{n} \alpha_i f(t_i) \quad \forall f \in L.$$

Exercise 1.17 ^{7} *1. Prove (*) for the case of finite Δ.*

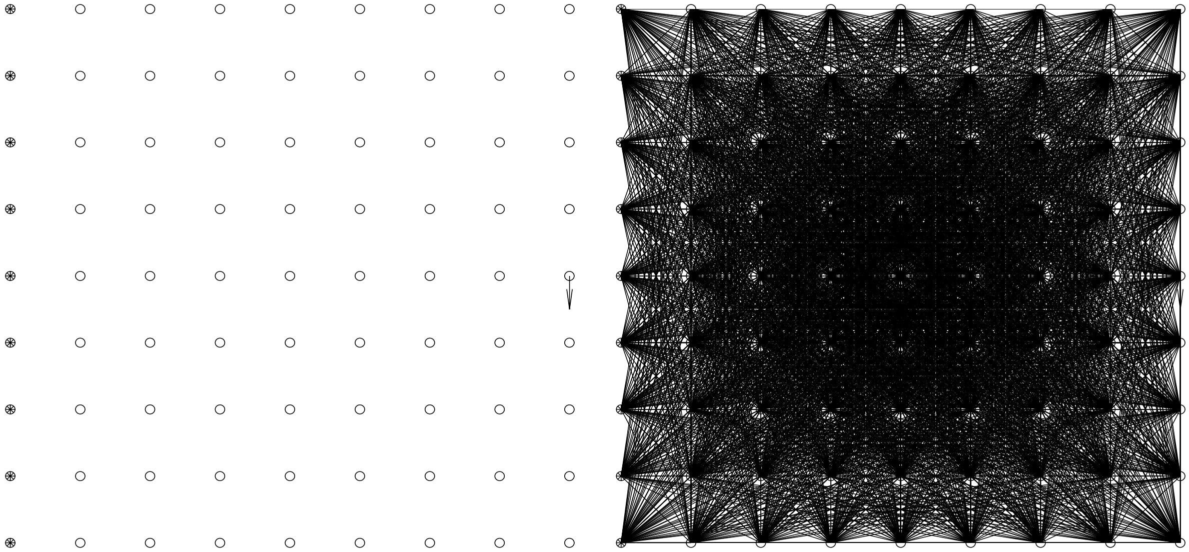
Hint: Assuming that I is not identically zero, associate with points $t \in \Delta$ the convex sets $A_t = \{f \in L \mid f(t) \leq 0, I(f) = 1\}$ and prove that there exist n sets $A_{t_1}, ..., A_{t_n}$ of this type with empty intersection. Apply the Homogeneous Farkas Lemma to the linear forms $f(t_1), ..., f(t_n), I(f)$ of $f \in L$.

2. Prove (*) for the general case.

[8)] In fact this additional assumption is redundant

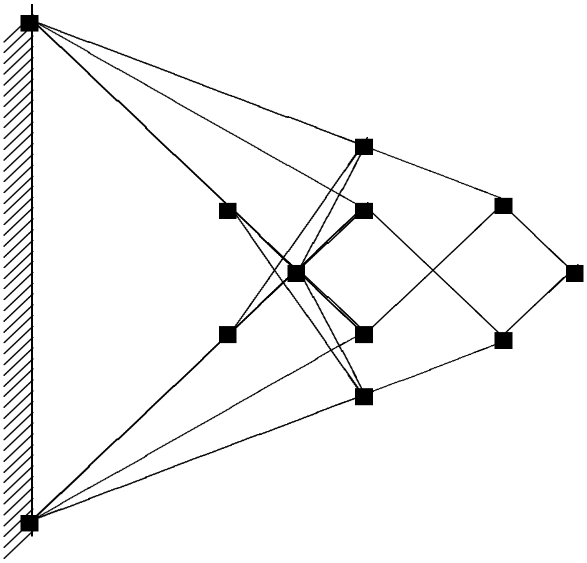
1.4.5 How many bars are there in an optimal truss?

Let us look at the following ground structure:



9 × 9 planar nodal grid and the load (left); 3204 tentative bars (right)

(the most left nodes are fixed; the dimension of the space of displacments is $M = 2 \times 8 \times 9 = 144$)



Optimal truss (24 bars)

We see that the optimal bar uses just 24 of 3204 tentative bars. Is this phenomenon typical or not? As you shall see in a while, the answer is positive: there always exists an optimal truss with no more than $M + 1$ bars, M being the dimension of the space displacements. Thus, with the above ground structure we may be sure in advance that there exists an optimal truss if not with 22, then with 145 bars; this still is by order of magnitude less than the number of tentative bars.

Exercise 1.18 [7] *Consider a Linear Programming problem*

$$c^T x \to \min \mid Ax = b, x \geq 0$$

with $k \times n$ matrix A of rank l. Assuming the program solvable, prove that there exists an optimal solution x^* to the problem with at least $n - l$ zero coordinates.

Hint: Look at an optimal solution with as small as possible number of positive coordinates.

Exercise 1.19 [5] *Consider a TTD problem with M-dimensional space of virtual displacements. Prove that there exists an optimal solution to this problem with no more than $M + 1$ bars of positive volume.*

Hint: given an optimal truss t^* along with the associated displacement v^*, demonstrate that every solution t to the system

$$\begin{array}{rcl}\sum_{i=1}^n t_i(b_i^T v^*)b_i & = & f, \\ \sum_{i=1}^n t_i & = & w, \\ t & \geq & 0\end{array}$$

also is an optimal truss.

Lecture 2

From Linear Programming to Conic Programming

Linear Programming models cover numerous applications. Whenever applicable, LP allows to obtain a lot of useful quantitative and qualitative information on the corresponding application. The specific analytic structure of LP programs gives rise to a number of general results (e.g., those of the LP Duality Theory), and these results, as applied to particular models, in many cases provide us with valuable insight and understanding (cf., e.g., Exercise 1.19). The same analytic structure underlies specific computational techniques for LP; these perfectly well developed techniques allow to solve routinely quite large (tens/hundreds of thousands of variables and constraints) LP programs. At the same time, there are situations in reality which cannot be covered by LP models; to handle these "essentially nonlinear" cases, one needs to extend the basic theoretical results and computational techniques known for LP beyond the bounds of Linear Programming.

For the time being, the widest class of optimization problems to which the basic results of LP were extended, is the class of *convex* optimization programs. There are several equivalent ways to define a general convex optimization problem; the one we are about to use is not that traditional, but it serves best of all the purposes of our course.

When passing from a generic LP problem

$$c^T x \to \min \mid Ax \geq b \quad [A : m \times n] \tag{LP}$$

to its nonlinear extensions, we should make some components of the problem nonlinear. The traditional way here is to say: "Well, in (LP) there are linear objective function $f_0(x) = c^T x$ and inequality constraints $f_i(x) \geq b_i$ with linear functions $f_i(x) = a_i^T x$, $i = 1, ..., m$. What happens when we allow some/all of these functions $f_0, f_1, ..., m$ to be nonlinear?" For our purposes, however, it is more convenient to keep the constraint functions linear and to modify the interpretation of the inequality sign $\geq$.

2.1 Orderings of $\mathbf{R}^m$ and convex cones

The constraint inequality $Ax \geq b$ in (LP) is an inequality between *vectors*; as such, it requires a definition, and the definition is well-known: given two vectors $a, b \in \mathbf{R}^m$, we write $a \geq b$, if the coordinates of a majorate the corresponding coordinates of b:

$$a \geq b \Leftrightarrow \{a_i \geq b_i,\ i = 1, ..., m\}. \tag{"$\geq$"}$$

In the latter relation, we again meet with the inequality sign $\geq$, but now it stands for the "arithmetic $\geq$" – the relation $\geq$ between real numbers. The above "coordinate-wise" partial ordering of vectors from $\mathbf{R}^m$ satisfies a number of basic properties of the standard ordering of reals; namely, for all vectors $a, b, c, d, ... \in \mathbf{R}^m$ one has

1. *Reflexivity:* $a \geq a$;

2. *Anti-symmetry:* if both $a \geq b$ and $b \geq a$, then $a = b$;

3. *Transitivity:* if both $a \geq b$ and $b \geq c$, then $a \geq c$;

4. *Compatibility with linear operations:*

   (a) *Homogeneity:* if $a \geq b$ and λ is a nonnegative real, then $\lambda a \geq \lambda b$
   ("One can multiply both sides of an inequality by a nonnegative real")

   (b) *Additivity:* if both $a \geq b$ and $c \geq d$, then $a + c \geq b + d$
   ("One can add two inequalities of the same sign").

It turns out that

- *A significant part of nice features of LP programs comes from the fact that the vector inequality $\geq$ in the constraint of* (LP) *satisfies the properties* 1. – 4.*;*

- *The definition* (" $\geq$ ") *is neither the only possible, nor the only interesting way to define the notion of a vector inequality fitting the axioms* 1. – 4.

As a result,

> *A generic optimization problem which looks exactly the same as* (LP)*, up to the only fact that the inequality $\geq$ in* (LP) *is now understood in a way different from* (" $\geq$ ")*, inherits a significant part of nice properties of usual LP problems. Specifying properly the notion of a vector inequality, one can obtain from* (LP) *generic optimization problems which cover a lot of important applications which cannot be treated by the standard LP.*

To the moment what is said is nothing but a declaration. Let us look how this declaration is converted to reality.

We start with clarifying the "geometry" of a "vector inequality" satisfying the axioms 1. – 4. Thus, we consider vectors from $\mathbf{R}^m$ and assume that this set is equipped with a partial ordering, let it be denoted by $\succeq$: in other words, we say what are the pairs of vectors a, b from $\mathbf{R}^m$ linked by the inequality $a \succeq b$. When our ordering is "good" – fits the axioms 1. – 4.?

Our first observation is as follows:

> A. *A good inequality $\succeq$ is completely identified by the set* $\mathbf{K}$ *of $\succeq$-nonnegative vectors:*
> $$\mathbf{K} = \{a \in \mathbf{R}^n \mid a \succeq 0\}.$$
> *Namely,*
> $$a \succeq b \Leftrightarrow a - b \succeq 0 \quad [\Leftrightarrow a - b \in \mathbf{K}].$$
>
> Indeed, let $a \succeq b$. By 1 we have $-b \succeq -b$, and by 4.4b we may add the latter inequality to the former one to get $a - b \succeq 0$. Vice versa, if $a - b \succeq 0$, then, adding to this inequality the one $b \succeq b$, we get $a \succeq b$.

Thus, a good inequality $\succeq$ is completely specified by the set of $\succeq$-nonnegative vectors $\mathbf{K}$. This set, however, cannot be arbitrary:

B. *In order for a set $\mathbf{K} \subset \mathbf{R}^m$ to define, via the rule*

$$a \succeq b \Leftrightarrow a - b \in \mathbf{K}, \qquad (*)$$

a good partial ordering on $\mathbf{R}^m$, it is necessary and sufficient for $\mathbf{K}$ to be a pointed convex cone, i.e., to satisfy the following conditions:

1. $\mathbf{K}$ is nonempty and closed with respect to addition of its elements:

$$a, a' \in \mathbf{K} \Rightarrow a + a' \in \mathbf{K};$$

2. $\mathbf{K}$ is a conic set – along with any its point a, it contains the ray $\{\lambda a \mid \lambda \geq 0\}$ spanned by the point.
3. $\mathbf{K}$ is pointed – the only vector a such that both a and $-a$ belong to $\mathbf{K}$ is the zero vector.

   Geometrically: $\mathbf{K}$ does not contain straight lines passing through the origin.

We skip the proof of B – the reader definitely can prove this statement

Thus, every pointed convex cone $\mathbf{K}$ in $\mathbf{R}^m$ defines, via the rule (*), a partial ordering on $\mathbf{R}^m$ which satisfies the axioms 1. – 4.; let us denote this ordering by $\geq_{\mathbf{K}}$:

$$a \geq_{\mathbf{K}} b \Leftrightarrow a - b \geq_{\mathbf{K}} 0 \Leftrightarrow a - b \in \mathbf{K}.$$

What is the cone responsible for the standard coordinate-wise ordering $\geq$ we have started with? The answer is clear: this is the cone comprised of vectors with nonnegative components – the nonnegative orthant

$$\mathbf{R}^m_+ = \{x = (x_1, ..., x_m)^T \in \mathbf{R}^m : x_i \geq 0, \ i = 1, ..., m\}.$$

(Thus, in order to express the fact that a vector a is greater than or equal to, in the component-wise sense, to a vector b, we were supposed to write $a \geq_{\mathbf{R}^m_+} b$. We, however, are not going to be that tedious and will use the standard shorthand notation $a \geq b$.)

The nonnegative orthant $\mathbf{R}^m_+$ is not just a pointed convex cone; it possesses two useful additional properties:

I. The cone is closed: if a sequence of vectors a^i from the cone has a limit, the latter also belongs to the cone.

II. The cone possesses a nonempty interior: there exists a vector such that a ball of positive radius centered at this vector is contained in the cone.

These additional properties are very important. For example, the first of them is responsible for the possibility to pass to the term-wise limit in an inequality:

$$a^i \geq b^i \quad \forall i, a^i \rightarrow a, b^i \rightarrow b \text{ as } i \rightarrow \infty \Rightarrow a \geq b.$$

It makes sense to restrict ourselves with good partial orderings coming from cones $\mathbf{K}$ sharing the properties **I**, **II**. Thus,

From now on, speaking about good partial orderings $\geq_{\mathbf{K}}$, we always assume that the underlying set $\mathbf{K}$ is a pointed and <u>closed</u> convex cone <u>with a nonempty interior</u>.

Note that the closedness of $\mathbf{K}$ makes it possible to pass to limits in $\geq_{\mathbf{K}}$-inequalities:

$$a^i \geq_{\mathbf{K}} b^i, \ a^i \to a, b^i \to b \text{ as } i \to \infty \Rightarrow a \geq_{\mathbf{K}} b.$$

The nonemptiness of the interior of $\mathbf{K}$ allows to define, along with the "non-strict" inequality $a \geq_{\mathbf{K}} b$, also the strict inequality according to the rule

$$a >_{\mathbf{K}} b \Leftrightarrow a - b \in \operatorname{int} \mathbf{K},$$

where int K is the interior of the cone $\mathbf{K}$. E.g., the strict coordinate-wise inequality $a >_{\mathbf{R}^m_+} b$ (shorthand: $a > b$) simply says that the coordinates of a are strictly greater, in the usual arithmetic sense, than the corresponding coordinates of b.

Examples of partial orderings we are especially interested in are given by the following cones:

- The nonnegative orthant $\mathbf{R}^m_+$;
- The *Lorentz* (or the second-order; less scientific name: the ice-cream) cone

$$\mathbf{L}^m = \left\{ x = (x_1, ..., x_{m-1}, x_m)^T \in \mathbf{R}^m : x_m \geq \sqrt{\sum_{i=1}^{m-1} x_i^2} \right\}$$

- The *positive semidefinite cone* $\mathbf{S}^m_+$. This cone belongs to the space $\mathbf{S}^m$ of $m \times m$ symmetric matrices and is comprised of all $m \times m$ symmetric positive semidefinite matrices, i.e., of $m \times m$ matrices A such that

$$A = A^T, \quad x^T A x \geq 0 \quad \forall x \in \mathbf{R}^m.$$

2.2 "Conic programming" – what is it?

Let $\mathbf{K}$ be a cone in $\mathbf{R}^m$ (convex, pointed, closed and with a nonempty interior). Given an *objective* $c \in \mathbf{R}^n$, an $m \times n$ *constraint matrix* A and a *right hand side* $b \in \mathbf{R}^m$, consider the optimization program

$$c^T x \to \min \mid Ax - b \geq_{\mathbf{K}} 0 \qquad \text{(CP)}.$$

We shall refer to (CP) as to a *conic* problem associated with the cone $\mathbf{K}$. Note that the only difference between this program and an LP problem is that the latter deals with a particular choice of $\mathbf{K}$ – with the case when $\mathbf{K}$ is the nonnegative orthant $\mathbf{R}^m_+$. Replacing this particular cone with other cones, we get possibility to cover a lot of important applications which cannot be captured by LP. Just to get an idea of how wide is the spectrum of applications covered by conic problems coming from simple non-polyhedral (i.e., different from $\mathbf{R}^m_+$) cones, let us look at two examples as follows:

Example 2.2.1 [Synthesis of arrays of antennae, see Section 1.2.4] Given "building blocks" – antennae $S_1, ..., S_N$ with diagrams $Z_1(\delta), ..., Z_N(\delta)$, a target diagram $Z_*(\delta)$, and a finite grid T in the space of directions, find (complex-valued) amplification coefficients $z_i = u_i + \mathrm{i} v_i$ minimizing the quantity

$$\| Z_* - \sum_{j=1}^{N} z_j Z_j \|_{T,\infty} = \max_{\delta \in T} |Z_*(\delta) - \sum_{j=1}^{N} z_j Z_j(\delta)|.$$

We already dealt with a particular case of this problem – the one where Z_* and all Z_i were real-valued – in Lecture 1; in the latter case it was sufficient to deal with real design variables z_i, and the problem could be posed as an LP program. In the general case, some of all of the functions Z_*, Z_j are complex-valued; as a result, the conversion of the problem to an LP program fails. The problem of interest, however, can be immediately posed as (CP). Indeed, let $w = (u_1, v_1, ..., u_N, v_N)^T \in \mathbf{R}^{2N}$ be a collection of our design variables – the real and the imaginary parts of the complex-valued amplification coefficients z_N. For a particular direction δ, the complex number

$$Z_*(\delta) - \sum_{j=1}^{N} z_j Z_j(\delta)$$

treated as a two-dimensional real vector, is an *affine* function of x:

$$Z_*(\delta) - \sum_{j=1}^{N} z_j Z_j(\delta) = \alpha_\delta w + \beta_\delta \quad [\alpha_\delta \text{ is } 2 \times 2k \text{ matrix}, \beta_\delta \in \mathbf{R}^2]$$

Consequently, the problem of interest can be posed as

$$t \to \min \mid \| \alpha_\delta w + \beta_\delta \|_2 \leq t, \forall \delta \in T. \tag{An}$$

Now, a constraint

$$\| \alpha_\delta w + \beta_\delta \|_2 \leq t$$

says that the 3-dimensional vector

$$\begin{pmatrix} \alpha_\delta w + \beta_\delta \\ t \end{pmatrix}$$

affinely depending on the design vector $x = (w, t)$ of (An):

$$\begin{pmatrix} \alpha_\delta w + \beta_\delta \\ t \end{pmatrix} \equiv A_\delta x - b_\delta$$

must belong to the 3-dimensional ice-cream cone $\mathbf{L}^3$. Consequently, (An) is nothing but the problem

$$c^T x \equiv t \to \min \mid A_\delta x + b_\delta \in \mathbf{L}^3, \quad \forall \delta \in T \quad [x = (w, t)];$$

introducing the cone

$$\mathbf{K} = \prod_{\delta \in T} \mathbf{L}^3$$

along with the affine mapping

$$Ax - b = \{A_\delta x + b_\delta\}_{\delta \in T},$$

we can write down our problem as the conic problem

$$c^T x \to \min \mid Ax - b \geq_{\mathbf{K}} 0.$$

What we end up with is a *conic quadratic* program – a conic program associated with a cone $\mathbf{K}$ which is a direct product of (finitely many) ice-cream cones.

Of course, the same reduction to a conic quadratic problem can be used for the synthesis of filters in the frequency domain.

Example 2.2.2 [Stability analysis for an uncertain linear time-dependent dynamic system] Consider a linear time-varying system

$$\frac{d}{dt}v(t) = Q(t)v(t) \tag{S}$$

with $m \times m$ time-varying matrix $Q(t)$ and assume that all we know about the matrix $Q(t)$ of the system is that the matrix, at every time instant t, belongs to a given polytope:

$$Q(t) \in \mathrm{Conv}(Q_1, ..., Q_k).$$

Imagine, e.g., that all but one entries of Q are constant, while say, the entry Q_{11} varies somehow within known bounds $Q_{11}^{\min}$ and $Q_{11}^{\max}$. This is exactly the same as to say that $Q(t)$, for every t, belongs to the polytope spanned by two matrices Q_1 and Q_2, the (1,1)-entry of the matrices being $Q_{11}^{\min}$ and $Q_{11}^{\max}$, respectively, and the remaining entries being the same as in $Q(\cdot)$.

When speaking about system (S), the question of primary interest is whether the system is stable, i.e., whether all trajectories of the system tend to 0 as $t \to \infty$. A simple *sufficient* condition for stability is the existence of a *quadratic Lyapunov function* – a function

$$L(v) = v^T X v,$$

X being a symmetric positive definite matrix, such that

$$\frac{d}{dt}L(v(t)) \le -\alpha L(v(t)) \tag{Ly}$$

for every trajectory of the system; here $\alpha > 0$ is the "decay rate". (Ly) clearly implies that

$$L(v(t)) \equiv v^T(t)Xv(t) \le \exp\{-\alpha t\}L(v(0)),$$

and since X is positive definite, the latter inequality, in turn, says that $v(t) \to 0, t \to \infty$. Thus, whenever (Ly) can be satisfied by a pair (X, α) with positive definite X and positive α, the pair can be treated as a "stability certificate" for (S).

Now, the left hand side in (Ly) can be easily computed: it is nothing but

$$v^T(t)[Q^T(t)X + XQ(t)]v(t).$$

Consequently, (Ly) requires that

$$v^T(t)[Q^T(t)X + XQ(t)]v(t) \le -\alpha v^T(t)Xv(t) \Leftrightarrow -v^T(t)[Q^T(t)X + XQ(t) + \alpha X]v(t) \ge 0$$

For t given, the matrix $Q(t)$ may be an arbitrary matrix from the polytope $\mathrm{Conv}(Q_1, ..., Q_k)$, while $v(t)$ may be an arbitrary vector. Thus, (Ly) requires the the matrix $[-Q^T X - XQ - \alpha X]$ to be positive semidefinite whenever $Q \in \mathrm{Conv}(Q_1, ..., Q_k)$, or, which is the same (why?), requires the validity of the inequalities

$$-Q_i^T X - XQ_i - \alpha X \ge_{\mathbf{S}^m_+} 0, \; i = 1, ..., k.$$

Now, if a positive definite matrix X can be extended, by a positive α, to a pair (X, α) satisfying the indicated inequalities, if and only if the matrices

$$-Q_i^T X - XQ_i, \; i = 1, ..., k,$$

are positive definite (why?). We see that

In order to certify the stability of (S) *by a quadratic Lyapunov function, it suffices to find a symmetric matrix X satisfying the following system of* <u>*strict*</u> $\mathbf{S}^m_+$*-inequalities:*

$$X >_{\mathbf{S}^m_+} 0; \quad -Q_i^T X - XQ_i >_{\mathbf{S}^m_+} 0. \tag{2.2.1}$$

Now, a symmetric matrix A is positive definite if and only if the matrix $A - \tau I$, I being the unit matrix of the same size as A, is positive semidefinite for some positive τ (see Assignment to the lecture). Consequently, to verify whether (2.2.1) is solvable, is the same as to verify whether the optimal value in the program

$$t \to \min \mid \begin{pmatrix} X + tI & & & \\ & -Q_1^T X - XQ_1 + tI & & \\ & & \ddots & \\ & & & -Q_k^T X - XQ_k + tI \end{pmatrix} \geq_{\mathbf{S}_+^{m(k+1)}} 0, \tag{2.2.2}$$

the design variables being t and the $\frac{m(m+1)}{2}$ "free" entries of the symmetric matrix X, is or is not negative. If the optimal value in the problem is negative, then (2.2.1) is solvable, and one can use as a solution to (2.2.1) the X-component of an arbitrary feasible solution (X, t) to (2.2.2) with negative t. Whenever this is the case, (S) is stable, and the stability can be certified by a quadratic Lyapunov function. On the other hand, if the optimal value in (2.2.2) is nonnegative, then (2.2.1) is infeasible. Whether in the latter case (S) is or is not stable, this remains unclear; all can be said is that the stability, if it is present, cannot be certified by a *quadratic* Lyapunov function.

It remains to note that (2.2.2) is a conic problem associated with the positive semidefinite cone $\mathbf{S}^{m(k+1)}$. Indeed, the left hand side in the constraint inequality in (2.2.2) affinely depends on the design variables, exactly as required in the definition of a conic program.

2.3 Conic Duality

Aside of algorithmic issues, the most important theoretical result in Linear Programming is the LP Duality Theorem. The questions we are about to answer now is: whether this theorem can be extended to conic problems? What is the extension?

The source of the LP Duality Theorem was the desire to get a systematic way to bound from below the optimal value in an LP program

$$c^T x \to \min \mid Ax \geq b \tag{LP}$$

and the way was as follows: we were looking at the inequalities of the type

$$\lambda^T Ax \geq \lambda^T b \tag{Cons(λ)}$$

coming from nonnegative weight vectors λ; by its origin, an inequality of this such a type is a consequence of the system of constraints $Ax \geq b$ of the problem, i.e., it is satisfied at every solution to the system. Consequently, whenever we are lucky to get, as the left hand side of (Cons(λ)), the expression $c^T x$, i.e., whenever a nonnegative weight vector λ satisfies the relation

$$A^T \lambda = c,$$

the inequality (Cons(λ)) yields a lower bound $b^T\lambda$ on the optimal value in (LP). And the dual problem

$$\max b^T\lambda \mid \lambda \geq 0, A^T\lambda = c$$

was nothing but the problem of finding the best lower bound one can get in this fashion.

The same scheme can be used to develop the dual to a conic problem

$$c^T x \to \min \mid Ax \geq_{\mathbf{K}} b. \tag{CP}$$

The only step of the construction to be understood is the following one:

> (?) *What are the "admissible" weight vectors λ, i.e., the vectors such that the scalar inequality*
>
> $$\lambda^T Ax \geq \lambda^T b$$
>
> *indeed is a consequence of the vector inequality $A^T x \geq_{\mathbf{K}} b$?*

In the particular case of coordinate-wise partial ordering, i.e., in the case of $\mathbf{K} = \mathbf{R}^m_+$, the admissible vectors were those with nonnegative coordinates. Those vectors, however, not necessarily are admissible for orderings $\geq_{\mathbf{K}}$ given by cones $\mathbf{K}$ different from the nonnegative orthant:

Example 2.3.1 *Consider the ordering $\geq_{\mathbf{L}^3}$ on $\mathbf{R}^3$ given by the 3-dimensional ice-cream cone:*

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \geq_{\mathbf{L}^3} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \Leftrightarrow a_3 \geq \sqrt{a_1^2 + a_2^2}.$$

We, e.g., have

$$\begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix} \geq_{\mathbf{L}^3} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

However, aggregating this inequality with the aid of a positive weight vector $\lambda = \begin{pmatrix} 1 \\ 1 \\ 0.1 \end{pmatrix}$, we get a <u>false</u> inequality

$$-1.8 \geq 0.$$

Thus, not every nonnegative weight vector is admissible for the partial ordering $\geq_{\mathbf{L}^3}$.

To answer the question (?) is the same as to say what, for a given cone $\mathbf{K}$, are the weight vectors λ such that

$$\forall a \geq_{\mathbf{K}} 0: \quad \lambda^T a \geq 0. \tag{2.3.1}$$

Whenever λ possesses the property (2.2.2), the scalar inequality

$$\lambda^T a \geq \lambda^T b$$

is a consequence of the vector inequality $a \geq_{\mathbf{K}} b$:

$$\begin{array}{rrcll} & a & \geq_{\mathbf{K}} & b & \\ \Leftrightarrow & a-b & \geq_{\mathbf{K}} & 0 & [\text{additivity of } \geq_{\mathbf{K}}] \\ \Rightarrow & \lambda^T(a-b) & \geq & 0 & [\text{by } (2.3.1)] \\ \Leftrightarrow & \lambda^T a & \geq & \lambda^T b. & \blacksquare \end{array}$$

Vice versa, if λ is an admissible weight vector for the partial ordering $\geq_{\mathbf{K}}$:

$$\forall (a, b : a \geq_{\mathbf{K}} b) : \quad \lambda^T a \geq \lambda^T b$$

then, of course, λ satisfies (2.3.1).

Thus, the admissible for a partial ordering $\geq_{\mathbf{K}}$ weight vectors λ are exactly the vectors satisfying (2.3.1), or, which is the same, the vectors from the set

$$\mathbf{K}_* = \{\lambda \in \mathbf{R}^m : \lambda^T a \geq 0 \quad \forall a \in \mathbf{K}\}$$

(recall that this is the same – to say that $a \geq_{\mathbf{K}} 0$ and to say that $a \in \mathbf{K}$). The set $\mathbf{K}_*$ is comprised of the vectors with nonnegative inner products with all vectors from $\mathbf{K}$ and called the *cone dual to* $\mathbf{K}$. The name is correct due to the following fact:

Theorem 2.3.1 [Properties of the dual cone] *Let $K \subset \mathbf{R}^m$ be a nonempty set. Then*

(i) *The set*

$$K_* = \{\lambda \in \mathbf{R}^m : \lambda^T a \geq 0 \quad \forall a \in K\}$$

is a closed convex cone.

(ii) *If K possesses a nonempty interior, then K_* is pointed.*

(iii) *If K is a closed convex cone, then so is K_*, and the cone dual to K_* is exactly K:*

$$(K_*)_* = K.$$

(iv) *If K is a closed convex pointed cone, then K_* has a nonempty interior.*

(v) *If K is a closed convex pointed cone with a nonempty interior, then so is K_*.*

The proof of the theorem is one of the subjects of the Assignment to the Lecture.

From the dual cone to the problem dual to (CP). Now we are equipped to derive the problem dual to a conic problem (CP). Same as in the case of Linear Programming, we start from the observation that whenever x is a feasible solution to (CP) and λ is an admissible weight vector, i.e., $\lambda \in \mathbf{K}_*$, x satisfies the scalar inequality

$$\lambda^T A x \geq \lambda^T b$$

– this observation is an immediate consequence of the origin of $\mathbf{K}_*$. It follows that whenever λ_* is an admissible weight vector satisfying the relation

$$A^T \lambda = c,$$

one has

$$c^T x = (A^T \lambda)^T x = \lambda^T A x \geq \lambda^T b = b^T \lambda$$

for all x feasible for (CP), so that the quantity $b^T \lambda$ is a lower bound on the optimal value in (CP). The best bound one can get in such a manner is the optimal value in the problem

$$b^T \lambda \to \max \mid A^T \lambda = c, \lambda \geq_{\mathbf{K}_*} 0 \quad [\Leftrightarrow \lambda \in \mathbf{K}_*] \tag{D}$$

and this program, *by definition*, is the program dual to (CP).

All we know about the duality we have just introduced is the following

Proposition 2.3.1 [Weak Duality Theorem] *The optimal value in* (D) *is a lower bound on the optimal value in* (CP).

Indeed, this statement is a direct consequence of the construction which led us to (D).

2.3.1 Geometry of the primal and the dual problems

Problem (D) looks as a problem of a structure essentially different from the one of (CP); more careful analysis, however, demonstrates that the difference in structures comes just from the way we represent the data: geometrically, the problems are completely similar. Indeed, in (D) we are asked to maximize a linear objective $b^T\lambda$ over the intersection of an affine plane $L_* = A^T\lambda = c$ with the cone $\mathbf{K}_*$. And what is (CP)? Let us pass in this latter problem from the "true design variables" x to their images $y = Ax - b$. When x runs through $\mathbf{R}^n$, y runs through affine plane $L = \{y = Ax - b \mid x \in \mathbf{R}^n\}$. x is feasible if the corresponding $y = Ax - b$ belongs to the cone $\mathbf{K}$. Thus, in (CP) we also deal with an intersection of an affine plane L with a cone – namely, $\mathbf{K}$. Now assume that our objective c^Tx can be expressed via $y = Ax - b$:

$$c^Tx = d^T(Ax - b) + \text{const}.$$

This assumption clearly is equivalent to the possibility to represent c as A^Td:

$$c \in \operatorname{Im} A^T \Rightarrow \exists d: \quad c^Tx = d^T(Ax - b) + d^Tb \quad \forall x. \tag{2.3.2}$$

Under the premise of (2.3.2) the primal problem (CP) can be posed equivalently in terms of y, namely, as the problem

$$\begin{array}{c} d^Ty \to \min \mid y \in L, \ y \geq_{\mathbf{K}} 0. \\ {[L = \{y = Ax - b \mid x \in \mathbf{R}^n\}]} \end{array}$$

Thus,

> *under the premise in (2.3.2) the primal problem is, geometrically, to minimize a linear form over the intersection of an affine plane L with the cone $\mathbf{K}$, and the dual problem is to maximize another linear form over the intersection of affine plane L_* with the dual cone $\mathbf{K}_*$.*

We see that under the premise of (2.3.2) (CP) and (D) are, geometrically, completely similar to each other.

Now, what happens if the premise in (2.3.2) is not satisfied? The answer is very simple: in this case (CP) makes no sense at all – it is either below unbounded, or infeasible.

> Indeed, from Linear Algebra it is known that the system
>
> $$A^Td = c$$
>
> with unknown d is unsolvable if and only if c is not orthogonal to the kernel of A, in other words, if and only if there exist e such that $Ae = 0$ and $c^Te > 0$. If it is the case and (CP) is feasible, then (CP) is below unbounded – subtracting from a feasible solution x a large multiple of e, we do not violate feasibility (since $Ae = 0$) and may make the value of the objective as negative as we wish.

Thus, if (2.3.2) is not satisfied, we may reject (CP) from the very beginning. In view of this observation, speaking about conic problem (CP), it makes sense to assume that the premise in (2.3.2) is satisfied. In fact from now one we make a bit stronger assumption

A. *The matrix A is of full column rank, i.e., its columns are linearly independent.*

In other words, we assume that the mapping $x \mapsto Ax$ is with the trivial kernel ("we have eliminated from the very beginning the redundant design degrees of freedom – those not affecting the value of Ax"). Under this assumption, the equation

$$A^T d = c$$

is solvable for every possible objective c, not only for the one we actually are interested in.

Under assumption **A** (CP) and (D) share the same geometry: both problems are to optimize a linear objective over the intersection of an affine plane and a cone.

We have seen that a conic problem (CP) can be reformulated as a problem (P) of minimizing a linear objective $d^T y$ over the intersection of an affine plane L and a cone $\mathbf{K}$. And of course a problem (P) of this latter type can be posed in the form of (CP) – to this end it suffices to represent the plane L as the image of an affine mapping $x \mapsto Ax - b$ (i.e., to parameterize somehow the feasible plane) and to "translate" the objective $d^T y$ to the space of x-variables – to set $c = A^T d$, which yields

$$y = Ax + l \Rightarrow d^T y = c^T x + \text{const}.$$

When speaking about a conic problem, we may pass from its "analytic form" (CP) to the "geometric form" (P) and vice versa.

Now, what are the relations between the "geometric data" of the primal and the dual problems? We already know that the cone associated with the dual problem is dual to the cone associated with the primal one. What about the feasible planes L and L_*? The answer is simple: they are orthogonal to each other! More exactly, the affine plane L is the translation, by vector $-b$, of the linear subspace

$$\mathcal{L} = \{y = Ax \mid x \in \mathbf{R}^n\}.$$

And L_* is the translation, by any solution λ_0 of the system $A^T\lambda = c$, *e.g., by the solution d to the system*, of the linear subspace

$$\mathcal{L}_* = \{\lambda \mid A^T\lambda = 0\}.$$

A well-known fact of Linear Algebra says that the linear subspaces $\mathcal{L}$ and $\mathcal{L}_*$ are orthogonal complements of each other:

$$\mathcal{L} = \{y \mid y^T\lambda = 0 \quad \forall \lambda \in \mathcal{L}_*\}; \mathcal{L}_* = \{\lambda \mid y^T\lambda = 0 \quad \forall y \in \mathcal{L}\}.$$

Thus, we come to a nice geometrical conclusion:

A conic problem[1] (CP) *is the problem of minimizing a linear objective $c^T x$ over the intersection of a cone $\mathbf{K}$ with an affine plane – the translation $L = \mathcal{L} - b$ of a linear subspace $\mathcal{L}$ by a vector $(-b)$:*

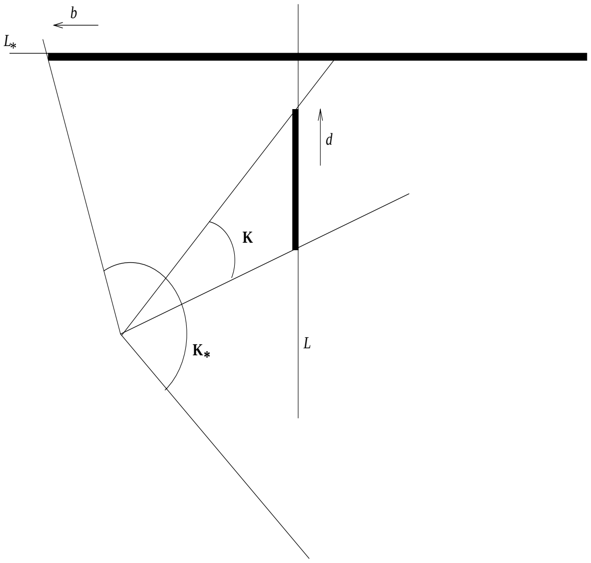
$$d^T y \to \min \mid y \in \mathcal{L} - b, \ y \geq_{\mathbf{K}} 0 \tag{P}$$

The dual problem is to maximize the linear objective $b^T\lambda$ over the intersection of the dual cone $\mathbf{K}_*$ with an affine plane – the translation $L_* = \mathcal{L}_* + d$ of the orthogonal complement of $\mathcal{L}$ by the primal objective d:

$$b^T\lambda \to \max \mid \lambda \in \mathcal{L}_* + d, \ \lambda \geq_{\mathbf{K}_*} 0. \tag{D}$$

[1] recall that we have restricted ourselves to the problems satisfying the assumption **A**

What we get is an extremely transparent geometric description of *the primal-dual pair* of conic problems (P), (D). Note that the duality is completely symmetric: the problem dual to (D) is (P)! Indeed, as we know from Theorem 2.3.1, the cone dual to $\mathbf{K}_*$ is exactly $\mathbf{K}$, while the orthogonal complement to $\mathcal{L}_*$ – to the orthogonal complement of $\mathcal{L}$ – is $\mathcal{L}$ itself. Switch from maximization to minimization corresponds to the fact that the "shifting vector" in (P) is $(-b)$, while the "shifting vector" in (D) is d. The geometry of the primal-dual pair (P), (D) is illustrated on the below picture:



Primal-dual pair of conic problems
[bold: primal (vertical segment) and dual (horyzontal ray) feasible sets]

Finally, note that in the case when (CP) is (LP) (i.e., in the case when $\mathbf{K}$ is the nonnegative orthant) the "conic dual" problem (D) is exactly the usual LP dual; this fact immediately follows from the observation that the cone dual to a nonnegative orthant is the orthant itself.

We have understood the geometry of a primal-dual pair of conic problems: the "geometric data" of such a pair are given by a pair of dual to each other cones $\mathbf{K}, \mathbf{K}_*$ in $\mathbf{R}^n$ and a pair of affine planes $L = \mathcal{L} - b$, $L_* = \mathcal{L}^\perp + d$, where $\mathcal{L}$ is a linear subspace in $\mathbf{R}^n$ and $\mathcal{L}^\perp$ is its orthogonal complement. The first problem – let it be called (P) – from the pair is to minimize $b^T y$ over $y \in \mathbf{K} \cap L$, and the second (D) is to maximize $d^T \lambda$ over $\lambda \in \mathbf{K}_* \cap L_*$. Note that the "geometric data" $(\mathbf{K}, \mathbf{K}_*, L, L_*)$ of the pair do not completely specify the problems of the pair: given L, L_*, we can uniquely define $\mathcal{L}$, but not the shift vectors $(-b)$ and d: b is known up to shift by a vector from $\mathcal{L}$, and d is known up to shift by a vector from $\mathcal{L}^\perp$. Note, however, that the indicated non-uniqueness is of absolutely no importance: replacing a somehow chosen vector $d \in L_*$ by another vector $d' \in L_*$, we pass from (P) to a new problem (P′) which is completely equivalent to (P): indeed, both (P) and (P′) have the same feasible set, and *at the (common) feasible plane L of the problems their objectives $d^T y$ and $(d')^T y$ differ from each other by a constant:*

$$y \in L = \mathcal{L} - b, d - d' \in \mathcal{L}^\perp \Rightarrow (d - d')^T (y + b) = 0 \Rightarrow (d - d')^T y = -(d - d')^T b \quad \forall y \in L.$$

Similarly, shifting b along $\mathcal{L}$, we do modify the objective in (D), but in a trivial way – *on the feasible plane L_** of the problem the new objective differs from the old one by a constant.

2.4 The Conic Duality Theorem

All we know to the moment on the conic duality is the Weak Duality Theorem 2.3.1, which is much weaker than the Linear Programming Duality Theorem. Is it possible to get results similar to those of the LP Duality Theorem in the general conic case as well? The answer is affirmative, *provided that the primal problem* (CP) *is strictly feasible*, i.e., that there exists x such that $Ax - b >_{\mathbf{K}} 0$. Geometrically:

> A conic problem (P) is called strictly feasible, if its feasible plane L intersects the interior of the corresponding cone $\mathbf{K}$.

The advantage of the geometrical definition of strict feasibility is that it is independent of the particular way in which the feasible plane is defined; with this definition, it is clear, e.g., what does it mean that the dual problem (D) is strictly feasible.

Our main result is the following

Theorem 2.4.1 [Conic Duality Theorem] *Consider a conic problem*

$$c^T x \to \min \mid Ax \geq_{\mathbf{K}} b \tag{CP}$$

along with its conic dual

$$b^T\lambda \to \max \mid A^T\lambda = c, \lambda \geq_{\mathbf{K}_*} 0. \tag{D}$$

1) *The duality is symmetric: the dual problem is conic, and the problem dual to dual is the primal.*

2) *The value of the dual objective at every dual feasible solution is $\leq$ the value of the primal objective at every primal feasible solution, so that the duality gap*

$$c^T x - b^T\lambda$$

is nonnegative at every "primal-dual feasible pair" (x, λ) – a pair comprised of primal feasible solution x and dual feasible solution λ.

3.a) *If the primal* (CP) *is below bounded and strictly feasible (i.e. $Ax >_{\mathbf{K}} b$ for some x), then the dual* (D) *is solvable and the optimal values in the problems are equal to each other*

3.b) *If the dual* (D) *is above bounded and strictly feasible (i.e., exists $\lambda >_{\mathbf{K}_*} 0$ such that $A^T\lambda = c$), then the primal* (CP) *is solvable, and the optimal values in the problems are equal to each other.*

4) *Assume that at least one of the problems* (CP), (D) *is bounded and strictly feasible. Then a primal-dual feasible pair (x, λ) is comprised of optimal solutions to the respective problems*

4.a) if and only if

$$b^T\lambda = c^T x \qquad \text{[zero duality gap]}$$

and

4.b) if and only if

$$\lambda^T[Ax - b] = 0 \qquad \text{[complementary slackness]}$$

Proof. 1): The result was already obtained when discussing the geometry of the primal and the dual problems.

2): This is the Weak Duality Theorem.

3): Assume that (CP) is strictly feasible and below bounded, and let c^* be the optimal value in the problem. We should prove that the dual is solvable with the same optimal value. Since we already know that the optimal value in the dual is $\leq c^*$ (see 2)), all we need is to point out a dual feasible solution λ_* with $b^T\lambda_* \geq c^*$.

Consider the convex set

$$M = \{y = Ax - b \mid x \in \mathbf{R}^n, c^T x \leq c^*\}.$$

Let us start with the case of $c \neq 0$. We claim that in this case

(i) The set M is nonempty;

(ii) the plane M does not intersect the interior K of the cone $\mathbf{K}$.

(i) is evident (why?). To verify (ii), assume, on contrary, that there exists a point $\bar{x}$, $c^T\bar{x} \leq c^*$, such that the point $\bar{y} = A\bar{x} - b$ is $>_{\mathbf{K}} 0$. Then, of course, $Ax - b >_{\mathbf{K}} 0$ for all x close enough to $\bar{x}$, i.e., all points x from a small enough neighbourhood of $\bar{x}$ also are feasible for (CP). Since $c \neq 0$, there are points x in this neighbourhood with $c^T x < c^T\bar{x} \leq c^*$, which is impossible, since c^* is the optimal value in (CP).

Now let us make use of the following basic fact:

Theorem 2.4.2 [Separation Theorem for Convex Sets] *Let M, K be nonempty non-intersecting convex subsets of $\mathbf{R}^m$. Then M and K can be separated by a linear functional: there exists a nonzero vector λ such that*

$$\sup_{u\in M} \lambda^T u \leq \inf_{u\in K} \lambda^T u.$$

Applying the Separation Theorem to our M and K, we conclude that there exists $\lambda \in \mathbf{R}^m$ such that

$$\sup_{y\in M} \lambda^T y \leq \inf_{y\in \mathrm{int}\,\mathbf{K}} \lambda^T y. \tag{2.4.1}$$

From the inequality it follows that the linear form $\lambda^T y$ is below bounded on the interior K of the cone $\mathbf{K}$. Since this interior is a conic set:

$$y \in K, \mu > 0 \Rightarrow \mu y \in K$$

(why?), this boundedness implies that $\lambda^T y \geq 0$ for all $y \in K$. Consequently, $\lambda^T y \geq 0$ for all y from the closure of K, i.e., from all y from $\mathbf{K}$. We conclude that $\lambda \geq_{\mathbf{K}_*} 0$, so that the inf in (2.4.1) is nonnegative. On the other hand, the infimum of a linear form over a conic set clearly cannot be positive; we conclude that the inf in (2.4.1) is 0, so that the inequality reads

$$\sup_{u\in M} \lambda^T u \leq 0.$$

Recalling the definition of M, we get

$$[A^T\lambda]^T x \leq \lambda^T b \tag{2.4.2}$$

for all x from the half-space $c^T x \leq c^*$. But the linear form $[A^T\lambda]^T x$ can be above bounded on the half-space if and only if the vector $A^T\lambda$ is proportional, with a nonnegative coefficient, to the vector c:

$$A^T\lambda = \mu c$$

for some $\mu \geq 0$. We claim that $\mu > 0$. Indeed, assuming $\mu = 0$, we get $A^T\lambda = 0$, whence $\lambda^T b \geq 0$ in view of (2.4.2). It is time now to recall that (CP) is strictly feasible, i.e., $A\bar{x} - b >_{\mathbf{K}} 0$ for some $\bar{x}$. Since $\lambda \geq_{\mathbf{K}_*} 0$ and $\lambda \neq 0$, the product $\lambda^T[A\bar{x} - b]$ should be strictly positive (why?), while in fact we know that the product is $-\lambda^T b \leq 0$ (since $A^T\lambda = 0$ and, as we have seen, $\lambda^T b \geq 0$).

Thus, $\mu > 0$. Setting $\lambda_* = \mu^{-1}\lambda$, we get

$$\begin{array}{rll} \lambda_* & \geq_{\mathbf{K}_*} 0 & [\text{since } \lambda \geq_{\mathbf{K}_*} 0 \text{ and } \mu > 0] \\ A^T\lambda_* & = c & [\text{since } A^T\lambda = \mu c] \\ c^T x & \leq \lambda_*^T b \quad \forall x : c^T x \leq c^* & [\text{see (2.4.2)}] \end{array}$$

i.e., we see that λ_* is feasible for (D), the value of the dual objective at λ_* being at least c^*, as is required.

It remains to verify the case of $c = 0$. Here, of course, $c^* = 0$, and the existence of the dual feasible solution with the value of the objective $\geq c^* = 0$ is evident: the required solution is $\lambda = 0$. 3.a) is proved.

3.b): the result follows from 3.a) in view of the primal-dual symmetry.

4): Let x be primal feasible, and λ be dual feasible. Then

$$c^T x - b^T\lambda = (A^T\lambda)^T x - b^T\lambda = [Ax - b]^T\lambda.$$

We get a useful identity as follows:

> (!) *For primal-dual feasible pair (x, λ) the duality gap $c^T x - b^T\lambda$ always is equal to the inner product of the primal slack $y = Ax - b$ and λ.*
>
> Note that (!) in fact does not require "full" primal-dual feasibility: x may be arbitrary, and λ should belong to the dual feasible plane $A^T\lambda = c$, but not necessary to be $\geq_{\mathbf{K}_*} 0$.

In view of (!) it is absolutely the same – to say that the complementary slackness holds or to say that the duality gap is zero; thus, all we need is to prove 4.a).

Note that the "primal residual" $c^T x - c^*$ and the "dual residual" $b^* - b^T\lambda$ (b^* is the optimal value in the dual) always are nonnegative, provided that x is primal feasible, and λ is dual feasible. It follows that the duality gap

$$c^T x - b^T\lambda = [c^T x - c^*] + [b^* - b^T\lambda] + [c^* - b^*]$$

always is nonnegative (recall that $c^* \geq b^*$ by 2)), and it is zero if and only if $c^* = b^*$ and both primal and dual residuals are zero (i.e., x is primal optimal, and λ is dual optimal); all these considerations are valid independently of any assumptions of strict feasibility. We see that the condition "the duality gap at a primal-dual feasible pair is zero is always sufficient for primal and dual optimality of the components of the pair; and if $c^* = b^*$, this sufficient condition is also necessary. Since in the case of 4) we indeed have $c^* = b^*$ (it is stated by 3)), 4.a) follows. ■

The Conic Duality Theorem admits a useful

Corollary 2.4.1 *Assume that both* (CP) *and* (D) *are strictly feasible. Then both problems are solvable, the optimal values are equal to each other, and each one of the conditions 4.a), 4.b) is necessary and sufficient for optimality of a primal-dual feasible pair.*

> Indeed, by the Weak Duality Theorem, if one of the problems is feasible, the other is bounded, and it remains to use the items 3) and 4) of the Conic Duality Theorem.

2.4.1 Is something wrong with conic duality?

The statement of the Conic Duality Theorem is weaker than the one of the LP Duality theorem: in the LP case, feasibility (even non-strict) and boundedness of one of the problems of a primal-dual pair implies solvability of both the primal and the dual, equality between optimal values of the problems and all other nice things. In the general conic case something "nontrivial" is stated only in the case of strict feasibility (and boundedness) of one of the problems. It can be demonstrated by examples that this phenomenon reflects the nature of the situation, and not our ability to analyze it. The case of non-polyhedral cone **K** indeed is more complicated than the one of the nonnegative orthant **K**; as a result, "word-by-word" extension of the LP Duality Theorem to the conic case fails to be true.

Example 2.4.1 Consider the following conic problem with 2 variables $x = (x_1, x_2)^T$ and **K** being the 3-dimensional ice-cream cone:

$$x_1 \to \min \mid Ax - b \equiv \begin{bmatrix} x_1 - x_2 \\ 1 \\ x_1 + x_2 \end{bmatrix} \geq_{\mathbf{L}^3} 0$$

Recalling the definition of $\mathbf{L}^3$, we can write the problem down equivalently as

$$x_1 \to \min \mid \sqrt{(x_1 - x_2)^2 + 1} \leq x_1 + x_2,$$

i.e., as the problem

$$x_1 \to \min \mid 4x_1 x_2 \geq 1, x_1 + x_2 > 0.$$

Geometrically we are interested to minimize x_1 over the intersection of the 3D ice-cream cone with a 2D plane; the projection of this intersection onto the (x_1, x_2)-plane is part of the 2D nonnegative orthant bounded by the hyperbola $x_1 x_2 \geq 1/4$. The problem clearly is strictly feasible (a strictly feasible solution is, e.g., $x = (1,1)^T$) and below bounded, the optimal value being equal to 0. This optimal value, however, is not achieved – the problem is unsolvable!

Example 2.4.2 Consider the following conic problem with two variables $x = (x_1, x_2)^T$ and **K** being the 3-dimensional ice-cream cone:

$$x_2 \to \min \mid Ax - b = \begin{bmatrix} x_1 \\ x_2 \\ x_1 \end{bmatrix} \geq_{\mathbf{L}^3} 0.$$

Recalling the definition of $\mathbf{L}^3$, we can write down the problem equivalently as

$$x_2 \to \min \mid \sqrt{x_1^2 + x_2^2} \leq x_1,$$

i.e., as the problem

$$x_2 \to \min \mid x_2 = 0, x_1 \geq 0.$$

The problem clearly is solvable, and its optimal set is the ray $\{x_1 \geq 0, x_2 = 0\}$ on the design plane $\mathbf{R}^2$. Geometrically our problem is to minimize the linear objective on the intersection of the 3D ice-cream cone with its tangent plane passing through the ray $\{x_3 = x_1 \geq 0, x_2 = 0\}$, and the objective is constant on the ray.

Now let us build the conic dual to our (solvable!) primal. To this end it is worthy to note that the cone dual to an ice-cream cone is this ice-cream cone itself (see exercises to the lecture). Thus, the dual problem is

$$b^T\lambda \equiv 0 \to \max \mid A^T\lambda \equiv \begin{bmatrix} \lambda_1+\lambda_3 \\ \lambda_2 \end{bmatrix} = c \equiv \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \lambda \geq_{\mathbf{L}^3} 0.$$

In spite of the fact that primal is solvable, the dual is infeasible: indeed, from $\lambda \geq_{\mathbf{L}^3} 0$ it follows that $\lambda_3 \geq \sqrt{\lambda_1^2+\lambda_2^2}$; consequently, for such a λ one can have $\lambda_1+\lambda_3=0$, as required by the first equality in the vector equation $A\lambda = c$, only if $\lambda_2 = 0$; but then the second equality in our vector equation is wrong...

We see that the weakness of the Conic Duality Theorem as compared to the LP Duality one reflects pathologies which indeed may happen in the general conic case.

2.4.2 Consequences of the Conic Duality Theorem

Sufficient condition for infeasibility. As we remember, the necessary and sufficient condition for infeasibility of a (finite) system of scalar linear inequalities (i.e., for a vector inequality with respect to the partial ordering $\geq$) is the possibility to combine these inequalities in linear fashion in such a way that the resulting scalar linear inequality is contradictory. In the case of cone-generated vector inequalities a slightly weaker result can be obtained:

Proposition 2.4.1 *Consider a linear vector inequality*

$$Ax - b \geq_{\mathbf{K}} 0. \tag{I}$$

(i) *If there exists λ satisfying*

$$\lambda \geq_{\mathbf{K}_*} 0, A^T\lambda = 0, \lambda^T b > 0, \tag{II}$$

then (I) *has no solutions.*

(ii) *If there does not exist λ satisfying* (II), *then* (I) *is "almost solvable" – for every positive ϵ there exists b' such that $\| b' - b \|_2 < \epsilon$ and the perturbed system*

$$Ax - b' \geq_{\mathbf{K}} 0$$

is solvable.

Moreover,

(iii) (II) *is solvable* *if and only if* (I) *is* *not* *"almost solvable".*

Note the difference between the simple case when $\geq_{\mathbf{K}}$ is the usual partial ordering $\geq$ and the general case: in the former, one can replace in (ii) "nearly solvable" by "solvable"!

In the general conic case, however, "almost" is unavoidable.

Example 2.4.3 Consider the following linear vector inequality with one variable and the partial order given by the 3D ice-cream cone:

$$Ax - b \equiv \begin{bmatrix} x+1 \\ x-1 \\ \sqrt{2}x \end{bmatrix} \geq_{\mathbf{L}^3} 0.$$

Recalling the definition of the ice-cream cone, we can write the inequality equivalently as

$$\sqrt{2}x \geq \sqrt{(x+1)^2+(x-1)^2} \equiv \sqrt{2x^2+2}, \tag{i}$$

which of course in unsolvable. The corresponding system (II) is

$$\begin{array}{cc} \lambda_3 \geq \sqrt{\lambda_1^2+\lambda_2^2} & \left[\Leftrightarrow \lambda \geq_{\mathbf{L}^3_*} 0\right] \\ \lambda_1+\lambda_2+\sqrt{2}\lambda_3 = 0 & \left[\Leftrightarrow A^T\lambda = 0\right] \\ \lambda_2-\lambda_1 > 0 & \left[\Leftrightarrow b^T\lambda > 0\right] \end{array} \tag{ii}$$

By the Cauchy inequality, from the first relation in (ii) it follows that $-\lambda_1-\lambda_2 \leq \sqrt{2}\lambda_3$, the inequality being equality if and only if $\lambda_1=\lambda_2=-\lambda_3/\sqrt{2}$. The second equation in (ii) therefore implies that $\lambda_1=\lambda_2$. But then the third inequality in (ii) is impossible! We see that here both (i) and (ii) have no solutions.

The geometry of the outlined example is as follows. (i) asks to find a point in the intersection of the 3D ice-cream cone and a line. This line is an asymptote of the cone (it belongs to a 2D plane which crosses the cone in such way that the boundary of a the cross-section is a branch of a hyperbola, and the line is one of two asymptotes of the hyperbola). Although the intersection is empty ((i) is unsolvable), small shifts of the line make the intersection nonempty (i.e., (i) is unsolvable and "almost solvable" at the same time). And it turns out that one cannot certify the fact that (i) itself is unsolvable by providing a solution to (ii).

Proof of the Proposition. (i) is evident (why?).

Let us prove (ii). To this end it suffices to verify that if (I) is not "almost solvable", then (II) is solvable. Let us fix a vector $\sigma >_{\mathbf{K}} 0$ and look at the conic problem

$$t \to \min \mid Ax + t\sigma - b \geq_{\mathbf{K}} 0, \tag{CP}$$

the design variables being (x,t). This problem clearly is strictly feasible (why?). Now, if (I) is not almost solvable, then, first, the matrix of the problem $[A;\sigma]$ satisfies $\mathbf{A}$ (otherwise the image of the mapping $(x,t) \mapsto Ax+t\sigma-b$ would coincide with the image of the mapping $x \mapsto Ax-b$, which is not he case – the first of these images does intersect $\mathbf{K}$, while the second does not). Second, the optimal value in (CP) is strictly positive (otherwise the problem would admit feasible solutions with t close to 0, and this would mean that (I) is almost solvable). From the Conic Duality Theorem it follows that the problem

$$b^T\lambda \to \max \mid A^T\lambda = 0, \sigma^T\lambda = 1, \lambda \geq_{\mathbf{K}_*} 0$$

has a feasible solution with positive $b^T\lambda$, i.e., (II) is solvable.

It remains to prove (iii). Assume first that (I) is not almost solvable; then (II) must be solvable by (ii). Vice versa, assume that (II) is solvable, and let λ be a solution to (II). Then λ solves also all systems of the type (II) associated with small enough perturbations of b instead of b itself; by (i), it implies that all inequalities obtained from (I) by small enough perturbation of b are unsolvable. ■

When a scalar linear inequality is a consequence of a given linear vector inequality? The question we are interested in is as follows: given a linear vector inequality

$$Ax \geq_{\mathbf{K}} b \tag{V}$$

and a scalar inequality

$$c^Tx \geq d \tag{S}$$

we are interested to check whether (S) is a consequence of (V). If (V) were the usual inequality $\geq$, the answer would be given by the Farkas Lemma:

Inequality (S) *is a consequence of a* feasible *system of linear inequalities* $Ax \geq b$ *if and only if* (S) *can be obtained from the system and the trivial inequality* $1 \geq 0$ *in a linear fashion (by taking weighted sum with nonnegative weights).*

In the general conic case we can get a slightly weaker result:

Proposition 2.4.2 (i) *If* (S) *can be obtained from* (V) *and from the trivial inequality* $1 \geq 0$ *by admissible aggregation, i.e., there exist weight vector* $\lambda \geq_{\mathbf{K}_*} 0$ *such that*

$$A^T\lambda = c, \lambda^T b \geq d,$$

then (S) *is a consequence of* (V).

(ii) *If* (S) *is a consequence of a* strictly feasible *linear vector inequality* (V), *then* (S) *can be obtained from* (V) *by an admissible aggregation.*

The difference between the case of the partial ordering $\geq$ and a general partial ordering $\geq_{\mathbf{K}}$ is in the word "strictly" in (ii).

Proof of the proposition. (i) is evident (why?). To prove (ii), assume that (V) is strictly feasible and (S) is a consequence of (V) and consider the conic problem

$$\begin{array}{c} t \to \min \mid \bar{A}\begin{pmatrix} x \\ t \end{pmatrix} - \bar{b} \equiv \begin{bmatrix} Ax - b \\ d - c^T x + t \end{bmatrix} \geq_{\bar{\mathbf{K}}} 0, \\ \bar{\mathbf{K}} = \{(x,t) \mid x \in \mathbf{K}, t \geq 0\} \end{array}$$

The problem clearly is strictly feasible (choose x to be a strictly feasible solution to (V) and then choose t to be large enough). The fact that (S) is a consequence of (V) says exactly that the optimal value in the problem is nonnegative. By the Conic Duality Theorem, the dual problem

$$b^T\lambda - d\mu \to \max \mid A^T\lambda - c = 0, \mu = 1, \begin{pmatrix} \lambda \\ \mu \end{pmatrix} \geq_{\bar{\mathbf{K}}_*} 0$$

has a feasible solution with the value of the dual objective ≥ 0. Since, as it is easily seen, $\bar{\mathbf{K}}_* = \{(\lambda,\mu) \mid \lambda \in \mathbf{K}_*, \mu \geq 0\}$, the indicated solution satisfies the requirements

$$\lambda \geq_{\mathbf{K}_*} 0, A^T\lambda = c, b^T\lambda \geq d,$$

i.e., (S) can be obtained from (V) by an admissible aggregation. ■

"Robust solvability status". Examples 2.4.2 – 2.4.3 make it clear that in the general conic case we may meet "pathologies" which do not occur in LP. E.g., a feasible and below bounded problem may be unsolvable, the dual to a solvable conic problem may be infeasible, etc. Where the pathologies come from? Looking at our "pathological examples", we come to the following guess: the source of the pathologies is that *in these examples, the "solvability status" of the primal problem is non-robust – it can be changed by small perturbations of the data.* The issue of *robustness* we are led to is very important in modeling, and it deserves a careful investigation.

Data of a conic problem. When asked "What are the data of an LP program $\min\{c^T x \mid Ax - b \geq 0\}$", everybody will give the same answer: "the objective c, the constraint matrix A and the right hand side vector b". Similarly, speaking about a conic problem

$$c^T x \to \min \mid Ax - b \geq_{\mathbf{K}} 0, \tag{CP}$$

it makes sense to treat as its data the triple (c, A, b); the sizes of the problem – the dimension n of x and the dimension m of $\mathbf{K}$, same as the underlying cone $\mathbf{K}$, may be treated as the structure of (CP).

Robustness. When investigating (CP), the question of primary importance is whether the properties of the program are stable with respect to perturbations of the data. The reasons which make this question important are as follows:

- In actual applications, especially those arising in Engineering, the data normally are inexact: their true values, even when they "exist in the nature", are not known exactly when the problem is processed. Consequently, the results of the processing say something definite about the "true" problem only if these results are robust with respect to small data perturbations: the property of (CP) expressed by the result (feasibility, solvability, etc.) is shared not only by the particular problem we are processing, but also by all problems with close data.

- Even assuming that the exact data are available, we should take into account that processing them computationally we unavoidably add "noise" like rounding errors (you simply cannot load something like 1/7 to the standard computer). As a result, a real-life computational routine can recognize only those properties of the input problem which are stable with respect to small perturbations of the data.

Due to the indicated reasons, we should be interested not only in whether a given problem (CP) is feasible/bounded/solvable, etc., but also whether these properties are robust – remain unchanged under small data perturbations. And it turns out that the Conic Duality Theorem allows to recognize "robust feasibility/boundedness/solvability...".

Let us start with introducing the relevant concepts. Let us say that (CP) is

- *robust feasible*, if all "enough close" problems (i.e., all problems of the same structure $(n, m, \mathbf{K})$ with data enough close to those of (CP)) are feasible;

- *robust infeasible*, if all enough close problems are infeasible;

- *robust below bounded*, if all "enough close" problems are below bounded (i.e., their objectives are below bounded on their feasible sets);

- *robust unbounded*, if all "enough close" problems are not bounded;

- *robust solvable*, if all "close enough" problems are solvable.

Note that a problem which is not feasible, is infeasible; in contrast to this, a problem which not robust feasible, *not necessarily* is robust infeasible, since among close problems there may be both feasible and infeasible (look at Example 2.4.2 – slightly shifting and rotating the plane $\mathrm{Im}\, A - b$, we may get whatever we want – feasible bounded problem, feasible unbounded problem, infeasible problem...). This is why we need two kinds of definitions: one of "robust presence of a property" and one more of "robust absence of the same property".

Now let us look what are necessary and sufficient conditions for the most important robust forms of the "solvability status".

Proposition 2.4.3 [Robust feasibility] (CP) *is robust feasible if and only if it is strictly feasible. Whenever it is the case, the dual to* (CP) *problem* (D) *is robust (above) bounded.*

Proof. The statement is nearly tautological. Let usn fix $\delta >_{\mathbf{K}} 0$. If (CP) is robust feasible, then for small enough $t > 0$ the perturbed problem $\min\{c^T x \mid Ax - b - t\delta \geq_{\mathbf{K}} 0\}$ should be feasible; a feasible solution to the perturbed problem clearly is a strictly feasible solution to (CP). The inverse implication is evident (a strictly feasible solution to (CP) remains feasible for all problems with close enough data). It remains to note that if all problems "enough close" to (CP) are feasible, then their duals, by the Weak Duality Theorem, are above bounded, so that (D) is robust above bounded. ■

Proposition 2.4.4 [Robust infeasibility] (CP) *is robust infeasible if and only if the problem*

$$b^T\lambda = 1, A^T\lambda = 0, \lambda \geq_{\mathbf{K}_*} 0$$

is robust feasible, or, which is the same by Proposition 2.4.3, if the problem

$$b^T\lambda = 1, A^T\lambda = 0, \lambda >_{\mathbf{K}_*} 0 \qquad (2.4.3)$$

has a solution.

Proof. First assume that (2.4.3) is solvable, and let us prove that all close enough to (CP) problems are infeasible. Let us fix a solution $\bar{\lambda}$ to (2.4.3). Since A is of full column rank, simple Linear Algebra says that the systems $[A']^T\lambda = 0$ are solvable for all matrices A' from a small enough neighbourhood U of A; moreover, the corresponding solution $\lambda(A')$ can be chosen to satisfy $\lambda(A) = \bar{\lambda}$ and to be continuous in $A' \in U$. Since $\lambda(A')$ is continuous and $\lambda(A) >_{\mathbf{K}_*} 0$, $\lambda(A')$ is $>_{\mathbf{K}_*} 0$ in a neighbourhood of A; shrinking U appropriately, we may assume that $\lambda(A') >_{\mathbf{K}_*} 0$ for all $A' \in U$. Now, $b^T\bar{\lambda} = 1$; by continuity reasons, there exists a neighbourhood V of b and a neighbourhood U' of A such that $b' \in V$ and all $A' \in U'$ one has $(b')^T\lambda(A') > 0$.

Thus, we have seen that there exist a neighbourhood U' of A and a neighbourhood V of b, along with a function $\lambda(A')$, $A' \in U'$, such that

$$(b')^T\lambda(A') > 0, [A']^T\lambda(A') = 0, \lambda(A') \geq_{\mathbf{K}_*} 0$$

for all $b' \in V$ and $A' \in U$. By Proposition 2.4.1.(i) it means that all problems

$$[c']^T x \to \min \mid A'x - b' \geq_{\mathbf{K}} 0$$

with $b' \in V$ and $A' \in U'$ are infeasible, so that (CP) is robust infeasible. Now let us assume that (CP) is robust infeasible, and let us prove that then (2.4.3) is solvable. Indeed, by definition of robust infeasibility, there exist neighbourhoods U of A and V of b such that all vector inequalities

$$A'x - b' \geq_{\mathbf{K}} 0$$

with $A' \in U$ and $b' \in V$ are unsolvable. It follows that whenever $A' \in U$ and $b' \in V$, the vector inequality

$$A'x - b' \geq_{\mathbf{K}} 0$$

is not almost solvable (see Proposition 2.4.1). We conclude from Proposition 2.4.1.(ii) that for every $A' \in U$ and $b' \in V$ there exists $\lambda = \lambda(A', b')$ such that

$$[b']^T\lambda(A', b') > 0, [A']^T\lambda(A', b') = 0, \lambda(A', b') \geq_{\mathbf{K}_*} 0.$$

Now let us choose $\lambda_0 >_{\mathbf{K}_*} 0$. For all small enough positive ϵ we have $A_\epsilon = A + \epsilon b[A^T\lambda_0]^T \in U$. Let us choose an ϵ with the latter property to be so small that $\epsilon b^T\lambda_0 > -1$ and set $A' = A_\epsilon$, $b' = b$. According to the previous observation, there exists $\lambda = \lambda(A', b)$ such that

$$b^T\lambda > 0, [A']^T\lambda \equiv A^T[\lambda + \epsilon\lambda_0(b^T\lambda)] = 0, \lambda \geq_{\mathbf{K}_*} 0.$$

Setting $\bar{\lambda} = \lambda + \epsilon\lambda_0(b^T\lambda)$, we get $\bar{\lambda} >_{\mathbf{K}_*} 0$ (since $\lambda \geq_{\mathbf{K}_*} 0, \lambda_0 >_{\mathbf{K}_*} 0$ and $b^T\lambda > 0$), while $A\bar{\lambda} = 0$ and $b^T\bar{\lambda} = (b^T\lambda)(1 + \epsilon b^T\lambda_0) > 0$. Multiplying $\bar{\lambda}$ by appropriate positive factor (namely, by $1/(b^T\bar{\lambda})$), we get a solution to (2.4.3). ■

Now we are able to formulate our main result on "robust solvability".

Proposition 2.4.5 *For a conic problem* (CP) *the following conditions are equivalent to each other*

(i) (CP) *is robust feasible and robust (below) bounded;*

(ii) (CP) *is robust solvable;*

(iii) (D) *is robust solvable;*

(iv) (D) *is robust feasible and robust (above) bounded;*

(v) *Both* (CP) *and* (D) *are strictly feasible.*

In particular, under every one of these equivalent assumptions, both (CP) *and* (D) *are solvable with equal optimal values.*

Proof. (i) $\Rightarrow$ (v): If (CP) is robust feasible, it also is strictly feasible (Proposition 2.4.3). If, in addition, (CP) is robust below bounded, then (D) is robust solvable (by the Conic Duality Theorem); in particular, (D) is robust feasible and therefore strictly feasible (the same Proposition 2.4.3).

(v) $\Rightarrow$ (ii): The implication is given by the Conic Duality Theorem.

(ii) $\Rightarrow$ (i): trivial.

We have proved that (i)$\equiv$(ii)$\equiv$(v). Due to the primal-dual symmetry, we also have proved that (iii)$\equiv$(iv)$\equiv$(v). ∎

2.5 Conic Duality revisited

To understand what is our concern now, consider a simple example: an optimization program with just two variables and four constraints:

$$\begin{array}{rcl} x_1 + 2x_2 & \to & \max \\ x_1 + x_2 & = & 4 \\ x_1 - x_2 & \leq & 3 \\ x_1 & \geq & 0 \\ x_2 & \geq & 0 \end{array}$$

Is this an LP program? Of course, yes! And why? As stated, it is *not* a problem of minimizing a linear form over the intersection of an affine plane and the nonnegative orthant, as an LP program should be... What is meant when saying that our problem "is" an LP program, is that it can be quite routinely *converted* to a "true" LP program – the one which indeed is to minimize a linear form on the intersection of a plane and the orthant. In principle, there are two conversion policies:

First, we can use the equality constraint(s) in order to express in an affine fashion part of the variables via the remaining "free" variables. What we end up with will be an *inequality constrained* – without equations – problem of optimizing a linear objective of the free variables. Of course, the resulting problem is a "true" LP program – a conic program associated with the nonnegative orthant $\mathbf{R}^n_+$ – (in LP the latter form is called *canonical*).

Second, we can add to our original design variables a number of artificial variables – "slacks" – and to convert all "nontrivial" inequality constraints – those saying more than that a particular variable should be nonnegative – into equality constraints. In our example this manipulation looks as follows:

$$\begin{array}{rcl} x_1 + 2x_2 & \to & \max \\ x_1 + x_2 & = & 4 \\ x_1 - x_2 + s & = & 3 \\ x_1 & \geq & 0 \\ x_2 & \geq & 0 \\ s & \geq & 0 \end{array}$$

What we end up with again is a "true" LP program – now in the dual form (D) (in LP, this form is called *standard*).

To process the problem analytically (e.g., to build its dual), the second option is incomparably better – it does not require messy computations.

What is said about LP, is valid in the general conic case as well. The fact that we finally can convert an optimization program to a conic form normally does not mean that the original form of the problem reads "minimize a linear form over the intersection of an affine plane and a cone". A typical original form it is something like

$$\begin{array}{rcl} c^T x & \to & \min \\ Px & = & p \\ A_i x - b_i & \geq_{\mathbf{K}^i} & 0, \ i = 1, ..., m \end{array} \qquad \text{(Ini)}$$

where $\mathbf{K}^i$ are different cones.

Let us look how to convert (Ini) to a "really conic" form, like (CP) or (D), and how to build the dual problem

If (Ini) has <u>no</u> equality constraints, it *already is* a conic problem in the form (CP). Indeed, it suffices to define a new cone $\mathbf{K}$ as the direct product of the cones $\mathbf{K}^i$, $i = 1, ..., m$:

$$\mathbf{K} = \{(y_1, ..., y_m) \mid y_i \in \mathbf{K}^i, \ i = 1, ..., m\}$$

and to write the problem

$$c^T x \to \min \mid Ax - b \equiv \begin{bmatrix} A_1 x - b_1 \\ A_2 x - b_2 \\ \dots \\ A_m x - b_m \end{bmatrix} \geq_{\mathbf{K}} 0.$$

Exercise 2.4 states that the direct product of cones $\mathbf{K}_i$ is a cone, and its dual is the direct product of the dual cones $\mathbf{K}_i^*$, so that what we get indeed is a conic problem.

Now, what to do if (Ini) does have equality constraints? Then we may act as in the our LP example; and by the same reasons as above, we prefer to add slack variables than to eliminate variables; thus, our final target is a conic program of the form (D).

It is clear that our target is achievable: a trivial way to reach it is:

1. To pass from (Ini) to an equivalent problem where the design vector is restricted to belong to some cone $\mathbf{K}_0$. This is exactly what people do in LP, replacing "free" variables – those without restrictions on their signs – as differences of two nonnegative variables. Let us do the same, and let us also switch from minimization to maximization:

$$\text{(Ini)} \mapsto \left\{ \begin{array}{rcl} \multicolumn{3}{l}{c^T(v - u) \to \max} \\ Pu - Pv & = & p \\ A_i(u - v) - b_i & \geq_{\mathbf{K}^i} & 0, \ i = 1, ..., m; \\ u & \geq & 0 \\ v & \geq & 0 \end{array} \right. \qquad \text{(Med)}$$

(the optimal value in (Med) is the negation of the one in (Ini))

2. It remains to add to (Med) slack variables. These variables should correspond to the vector inequality constraints $A_i(u-v)-b_i \geq_{\mathbf{K}_i} 0$ and should therefore be vectors of the same dimensions as b_i. Denoting these slack vectors by s_i, we transform (Med) as follows:

$$(\text{Med}) \mapsto \left\{\begin{array}{rcl} c^T(v-u) \to \max & & \\ Pu - Pv & = & p \\ A_i(u-v) - s_i & = & b_i,\ i=1,...,m \\ u & \geq & 0 \\ v & \geq & 0 \\ s_i & \geq_{\mathbf{K}^i} & 0,\ i=1,...,m. \end{array}\right. \qquad (\text{Fin})$$

We end up with problem (Fin) which is equivalent to (Ini) and clearly is a conic problem in the form of (D).

Of course, normally we can act in a smarter way than in the outlined quite straightforward scheme. E.g., in many cases one can extract from the original inequality constraints $A_ix - b_i \geq_{\mathbf{K}_i} 0$ a subset I of constraints saying that x belongs to some cone $\mathbf{K}_0$ (look at the inequality constraints $x_1, x_2 \geq 0$ in our LP example). If it is the case, there is no necessity in the updating (Ini) ⇒ (Med), same as there is no necessity in introducing slacks for the constraints from I. Sometimes there is no subset of constraints saying that x belongs to a cone, but there is a subset I saying that certain <u>sub</u>vector x' of x belongs to a certain cone; whenever this is the case, we can modify the first step of the above scheme – to represent as $u-v$ the complement of x' in x, not the entire x – and are not obliged to introduce slacks for the constraints from I at the second step, etc.

Now, what is the conic dual to (Fin)? The cone associated with (Fin) is

$$\mathbf{K}_* = \mathbf{R}^n_+ \times \mathbf{R}^n_+ \times \mathbf{K}^1 \times \mathbf{K}^2 \times ... \times \mathbf{K}^m,$$

the objective (to be maximized!) in (Fin) is

$$\begin{pmatrix} -c \\ c \\ 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}^T \begin{pmatrix} u \\ v \\ s_1 \\ s_2 \\ \dots \\ s_m \end{pmatrix},$$

and the equality constraints are

$$\begin{pmatrix} P & -P & & & & \\ A_1 & -A_1 & -I_1 & & & \\ A_2 & -A_2 & & -I_2 & & \\ \vdots & \vdots & & & \ddots & \\ A_m & -A_m & & & & -I_m \end{pmatrix} \begin{pmatrix} u \\ v \\ s_1 \\ s_2 \\ \dots \\ s_m \end{pmatrix} = \begin{pmatrix} p \\ b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix},$$

I_i being unit matrices of appropriate sizes. Taking into account that the cone $\mathbf{K}_*$ is dual to the cone

$$\mathbf{K} = \mathbf{R}^n_+ \times \mathbf{R}^n_+ \times \mathbf{K}^1_* \times \mathbf{K}^2_* \times ... \times \mathbf{K}^m_*,$$

we conclude that (Fin) is dual to the following conic problem of the form (CP):

$$\begin{array}{rcl} p^T\mu + \sum_{i=1}^m b_i^T\xi_i & \to & \min \\ P^T\mu + \sum_{i=1}^m A_i^T\xi_i + c & \geq & 0; \\ -P^T\mu - \sum_{i=1}^m A_i^T\xi_i - c & \geq & 0; \\ -\xi_i & \geq_{\mathbf{K}^i_*} & 0,\ i = 1, ..., m, \end{array}$$

the variables being μ (a vector of the same dimension as p) and ξ_i, $i = 1, ..., m$, of the same dimensions as $b_1, ..., b_m$. The first two $\geq$-constrains in the resulting problem are equivalent to the vector equation

$$P\mu + \sum_{i=1}^m A_i^T\xi_i = -c;$$

it also makes sense to pass from variables ξ_i to their negations $\eta_i = -\xi_i$, from μ to its negation $\nu = -\mu$ and to switch from minimization to maximization, thus coming to the problem

$$\begin{array}{rcl} p^T\nu + \sum_{i=1}^m b_i^T\eta_i & \to & \max \\ P^T\nu + \sum_{i=1}^m A_i^T\eta_i & = & c \\ \eta_i & \geq_{\mathbf{K}^i_*} & 0,\ i = 1, ..., m. \end{array} \qquad \text{(Dl)}$$

with design variables $\nu, \eta_1, ..., \eta_m$; the resulting problem will be called the problem *dual* to the *primal* problem (Ini). Note that we have extended somehow our duality scheme – previously it required from the primal problem to be "purely conic" – not to have linear equality constraints; from now on this restriction is eliminated.

Summary on building the dual

Same as in the LP textbooks, let us summarize the rules for building the dual:

> *Consider a "primal" problem – an optimization problem with linear objective and linear vector equality/inequality constraints*
>
> $$c^Tx \to \min$$
>
> *s.t.*
>
> $$\begin{array}{rcll} Px & = & p & \text{[dim } p \text{ scalar linear equations]} \\ A_1x - b_1 & \geq_{\mathbf{K}^1} & 0 & \text{[linear vector inequality \# 1]} \\ \dots & & & \\ A_mx - b_m & \geq_{\mathbf{K}^m} & 0 & \text{[linear vector inequality \# } m\text{]} \end{array} \qquad \text{(Pr)}$$
>
> *The dual to* (Pr) *problem is*
>
> $$p^T\nu + \sum_{i=1}^m b_i^T\eta_i \quad \to \quad \max$$
>
> *s.t.*
>
> $$\begin{array}{rcll} P^T\nu + \sum_{i=1}^m A_i^T\eta_i & = & c & \text{[dim } x \text{ scalar equations]} \\ \eta_1 & \geq_{\mathbf{K}^1_*} & 0 & \text{[linear vector inequality \# 1]} \\ \dots & & & \\ \eta_m & \geq_{\mathbf{K}^m_*} & 0 & \text{[linear vector inequality \# } m\text{]} \end{array} \qquad \text{(Dl)}$$

Note that

1. Dual variables correspond to the constraints of the primal problem: the dual design vector is comprised of a vector variable ν of the same dimension as the right hand side p of the system

of primal equalities and of m vector variables η_i, $i = 1, ..., m$, of the same dimensions as those of the primal vector inequalities

2. There is a natural one-to-one correspondence between the vector inequalities of the primal and those of the dual problem, and the cones underlying corresponding to each other primal and dual vector inequalities are dual to each other.

3. The problem dual to (Dl) is (equivalent to) the primal problem (Pr).

> Indeed, (Dl) is of the same structure as (Pr), so that we can apply the outlined construction to (Dl). The (vector) variables of the resulting problem are as follows:
>
> – the first of them, let it be called x', is responsible for the system of equations in (Dl); the dimension of this variable is dim c, i.e., it is equal to the design dimension of (Pr);
>
> – The remaining m vector variables, let them be called w_i, $i = 1, ..., m$, are responsible each for its own linear vector inequality of (Dl).
>
> Applying the outlined scheme to (Dl) (please do it and do not forget to start with passing from the maximization problem (Dl) to an equivalent minimization problem; this is preassumed by our scheme), we come to the problem
>
> $$c^T x' \to \max$$
>
> s.t.
>
> $$\begin{array}{rcl} Px' & = & -p \\ A_i x' + w_i & = & -b_i, i = 1, ..., m, \\ w_i & \geq_{\mathbf{K}^i} & 0,\ i = 1, ..., m, \end{array}$$
>
> and the resulting problem is equivalent to (Pr) (to see it, set $x = -x'$).

Summary on Conic Duality

The results on Conic Duality we have developed so far deal with the pair of "pure" conic problems (Fin) – (Dl), not with the pair of problems (Pr) – (Dl); one, however, can easily express these results directly in terms of (Pr) and (Dl). Here is the translation:

1. The role of Assumption **A** is now played by the pair of requirements as follows:

   A.1 *The rows of the matrix P in (Pr) are linearly independent;*

   A.2 *There is no nonzero vector x such that $Px = 0$, $A_i x = 0$, $i = 1, ..., m$.*

   From now on, speaking about problem (Pr), we *always* assume that **A.1**, **A.2** take place.

   > Note that **A.1**, **A.2** imply that both (Fin) and (Dl) satisfy **A** (why?).

2. Strict feasibility: A problem of the form (Pr) is called *strictly feasible*, if there exist a feasible solution $\bar{x}$ which satisfies the strict versions of all vector inequality constraints of the problem, i.e., is such that $A_i\bar{x} - b_i >_{\mathbf{K}_i} 0$, $i = 1, ..., m$.

   > Note that (Pr) is strictly feasible if and only if (Fin) is.

3. Weak Duality: *The optimal value in* (Dl) *is less than or equal to the optimal value in* (Pr).

4. Strong Duality: *If one of the problems* (Pr), (Dl) *is strictly feasible and bounded, then the other problem is solvable, the optimal values in the problems being equal to each other.*

   If both problems are strictly feasible, both are solvable, the optimal values being equal to each other.

5. Optimality Conditions: *Let x be a feasible solution to* (Pr) *and* $\lambda = (\nu, \{\eta_i\}_{i=1}^m)$ *be a feasible solution to* (Dl)*. Then the* duality gap *at the pair – the quantity*

$$\Delta(x,\lambda) = c^T x - \left[p^T\nu + \sum_{i=1}^m b_i^T\eta_i\right]$$

always is nonnegative and is equal to

$$\sum_{i=1}^m \eta_i^T[A_i x - b_i].$$

The duality gap is zero if and only if the complementary slackness holds:

$$\eta_i^T[A_i x - b_i] = 0,\ \ i = 1, ..., m.$$

If the duality gap $\Delta(x,\lambda)$ is zero, then x is an optimal solution to (Pr) *and λ is an optimal solution to* (Dl)*.*

If x is an optimal solution to (Pr) *and λ is an optimal solution to* (Dl) *and the optimal values in the problems are equal, then the duality gap $\Delta(x,\lambda)$ is zero.*

2.6 Assignments to Lecture # 2

2.6.1 Around cones

Recall that a "cone" for us always means "pointed closed convex cone with a nonempty interior in certain $\mathbf{R}^n$.

Theorem 2.3.1

Exercise 2.1 [10] *1. Prove the following statement:*

> *Let S be a nonempty closed convex set in $\mathbf{R}^n$ and x be a point in $\mathbf{R}^n$ outside of S. Then the problem*
>
> $$\min\{(x-y)^T(x-y) \mid y \in S\}$$
>
> *has a unique solution x^*, and $e \equiv x - x^*$ strictly separates x and S:*
>
> $$e^T x \geq e^T e + \sup_{y\in S} e^T y > \sup_{y \in S} e^T y.$$

*2. *Derive from 1) Theorem 2.3.1.*

3. Derive from Theorem 2.3.1 that whenever $0 \neq x \geq_{\mathbf{K}} 0$, there exists $\lambda \geq_{\mathbf{K}_*} 0$ such that $\lambda^T x > 0$.

The interior of a cone

Exercise 2.2 [3] *Let $\mathbf{K}$ be a cone, and let $\bar{x} >_{\mathbf{K}} 0$. Prove that $x >_{\mathbf{K}} 0$ if and only if there exists positive t such that $x \geq_{\mathbf{K}} t\bar{x}$.*

Exercise 2.3 [5] *1. Prove that if $0 \neq x \geq_{\mathbf{K}} 0$ and $\lambda >_{\mathbf{K}_*} 0$, then $\lambda^T x > 0$.*

> Hint: Use the result of Exercises 2.1 and 2.2.

2. Prove that if $\lambda >_{\mathbf{K}_*} 0$, then for every real a the set

$$\{x \geq_{\mathbf{K}} 0 \mid \lambda^T x \leq a\}$$

is bounded.

"Calculus" of cones

Exercise 2.4 [20] *Prove the following statements:*

1. [stability with respect to direct multiplication] *Let $\mathbf{K}_i \subset \mathbf{R}^{n_i}$ be cones, $i = 1, ..., k$. Prove that the direct product of the cones – the set*

   $$\mathbf{K} = \mathbf{K}_1 \times ... \times \mathbf{K}_k = \{(x_1, ..., x_k) \mid x_i \in \mathbf{K}_i, \ i = 1, ..., k\}$$

   is a cone in $\mathbf{R}^{n_1+...+n_k} = \mathbf{R}^{n_1} \times ... \times \mathbf{R}^{n_k}$.

   Prove that the cone dual to $\mathbf{K}$ is the direct product of the cones dual to $\mathbf{K}_i$, $i = 1, .., k$.

2. [stability with respect to taking inverse image] *Let* $\mathbf{K}$ *be a cone in* $\mathbf{R}^n$ *and* $u \mapsto Au$ *be a linear mapping from certain* $\mathbf{R}^k$ *to* $\mathbf{R}^n$ *with trivial kernel and the image intersecting the interior of* $\mathbf{K}$. *Prove that the inverse image of* $\mathbf{K}$ *under the mapping – the set*

$$\mathbf{K}^{\leftarrow} = \{u \mid Au \in \mathbf{K}\}$$

– is a cone in $\mathbf{R}^k$.

* *Prove that the cone dual to* $\mathbf{K}^{\leftarrow}$ *is the image of* $\mathbf{K}_*$ *under the mapping* $\lambda \mapsto A^T\lambda$

$$(\mathbf{K}^{\leftarrow})_* = \{A^T\lambda \mid \lambda \in \mathbf{K}_*\}.$$

3. [stability with respect to taking linear image] *Let* $\mathbf{K}$ *be a cone in* $\mathbf{R}^n$ *and* $y = Ax$ *be a linear mapping from* $\mathbf{R}^n$ *<u>onto</u>* $\mathbf{R}^N$ *(i.e., the image of* A *is the entire* $\mathbf{R}^N$*). Assume that the intersection of* $\operatorname{Ker} A$ *and* $\mathbf{K}$ *is the singleton* $\{0\}$.

**Prove that then the image* $\mathbf{K}^{\rightarrow}$ *of* $\mathbf{K}$ *under the mapping* A *– the set*

$$\mathbf{K}^{\rightarrow} = \{Ax \mid x \in \mathbf{K}\}$$

is a cone in $\mathbf{R}^N$.

* *Prove that the cone dual to* $\mathbf{K}^{\rightarrow}$ *is*

$$(\mathbf{K}^{\rightarrow})_* = \{\lambda \in \mathbf{R}^N \mid A^T\lambda \in \mathbf{K}_*\}.$$

* *Demonstrate by example that if in the above statement the assumption* $\operatorname{Ker} A \cap \mathbf{K} = \{0\}$ *is weakened to* $\operatorname{Ker} A \cap \operatorname{int} \mathbf{K} = \emptyset$, *then the image of* $\mathbf{K}$ *under the mapping* A *may happen to be non-closed.*

> Hint. Look what happens when the 3D ice-cream cone is projected onto its tangent plane.

Primal-dual pairs of cones and orthogonal pairs of subspaces

Exercise 2.5 [5] *Let* A *be a* $m \times n$ *matrix of full column rank and* $\mathbf{K}$ *be a cone in* $\mathbf{R}^m$.

1. *Prove that <u>at least</u> one of the following facts always takes place:*

   (i) *There exists a nonzero* $x \in \operatorname{Im} A$ *which is* $\geq_{\mathbf{K}} 0$*;*

   (ii) *There exists a nonzero* $\lambda \in \operatorname{Ker} A^T$ *which is* $\geq_{\mathbf{K}_*} 0$.

   Geometrically: given a primal-dual pair of cones $\mathbf{K}$, $\mathbf{K}_*$ *and a pair* $L, L^\perp$ *of linear subspaces which are orthogonal complements of each other, we either can find a nontrivial ray in the intersection* $L \cap \mathbf{K}$, *or in the intersection* $L^\perp \cap \mathbf{K}_*$, *or both.*

2. *Prove that the "strict" version of* (ii) *takes place (i.e., there exists* $\lambda \in \operatorname{Ker} A^T$ *which is* $>_{\mathbf{K}} 0$*) if and only if* (i) *does not take place, and vice versa: the strict version of* (i) *takes place if and only if* (ii) *does not take place.*

   Geometrically: if $\mathbf{K}, \mathbf{K}_*$ *is a primal-dual pair of cones and* $L, L^\perp$ *are linear subspaces which are orthogonal complements of each other, then the intersection* $L \cap \mathbf{K}$ *is trivial (is the singleton* $\{0\}$*) if and only if the intersection* $L^\perp \cap \operatorname{int} \mathbf{K}_*$ *is nonempty. And vice versa: if the "strict" version of* (ii) *takes place, than* (i) *does not take place.*

Several interesting cones

Given a cone $\mathbf{K}$ along with its dual $\mathbf{K}_*$, let us call a *complementary pair* every pair $x \in \mathbf{K}$, $\lambda \in \mathbf{K}_*$ such that

$$\lambda^T x = 0.$$

Recall that in "good cases" (e.g., under the premise of item 4 of the Conic Duality Theorem) a pair of feasible solutions (x, λ) of a primal-dual pair of conic problems

$$c^T x \to \min \mid Ax - b \geq_{\mathbf{K}} 0$$

$$b^T \lambda \to \max \mid A^T \lambda = c, \lambda \geq_{\mathbf{K}_*} 0$$

is comprised of optimal solutions if and only if the "primal slack" $y = Ax - b$ and λ are complementary.

Exercise 2.6 ^{3} [Nonnegative orthant] *Prove that the n-dimensional nonnegative orthant $\mathbf{R}^n_+$ indeed is a cone and that it is self-dual:*

$$(\mathbf{R}^n_+) = \mathbf{R}^n_+.$$

What are complementary pairs?

Exercise 2.7 ^{5} [Ice-cream cone] *Let $\mathbf{L}^n$ be the n-dimensional ice-cream cone:*

$$\mathbf{L}^n = \{x \in \mathbf{R}^n \mid x_n \geq \sqrt{x_1^2 + \ldots + x_{n-1}^2}\}.$$

1. *Prove that $\mathbf{L}^n$ indeed is a cone.*

2. *Prove that the ice-cream cone is self-dual:*

$$(\mathbf{L}^n)_* = \mathbf{L}^n.$$

3. *Characterize the complementary pairs.*

Exercise 2.8 ^{5} [Positive semidefinite cone] *Let $\mathbf{S}^n_+$ be the cone of $n \times n$ positive semidefinite matrices in the space $\mathbf{S}^n$ of symmetric $n \times n$ matrices. Assume that $\mathbf{S}^n$ is equipped with the Frobenius inner product*

$$\langle X, Y \rangle \;= \mathrm{Tr}(XY) = \sum_{i,j=1}^{n} X_{ij} Y_{ij}.$$

1. *Prove that $\mathbf{S}^n_+$ indeed is a cone.*

2. *Prove that the semidefinite cone is self-dual:*

$$(\mathbf{S}^n_+)_* = \mathbf{S}^n_+,$$

i.e., that the Frobenius inner products of a symmetric matrix Λ with all positive semidefinite matrices X of the same size are nonnegative if and only if the matrix itself is nonnegative.

3. *Prove the following characterization of the complementary pairs:*

> *Two matrices $X \in \mathbf{S}^n_+$, $\Lambda \in (\mathbf{S}^n_+)_* \equiv \mathbf{S}^n_+$ are complementary (i.e., $\langle \Lambda, X \rangle = 0$) if and only if their product is zero: $\Lambda X = 0$. In particular, matrices from a complementary pair commutate and share therefore a common orthonormal eigenbasis.*

2.6.2 Around conic problems

Several primal-dual pairs

Exercise 2.9 [5] [The min-max Steiner problem] *Consider the problem as follows:*

> *Given N points $b_1, ..., b_N$ in $\mathbf{R}^n$, find a point $x \in \mathbf{R}^n$ which minimizes the maximum, over $i = 1, ..., N$, (Euclidean) distance from itself to the points $b_1, ..., b_N$, i.e., solve the problem*
>
> $$\max_{i=1,...,N} \| x - b_i \|_2 \rightarrow \min .$$
>
> Imagine, e.g., that $n = 2$, $b_1, ..., b_N$ are locations of villages and you are interested to locate a fire station for which the worst-case distance to a possible fire is as small as possible.

1. *Pose the problem as a conic quadratic one – a conic problem associated with a direct product of ice-cream cones.*

2. *Build the dual problem.*

3. *What is the geometric interpretation of the dual? Whether the primal and the dual are strictly feasible? Solvable? With equal optimal values? What the complementary slackness says?*

Exercise 2.10 [5] [The weighted Steiner problem] *Consider the problem as follows:*

> *Given N points $b_1, ..., b_N$ in $\mathbf{R}^n$ along with positive weights ω_i, $i = 1, ..., N$, find a point $x \in \mathbf{R}^n$ which minimizes the weighted sum of its (Euclidean) distances to the points $b_1, ..., b_N$, i.e., solve the problem*
>
> $$\sum_{i=1}^{N} \omega_i \| x - b_i \|_2 \rightarrow \min .$$
>
> Imagine, e.g., that $n = 2$, $b_1, ..., b_N$ are locations of villages and you are interested to locate a telephone station for which the total cost of cables linking the station and the villages is as small as possible; the weights can be interpreted as the per mile cost of the cables; they may differ from each other because of differences in village populations and, consequently, in the capacities of the required cables.

1. *Pose the problem as a conic quadratic one – a conic problem associated with a direct product of ice-cream cones.*

2. *Build the dual problem.*

3. *What is the geometric interpretation of the dual? Whether the primal and the dual are strictly feasible? Solvable? With equal optimal values? What the complementary slackness says?*

2.6.3 Feasible and level sets of conic problems

Consider a feasible conic problem

$$c^T x \to \min \mid Ax - b \geq_{\mathbf{K}} 0 \tag{CP}$$

In many cases it is important to know whether the problem has

1) *bounded feasible set* $\{x \mid Ax - b \geq_{\mathbf{K}} 0\}$

2) *bounded level sets*

$$\{x \mid Ax - b \geq_{\mathbf{K}} 0, c^T x \leq a\}$$

for all real a.

Exercise 2.11 ^{7} *Let* (CP) *be feasible. Then the following properties are equivalent to each other:*

(i) *the feasible set of the problem is bounded;*

(ii) *the set of primal slacks* $K = \{y \geq_{\mathbf{K}} 0, y = Ax - b\}$ *is bounded*[2)]

(iii) $\mathrm{Im}\, A \cap \mathbf{K} = \{0\}$

(iv) *the system of vector inequalities*

$$A^T \lambda = 0, \lambda >_{\mathbf{K}_*} 0$$

is solvable.

Corollary. The property of (CP) *to have bounded feasible set is independent of the particular value of* b *such that* (CP) *is feasible!*

Exercise 2.12 ^{10} *Let problem* (CP) *be feasible. Prove that the following two conditions are equivalent to each other:*

(i) (CP) *has bounded level sets*

(ii) *The dual problem*

$$b^T \lambda \to \max \mid A^T \lambda = c, \lambda \geq_{\mathbf{K}_*} 0$$

is strictly feasible.

Corollary. The property of (CP) *to have bounded level sets is independent of the particular value of* b *such that* (CP) *is feasible!*

[2)] recall that we always assume that **A** holds!

Lecture 3

Conic Quadratic Programming

There are several "generic" families of conic problems which are of especial interest, both from the viewpoint of theory and applications. The cones underlying these problems are enough simple, so that one can describe explicitly the dual cone; as a result, the general duality machinery we have developed becomes as algorithmic as the Linear Programming duality. And it turns out that the "algorithmic duality machinery" in many cases allows to understand a lot the original model, to convert it into equivalent forms better suited for numerical processing, etc. Moreover, relative simplicity of the underlying cones enables to develop efficient computational methods for the corresponding conic problems. The most famous example of a "nice" generic conic problem is, doubtless, the Linear Programming; however, this is not the only nice problem of this sort. Two other nice generic conic problems of extreme importance are *Conic Quadratic* and *Semidefinite* programs. We are about to consider the first of these two problems.

3.1 Conic Quadratic problems: preliminaries

Recall that the m-dimensional Lorentz ($\equiv$second-order$\equiv$ice-cream) cone $\mathbf{L}^m$ is the cone given by

$$\mathbf{L}^m = \{x = (x_1, ..., x_m) \in \mathbf{R}^m \mid x_m \geq \sqrt{x_1^2 + ... + x_{m-1}^2}\}.$$

Here $m \geq 1$; in the extreme case of $m = 1$ we, as usual, interpret the empty sum $\sum_{i=1}^{0} x_i^2$ under the square root as 0, so that $\mathbf{L}^1$ is just the nonnegative ray on the axis.

A *conic quadratic problem* is a conic problem

$$c^T x \to \min \mid Ax - b \geq_{\mathbf{K}} 0 \tag{CP}$$

for which the cone $\mathbf{K}$ is a direct product of several ice-cream cones:

$$\begin{array}{rcl} \mathbf{K} & = & \mathbf{L}^{m_1} \times \mathbf{L}^{m_2} \times ... \times \mathbf{L}^{m_k} \\ & = & \left\{ y = \begin{pmatrix} y[1] \\ y[2] \\ ... \\ y[k] \end{pmatrix} \mid y[i] \in \mathbf{L}^{m_i},\ i = 1, ..., k \right\}. \end{array} \tag{3.1.1}$$

In other words, a conic quadratic problem is an optimization problem with linear objective and finitely many *"ice-cream constraints"*

$$A_i x - b_i \geq_{\mathbf{L}^{m_i}} 0,\ i = 1, ..., k,$$

where

$$[A;b] = \left[\begin{array}{c} [A_1;b_1] \\ \hline [A_2;b_2] \\ \hline \dots \\ \hline [A_k;b_k] \end{array}\right]$$

is the partition of the data matrix $[A;b]$ corresponding to the partition of y in (3.1.1). Thus, a conic quadratic program can be written down also as

$$c^T x \to \min \mid A_i x - b_i \geq_{\mathbf{L}^{m_i}} 0,\ i = 1, ..., k. \tag{3.1.2}$$

Now it is worthy to recall what exactly is the partial order $\geq_{\mathbf{L}^m} 0$. For a vector $z \in \mathbf{R}^m$, the inequality $z \geq_{\mathbf{L}^m} 0$ means that the last entry in z is $\geq$ the Euclidean norm $\| \cdot \|_2$ of the sub-vector of z comprised of the first $m-1$ entries of z. Consequently, the $\geq_{\mathbf{L}^{m_i}}$ 0-inequalities in (3.1.2) can be written down as

$$\| D_i x - d_i \|_2 \leq p_i^T x - q_i,$$

where

$$[A_i; b_i] = \left[\begin{array}{cc} D_i & d_i \\ p_i^T & q_i \end{array}\right]$$

is the partitioning of the data matrix $[A_i, b_i]$ into the sub-matrix $[D_i; d_i]$ comprised of the first $m_i - 1$ rows and the last row $[p_i^T; q_i]$. We conclude that a conic quadratic problem can be written down as the optimization program

$$c^T x \to \min \mid \| D_i x - d_i \|_2 \leq p_i^T x - q_i,\ i = 1, ..., k, \tag{QP}$$

and this the "most explicit" form we prefer to use; in this form, D_i are matrices of the same row dimension as x, d_i are vectors of the same dimensions as the column dimensions of the matrices D_i, p_i are vectors of the same dimension as x and q_i are reals.

As we know from Exercises 2.7, 2.4, (3.1.1) indeed is a cone, moreover, a self-dual one: $\mathbf{K}_* = \mathbf{K}$. Consequently, the problem dual to (CP) is

$$b^T \lambda \to \max \mid A^T \lambda = c,\ \lambda \geq_{\mathbf{K}} 0.$$

Denoting $\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_k \end{pmatrix}$ with m_i-dimensional blocks λ_i (cf. (3.1.1)), we can write the dual problem as

$$\sum_{i=1}^k b_i^T \lambda_i \to \max \mid \sum_{i=1}^k A_i^T \lambda_i = c,\ \lambda_i \geq_{\mathbf{L}^{m_i}} 0,\ i = 1, ..., k;$$

recalling what does $\geq_{\mathbf{L}^{m_i}} 0$ mean and representing $\lambda_i = \binom{\mu_i}{\nu_i}$ with scalar component ν_i, we finally come to the following form of the dual problem:

$$\sum_{i=1}^k [\mu_i^T d_i + \nu_i q_i] \to \max \mid \sum_{i=1}^k [D_i^T \mu_i + \nu_i p_i] = c,\ \mid \| \mu_i \| \leq \nu_i,\ i = 1, ..., k. \tag{QD}$$

The design variables in the dual are vectors μ_i of the same dimensions as the vectors d_i and reals ν_i, $i = 1, ..., k$.

Since from now on we will treat (QP) and (QD) as the standard forms of a conic quadratic problem and its dual, it makes sense to say how our basic assumption **A** from Lecture 2 and notions like feasibility, strict feasibility, boundedness, etc., "read" in our new format. Assumption **A** from Lecture 2 now reads:

> *There is no nonzero x which is orthogonal to all rows of all matrices D_i and to all vectors p_i, $i = 1, , ., k$*

and we always make this assumption by default. Now, among notions like feasibility, solvability, etc., the only notion which does need a "translation" is the one of strict feasibility, which now reads as follows:

> *Strict feasibility of* (QP) *means that there exist $\bar{x}$ such that all inequality constraints $\| D_i x - d_i \|_2 \leq p_i^T x - q_i$ of the problem are satisfied as $\bar{x}$ as strict inequalities.*
>
> *Strict feasibility of* (QD) *means that there exists a feasible solution $\{\bar{\mu}_i, \bar{\nu}_i\}_{i=1}^k$ to the problem such that $\| \mu_i \|_2 < \bar{\nu}_i$ for all $i = 1, ..., k$.*

3.2 Examples of conic quadratic problems

Best linear approximation of complex-valued functions

Recall the Tschebyshev approximation problem from Lecture 1 which we now formulate as follows:

> *Given a finite set T, a "target" function f_* on this set and n "building blocks" – functions $f_1, ..., f_n$ on T – find linear combination of the functions $f_1, ..., f_n$ which is as close as possible, in the uniform on T norm, to the target function f_*, i.e., solve the problem*
>
> $$\min_x \{\max_{t \in T} |f_*(t) - \sum_{j=1}^n x_j f_j(t)|\} \qquad \text{(T)}$$

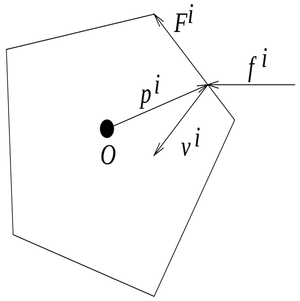
We have seen that in the case of *real-valued* functions $f_*, f_1, ..., f_n$ the problem can be posed as an LP program. We have also seen that in some applications the functions in question are *complex-valued*; this, e.g., is the case in the general Antennae Array problem (Section 1.2.4) and in the Filter Synthesis problem when the design specifications have to do with the transfer function (Section 1.2.3). What to do in these situations? Our approach in Lecture 1 was to approximate the modulus of a complex number (i.e., the Euclidean norm of a real 2D vector) by a "polyhedral norm" – the maximum of several linear functions of the vector; with this approximation, (T) becomes an LP program. If we prefer to avoid approximation, we may easily pose the complex-valued Tschebyshev problem as a conic quadratic program:

$$\tau \to \min \mid \| f_*(t) - \sum_{j=1}^n x_j f_j(t) \|_2 \leq \tau, \ t \in T \qquad (3.2.1)$$

with design variables x, τ; in (3.2.1) we treat complex numbers $f_*(t), f_j(t)$ as real 2D vectors.

Contact problems with static friction

The examples to follow, along with their analysis, are taken from [2]. Consider a rigid body in $\mathbf{R}^3$ held by N robot fingers. Whether the robot is able to hold the body? To pose the question mathematically, let us look what happens at i-th contact:



Geometry of i-th contact

[p^i is the contact point; f^i is the contact force; v^i is the inward normal to the surface]

Let p^i be the position of the contact, f^i be the contact force exerted by i-th finger, v^i be the unit inward normal to the surface of the body at the point p^i, and F^i be the friction force caused by the contact. Physics says that this force is tangential to the surface of the body:

$$(F^i)^T v^i = 0 \tag{3.2.2}$$

and its magnitude cannot exceed constant times the magnitude of the normal component of the contact force:

$$\| F^i \|_2 \leq \mu (f^i)^T v^i, \tag{3.2.3}$$

the friction coefficient μ being a given constant.

Assume that the body is subject to additional external forces (e.g., the gravity one); as far as their mechanical consequences are concerned, all these forces can be represented by a single force – their sum – $F^{\rm ext}$ along with the *torque* $T^{\rm ext}$ – the sum of vector products of the external forces and the points where they are applied.

In order for the body to be in static equilibrium, the total force acting at the body and the total torque should be zero:

$$\begin{array}{rcl} \sum_{i=1}^N (f^i + F^i) + F^{\rm ext} & = & 0 \\ \sum_{i=1}^N p^i \times (f^i + F^i) + T^{\rm ext} & = & 0, \end{array} \tag{3.2.4}$$

where $p \times q$ stands for the vector product of two 3D vectors p and q.

Stable grasp analysis problem. The question "whether the robot is capable to hold the body" can be interpreted as follows. Assume that $f^i, F^{\rm ext}, T^{\rm ext}$ are given. Then the friction forces F^i will adjust themselves to satisfy the friction constraints (3.2.2) – (3.2.3) and the equilibrium equations (3.2.4). If it is possible – i.e., if the system of constraints (3.2.2), (3.2.3), (3.2.4) with respect to unknowns F^i is solvable – then the robot holds the body (the scientific wording for this is "the body is in a *stable grasp*"), otherwise the body cannot be held.

Now, the question whether, for given $p^i, f^i, F^{\rm ext}, T^{\rm ext}$, the body is held in a stable grasp, mathematically is the question whether the system S of constraints (3.2.2), (3.2.3), (3.2.4) with unknowns $F^i \in \mathbf{R}^3$ is solvable. And the latter problem in fact is nothing but a *conic quadratic* *feasibility problem* – conic quadratic problem with trivial (identically zero) objective. We say "in fact", since the problem, as it arises is not in our canonical form. This is typical: "nice" in their essence problems normally arise not exactly in their "catalogue" forms, and one should know how to recognize what in fact he is dealing with. In our case this "recognition problem" is easy to solve. One way to see that S is a conic quadratic problem is to note that we can use the system of linear equations (3.2.2), (3.2.4) to express part of the unknowns via the remaining ones, let the latter be denoted by x. With this parameterization, every F^i becomes an affine vector-valued function $D_i x - d_i$ of the "free design vector" x, and the question we are interested in becomes whether the primal conic quadratic problem

$$0^T x \to \min \parallel D_i x - d_i \parallel_2 \leq \mu(f^i)^T v^i, \ i = 1, ..., N$$

is or is not feasible.

Another way to realize that S is a conic quadratic problem is to note that the problem from the very beginning is dual to a conic quadratic problem (see Exercise 3.1).

Stable grasp synthesis problems. To the moment we treated the contact forces f^i as given. Sometimes this is not the case, i.e., the robot can, to some extent, control tensions in its fingers. As a simple example, assume that the directions u^i of the contact forces – "directions of fingers" – are fixed, but the magnitudes of these forces can be controlled:

$$f^i = \nu_i u^i,$$

where the reals ν_i are allowed to vary in a given segment $[0, F_{\max}]$. We may now ask whether the robot can choose admissible magnitudes of the contact forces to ensure a stable grasp. Mathematically, the question is whether the system

$$\begin{array}{rcl} \sum_{i=1}^N (\nu_i u^i + F^i) + F^{\rm ext} & = & 0 \\ \sum_{i=1}^N p_i \times (\nu_i u^i + F^i) + T^{\rm ext} & = & 0 \\ (F^i)^T v^i & = & 0 \\ \hline \parallel F^i \parallel_2 & \leq & [\mu (u^i)^T v^i] \nu_i, \ i = 1, ..., N \\ 0 \leq \nu_i & \leq & F_{\max}, \ i = 1, ..., N \end{array} \qquad (3.2.5)$$

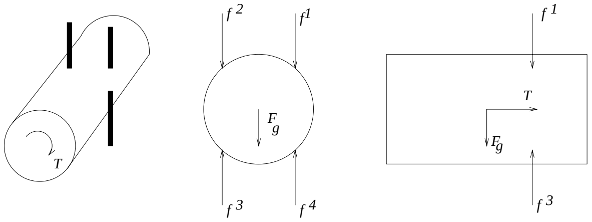
is solvable. And it is again a conic quadratic feasibility problem: same as above, we may eliminate the linear equations to end up with a system of conic quadratic and linear (i.e., also conic quadratic) constraints "the Euclidean norm of something affinely depending on the design variables should be $\leq$ something else, also affinely depending on the design variables".

We could also add to our feasibility problem a meaningful objective, thus coming to "a true" – with a nontrivial objective – conic quadratic problem. We may, e.g., think of the quantity $\sum_{i=1}^N \nu_i$ as of a measure of the power dissipation of robot's actuators and pose the problem of minimizing this objective under the constraints (3.2.5). Another, and perhaps more adequate, measure of power dissipation is $\sqrt{\sum_{i=1}^N \nu_i^2}$. With this objective, we again end up with a conic quadratic problem:

$$t \to \min \mid \ (3.2.5) \ \& \ \parallel \nu \parallel_2 \leq t \qquad \left[\nu = (\nu_1, ..., \nu_N)^T\right]$$

the design variables being $t, \{\nu_i\}_{i=1}^N, \{F^i\}_{i=1}^N$.

As a concluding example of this series, consider the following situation: the robot should hold a cylinder by four fingers, all acting in the vertical direction. The external forces and torques acting at the cylinder are the gravity F_g and an externally applied torque T along the cylinder axis, as shown on the below picture:



Perspective, front and side views

The magnitudes of the contact forces may vary in a given segment $[0, F_{\max}]$. The question is what can be the largest magnitude τ of the external torque T such that a stable grasp still is possible? The problem, mathematically, is

$$
\begin{array}{lrcl}
 & & \max \tau & \\
\text{s.t.} & & & \\
 & \sum_{i=1}^4 (\nu_i u^i + F^i) + F_g & = & 0, \\
 & \sum_{i=1}^4 p^i \times (\nu_i u^i + F^i) + \tau u & = & 0, \\
 & & & [u \text{ is the direction of the cylinder axis}] \\
 & (v^i)^T F_i & = & 0, \ i = 1, ..., 4, \\
 & \| F_i \|_2 & \leq & [\mu [u^i]^T v^i] \nu_i, \ i = 1, ..., 4, \\
 & 0 \leq \nu_i & \leq & F_{\max}, \ i = 1, ..., 4,
\end{array} \qquad \text{(G)}
$$

where the design variables are τ, ν_i, F_i, $i = 1, ..., 4$.

3.3 What can be expressed via conic quadratic constraints?

As it was already mentioned, optimization problems arising in applications normally are not in their "catalogue" forms, and an important skill of everybody interested in applications of Optimization is his/her ability to recognize what in fact is met. Normally, an initial form of an applied optimization model is

$$\min\{f(x) \mid x \in X\}, \tag{3.3.1}$$

where f is the "loss function", and the set X of admissible design vectors typically is given as

$$X = \cap_{i=1}^m X_i; \tag{3.3.2}$$

every X_i is the set of design vectors admissible for a particular design restriction, and their intersection X is the set of designs admissible for all m design restrictions we take into consideration. The sets X_i in many cases – although not always – are given by

$$X_i = \{x \in \mathbf{R}^n \mid g_i(x) \leq 0\}, \tag{3.3.3}$$

$g_i(x)$ being i-th *constraint function*[1]. You may treat $g_i(x)$ as the amount of i-th resource required for design x, so that the constraint $g_i(x) \leq$ const says that the resource should not

[1] Below, speaking about a real-valued function on $\mathbf{R}^n$, we assume that the function is allowed to take real values and the value $+\infty$ and is defined on the entire space. The set of those x where the function takes real values is called the domain Dom f of the function.

exceed a given level; shifting g_i appropriately, we may make this level to be 0, thus coming to the representation (3.3.3).

The objective f in (3.3.1) – (3.3.2) may be non-linear, and one might think that in these cases the problem cannot be posed in conic form, no matter "nice" or not. This conclusion is wrong: we *always* can pass from an optimization problem to an equivalent one with *linear* objective. To this end it suffices to add a new design variable, say, t, and rewrite the problem equivalently as

$$t \to \min \mid (x,t) \in \hat{X} = \{(x,t) \mid f(x) - t \le 0\} \cap \{(x,t) \mid x \in X_1\} \cap ... \cap \{(x,t) \mid x \in X_m\};$$

note that our new objective is linear in the new design variables (x,t).

It makes sense to assume that the indicated transformation is done from the very beginning, so that (3.3.1) – (3.3.2) is of the form

$$c^T x \to \min \mid x \in X = \cap_{i=1}^m X_i. \tag{P}$$

Now the question is what is the "catalogue" form of the set X; in order to know how to recognize this form, one needs a kind of dictionary where different forms of the same structure are listed. Let us build such a dictionary for the conic quadratic programs. Thus, our goal is to understand when a given set X can be represented by *conic quadratic inequalities* (c.q.i.'s) – one or several constraints of the type $\| Dx - d \|_2 \le p^T x - q$. The word "represented" needs clarification, and here it is:

> *We say that a set $X \subset \mathbf{R}^n$ can be represented via conic quadratic inequalities* (for short: is CQr – Conic Quadratic representable), *if there exists vector of additional variables u and a system S of finitely many vector inequalities of the form $A_j \begin{pmatrix} x \\ u \end{pmatrix} - b_j \ge_{\mathbf{L}^{m_j}} 0$ ($x \in \mathbf{R}^n$) such that X is the projection of the solution set of S onto the x-space, i.e., $x \in X$ if and only if one can extend x by a properly chosen u to a solution (x,u) of the system S:*
>
> $$x \in X \Leftrightarrow \exists u : A_j \begin{pmatrix} x \\ u \end{pmatrix} - b_j \ge_{\mathbf{L}^{m_j}} \ge 0, \ j = 1, ..., N.$$
>
> *Every system S of the indicated type is called a conic quadratic representation* (for short: a CQR) *of the set X*[2)]

The meaning of this definition is clarified by the following observation:

> *Consider an optimization problem*
>
> $$c^T x \to \min \mid x \in X.$$
>
> *and assume that X is CQr. Then the problem is equivalent to a conic quadratic program. The latter program can be written down explicitly, provided that we are given a* CQr *of X.*

[2)] Note that here we do not impose on the representing system of conic quadratic inequalities S the requirement to satisfy assumption **A**; e.g., the entire space is CQr – it is a solution set of the "system" $|0^T x| \le 1$ comprised of a single conic quadratic inequality.

Indeed, let S be a CQr of X, and u be the corresponding vector of additional variables. The problem

$$c^T x \to \min \mid (x,u) \text{ satisfy } S$$

with design variables x, u is equivalent to the original problem (P), on one hand, and is a conic quadratic program, on the other hand.

Let us call a problem of the form (P) with CQ-representable X a *good* problem.

The question we are interested in is how to recognize good problems, i.e., how to recognize CQ-representable sets. Well, how we recognize continuity of a given complicated function? Normally not by a straightforward verification of the definition of continuity. We use two types of tools:

A We know a number of simple functions – a constant, $f(x) = x$, $f(x) = \exp\{x\}$, etc. – which indeed are continuous: we have verified it directly, by demonstrating that the functions fit the definition of continuity;

B We know a number of basic continuity-preserving operations, like taking products, sums, etc.

And when we see that a function is obtained from simple functions of the type A by a number of operations of the type B, we immediately conclude that the function is continuous.

The outlined approach is very typical for Mathematics, and this exactly the approach we are about to follow. In fact it makes sense to ask two kinds of questions:

(?) What are CQ-representable sets

(??) What are CQ-representable functions $g(x)$, i.e., functions which possess CQ-representable *epigraphs*

$$\mathrm{Epi}\{f\} = \{(x,t) \in \mathbf{R}^n \times \mathbf{R} \mid g(x) \le t\}.$$

Our interest in the second question is motivated by the fact that CQ-representability of a function g automatically implies CQ-representability of all its level sets – the sets of the form $\{x \mid g(x) \le \text{const}\}$:

Observation: *If a function g is CQ-representable, then all it level sets $\{x \mid g(x) \le a\}$ are so, and every CQ-representation of (the epigraph of) g explicitly induces CQ-representations of the level sets.*

Indeed, assume that we have a CQ-representation of the epigraph of g:

$$g(x) \le t \Leftrightarrow \exists u : \| \alpha_j(x,t,u) \|_2 \le \beta_j(x,t,u),\ j = 1,...,N,$$

where α_j and β_j are, respectively, vector-valued and scalar *affine* functions of their arguments. In order to get from this representation a CQ-representation of a level set $\{x \mid g(x) \le a\}$, it suffices to fix, at the value a, the variable t in the conic quadratic inequalities $\| \alpha_j(x,t,u) \|_2 \le \beta_j(x,t,u)$.

Now let us look what are our "raw materials" – simple functions and sets admitting CQR's. Here they are:

Elementary CQ-representable functions/sets

1. A constant function $g(x) \equiv a$

Indeed, the epigraph of the function $\{(x,t) \mid a \leq t\}$ is given by a linear inequality, and a linear inequality $0 \leq p^T z - q$ is at the same time conic quadratic inequality $\| 0 \|_2 \leq p^T z - q$.

2. A linear function $g(x) = a^T x$

Indeed, the epigraph of a linear function is given by a linear inequality.

3. The Euclidean norm $g(x) = \| x \|_2$

Indeed, the epigraph of g is given by the conic quadratic inequality $\| x \|_2 \leq t$ in variables x, t.

4. The squared Euclidean norm $g(x) = x^T x$.

Indeed, $t = \frac{(t+1)^2}{4} - \frac{(t-1)^2}{4}$, so that

$$x^T x \leq t \Leftrightarrow x^T x + \frac{(t-1)^2}{4} \leq \frac{(t+1)^2}{4} \Leftrightarrow \left\| \begin{pmatrix} x \\ \frac{t-1}{2} \end{pmatrix} \right\|_2 \leq \frac{t+1}{2}$$

(check the second $\Leftrightarrow$!), and the concluding relation is a conic quadratic inequality.

5. The fractional-quadratic function $g(x,s) = \begin{cases} \frac{x^T x}{s}, & s > 0 \\ 0, & s = 0, x = 0 \\ +\infty, & \text{in all remaining cases} \end{cases}$ (x vector, s scalar).

Indeed, with the convention that $(x^T x)/0$ is 0 or $+\infty$, depending on whether $x = 0$ or not, and taking into account that $ts = \frac{(t+s)^2}{4} - \frac{(t-s)^2}{4}$, we have:

$$\{\tfrac{x^T x}{s} \leq t, s \geq 0\} \Leftrightarrow \{x^T x \leq ts, t \geq 0, s \geq 0\} \Leftrightarrow \{x^T x + \tfrac{(t-s)^2}{4} \leq \tfrac{(t+s)^2}{4}, t \geq 0, s \geq 0\}$$
$$\Leftrightarrow \left\| \begin{pmatrix} x \\ \frac{t-s}{2} \end{pmatrix} \right\|_2 \leq \tfrac{t+s}{2}$$

(check the third $\Leftrightarrow$!), and the concluding relation is a conic quadratic inequality.

The level sets of the indicated CQr functions provide us with a spectrum of "elementary" CQr sets. It makes sense to add to this spectrum a set more:

6. (A branch of) Hyperbola $\{(t,s) \in \mathbf{R}^2 \mid ts \geq 1, t > 0\}$.

Indeed,

$$\{ts \geq 1, t > 0\} \Leftrightarrow \{\tfrac{(t+s)^2}{4} \geq 1 + \tfrac{(t-s)^2}{4}\} \Leftrightarrow \{\left\| \begin{pmatrix} \frac{t-s}{2} \\ 1 \end{pmatrix} \right\|_2^2 \leq \tfrac{(t+s)^2}{4}\}$$
$$\Leftrightarrow \{\left\| \begin{pmatrix} \frac{t-s}{2} \\ 1 \end{pmatrix} \right\|_2 \leq \tfrac{t+s}{2}\}$$

(check the last $\Leftrightarrow$!), and we see that a hyperbola is given by a conic quadratic inequality.

Now let us look what are simple operations preserving CQ-representability of functions/sets.

Operations preserving CQ-representability of sets

A. Intersection: if sets $X_i \subset \mathbf{R}^n$, $i = 1, ..., N$, are CQr, so is their intersection $X = \cap_{i=1}^N X_i$. Indeed, let S_i be CQ-representation of X_i, and u_i be the corresponding vector of additional variables. Then the system S of constraints of the variables $(x, u_1, ..., u_N)$:

$$\{(x, u_i) \text{ satisfies } S_i\}, \ i = 1, ..., N$$

is a system of conic quadratic inequalities, and this system clearly is a CQ-representation of X.

Corollary 3.3.1 *If every one of the sets X_i in problem* (P) *is CQr, the problem is good – it can be rewritten in the form of a conic quadratic problem, and such a transformation is readily given by CQR's of the sets X_i, $i = 1, ..., m$.*

Corollary 3.3.2 *Adding to a good problem (finitely many) CQr constraints $x \in X_i$, (e.g., finitely many scalar linear inequalities), we again get a good problem.*

B. Direct product: If sets $X_i \subset \mathbf{R}^{n_i}$, $i = 1, ..., k$, are CQr, then so is their direct product $X_1 \times ... \times X_k$.
Indeed, if $S_i = \{\alpha^i_j(x_i, u_i) \|_2 \leq \beta^i_j(x_i, u_i\}_{j=1}^{N_j}$, $i = 1, ..., k$, are CQR's of the sets X_i, then the union over i of this system of inequalities, regarded as a system with design variables $x = (x_1, ..., x_k)$ and additional variables $u = (u_1, ..., u_k)$ is a CQR for the direct product of $X_1, ..., X_k$.

C. Affine image ("Projection"): Assume that the set $X \subset \mathbf{R}^n$ is CQr, and that $x \mapsto y = Ax + b$ is an affine mapping of $\mathbf{R}^n$ onto $\mathbf{R}^k$. Then the image $X^{\rightarrow}$ of the set X under the mapping is CQr.
Indeed, passing to an appropriate basis in $\mathbf{R}^n$, we may assume that the kernel $\mathrm{Ker}\, A$ of A is comprised of the last $n-k$ vectors of the basis; in other words, we may assume that $x \in \mathbf{R}^n$ can be partitioned as $x = (v, w)$ (v is k-, and w is $(n-k)$-dimensional) in such a way that for $x = (u, v)$ one has $Ax = Qv$ with nonsingular $k \times k$ matrix Q; thus,

$$y = Ax + b \Leftrightarrow x = (Q^{-1}(y-b), w) \text{ for some } w.$$

Now let $S = \{\| \alpha_j(x, u) \|_2 \leq \beta_j(x, u)\}_{j=1}^N$ be CQ-representation of X, u being the corresponding vector of design variables and α_j, β_j being affine in (x, u). Then the system of c.q.i.'s in the variables $y \in \mathbf{R}^k$, $w \in \mathbf{R}^{n-k}$, u:

$$S^+ = \{\| \alpha_j((Q^{-1}(y-b), w), u) \|_2 \leq \beta_j((Q^{-1}(y-b), w), u)\}_{j=1}^N,$$

the vector of additional variables being (w, u), the is a CQR of $X^{\rightarrow}$. Indeed, $y \in X^{\rightarrow}$ if and only if there exists $w \in \mathbf{R}^{n-k}$ such that the point $x = (Q^{-1}(y-b), w)$ belongs to X, and the latter happens if and only if there exist u such that the point $(x, u) = ((Q^{-1}(y-b), w), u)$ solves S.

D. Inverse affine image: Let $X \subset \mathbf{R}^n$ be a CQr set, and let $x = Ay + b$ be an affine mapping from $\mathbf{R}^k$ to $\mathbf{R}^n$. Then the inverse image $X^{\leftarrow} = \{y \in \mathbf{R}^k \mid Ay + b \in X\}$ also is CQr.
Indeed, let $S = \{\| \alpha_j(x, u) \|_2 \leq \beta_j(x, u)\}_{i=1}^N$ be a CQR for X. Then the system of c.q.i.'s $S = \{\| \alpha_j(Ay + b, u) \|_2 \leq \beta_j(Ay + b, u)\}_{i=1}^N$ with variables y, u clearly is a CQR for $X^{\leftarrow}$.

Corollary 3.3.3 *Consider a good problem* (P) *and assume that we restrict its design variables to be given affine functions of a new design vector y. Then the induced problem with the design vector y also is good.*

In particular, adding to a good problem arbitrarily many linear equality constraints, we end up with a good problem[3)]

It should be stressed that the above statements are not just existence theorems – they are "algorithmic": given CQR's of the "operands" (say, m sets $X_1, ..., X_m$), we may *completely mechanically* build a CQR for the "result of the operation" (e.g., for the intersection $\bigcap_{i=1}^m X_i$).

Note that we have already used nearly all our Corollaries in the Grasp problem. E.g., to see that (G) is a conic quadratic problem, we in fact have carried out the following reasoning:

1. The problem
$$\min \tau \mid \| F^i \|_2 \leq s_i, \ i = 1, ..., N \qquad (\mathrm{P}_0)$$
with design variables τ, $(F^i, s_i) \in \mathbf{R}^3 \times \mathbf{R}$, $\nu_i \in \mathbf{R}$ is perhaps absolutely crazy (part of the variables does not appear at all, the objective and the constraints are not connected, etc.), but clearly is good;

[3)] Indeed, we may use the linear equations to express affinely the original design variables via part of them, let this part be y; the problem with added linear constraints can now be posed as a problem with design vector y, and this is exactly the transformation discussed in the "general" part of the Corollary.

2. Adding to the good problem (P$_0$) linear equality constraints

$$\begin{array}{rcl} \sum_{i=1}^N(\nu_i u^i + F^i) & = & -F_g \\ \sum_{i=1}^N p^i \times (\nu^i u^i + F^i) + \tau u & = & 0 \\ (v^i)^T F^i & = & 0,\ i = 1, ..., N \\ s_i - [\mu (u^i)^T v^i]\nu_i & = & 0,\ i = 1, ..., N \end{array}$$

   where u^i, u, F_g are given vectors, we get a good problem (P$_1$) (Corollary 3.3.3);

3. The problem of interest (G) is obtained from the good problem (P$_1$) by adding scalar linear inequalities

$$0 \le \nu_i \le F_{\max},\ i = 1, ..., N,$$

   so that (G) itself is good (Corollary 3.3.2).

Operations preserving CQ-representability of functions

Recall that a function $g(x)$ is called CQ-representable, if its epigraph $\{(x,t) \mid g(x) \le t\}$ is a CQ-representable set; a CQR of the epigraph of g is called conic quadratic representation of g. Recall also that a level set of a CQr function is CQ-representable. Here are transformations preserving CQ-representability of functions:

E. Taking maximum: if functions $g_i(x)$, $i = 1, ..., m$, are CQr, then so is their maximum $g(x) = \max_{i=1,...,m} g_i(x)$.

Indeed, the epigraph of the maximum is just intersection of the epigraphs of the operands, and an intersection of CQr sets again is CQr.

F. Summation with nonnegative weights: if functions $g_i(x)$, $x \in \mathbf{R}^n$, are CQr, $i = 1, ..., m$, and α_i are nonnegative weights, then the function $g(x) = \sum_{i=1}^m \alpha_i g_i(x)$ also is CQr.

Indeed, consider the set

$$\Pi = \{(x_1, t_1; x_2, t_2; ...; x_m, t_m; t) \mid x_i \in \mathbf{R}^n, t_i \in \mathbf{R}, t \in \mathbf{R}, g_i(x_i) \le t_i, i = 1, ..., m; \sum_{i=1}^m \alpha_i t_i \le t\}.$$

The set is CQr. Indeed, the set is the direct product of the epigraphs of g_i intersected with the half-space given by the linear inequality $\sum_{i=1}^m \alpha_i t_i \le t$. Now, a direct product of CQr sets also is CQr, a half-space is CQr (it is a level set of an affine function, and such a function is CQr), and the intersection of CQr sets also is CQr. Since Π is CQr, so is its projection on subspace of variables $x_1, x_2, ..., x_m, t$, i.e., the set

$$\{(x_1, ..., x_m, t) : \exists t_1, ..., t_m : g_i(x_i) \le t_i, i = 1, ..., m, \sum_{i=1}^m \alpha_i t_i \le t\} = \{(x_1, ..., x_m, t) : \sum_{i=1}^m \alpha_i g_i(x) \le t\}.$$

Since the latter set is CQr, so is its inverse image under the mapping

$$(x, t) \mapsto (x, x, ...x, t),$$

and this inverse image is exactly the epigraph of g.

G. Direct summation: If functions $g_i(x_i)$, $x_i \in \mathbf{R}^{n_i}$, $i = 1, ..., m$, are CQr, so is their direct sum

$$g(x_1, ..., x_m) = g_1(x_1) + ... + g_m(x_m).$$

Indeed, the functions $\hat{g}_i(x_1, ..., x_m) = g_i(x_i)$ clearly are CQr – their epigraphs are inverse images of the epigraphs of g_i under affine mappings $(x_1, ..., x_m, t) \mapsto (x_i, t)$. It remains to note that g is the sum of $\hat{g}_i$.

H. Affine substitution of argument: If a function $g(x)$, $x \in \mathbf{R}^n$, is CQr and $y \mapsto Ay + b$ is an affine mapping from $\mathbf{R}^k$ to $\mathbf{R}^n$, then the superposition $g^{\rightarrow}(y) = g(Ay + b)$ is CQr.

Indeed, the epigraph of $g^{\rightarrow}$ is the inverse image of the epigraph of g under the affine mapping $(y,t) \mapsto (Ay+b,t)$.

I. Partial minimization: Let $g(x)$ be CQr. Assume that x is partitioned into two sub-vectors: $x=(v,w)$, and let $\hat{g}$ be obtained from g by partial minimization in w:

$$\hat{g}(u) = \inf_{w} g(u,v),$$

and assume that for every v the minimum in w is achieved. Then $\hat{g}$ is CQr.
Indeed, under the assumption that the minimum in w always is achieved, the epigraph of $\hat{g}$ is the image of the epigraph of g under the projection $(v,w,t) \mapsto (v,t)$.

More operations preserving CQ-representability

Let us list a number of more "advanced" operations with sets/functions preserving CQ-representability.

J. Arithmetic summation of sets. Let X_i, $i=1,...,k$, be nonempty convex sets in $\mathbf{R}^n$. Their arithmetic sum $X_1+X_2+...+X_k$ is, by definition, the set of all k-term sums, the first term being from X_1, the second - from X_2, and so on:

$$X_1+...+X_k = \{x = x^1+...+x^k \mid x^i \in X_i,\ i=1,...,k\}.$$

We claim that

If all X_i are CQr, so is their sum.

Indeed, the direct product

$$X = X_1 \times X_2 \times ... \times X_k \subset \mathbf{R}^{nk}$$

is CQr by B.; it remains to note that $X_1+...+X_k$ is the image of X under the linear mapping

$$(x^1,...,x^k) \mapsto x^1+...+x^k : \mathbf{R}^{nk} \to \mathbf{R}^n,$$

and that the image of a CQr set under affine mapping also is CQr (see C.)

J.1. inf-convolution. The operation with functions related to the arithmetic summation of sets is the inf-convolution defined as follows. Let $f_i : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$, $i=1,...,n$, be functions. Their inf-convolution is the function

$$f(x) = \inf\{f_1(x^1)+...+f_k(x^k) \mid x^1+...+x^k = x\}. \qquad (*)$$

We claim that

If all f_i are CQr, their inf*-convolution is $> -\infty$ everywhere and for every x for which the* inf *in the right hand side of (*) is finite, this infinum is achieved, then f is CQr.*

Indeed, under the assumption in question the epigraph of f, as it is immediately seen, is the arithmetic sum of the epigraphs of $f_1,...,f_k$.

K. Taking conic hull of a closed convex set. Let $X \in \mathbf{R}^n$ be a nonempty convex set. Its *conic hull* is the set

$$X^+ = \{(x,t) \in \mathbf{R}^n \times \mathbf{R} : t > 0, t^{-1}x \in X\} \cup \{0\}.$$

Geometrically: we add to the coordinates of vectors from $\mathbf{R}^n$ a new coordinate equal to 1:

$$(x_1, ..., x_n)^T \mapsto (x_1, ..., x_n, 1)^T,$$

thus getting an affine embedding of $\mathbf{R}^n$ in $\mathbf{R}^{n+1}$. We take the image of X under this mapping – "lift" X by one along the $(n+1)$st axis – and them form the set X^+ comprised of all rays starting at the origin and passing through the points of the "lifted" X.

The conic hull of a closed convex set X is not necessarily closed; it indeed is the case if and only if X is not only closed, but is bounded as well. The <u>closed</u> convex hull of X is the closure of its conic hull:

$$\widehat{X}^+ = \mathrm{cl}\, X^+ = \left\{(x,t) \in \mathbf{R}^n \times \mathbf{R} : \exists \{(x_i,t_i)\}_{i=1}^{\infty} : t_i > 0, t_i^{-1}x_i \in X, t = \lim_i t_i, x = \lim_i x_i\right\}.$$

Note that if X is a closed convex set, then the parts of the conic hull X^+ of X and the closed convex hull $\widehat{X}^+$ belonging to the open half-space $\{t > 0\}$ are equal to each other (check!). Note also that if X is a closed convex set, you can obtain it from its (closed) convex hull by taking intersection with the hyperplane $\{t = 1\}$:

$$x \in X \Leftrightarrow (x,1) \in \widehat{X}^+ \Leftrightarrow (x,1) \in X^+.$$

We claim that

If a closed convex set X is CQr:

$$X = \{x \mid \exists u : Ax + Bu + b \geq_{\mathbf{K}} 0\}, \qquad (3.3.4)$$

$\mathbf{K}$ *being a direct product of Lorentz cones, then the CQr set*

$$\widetilde{X}^+ = \{(x,t) \mid \exists u : Ax + Bu + tb \geq_{\mathbf{K}} 0\} \qquad (3.3.5)$$

is "between" the conic hull X^+ and the closed conic hull $\widehat{X}^+$ of X:

$$X^+ \subset \widetilde{X}^+ \subset \widehat{X}^+.$$

In particular, if X is closed and bounded CQr set (so that $X^+ = \widehat{X}^+$), the conic hull of X is CQr.

If the CQR (3.3.4) is such that $Bu \in \mathbf{K}$ implies that $Bu = 0$, then $\widetilde{X}^+ = \widehat{X}^+$, so that $\widehat{X}^+$ is CQr.

We should prove that the set $\widetilde{X}^+$ (which by construction is CQr) is between X^+ and $\widehat{X}^+$. Indeed, $0 \in \widetilde{X}$, and if (x,t) is a pair with $t > 0$, $z = t^{-1}x \in X$, then there exists u such that

$$Az + Bu + b \geq_{\mathbf{K}} 0 \Rightarrow Ax + B(tu) + tb \geq_{\mathbf{K}} 0 \Rightarrow (x,t) \in \widetilde{X}^+.$$

Thus, $X^+ \subset \widetilde{X}^+$. Now let us prove that $\widetilde{X}^+ \subset \widehat{X}^+$. Let us choose somehow a point $\bar{x} \in X$, so that for properly chosen $\bar{u}$ it holds

$$A\bar{x} + B\bar{u} + b \geq_{\mathbf{K}} 0,$$

i.e., $(\bar{x}, 1) \in \widetilde{X}^+$. Since $\widetilde{X}^+$ is convex (as every CQr set), we conclude that whenever (x,t) belongs to $\widetilde{X}^+$, every pair $(x_\epsilon = x + \epsilon\bar{x}, t_\epsilon = t + \epsilon)$ with $\epsilon > 0$ also belongs to the same set:

$$\exists u = u_\epsilon : Ax_\epsilon + Bu_\epsilon + t_\epsilon b \geq_{\mathbf{K}} 0.$$

It follows that $t_\epsilon^{-1}x_\epsilon \in X$, whence $(x_\epsilon, t_\epsilon) \in X^+ \subset \widehat{X}^+$. As $\epsilon \to +0$, we have $(x_\epsilon, t_\epsilon) \to (x,t)$, and since $\widehat{X}^+$ is closed, we get $(x,t) \in \widehat{X}^+$. Thus, $\widetilde{X}^+ \subset \widehat{X}^+$.

Now assume that $Bu \in \mathbf{K}$ only if $Bu = 0$, and let us prove that $\widetilde{X}^+ = \widehat{X}^+$. All we should prove is that $\widetilde{X}^+$ is closed. Thus, assume that $(x_i, t_i) \in \widetilde{X}^+$ and $(x_i, t_i) \to (x,t)$, and let us prove that $(x,t) \in \widetilde{X}^+$. There is nothing to prove if $t > 0$, since in this case $t_i > 0$ for all large enough values of i, and for those i

$$(x_i, t_i) \in \widetilde{X}^+ \Rightarrow \exists u_i : Ax_i + Bu_i + t_i b \in \mathbf{K} \Rightarrow A(t_i^{-1}x_i) + B(t_i^{-1}u_i) + b \in \mathbf{K} \Rightarrow t_i^{-1}x_i \in X.$$

Since $x_i \to x$, $t_i \to t > 0$, we have $t_i^{-1}x_i \to t^{-1}x$; since X is closed and contains all points $t_i^{-1}x_i$, we get $t^{-1}x \in X$, so that $(x,t) \in X^+ \subset \widetilde{X}^+$.

Now assume that $t = 0$. Since $(x_i, t_i) \in \widetilde{X}^+$, there exists u_i such that

$$Ax_i + Bu_i + t_i b \in \mathbf{K}. \tag{3.3.6}$$

If the sequence $\{Bu_i\}$ is bounded, we can, passing to a subsequence, assume that $Bu_i \to Bu$, thus coming to

$$Ax + Bu \in \mathbf{K},$$

and, consequently, to $(x,t) = (x,0) \in \widetilde{X}^+$. To complete the proof, it suffices to demonstrate that the sequence $\{Bu_i\}$ is bounded. Indeed, let us set $v_i = \| Bu_i \|_2^{-1} u_i$, so that $\| Bv_i \|_2 = 1$; passing to a subsequence, we can assume that $Bv_i \to Bv \neq 0$. From (3.3.6) it follows that

$$Bv = \lim_i \left[Bv_i + \| Bu_i \|_2^{-1} \left[Bx_i + t_i b \right] \right] \in \mathbf{K}$$

(recall that $x_i \to x, t_i \to 0$). Thus, there exists v such that $Bv \neq 0$ and $Bv \in \mathbf{K}$, which is a contradiction.

K.1. "Projective transformation" of a CQr function. The operation with functions related to taking conic hull of a convex set is the "projective transformation" which converts a function $f(x) : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ [4] into the function

$$f^+(x,s) = sf(x/s) : \{s > 0\} \times \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}.$$

The epigraph of f^+ is the conic hull of the epigraph of f with excluded origin:

$$\begin{array}{rcl} \{(x,s,t) \mid s > 0, t \geq f^+(x,s)\} & = & \{(x,s,t) \mid s > 0, s^{-1}t \geq f(s^{-1}x)\} \\ & = & \left\{(x,s,t) \mid s > 0, s^{-1}(x,t) \in \mathrm{Epi}\{f\}\right\}. \end{array}$$

The closure $\mathrm{cl}\,\mathrm{Epi}\{f^+\}$ is the epigraph of certain function, let it be denoted $\widehat{f}^+(x,s)$; this function is called the *projective transformation* of f. E.g., the fractional-quadratic function from Example 5 is the projective transformation of the function $f(x) = x^Tx$. Note that the function $\widehat{f}^+(x,s)$ not necessarily coincides with $f^+(x,s)$ even in the open half-space $s > 0$; this is the case if and only if the epigraph of f is closed (or, which is the same, f is lower semicontinuous: whenever $x_i \to x$ and $f(x_i) \to a$, we have $f(x) \leq a$). We are about to demonstrate that the projective transformation "nearly preserves" CQ-representability:

Let $f : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ be a lower semicontinuous function which is CQr:

$$\{(x,t) \mid t \geq f(x)\} = \{(t,x) \mid \exists u : Ax + tp + Bu + b \geq_{\mathbf{K}} 0\}, \tag{3.3.7}$$

[4] Recall that "a function" for us means a proper function – one which takes a finite value at least at one point

K *being a direct product of Lorentz cones. Assume that the CQR is such that* $Bu \geq_{\mathbf{K}} 0$ *implies that* $Bu = 0$. *Then the projective transformation* $\widehat{f}^{+}$ *of* f *is CQr, namely,*

$$\mathrm{Epi}\{\widehat{f}^{+}\} = \{(x,t,s) \mid s \geq 0, \exists u : Ax + tp + Bu + sb \geq_{\mathbf{K}} 0\}.$$

Indeed, let us set

$$G = \{(x,t,s) \mid \exists u : s \geq 0, Ax + tp + Bu + sb \geq_{\mathbf{K}} 0\}.$$

As we remember from the previous combination rule, G is exactly the closed conic hull of the epigraph of f, i.e., is the epigraph of $\widehat{f}^{+}$.

L. The polar of a convex set. Let $X \subset \mathbf{R}^n$ be a convex set containing the origin. The *polar* of X is the set

$$X_* = \left\{y \in \mathbf{R}^n \mid y^T x \leq 1 \; \forall x \in X\right\}.$$

E.g.,

- the polar of the singleton $\{0\}$ is the entire space;
- the polar of the entire space is the singleton $\{0\}$;
- the polar of a linear subspace is its orthogonal complement (why?);
- the polar of a closed convex pointed cone with a nonempty interior is *minus* the dual cone (why?).

It is worthy of mentioning that the polarity is "symmetric": if X is a closed convex set containing the origin, then so is X_*, and twice taken polar is the original set: $(X_*)_* = X$.

We are about to prove that the polarity $X \mapsto X_*$ "nearly preserves" CQ-representability:

Let $X \subset \mathbf{R}^n$, $0 \in X$, *be a CQr set:*

$$X = \{x \mid \exists u : Ax + Bu + b \geq_{\mathbf{K}} 0\}, \tag{3.3.8}$$

K *being a direct product of Lorentz cones.*

Assume that there exists $\bar{x}, \bar{u}$ *such that*

$$A\bar{x} + B\bar{u} + b >_{\mathbf{K}} 0.$$

Then the polar of X *is the CQr set*

$$X_* = \left\{y \mid \exists \xi : A^T\xi + y = 0, B^T\xi = 0, b^T\xi \leq 1, \xi \geq_{\mathbf{K}} 0\right\} \tag{3.3.9}$$

Indeed, consider the following conic quadratic problem:

$$-y^T x \to \min \mid Ax + Bu + b \geq_{\mathbf{K}} 0 \tag{P$_y$}$$

the design variables being x, u. A vector y belongs to X_* if and only if (P_y) is below bounded, and its optimal value is at least -1. Since (P_y) is strictly feasible, by Conic Duality Theorem this is the case if and only if the dual problem

$$-b^T\xi \to \max \mid A^T\xi = -y, B^T\xi = 0, \xi \geq_{\mathbf{K}} 0$$

(recall that $\mathbf{K}$ is self-dual) has a feasible solution with the value of the dual objective at least -1. Thus,

$$X_* = \left\{y \mid \exists \xi : A^T\xi + y = 0, B^T\xi = 0, b^T\xi \leq 1, \xi \geq_{\mathbf{K}} 0\right\},$$

as claimed in (3.3.9). It remains to note that X_* is obtained from the CQr set $\mathbf{K}$ by operations preserving CQ-representability: intersection with the CQr set $\{\xi \mid B^T\xi = 0, b^T\xi \leq 1\}$ and subsequent affine mapping $\xi \mapsto -A^T\xi$.

L.1. The Legendre transformation of a CQr function. The operation with functions related to taking polar of a convex set is the *Legendre transformation*. The Legendre transformation of a function $f(x) : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ is the function

$$f_*(y) = \sup_x \left[y^T x - f(x)\right].$$

E.g.,

- the Legendre transformation of a constant $f(x) \equiv c$ is the function

$$f_*(y) = \begin{cases} -c, & y = 0 \\ +\infty, & y \neq 0 \end{cases};$$

- the Legendre transformation of an affine function $f(x) \equiv a^T x + b$ is the function

$$f_*(y) = \begin{cases} -b, & y = a \\ +\infty, & y \neq a \end{cases};$$

- the Legendre transformation of a convex quadratic form $f(x) \equiv \frac{1}{2}x^T D^T D x + b^T x + c$ with rectangular D such that $\mathrm{Ker}\, D^T = \{0\}$ is the function

$$f_*(y) = \begin{cases} \frac{1}{2}(y-b)^T D^T (DD^T)^{-2} D(y-b) - c, & y - b \in \mathrm{Im} D^T \\ +\infty, & \text{otherwise} \end{cases};$$

It is worthy of mentioning that the Legendre transformation is symmetric: if f is a proper convex lower semicontinuous funcion (i.e., a function with nonempty closed convex epigraph), then so is f_*, and twice take, the Legendre transformation recovers the original function: $(f_*)_* = f$.

We are about to prove that the Legendre transformation "nearly preserves" CQ-representability:

Let $f : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ be CQr:

$$\{(x,t) \mid t \geq f(x)\} = \{(t,x) \mid \exists u : Ax + tp + Bu + b \geq_{\mathbf{K}} 0\},$$

$\mathbf{K}$ *being a direct product of Lorentz cones. Assume that there exist $\bar{x}, \bar{t}, \bar{u}$ such that*

$$A\bar{x} + \bar{t}p + B\bar{u} + b >_{\mathbf{K}} 0.$$

Then the Legendre transformation of f is the CQr set

$$\mathrm{Epi}\{f_*\} = \left\{(y,s) \mid \exists \xi : A^T\xi = -y, B^T\xi = 0, p^T\xi = 1, s \geq b^T\xi, \xi \geq_{\mathbf{K}} 0\right\}. \quad (3.3.10)$$

Indeed, we have

$$\mathrm{Epi}\{f_*\} = \left\{(y,s) \mid y^T x - f(x) \le s \ \forall x\right\} = \left\{(y,s) \mid y^T x - t \le s \ \forall (x,t) \in \mathrm{Epi}\{f\}\right\}. \tag{3.3.11}$$

Consider the conic quadratic program

$$-y^T x + t \to \min \mid Ax + tp + Bu + b \ge \mathbf{K}_0 \tag{P$_y$}$$

with design variables x, t, u. By (3.3.11), a pair (y,s) belongs to the epigraph of f_* if and only if (P_y) is below bounded with optimal value $\ge -s$. Since (P_y) is strictly feasible, this is the case if and only if the dual problem

$$-b^T\xi \to \max \mid A^T\xi = -y, B^T\xi = 0, p^T\xi = 1, \xi \ge_{\mathbf{K}} 0$$

has a feasible solution with the value of the dual objective $\ge -s$. Thus,

$$\mathrm{Epi}\{f_*\} = \left\{(y,s) \mid \exists \xi : A^T\xi = -y, B^T\xi = 0, p^T\xi = 1, s \ge b^T\xi, \xi \ge_{\mathbf{K}} 0\right\}$$

as claimed in (3.3.10). It remains to note that the right hand side set in (3.3.10) is CQr (as a set obtained from the CQr set $\mathbf{K} \times \mathbf{R}_s$ by operations preserving CQ-representability – intersection with the set $\{\xi \mid B^T\xi = 0, p^T\xi = 1, b^T\xi \le s\}$ and subsequent affine mapping $\xi \mapsto -A^T\xi$).

M. Taking convex hull of a number of sets. Let $Y \subset \mathbf{R}^n$ be a set. Its convex hull is the smallest convex set which contains Y:

$$\mathrm{Conv}(Y) = \left\{ x = \sum_{i=1}^{k_x} \alpha_i x_i \mid x_i \in x, \alpha_i \ge 0, \sum_i \alpha_i = 1 \right\}$$

The closed convex hull $\overline{\mathrm{Conv}}(Y) = \mathrm{cl}\,\mathrm{Conv}(Y)$ of Y is the smallest *closed* convex set containing Y.

Following Yu. Nesterov, let us prove that taking convex hull "nearly preserves" CQ-representability:

Let $X_1, ..., X_k \subset \mathbf{R}^n$ be closed convex CQr sets:

$$X_i = \{x \mid A_i x + B_i u_i + b_i \ge_{\mathbf{K}_i} 0,\ i = 1, ..., k\}, \tag{3.3.12}$$

$\mathbf{K}_i$ being direct product of Lorentz cones.

Then the CQr set

$$\begin{array}{rcl} Y & = & \{x \mid \exists \xi^1, ..., \xi^k, t_1, ..., t_k, \eta^1, ..., \eta^k : \\ & & \begin{array}{rcl} \left[\begin{array}{c} A_1 x^1 + B_1\eta^1 + t_1 b_1 \\ A_2 x^2 + B_2\eta^2 + t_2 b_2 \\ \ldots \\ A_k x^k + B_k\eta^k + t_k b_k \end{array}\right] & \ge_{\mathbf{K}} & 0, \\ t_1, ..., t_k & \ge & 0, \\ \xi^1 + ... + \xi^k & = & x \\ t_1 + ... + t_k & = & 1\}, \end{array} \\ \mathbf{K} & = & \mathbf{K}_1 \times ... \times \mathbf{K}_k \end{array} \tag{3.3.13}$$

is between the convex hull and the closed convex hull of the set $X_1 \cup ... \cup X_k$*:*

$$\mathrm{Conv}(\bigcup_{i=1}^k X_i) \subset X \subset \overline{\mathrm{Conv}}(\bigcup_{i=1}^k X_i).$$

In particular, if, in addition to CQ-representability,

(i) *all* X_i *are bounded,*

or

(ii) $X_i = Z_i + W$*, where* Z_i *are closed and bounded sets and* W *is a convex closed set,*

then

$$\mathrm{Conv}(\bigcup_{i=1}^k X_i) = Y = \overline{\mathrm{Conv}}(\bigcup_{i=1}^k X_i)$$

is CQr.

First, the set Y clearly contains $\mathrm{Conv}(\bigcup\limits_{i=1}^k X_i)$. Indeed, since the sets X_i are convex, the convex hull of their union is

$$\left\{ x = \sum_{i=1}^k t_i x^i \mid x^i \in X_i, t_i \geq 0, \sum_{i=1}^k t_i = 1 \right\}$$

(why?); for a point

$$x = \sum_{i=1}^k t_i x^i \qquad \left[x^i \in X_i, t_i \geq 0, \sum_{i=1}^k t_i = 1 \right],$$

there exist u^i, $i = 1, ..., k$, such that

$$A_i x^i + B_i u^i + b_i \geq_{\mathbf{K}_i} 0.$$

We get

$$\begin{array}{rcl} x & = & (t_1 x^1) + ... + (t_k x^k) \\ & = & \xi^1 + ... + \xi^k, \\ & & [\xi^i = t_i x^i]; \\ t_1, ..., t_k & \geq & 0; \\ t_1 + ... + t_k & = & 1; \\ A_i \xi^i + B_i \eta^i + t_i b_i & \geq_{\mathbf{K}_i} & 0, \ i = 1, ..., k, \\ & & [\eta^i = t_i u^i], \end{array} \tag{3.3.14}$$

so that $x \in Y$ (see the definition of Y).

To complete the proof that Y is between the convex hull and the closed convex hull of $\bigcup\limits_{i=1}^k X_i$, it remains to verify that if $x \in Y$, then x is contained in the closed convex hull of $\bigcup\limits_{i=1}^k X_i$. Let us somehow choose $\bar{x}^i \in X_i$; for properly chosen $\bar{u}^i$ we have

$$A_i \bar{x}^i + B_i \bar{u}^i + b_i \geq_{\mathbf{K}_i} 0, \ \ i = 1, ..., k. \tag{3.3.15}$$

Since $x \in Y$, there exist t_i, ξ^i, η^i satisfying the relations

$$\begin{array}{rcl} x & = & \xi^1 + ... + \xi^k, \\ t_1, ..., t_k & \geq & 0, \\ t_1 + ... + t_k & = & 1, \\ A_i \xi^i + B_i \eta^i + t_i b_i & \geq_{\mathbf{K}_i} & 0, \ \ i = 1, ..., k. \end{array} \tag{3.3.16}$$

In view of the latter relations and (3.3.15), we have for $0 < \epsilon < 1$:

$$A_i[(1-\epsilon)\xi^i + \epsilon k^{-1}\bar{x}^i] + B_i[(1-\epsilon)\eta^i + \epsilon k^{-1}\bar{u}^i] + [(1-\epsilon)t_i + \epsilon k^{-1}]b_i \geq_{\mathbf{K}_i} 0;$$

setting

$$\begin{array}{rcl} t_{i,\epsilon} & = & (1-\epsilon)t_i + \epsilon k^{-1}; \\ x^i_\epsilon & = & t^{-1}_{i,\epsilon}\left[(1-\epsilon)\xi^i + \epsilon k^{-1}\bar{x}^i\right]; \\ u^i_\epsilon & = & t^{-1}_{i,\epsilon}\left[(1-\epsilon)\eta^i + \epsilon k^{-1}\bar{u}^i\right], \end{array}$$

we get

$$\begin{array}{rcl} A_i x^i_\epsilon + B_i u^i_\epsilon + b_i & \geq_{\mathbf{K}_i} & 0 \Rightarrow \\ x^i_\epsilon & \in & X_i, \\ t_{1,\epsilon}, ..., t_{k,\epsilon} & \geq & 0, \\ t_{1,\epsilon} + ... + t_{k,\epsilon} & = & 1 \\ & \Rightarrow & \\ x_\epsilon & \equiv & \sum_{i=1}^k t_{i,\epsilon} x^i_\epsilon \\ & \in & \mathrm{Conv}(\bigcup\limits_{i=1}^k X_i). \end{array}$$

On the other hand, by construction we have

$$x_\epsilon = \sum_{i=1}^k \left[(1-\epsilon)\xi^i + \epsilon k^{-1}\bar{x}^i\right] \to x = \sum_{i=1}^k \xi^i,\ \epsilon \to +0,$$

so that x belongs to the closed convex hull of $\bigcup\limits_{i=1}^k X_i$, as claimed.

It remains to verify that in the cases of (i), (ii) the convex hull of $\bigcup\limits_{i=1}^k X_i$ is the same as the closed convex hull of this union. (i) is a particular case of (ii) corresponding to $W = \{0\}$, so that it suffices to prove (ii). Assume that

$$\begin{array}{c} x_t = \sum_{i=1}^k \mu_{ti}[z_{ti} + p_{ti}] \to x,\ i \to \infty \\ \left[z_{ti} \in Z_i, p_{ti} \in W, \mu_{ti} \geq 0, \sum_i \mu_{ti} = 1\right] \end{array}$$

and let us prove that x belongs to the convex hull of the union of X_i. Indeed, since Z_i are closed and bounded, passing to a subsequence, we may assume that

$$z_{ti} \to z_i \in Z_i \text{ and } \mu_{ti} \to \mu_i \text{ as } t \to \infty.$$

It follows that the vectors

$$p_t = \sum_{i=1}^m \mu_{ti} p_{ti} = x_t - \sum_{i=1}^k \mu_{ti} z_{ti}$$

converge as $t \to \infty$; since W is closed and convex, the limit p of these vectors belongs to W. We now have

$$x = \lim_{i\to\infty}\left[\sum_{i=1}^k \mu_{ti} z_{ti} + p_t\right] = \sum_{i=1}^k \mu_i z_i + p = \sum_{i=1}^k \mu_i[z_i + p],$$

so that x belongs to the convex hull of the union of X_i (as a convex combination of points $z_i + p \in X_i$).

N. Theorem on superposition. Let $f_\ell : \mathbf{R}^m \to \mathbf{R} \cup \{+\infty\}$, $\ell = 1, ..., m$ be CQr functions:

$$t \geq f_\ell(x) \Leftrightarrow \exists u^\ell \mid A_\ell(x, t, u^\ell) \succeq_{\mathbf{K}_\ell} 0,$$

where $\mathbf{K}_\ell$ is a direct product of Lorentz cones, and let

$$f : \mathbf{R}^m \to \mathbf{R} \cup \{+\infty\}$$

be CQr:

$$t \geq f(y) \Leftrightarrow \exists v \mid A(y, t, v) \succeq_{\mathbf{K}} 0,$$

$\mathbf{K}$ being a direct produc of the Lorentz cones.

Assume that f is monotone with respect to the usual partial order:

$$y' \geq y'' \Rightarrow f(y') \geq f(y''),$$

and that the superposition

$$g(x) = \begin{cases} f(f_1(x), ..., f_m(x)) & f_\ell(x) < \infty, \ell = 1, ..., m \\ +\infty & \text{otherwise} \end{cases}$$

is a function (i.e., is finite at least at one point).

Theorem 3.3.1 *In the situation in question, the superposition g is CQr with CQR*

$$t \geq g(x) \Leftrightarrow \exists t_1, ..., t_m, u^1, ..., u^m, v : \begin{cases} A_\ell(x, t_\ell, u^\ell) \succeq_{\mathbf{K}_\ell} 0, \ \ell = 1, ..., m \\ A(t_1, ..., t_m, t, v) \succeq_{\mathbf{K}} 0 \end{cases} \tag{3.3.17}$$

Proof of this simple statement is left to the reader.

Remark 3.3.1 If part of the "inner" functions f_ℓ, say, $f_1, ..., f_k$, are affine, it suffices to require the monotonicity of the "outer" function f with respect to the variables $y_{k+1}, ..., y_m$ only. A CQR for the superposition in this case becomes

$$t \geq g(x) \Leftrightarrow \exists t_{k+1}, ..., t_m, u^{k+1}, ..., u^m, v : \begin{cases} A_\ell(x, t_\ell, u^\ell) \succeq_{\mathbf{K}_\ell} 0, \ \ell = k+1, ..., m \\ A(f_1(x), f_2(x), ..., f_k(x), t_{k+1}, t_{k+2}, ..., t_m, t, v) \succeq_{\mathbf{K}} 0 \end{cases} \tag{3.3.18}$$

3.3.1 More examples of CQ-representable functions/sets

Now we are enough equipped to build the dictionary of CQ-representable functions/sets. We already have "the elementary" part of the dictionary; now let us add a more "advanced" part of it.

7. Convex quadratic form $g(x) = x^T Q x + q^T x + r$, Q being a positive semidefinite symmetric matrix, is CQr.

Indeed, Q is positive semidefinite symmetric and therefore can be decomposed as $Q = D^T D$, so that $g(x) =\| Dx \|_2^2 + q^T x + r$. We see that g is obtained from our "raw materials" – the squared Euclidean norm and an affine function – by affine substitution of argument and addition.

Here is an explicit CQR of g:

$$\{(x, t) \mid x^T D^T D x + q^T x + r \leq t\} = \{(x, t) \mid \left\| \begin{matrix} Dx \\ \frac{t + q^T x + r}{2} \end{matrix} \right\|_2 \leq \frac{t - q^T x - r}{2}\} \tag{3.3.19}$$

8. The cone $K = \{(x,\sigma_1,\sigma_2) \in \mathbf{R}^n \times \mathbf{R} \times \mathbf{R} \mid \sigma_1,\sigma_2 \geq 0, \sigma_1\sigma_2 \geq x^T x\}$ is CQr.
Indeed, the set is just denotation of the epigraph of a fractional-quadratic function $x^T x/s$, see Example 5; we simply write σ_1 instead of s and σ_2 instead of t.

Here is an explicit CQR for the set:

$$K = \{(x,\sigma_1,\sigma_2) \mid \left\| \begin{pmatrix} x \\ \frac{\sigma_1-\sigma_2}{2} \end{pmatrix} \right\|_2 \leq \frac{\sigma_1+\sigma_2}{2}\} \tag{3.3.20}$$

Surprisingly, our set is just the Lorenz cone, more precisely, its inverse image under the one-to-one linear mapping

$$\begin{pmatrix} x \\ \sigma_1 \\ \sigma_2 \end{pmatrix} \mapsto \begin{pmatrix} x \\ \frac{\sigma_1-\sigma_2}{2} \\ \frac{\sigma_1+\sigma_2}{2} \end{pmatrix}.$$

9. The "half-cone" $K^2_+ = \{(x_1,x_2,t) \in \mathbf{R}^3 \mid x_1,x_2 \geq 0, 0 \leq t \leq \sqrt{x_1x_2}\}$ is CQr.
Indeed, our set is the intersection of the cone $\{t^2 \leq x_1x_2, x_1,x_2 \geq 0\}$ from the previous example and the half-space $t \geq 0$.

Here is the explicit CQR of K_+:

$$K_+ = \{(x_1,x_2,t) \mid t \geq 0, \left\| \begin{pmatrix} t \\ \frac{x_1-x_2}{2} \end{pmatrix} \right\|_2 \leq \frac{x_1+x_2}{2}\}. \tag{3.3.21}$$

10. The hypograph of the geometric mean – the set $K^2 = \{(x_1,x_2,t) \in \mathbf{R}^3 \mid x_1,x_2 \geq 0, t \leq \sqrt{x_1x_2}\}$ – is CQr.
Note the difference with the previous example – now t is not required to be nonnegative!

Here is the explicit CQR for K^2 (cf. Example 9):

$$K^2 = \left\{(x_1,x_2,t) \mid \exists \tau : t \leq \tau; \tau \geq 0, \left\| \begin{pmatrix} \tau \\ \frac{x_1-x_2}{2} \end{pmatrix} \right\|_2 \leq \frac{x_1+x_2}{2}\right\}.$$

11. The hypograph of the geometric mean of 2^l variables – the set $K^{2^l} = \{(x_1,...,x_{2^l},t) \in \mathbf{R}^{2^l+1} \mid x_i \geq 0, i = 1,...,2^l, t \leq (x_1x_2...x_{2^l})^{1/2^l}\}$ – is CQr.

To see that K^{2^l} is CQr and to get a CQR of this set, it suffices to iterate the construction of Example 10. Indeed, let us add to our initial variables a number of additional x-variables:

– let us call 2^l our original x-variables the variables of *level 0* and write $x_{0,i}$ instead of x_i. Let us add one new variable of *level 1* per every two variables of level 0. Thus, we add 2^{l-1} variables $x_{1,i}$ of level 1.

– similarly, let us add one new variable of level 2 per every two variables of level 1, thus adding 2^{l-2} variables $x_{2,i}$; then we add one new variable of level 3 per every two variables of level 2, and so on, until level l with a single variable $x_{l,1}$ is built.

Now let us look at the following system S of constraints:

$$\begin{array}{ll}
\text{layer 1:} & x_{1,i} \leq \sqrt{x_{0,2i-1}x_{0,2i}}, x_{1,i}, x_{0,2i-1}, x_{0,2i} \geq 0, \quad i = 1,...,2^{l-1} \\
\text{layer 2:} & x_{2,i} \leq \sqrt{x_{1,2i-1}x_{1,2i}}, x_{2,i}, x_{1,2i-1}, x_{1,2i} \geq 0, \quad i = 1,...,2^{l-2} \\
 & \dots\dots\dots\dots\dots\dots \\
\text{layer } l\text{:} & x_{l,1} \leq \sqrt{x_{l-1,1}x_{l-1,2}}, x_{l,1}, x_{l-1,1}, x_{l-1,2} \geq 0 \\
\hline
(*) & t \leq x_{l,1}
\end{array}$$

The inequalities of the first layer say that the variables of the zero and the first level should be nonnegative and every one of the variables of the first level should be $\leq$ the geometric mean of the corresponding pair of our original x-variables. The inequalities of the second layer add the requirement that the variables of the second level should be nonnegative, and every one of them should be $\leq$ the geometric mean of the corresponding pair of the first level variables, etc. It is clear that if all these inequalities and (*) are satisfied, then t is $\leq$ the geometric mean of $x_1,...,x_{2^l}$. Vice versa, given nonnegative $x_1,...,x_{2^l}$ and a real t which is $\leq$ the geometric mean of $x_1,...,x_{2^l}$, we always can extend these data to a solution of

S. In other words, K^{2^l} is the projection of the solution set of S onto the plane of our original variables $x_1, ..., x_{2^l}, t$. It remains to note that the set of solutions of S is CQr (as the intersection of CQr sets $\{(v, p, q, r) \in \mathbf{R}^N \times \mathbf{R}^3_+ \mid r \leq \sqrt{qp}\}$, see Example 9), so that its projection also is CQr. To get a CQR of K^{2^l}, it suffices to replace the inequalities in S with their conic quadratic equivalents explicitly given by Example 9.

What about functions which look very different from something quadratic, e.g., what about the function $g(x) = x^{7/3}$ on the axis? Is it CQr? If the question is to be interpreted "literally", the answer is definite "no" – the function in question is non-convex! An absolutely evident observation is as follows:

> (!) *A CQr set X always is convex (as the projection of the set of solutions of a system of convex inequalities $\| \alpha_j(x, u) \|_2 - \beta_j(x, y) \leq 0$ in the space of (x, u)-variables onto the space of x-variables).*
>
> *Consequently, a CQr function necessarily is convex – since its epigraph must be a CQr, and therefore convex, set.*

Our question about the function $x^{7/3}$ admits, however, a meaningful modification. Namely, the function $x^{7/3}$, same as every power function x^p with $p \geq 1$, is convex on the nonnegative ray; extending the function by the value 0 onto the negative ray, we get a convex function x_+^p, $x_+ = \max\{x, 0\}$, and we may ask whether this function is CQr. In particular, what about CQ-representativity of the function $x_+^{7/3}$? The answer is affirmative, and here is the construction:

We know from Example 11 that the set

$$K^{16} = \{(y_1, ..., y_{16}, s) \in \mathbf{R}^{17}_+ \mid s \leq (y_1 y_2 ... y_{16})^{1/16}\}$$

is CQr. Now let us make all our 17 variables $y_1, ..., y_{16}, s$ affine functions of just two variables ξ, t as follows:

– the variable s and first 9 of the variables y_i are identically equal to ξ;

– the next 3 of the variables y_i are identically equal to t;

– the rest of the variables (i.e., the last four variables y_i) are identically equal to 1.

We have defined certain affine mapping from $\mathbf{R}^2$ to $\mathbf{R}^{17}$. What is the inverse image of K^{16} under this mapping? It is the set

$$\begin{array}{rcl} K & = & \{(\xi, t) \in \mathbf{R}^2_+ \mid \xi \leq \xi^{9/16} t^{3/16}\} \\ & = & \{(\xi, t) \in \mathbf{R}^2_+ \mid t \geq \xi^{7/3}\}. \end{array}$$

Thus, the set K is CQr (as an inverse image of the CQr set K^{16}), and we can easily get explicit CQR of K from the one of K^{16} (see Example 11). On the other hand, the set K is "almost" the epigraph of $\xi_+^{7/3}$ – it is the part of this epigraph in the first quadrant. And it is easy to get the "complete" epigraph from this part: it suffices to note that the epigraph E of $x_+^{7/3}$ is the projection of the 3D set

$$K' = \{(x, \xi, t) \mid \xi \geq 0, x \leq \xi, \xi^{7/3} \leq t\}$$

onto the (x, t)-plane, and that the set K' clearly is CQr along with K. Indeed, to obtain K' from K one should first pass from K to its direct product with the x-axis – to the set

$$K^+ = \{(x, \xi, t) \mid (\xi, t) \in K\}$$

– and then to intersect K^+ with the half-space given by the inequality $x \leq \xi$. Thus, to obtain the epigraph E we are interested in from the CQr set K, one should successively

– pass from K to its direct product with the real axis $\mathbf{R}$ (note that the second factor clearly is CQr!)

– intersect the result with a half-space

– project the result onto 2D plane of the variables (x, t).

All these operations preserve the CQ-representability and yield an explicit CQr for E:

$$\{t \geq x_+^{7/3}\} \Leftrightarrow \exists(\xi, u) : \{x \leq \xi\} \ \& \ S(\underbrace{\xi, ..., \xi}_{9}, \underbrace{t, ..., t}_{3}, \underbrace{1, ..., 1}_{4}, \xi; u),$$

where $S(y, t; u)$ denotes the system of c.q.i.'s from the CQR of K^{16} [5)], u being the vector of additional variables for the latter CQR.

The outlined construction might look too sophisticated; well, with small experience the derivations of this type become much easier and transparent than, say, arithmetic manipulations required to solve a 3×3 system of linear equations...

Of course, the particular values 7 and 3 in our "$x_+^{7/3}$-exercise" play no significant role, and we arrive at the following

12. The convex increasing power function $x_+^{p/q}$ of rational degree $p/q > 1$ is CQr.
Indeed, given positive integers p, q, $p > q$, let us choose the smallest integer l such that $p + q \leq 2^l$, and consider the CQr set

$$K^{2^l} = \{(y_1, ..., y_{2^l}, s) \in \mathbf{R}_+^{2^l+1} \mid s \leq (y_1 y_2 ... y_{2^l})^{1/2^l}\}. \tag{3.3.22}$$

Setting $r = 2^l - p$, consider the following affine parameterization of the variables from $\mathbf{R}^{2^l+1}$ by two variables ξ, t:

– s and r first variables y_i are identically equal to ξ (note that we still have $2^l - r = p > q$ "unused" variables y_i);

– q next variables y_i are identically equal to t;

– the remaining y_i's are identically equal to 1.

The inverse image of K^{2^l} under this mapping is CQr and is the set

$$K = \{(\xi, t) \in \mathbf{R}_+^2 \mid \xi^{1-r/2^l} \leq t^{q/2^l}\} = \{(\xi, t) \in \mathbf{R}_+^2 \mid t \geq \xi^{p/q}\}.$$

The epigraph of $x_+^{p/q}$ can be obtained from the CQr set K by the same preserving CQ-representability operations as in the case of $p/q = 7/3$.

13. The decreasing power function $g(x) = \begin{cases} x^{-p/q}, & x > 0 \\ +\infty, & x \leq 0 \end{cases}$ of rational degree $-p/q < 0$ is CQr.
Same as in 12, given integer $p, q > 0$ we choose the smallest integer l such that $2^l \geq p + q$, consider the CQr set (3.3.22) and parameterize affinely the variables y_i, s by two variables (x, t) as follows:

– s and the first $2^l - p - q$ of y_i's are identically equal to one;

– p of the remaining y_i's are identically equal to x, and q last of y_i's are identically equal to t.

It is immediately seen that the inverse image of K^{2^l} under the indicated affine mapping is the epigraph of g.

14. The even power function $g(x) = x^{2p}$ on the axis (p is a positive integer) is CQr.
Indeed, we already know that the sets $P = \{(x, \xi, t) \in \mathbf{R}^3 \mid x^2 \leq \xi\}$ and $K' = \{(x, \xi, t) \in \mathbf{R}^3 \mid 0 \leq \xi, \xi^p \leq t\}$ are CQr (both sets are direct products of an axis and the sets with already known to us CQR's). It remains to note that the epigraph of g is the projection of $P \cap Q$ onto the (x, t)-plane.

[5)] i.e., $S(y, t; u)$ is a Boolean function taking values **true** of **false** depending on whether the (y, t, u) satisfy or does not satisfy the c.q.i.'s in question

Note that Example 14 along with our combination rules allows to build a CQR for an arbitrary "evidently convex" polynomial $p(x)$ on the axis, i.e., for every polynomial of the form

$$p(x) = \sum_{l=1}^{L} p_l x^{2l}, \;\; p_i \geq 0.$$

We see how strong are the "expressive abilities" of c.q.i.'s: they allow to handle a wide variety of essentially different functions and sets. We see also how powerful is our (in its essence absolutely trivial!) "calculus" of CQ-representable functions and sets.

3.4 More applications

We now are enough equipped to consider more applications of Conic Quadratic Programming.

3.4.1 Tschebyshev approximation in relative scale

The material of this Section originates from [2]. Recall that in the Tschebyshev approximation problem we are looking for a linear combination of given basic functions $f_i(t)$ which is as close as possible to a given target function $f_*(t)$, all functions being defined on a given finite set T. In the usual version of the problem, the quality of an approximation $\sum_i x_i f_i(t)$ is measured by the largest, over $t \in T$, *absolute* deviation between the approximation and the target function. In a number of applications absolute deviation does not make much sense: the target function is positive, so should be its approximation, and the deviation we indeed are interested in is the *relative* one. A natural way to measure the relative deviation between two positive reals a, b is to look at the smallest $\tau \equiv \tau(a, b)$ such that both

$$a/b \leq 1 + \tau, b/a \leq 1 + \tau,$$

or, which is the same, the smallest τ such that

$$\frac{1}{1+\tau} \leq \frac{a}{b} \leq 1 + \tau.$$

With this approach, the "relative" Tschebyshev problem becomes

$$\min_x \max_{t \in T} \tau(f_*(t), \sum_i x_i f_i(t)),$$

where we should add the constraints that $\sum_i x_i f_i(t) > 0, t \in T$, in order to meet the requirement of positivity of the approximation. The resulting problem can be written equivalently as

$$\tau \to \min \mid \sum_i x_i f_i(t) \leq (1+\tau) f_*(t), f_*(t) \leq (1+\tau) \sum_i x_i f_i(t), \;\forall t \in T,$$

the variables being τ, x. All nonlinear constraints we get are the hyperbolic constraints "the product of two nonnegative affine forms of the design variables must be $\geq$ a given positive constant", and the sets given by these constraints are CQr (see Example 6). Thus, the problem is equivalent to a conic quadratic program.

3.4.2 Robust Linear Programming

Consider an LP program

$$c^T x \to \min \mid Ax - b \geq 0 \qquad \text{(LP)}$$

When such a problem arises in applications, its data c, A, b not always are not known exactly; what is typically known is a domain $\mathcal{U}$ in the space of data – an "uncertainty set" – which for sure contains the "actual" (unknown) data. On the other hand, there are situations in reality where our decision x must satisfy the "actual" constraints, whether we know them or not. Assume, e.g., that (LP) is a model of a technological process in Chemical Industry, so that entries of x represent the amounts of different kinds of materials participating in the process. Our process typically is comprised of a number of decomposition-recombination stages, and we are supposed to take care of the natural balance restrictions: the amount of every material to be used at a particular stage cannot exceed the amount of the same material yielded by the preceding stages. In a meaningful production plan, the balance inequalities must be satisfied; at the same time these inequalities typically involve coefficients affected by unavoidable uncertainty of the contents of the raw materials, time-varying parameters of the technological devices, etc.

In the situation when all we know about the data is a set $\mathcal{U}$ they belong to, on one hand, and we *must* satisfy the actual constraints, on the other hand, the only way to meet the requirements is to restrict ourselves with *robust feasible* candidate solutions x – those satisfying *all possible* realizations of the uncertain constraints, i.e., such that

$$Ax - b \geq 0 \quad \forall [A; b]\, \exists c : (c, A, b) \in \mathcal{U}. \qquad (3.4.1)$$

In order to choose among these robust feasible solutions the best possible, we should decide how to "aggregate" various realizations of the objective in something single. To be methodologically consistent, it makes sense to use the same worst-case-oriented approach and to use the "guaranteed" objective $f(x)$ – the maximum, over all possible realizations of the objective c, value of $c^T x$:

$$f(x) = \sup\{c^T x \mid c : \exists [A : b] : (c, A, b) \in \mathcal{U}\}.$$

With this methodology, we can associate with out *uncertain LP program* $\mathcal{LP}(\mathcal{U})$ (the family of all usual – "certain" – LP programs (LP) with the data belonging to $\mathcal{U}$) its robust counterpart, where we are seeking for robust feasible solution with the smallest possible value of the guaranteed objective. In other words, the robust counterpart is the optimization problem

$$t \to \min \mid c^T x \leq t, Ax - b \geq 0 \quad \forall (c, A, b) \in \mathcal{U} \qquad \text{(R)}$$

Note that (R) is a usual – "certain" – optimization problem, but typically is not an LP program: the structure of (R) depends on the geometry of the uncertainty set $\mathcal{U}$ and can be very complicated.

Now, in many cases it makes sense to specify the uncertainty set (which is the set in some $\mathbf{R}^N$) as an *ellipsoid* – the image of the unit Euclidean ball under an affine mapping – or, more generally as an *intersection of finitely many ellipsoids*. And it is shown in [1] that in all these cases the robust counterpart of an uncertain LP problem is (equivalent to) an explicit conic quadratic program. Thus, Robust Linear Programming with ellipsoidal uncertainty sets can be viewed as a "generic source" of conic quadratic problems.

Let us look at the robust counterpart of an uncertain LP program

$$\left\{ \begin{array}{rcl} c^T x & \to & \min \\ a_i^T x - b_i & \geq & 0,\ i = 1, ..., m \end{array} \right\}_{(c,A,b)\in\mathbf{U}}$$

in the case of a simple ellipsoidal uncertainty – one where the data (a_i, b_i) of i-th inequality constraint

$$a_i^T x - b_i \geq 0,$$

same as the objective c, are allowed to run independently of each other through respective ellipsoids E_i, E. Thus, we assume that the uncertainty set is

$$\mathcal{U} = \left\{ (a_1, b_1; ...; a_m, b_m; c) \mid \exists (\{u_i, u_i^T u_i \leq 1\}_{i=0}^m) : \ c = c_* + P_0 u_0, \begin{pmatrix} a_i \\ b_i \end{pmatrix} = \begin{pmatrix} a_i^* \\ b_i^* \end{pmatrix} + P_i u^i, i = 1, ..., m \right\},$$

where c_*, a_i^*, b_i^* are the "nominal data" and $P_i u_i$, $i = 0, 1, ..., m$, represent the data perturbations; restrictions $u_i^T u_i \leq 1$ enforce these perturbations to vary in ellipsoids.

In order to realize that the robust counterpart of our uncertain LP problem is a conic quadratic program, note that x is robust feasible if and only if for every $i = 1, ..., m$ we have

$$\begin{array}{rcl} 0 & \leq & \min\limits_{u_i : u_i^T u_i \leq 1} \left[a_i^T[u]x - b_i[u] \mid \begin{pmatrix} a_i[u] \\ b_i[u] \end{pmatrix} = \begin{pmatrix} a_i^* \\ b_i^* \end{pmatrix} + P_i u_i \right] \\ & = & (a_i^* x)^T x - b_i^* + \min\limits_{u_i : u_i^T u_i \leq 1} u_i^T P_i^T \begin{pmatrix} x \\ -1 \end{pmatrix} \\ & = & (a_i^*)^T x - b_i^* - \left\| P_i^T \begin{pmatrix} x \\ -1 \end{pmatrix} \right\|_2 \end{array}$$

Thus, x is robust feasible if and only if it satisfies the system of c.q.i.'s

$$\left\| P_i^T \begin{pmatrix} x \\ -1 \end{pmatrix} \right\|_2 \leq [a_i^*]^T x - b_i^*, \ i = 1, ..., m.$$

Similarly, a pair (x, t) satisfies all realizations of the inequality $c^T x \leq t$ "allowed" by our ellipsoidal uncertainty set $\mathcal{U}$ if and only if

$$c_*^T x + \| P_0^T x \|_2 \leq t.$$

Thus, the robust counterpart (R) becomes the conic quadratic program

$$t \to \min \mid \| P_0^T x \|_2 \leq -c_*^T x + t; \ \left\| P_i^T \begin{pmatrix} x \\ -1 \end{pmatrix} \right\|_2 \leq [a_i^*]^T x - b_i^*, \ i = 1, ..., m \qquad \text{(RLP)}$$

Example: Robust synthesis of antenna array. Consider the same Antenna Synthesis example as in Section 1.2.4. Mathematically, the problem we were solving was an LP program with 11 variables

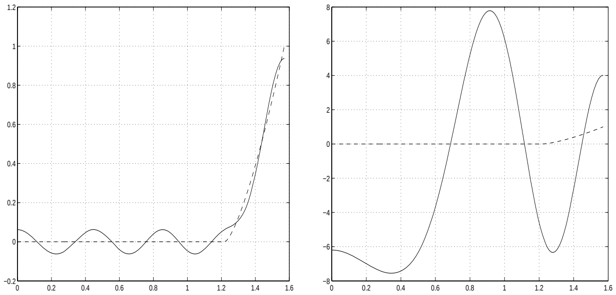
$$t \to \min \mid -t \leq Z_*(\theta_i) - \sum_{j=1}^{10} x_j Z_j(\theta_i) \leq t, \ i = 1, ..., N \qquad \text{(Nom)}$$

with given diagrams $Z_j(\cdot)$ of 10 "building blocks" and a given target diagram $Z_*(\theta)$. Let x_j^* be the optimal values of the design variables. Recall that our design variables are amplification coefficients – i.e., characteristics of certain physical devices. In reality, of course, we cannot tune the devices to have precisely the optimal characteristics x_j^*; the best we may hope at is that the actual characteristics $x_j^{\rm fct}$ of the amplifiers will coincide with the desired values x_j^* within small margin, say, 0.1% (this is a fairly high accuracy for a physical device):

$$x_j^{\rm fct} = p_j x_j^*, \ 0.999 \leq p_j \leq 1.001.$$

It is natural to assume that the factors p_j are random with the mean value equal to 1; it will be not a great sin to suppose that these factors are independent of each other.

Since the actual amplification coefficients differ from their desired values x_j^*, the actual (random) diagram of our array of antennae will differ from the "nominal" one we have found in Section 1.2.4. How large could be the difference? Look at the picture:



"Dream and reality": the nominal (left, solid) and an actual (right, solid) diagrams [dashed: the target diagram]

Note that what is shown to the right is not "the worst case": we just have taken as p_j a sample of 10 independent numbers distributed uniformly in $[0.999, 1.001]$ and have plotted the diagram corresponding to $x_j = p_j x_j^*$. Pay attention not only to the shape (completely opposite to what we need), but also to the scale: the target diagram varies from 0 to 1, the nominal diagram (the one which corresponds to the exactly optimal x_j) differs from the target one no more than by 0.0621 (this is the "nominal" optimal value – the one of the "nominal" problem (Nom)). The actual diagram varies from ≈ -8 to ≈ 8, and its uniform distance from the target is 7.79 (125 times larger !). We see that our nominal optimal design is completely meaningless: it looks as if we were trying to get the worse possible result, not the best possible one...

How could we get something better? Let us try to apply the Robust Counterpart approach. To this end let us note that if we want the amplification coefficients to be certain x_j, then the actual coefficients will be $x_j^{\rm fct} = p_j x_j$, $0.999 \le p_j \le 1.001$, and the actual discrepancies will be

$$\delta_i(x) = Z_*(\theta_i) - \sum_{j=1}^{10} p_j x_j Z_j(\theta_i).$$

Thus, we in fact are solving an uncertain LP problem where the uncertainty affects the coefficients of the constraint matrix (those corresponding to the variables x_j): these coefficients may vary within 0.1% margin of their nominal values.

In order to apply to our uncertain LP program the Robust Counterpart approach, we should specify the uncertainty set. The most straightforward way to do it is to say that our uncertainty is "an interval" one – every uncertain coefficient in a given inequality constraint may, independently of all other coefficients, run through its own uncertainty segment "nominal value ±0.1%". This approach, however, is too conservative: we have completely ignored the fact that our p_j's are of stochastic nature and are independent of each other, so that it is highly un-probable that all of them will simultaneously fluctuate in "dangerous" directions. In order to utilize statistical independence of perturbations, let us look what happens with a particular inequality

$$-t \le \delta_i(x) \equiv Z_*(\theta_i) - \sum_{j=1}^{10} p_j x_j Z_j(\theta_i) \le t \tag{3.4.2}$$

when p_j's are random. For a fixed x, the quantity $\delta_i(x)$ is a random variable with the expectation

$$\delta_i^*(x) = Z_*(\theta_i) - \sum_{j=1}^{10} x_j Z_j(\theta_i)$$

and the standard deviation

$$\begin{array}{c}\sigma_i(x)=\sqrt{E\{(\delta_i(x)-\delta_i^*(x))^2\}}=\sqrt{\sum_{j=1}^{10}x_j^2Z_j^2(\theta_i)E\{(p_j-1)^2\}}\leq\kappa\nu_i(x),\\ \nu_i(x)=\sqrt{\sum_{j=1}^{10}x_j^2Z_j^2(\theta_i)},\,\kappa=0.001.\end{array}$$

In other words, "a typical value" of $\delta_i(x)$ differs from $\delta_i^*(x)$ by a quantity of order of $\sigma_i(x)$. Now let us act as an engineer which believes that a random variable never differs from its mean by more than three times the standard deviation; since we are not obliged to be that concrete, let us choose a "safety parameter" ω and *ignore all events which result in* $|\delta_i(x)-\delta_i^*(x)|>\omega\nu_i(x)$ $^{6)}$. As about remaining events – those with $|\delta_i(x)-\delta_i^*(x)|\leq\omega\nu_i(x)$ – we do take upon ourselves full responsibility for these events. With this approach, a "reliable deterministic version" of uncertain constraint (3.4.2) becomes the pair of inequalities

$$\begin{array}{rcl}-t&\leq&d_i^*(x)-\omega\nu_i(x),\\ d_i^*(x)+\omega\nu_i(x)&\leq&t;\end{array}$$

Replacing all uncertain inequalities in (Nom) with their "reliable deterministic versions" and recalling the definition of $d_i^*(x)$ and $\nu_i(x)$, we end up with the optimization problem

$$\begin{array}{rrcl}&&&t\to\min\\ \text{s.t.}&&&\\ &\parallel Q_ix\parallel_2&\leq&[Z_*(\theta_i)-\sum_{j=1}^{10}x_jZ_j(\theta_i)]+t,\ i=1,...,N\\ &\parallel Q_ix\parallel_2&\leq&-[Z_*(\theta_i)-\sum_{j=1}^{10}x_jZ_j(\theta_i)]+t,\ i=1,...,N\\ &&&[Q_i=\omega\kappa\,\mathrm{Diag}(Z_1(\theta_i),Z_2(\theta_i),...,Z_{10}(\theta_i))]\end{array}\qquad\text{(Rob)}$$

It is immediately seen that (Rob) *is nothing but the robust counterpart of* (Nom) *corresponding to a simple ellipsoidal uncertainty*, namely, the one as follows:

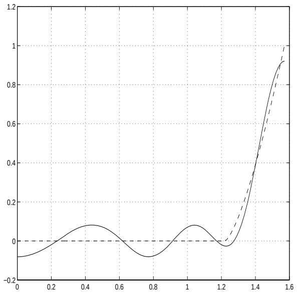
> The only data of a constraint
>
> $$\sum_{j=1}^{10}A_{ij}x_j\ \begin{array}{c}\geq\\ \leq\end{array}\ p_it+q_i$$
>
> (all constraints in (Nom) are of this form) affected by the uncertainty are the coefficients A_{ij} of the left hand side, and the difference $dA[i]$ between the vector of these coefficients and the nominal value $(Z_1(\theta_i),...,Z_{10}(\theta_i))$ of the vector of coefficients belongs to the ellipsoid
>
> $$\{dA[i]=\omega\kappa Q_iu\mid u\in\mathbf{R}^{10},u^Tu\leq1\}.$$

Thus, the above speculation was nothing but a reasonable way to specify uncertainty ellipsoids!

Now let us look what are the diagrams yielded by the Robust Counterpart approach – i.e., those given by the robust optimal solution. These diagrams also are random (neither the nominal nor the robust solution cannot be implemented exactly!); it, however, turns out that they are incomparably closer to the target (and to each other) than the diagrams associated with the optimal solution to the "nominal"

[6*] It would be better to use here σ_i instead of ν_i; we, however, did not assume that we know the distribution of p_j, this is why we replace unknown σ_i with its known upper bound ν_i

problem. Look at a typical "robust" diagram:



A "Robust" diagram. Uniform distance from the target is 0.0822.
[the safety parameter for the uncertainty ellipsoids is $\omega = 1$]

With the safety parameter $\omega = 1$, the robust optimal value is 0.0817; although it is by 30% larger than the nominal optimal value 0.0635, the robust optimal value has a definite advantage that it indeed says something about the quality of actual diagrams we can obtain when implementing the robust optimal solution: in a sample of 40 realizations of the diagrams corresponding to the robust optimal solution, the uniform distances from target were varying from 0.0814 to 0.0830.

We have built robust optimal solution under the assumption that the "implementation errors" do not exceed 0.1%. What happens if in reality the errors are larger – say, 1%? It turns out that nothing that dramatic: now in a sample of 40 diagrams given by the "old" robust optimal solution (affected by 10 times larger "implementation errors") the uniform distances from the target were varying from 0.0834 to 0.116. Imagine what will happen with the nominal solution under the same circumstances...

The last issue to be addressed here is: why the nominal solution is so instable? And why with the robust counterpart approach we were able to get a solution which is incomparably better, as far as "actual implementation" is concerned? The answer becomes clear when looking at the nominal and the robust optimal amplification coefficients:

j	1	2	3	4	5	6	7	8	9	10
$x_j^{\rm nom}$	1624.4	-14701	55383	-107247	95468	19221	-138622	144870	-69303	13311
$x_j^{\rm rob}$	-0.3010	4.9638	-3.4252	-5.1488	6.8653	5.5140	5.3119	-7.4584	-8.9140	13.237

It turns out that our nominal problem is "ill-posed" – although its optimal solution is far away from the origin, there is a "massive" set of "nearly optimal" solutions, and among the latter ones we can choose solutions of quite moderate magnitude. Indeed, here are the optimal values obtained when we add to the constraints of (Nom) the box constraints $|x_j| \leq L$, $j = 1, ..., 10$:

L	1	10	10^2	10^3	10^4	10^5	10^6	10^7
Opt_Val	0.09449	0.07994	0.07358	0.06955	0.06588	0.06272	0.06215	0.06215

Since the "implementation inaccuracies" for a solution are the larger the larger it is, there is no surprise that the nominal solution results in a very instable actual design. In contrast to this, the Robust Counterpart penalizes the (properly measured) magnitude of x (look at the terms $\| Q_i x \|_2$ in the constraints of (Rob)) and therefore yields a much more stable design. Note that this situation is typical for many applications: the nominal solution is on the boundary on the nominal feasible domain, and there are "nearly optimal" solutions to the nominal problem which are in "deep interior" of this domain. When solving the nominal problem, we do not take any care of a reasonable tradeoff between the "depth of feasibility" and the optimality: *any* improvement in the objective is sufficient to make the solution just marginally feasible for the nominal problem. And a solution which is only marginally feasible in the nominal problem can easily become "very infeasible" after the data are perturbed, which would not be the case for a "deeply interior" solution. With the Robust Counterpart approach, we do use certain tradeoff between the "depth of feasibility" and the optimality – we are trying to find something like the "deepest feasible nearly optimal solution"; as a result, we normally gain a lot in stability; and if, as in our example, there are "deeply interior nearly optimal" solutions, we do not loose that much in optimality.

3.4.3 Truss Topology Design

We already have dealt with the TTD (Truss Topology Design) problem in Lecture 1, where we managed to pose it as an LP program. Our knowledge, however, is not quite satisfactory. First, all we know is how to deal with the simplest case of the problem, where we are interested to withstand best of all a given external load, and the only restriction on the design is an upper bound on its total volume. What to do if we should control the compliances with respect to a number of different loads, if we have restrictions, like upper and lower bounds on the volumes of bars, etc.? Second, we do not understand what in fact was going on in Section 1.3.5, where a highly nonlinear TTD problem by a kind of miracle became an LP program. Miracles are not that useful as one could think – they do not yield understanding (an understood miracle is not a miracle at all) and therefore cannot be reproduced when necessary. We are about to improve our understanding of the TTD problem and, in particular, to process its more general settings.

As we remember, mathematically the TTD problem from Lecture 1 was as follows:

> Given $m \times m$ dyadic matrices $b_i b_i^T$, $i = 1, ..., n$, and a n-dimensional vector t with nonnegative entries ("a truss") and an m-dimensional vector f ("a load"), we define the *compliance* $\mathrm{Compl}_f(t)$ of the truss t with respect to the load f as the quantity $\frac{1}{2} f^T v$, where v ("the equilibrium displacement") solves the equilibrium equation
>
> $$A(t)v = f$$
>
> where the *bar-stiffness matrix* $A(t)$ is given by
>
> $$A(t) = \sum_{i=1}^{n} t_i b_i b_i^T.$$
>
> If the equilibrium equation has no solutions, the compliance, by definition, is $+\infty$.
>
> And the TTD problem was, given a *ground structure* $(n, m, \{b_i\}_{i=1}^{n})$, a load f and a "resource" w, to find a truss satisfying the resource restriction
>
> $$\sum_{i=1}^{n} t_i \le w$$
>
> (and, of course, the constraints $t \ge 0$) with the minimum possible compliance w.r.t. f.

There are two different parts in the story just recalled:

(a) An excursion to Mechanics – the definition of compliance.

(b) Formulation of a particular compliance-related optimization problem.

Instead of the particular problem (b), we could pose other quite meaningful problems (and we indeed shall do it in the mean time). In order to develop a unified approach to processing all these problems, we should better understand what, mathematically, is the compliance. This is the issue we are about to attack; and it makes sense to announce right now our final result:

> *For a given ground structure, the compliance* $\mathrm{Compl}_f(t)$*, regarded as a function of variables* (t, f)*, is CQ-representable.*

Our local goal is to justify the announced claim and to get explicit CQR for the compliance. Equipped with this result, we get a possibility to process "mechanically" numerous versions of the TTD problems via the machinery of Conic Quadratic Programming.

The analysis to follow basically does not depend on the fact that the components $b_i b_i^T$ of the bar-stiffness matrix are dyadic; the essence of the matter is that they are positive semidefinite symmetric matrices. This is why from now on assume that we are speaking about the case of bar-stiffness matrix $A(t)$ of the form

$$A(t) = \sum_{i=1}^{n} t_i B_i B_i^T, \tag{3.4.3}$$

where B_i are $m \times k_i$ matrices ($k_i = 1$ for the actual TTD problem). The compliance of the resulting "generalized truss" $\mathrm{Compl}_f(t)$ is defined exactly as above – as the quantity $\frac{1}{2} f^T v$, v being a solution to the equilibrium equation

$$A(t)v = f, \tag{3.4.4}$$

with the same as above convention "$\mathrm{Compl}_f(t) = +\infty$ when the equilibrium equation has no solutions".

The Variational Principle

Our first observation is as follows:

Proposition 3.4.1 [Variational Description of Compliance]
Consider a ground structure $(n, m, B_1, ..., B_n)$ along with a load $f \in \mathbf{R}^m$, and let $t \in \mathbf{R}^n_+$ be a truss. Let us associate with these data the quadratic form ("the potential energy of the closed system")

$$\mathcal{C}_{t,f}(v) = \frac{1}{2} v^T A(t) v - f^T v \tag{3.4.5}$$

of $v \in \mathbf{R}^m$. The compliance $\mathrm{Compl}_f(t)$ is finite if and only if the form is below bounded on $\mathbf{R}^m$, and whenever it is the case, one has

$$-\mathrm{Compl}_f(t) = \min_v \mathcal{C}_{t,f}(v). \tag{3.4.6}$$

Proof. Since $t \geq 0$, the matrix $A(t)$ is positive semidefinite. Now let us use the following general and well-known fact:

Lemma 3.4.1 *Let*

$$\mathcal{A}(v) = \frac{1}{2} v^T A v - b^T v$$

be a quadratic form on $\mathbf{R}^m$ with symmetric positive semidefinite matrix A. Then

(i) *The form is below bounded if and only if it attains its minimum;*

(ii) *The form attains its minimum if and only if the equation*

$$Av = b \tag{$*$}$$

is solvable, and if it is the case, the set of minimizers of the form is exactly the set of solutions to the equation;

(iii) *The minimum value of the form, if exists, is equal to $-\frac{1}{2}b^Tv$, v being (any) solution to (*).*

Proof. (i): There are two possibilities:

(a): b is orthogonal to $\mathrm{Ker}\,A$

(b): b has a nonzero projection b' onto $\mathrm{Ker}\,A$.

In the case of (b) the form clearly is below unbounded (look what happens when $v = tb'$ and $t \to \infty$). In the case of (a) the equation $Av = b$ is solvable[7]; at every solution to this equation, the gradient of $\mathcal{A}$ vanishes, so that such a solution is a minimizer of a *convex* (A is positive semidefinite!) function $\mathcal{A}(x)$. Thus, if the form is below bounded, then it attains its minimum, and, of course, vice versa.

(ii): Since the form is convex and smooth, its minimizers are exactly the same as its critical points – those were the gradient vanishes. The gradient of $\mathcal{A}(v)$ is $Av - b$, so that it vanishes exactly at the solutions to (*).

(iii): Let v be a solution to (*), or, which is the same, a minimizer of the form. From (*) we get $v^TAv = b^Tv$, so that $\mathcal{A}(v) = \frac{1}{2}b^Tv - b^Tv = -\frac{1}{2}b^Tv$. ■

In view of Lemma, the energy (3.4.5) is below bounded if and only if the equilibrium equation (3.4.4) is solvable, and if it is the case, the minimum of the energy is $-\frac{1}{2}f^Tv$, v being a solution to the equilibrium equation; recalling the definition of the compliance, we come to the desired result. ■

By the way, we have obtained the following

Variational Principle: *The equilibrium displacement of a truss t under an external load f is a minimizer of the quadratic form*

$$\frac{1}{2}v^TA(t)v - f^Tv$$

of a displacement vector v over v; if the form is below unbounded, there is no equilibrium at all.

This is a typical for Mechanics/Physics "variational principle"; these principles state that equilibria in certain physical systems are critical points (in good cases – even minimizers) of properly chosen "energy functionals". Variational principles are extremely powerful, and in mechanical, electrical,... applications the issue of primary importance is "whether the model is governed by a "tractable" variational principle?"

From Variational Principle to CQ-representation of compliance

Step 1. Let us look at the epigraph

$$\mathcal{C} = \{(t, f; \tau) \mid t \geq 0, \tau \geq \mathrm{Compl}_f(t)\}$$

of the compliance in the domain $t \geq 0$. Our goal is to find an explicit CQR of this set; to this purpose let us start with a little bit smaller set

$$\mathcal{C}' = \{(t, f, \tau) \mid t \geq 0, \tau > \mathrm{Compl}_f(t)\}.$$

Proposition 3.4.1 provides us with the following description of $\mathcal{C}$ and $\mathcal{C}'$:

[7] Linear Algebra says that a linear system $Px = q$ is solvable if and only if q is orthogonal to $\mathrm{Ker}\,P^T$; we use this fact for the particular case of $P = P^T$

$(\mathcal{C})$: *The set $\mathcal{C}$ is comprised of all triples $(t \geq 0, f, \tau)$ such that the quadratic form*

$$Q(v) = \frac{1}{2}v^T A(t)v - f^T v + \tau$$

of $v \in \mathbf{R}^m$ is nonnegative everywhere.

$(\mathcal{C}')$: *The set $\mathcal{C}$ is comprised of all triples $(t \geq 0, f, \tau)$ such that the form $Q(v)$ is positive everywhere.*

$(\mathcal{C}')$ says that certain convex quadratic inequality – namely, the inequality

$$Q(v) \leq 0 \tag{3.4.7}$$

– has *no* solutions. As we remember, a convex quadratic inequality can be represented via a conic quadratic inequality; what is the latter inequality in the case of (3.4.7)? The answer is immediate: let us set

$$B(t) = \sqrt{2}\begin{pmatrix} \sqrt{t_1}B_1^T \\ \sqrt{t_2}B_2^T \\ \dots \\ \sqrt{t_n}B_n^T \end{pmatrix}; \tag{3.4.8}$$

The bar-stiffness matrix is

$$A(t) = \sum_{i=1}^n t_i B_i B_i^T = \frac{1}{2}B^T(t)B(t),$$

so that the quadratic form $Q(v)$ can be written down as

$$Q(v) = \frac{1}{4}v^T B^T(t)B(t)v - f^T v + \tau = \frac{1}{4}\left[\| B(t)v \|_2^2 + (1 - f^T v + \tau)^2 - (1 + f^T v - \tau)^2\right]. \tag{3.4.9}$$

We come to the following

<u>Observation:</u> *Inequality (3.4.7) has no solutions if and only if the conic quadratic inequality*

$$\left\| \begin{pmatrix} B(t)v \\ 1 - f^T v + \tau \end{pmatrix} \right\|_2 \leq 1 + f^T v - \tau \tag{3.4.10}$$

with variables v has no solution.

Indeed, the relation between the inequalities (3.4.7) and (3.4.10) is as follows: the first, in view of (3.4.9), is the inequality $\frac{1}{4}P^2(v) \leq \frac{1}{4}p^2(v)$, while the second is $P(v) \leq p(v)$, $P(v)$ being the Euclidean norm of certain vector depending on v. Taking into account that $P(\cdot)$ always is nonnegative and that $p(v) = 1 + f^T v - \tau$ must be nonnegative at every solution of (3.4.7), we conclude that both inequalities have the same set of solutions.

Step 2. As we have seen, $\mathcal{C}'$ is exactly the set of values of the "parameter" (t, f, τ) for which the q.c.i. (3.4.10) is not solvable. We claim that in fact one can say more:

(!) *When the parameter (t, f, τ) is in $\mathcal{C}'$, the c.q.i. (3.4.10) is not even "almost solvable"* (see Proposition 2.4.1).

Indeed, (3.4.10) is of the form

$$Av - b \equiv \begin{pmatrix} B(t)v \\ -f^T v \\ +f^T v \end{pmatrix} - \begin{pmatrix} 0 \\ -1 - \tau \\ -1 + \tau \end{pmatrix} \geq_{\mathbf{L}^k} 0 \tag{3.4.11}$$

with certain k. Assume that $(t, f, \tau) \in \mathcal{C}$. What we should prove is that then all c.q.i.'s of the form

$$\begin{pmatrix} B(t)v \\ -f^T v \\ +f^T v \end{pmatrix} - \begin{pmatrix} \epsilon \\ -1-\tau+\epsilon_1 \\ -1+\tau+\epsilon_2 \end{pmatrix} \geq_{\mathbf{L}^k} 0 \tag{3.4.12}$$

with small enough perturbation vector ϵ and scalars ϵ_1, ϵ_2 also are not solvable. Assume that (3.4.12) with some fixed perturbations $\epsilon, \epsilon_1, \epsilon_2$ has a solution. Then the quadratic inequality

$$\frac{1}{4} \parallel B(t)v - \epsilon \parallel_2^2 + \frac{1}{4}(1-\tau-f^T v-\epsilon_1)^2 \leq \frac{1}{4}(1+\tau+f^T v-\epsilon_2)^2$$

with unknown v has a solution. The inequality is of the form

$$\begin{array}{rcl} \frac{1}{2}v^T A(t)v - F^T(\epsilon,\epsilon_1,\epsilon_2)v + T(\epsilon,\epsilon_1,\epsilon_2) & \equiv & \frac{1}{2}v^T A(t)v - \left[\frac{1}{2}B^T(t)\epsilon + \left(1-\frac{\epsilon_1+\epsilon_2}{2}\right)f\right]^T v \\ & & + \frac{(2\tau-\epsilon_1+\epsilon_2)(2-\epsilon_1\epsilon_2)}{4} \\ & \leq & 0. \end{array} \tag{3.4.13}$$

Now, since (3.4.7) has no solutions, we have $f = A(t)e$, e being a minimizer of the unperturbed quadratic form $Q(v)$ (see Lemma 3.4.1). Now, since $A(t) = B^T(t)B(t)$, the image of $A(t)$ is exactly the same as the image of $B^T(t)$, and $A(t)$ is invertible on its image; in other words, there exists a matrix R such that $B^T(t)z = A(t)RB^T(t)z$ for every z, and, in particular, $B^T(t)\epsilon = A(t)RB^T(t)\epsilon$. We see that the vector

$$v(\epsilon,\epsilon_1,\epsilon_2) = \left(1 - \frac{\epsilon_1+\epsilon_2}{2}\right)e + \frac{1}{2}RB^T(t)\epsilon$$

is a minimizer, over v, of the left hand side in (3.4.13); the only important for us observation is that it depends continuously on the perturbation $(\epsilon, \epsilon_1, \epsilon_2)$. The coefficients of the quadratic form of v in the left hand side of (3.4.13) also are continuous in the perturbation; consequently, the minimum, over v, value of the form also depends continuously on the perturbation. This is all we need: assuming that the conclusion in (!) fails to be true for a fixed triple $(t, f, \tau) \in \mathcal{C}'$, we would conclude that there exist arbitrarily small perturbations such that the (3.4.13) has a solution, so that the minimum of the left hand side in (3.4.13) over v for these perturbations is nonpositive. By continuity, it follows that the minimum value of the quadratic form in the left hand side of (3.4.13) is nonpositive when the perturbation is 0 as well, i.e., that the minimum value of the quadratic form $Q(v)$ is nonpositive. But then (3.4.7) has a solution (recall that below bounded quadratic form achieves its minimum), which is impossible – the "parameter" (t, f, τ) belongs to $\mathcal{C}'$!

Step 3. In fact (!) can be easily strengthened to the following statement:

(!!) *A triple* (t, f, τ) *with* $t \geq 0$ *belongs to* $\mathcal{C}'$ *if and only if* (3.4.10) *is* <u>*not*</u> *almost solvable.* Indeed, the (relatively difficult) "only if" part is given by (!). And the "if" part

is immediate: we should prove that if $t \geq 0$ and (t, f, τ) does <u>not</u> belong to $\mathcal{C}'$, i.e., if $\inf_v Q(v) \leq 0$ (see $(\mathcal{C}')$, then (3.4.10) is almost solvable. But in the case in question the c.q.i. (3.4.10) is simply solvable, even without "almost". Indeed, we remember from Observation that (3.4.10) is solvable if and only if (3.4.7) is so. Now, if the form Q is below bounded, its minimum is achieved, so that under the assumption $\inf_v Q(v) \leq 0$ (3.4.7) is solvable. And in the case when Q is not below bounded, (3.4.7) is solvable by evident reasons!

Step 4. Combining (!!) with Proposition 2.4.1.(iii), we come to the following result:

Lemma 3.4.2 *A triple (t, f, τ) with $\tau > 0$ belongs to the set $\mathcal{C}'$, i.e.,* $\mathrm{Compl}_f(t) < \tau$, *if and only if there exists a vector λ of proper dimension satisfying the relations (cf.* (3.4.11)*)*

$$A^T\lambda = 0, b^T\lambda > 0, \lambda \geq_{\mathbf{L}^k} 0. \tag{3.4.14}$$

Now let us look what in fact (3.4.14) says. Recalling the definition of $B(t)$ and looking at (3.4.11), we see that

$$\begin{array}{rcl} A^T & = & [\sqrt{2t_1}B_1; \sqrt{2t_2}B_2; ...; \sqrt{2t_n}B_n; -f; f] \\ b^T & = & [0; ...; 0; -1-\tau; -1+\tau] \end{array}$$

Partitioning λ accordingly: $\lambda^T = [w_1; ..., ; w_n; p; q]$, we can rewrite the relations in (3.4.14) equivalently as

$$\begin{array}{lrcl} (a) & \sum_{i=1}^n (2t_i)^{1/2} B_i w_i & = & (p-q)f; \\ (b) & p(-1-\tau) + q(-1+\tau) & > & 0; \\ (c) & \sqrt{[\sum_{i=1}^n w_i^T w_i] + p^2} & \leq & q. \end{array} \tag{3.4.15}$$

Given a solution $(\{w_i\}, p, q)$ to (3.4.15). let us note that in such a solution necessarily $p \neq q$ (and, therefore, $p < q$ by (c)). Indeed, otherwise (b) would imply $-2q > 0$, which is impossible in view of (c). Consequently, we can define the vectors

$$s_i = -(q-p)^{-1}\sqrt{2t_i}w_i,$$

and with respect to these vectors (3.4.15) becomes

$$\begin{array}{rcl} \sum_{i=1}^n B_i s_i & = & f \\ \sum_{i=1}^n \frac{s_i^T s_i}{2t_i} & \leq & \frac{q+p}{q-p} \\ & < & \tau, \end{array} \tag{3.4.16}$$

the concluding inequality being given by (3.4.15.(b))

We have covered 99% of the way to our target, namely, have basically proved the following

Lemma 3.4.3 *A triple $(t \geq 0, f, \tau)$ is such that* $\mathrm{Compl}_f(t) < \tau$ *if and only if there exist vectors s_i, $i = 1, ..., n$, satisfying the relations*

$$\begin{array}{rcl} \sum_{i=1}^n B_i s_i & = & f \\ \sum_{i=1}^n \frac{s_i^T s_i}{2t_i} & < & \tau \end{array} \tag{3.4.17}$$

(from now on, by definition, $0/0 = 0$ and $a/0 = +\infty$ when $a > 0$).

Proof. Lemma 3.4.2 says that if $\mathrm{Compl}_f(t) < \tau$, then (3.4.14) is solvable; and as we just have seen, a solution to (3.4.14) can be converted to a solution of (3.4.17). Vice versa, given a solution $\{s_i\}_{i=1}^n$ to (3.4.17), we can find $q > 1/2$ satisfying the relations

$$\sum_{i=1}^n \frac{s_i^T s_i}{2t_i} < 2q - 1 < \tau;$$

setting $p = q - 1$, $w_i = -(2t_i)^{-1/2}s_i$ ($w_i = 0$ when $t_i = 0$), we get, as it is immediately seen, a solution to (3.4.15), or, which is the same, to (3.4.14); by the same Lemma 3.4.2, it follows that $\mathrm{Compl}_f(t) < \tau$. ■

Step 5. It remains to cover the concluding 1% of the way to the target, and here it is:

Proposition 3.4.2 *A triple (t, f, τ) belongs to the epigraph of the function* $\mathrm{Compl}_f(t)$ *(extended by the value $+\infty$ to the set of those t's which are not nonnegative), i.e.,* $\mathrm{Compl}_f(t) \leq \tau$, *if and only if there exist vectors s_i, $i = 1, ..., n$, such that the following relations are satisfied:*

$$\begin{array}{lrcl} (a) & \sum_{i=1}^n B_i s_i & = & f \\ (b) & \sum_{i=1}^n \frac{s_i^T s_i}{2t_i} & \leq & \tau \\ (c) & t & \geq & 0 \end{array} \qquad (3.4.18)$$

In particular, the function $\mathrm{Compl}_f(t)$ *is CQ-representable.*

Proof. If we can extend a given triple (t, f, τ), by properly chosen s_i's, to a solution of (3.4.18), then, by Lemma 3.4.3, $\mathrm{Compl}_f(t) < \tau'$ for every $\tau' > \tau$, whence $\mathrm{Compl}_f(t) \leq \tau$. Vice versa, assume that $\mathrm{Compl}_f(t) \leq \tau$. Then for sure $\mathrm{Compl}_f(t) < \tau + 1$, and by Lemma 3.4.3 the optimization problem

$$\min_{s_1,...,s_n} \left\{ \sum_{i=1}^n \frac{s_i^T s_i}{2t_i} \mid \sum_{i=1}^n B_i s_i = f \right\} \qquad \text{(P)}$$

is feasible. But this is, essentially, a problem of minimizing a *convex quadratic* nonnegative objective over an *affine* plane (for those i with $t_i = 0$, s_i should be zeros, and we may simply ignore the corresponding terms). Therefore the problem is solvable. Let s_i^* form an optimal solution, and τ^* the optimal value in (P). By Lemma 3.4.3, if $\tau > \mathrm{Compl}_f(t)$, then $\sum_{i=1}^n (2t_i)^{-1}(s_i^*)^T s_i^* = \tau^* < \tau$. Consequently, if $\tau \geq \mathrm{Compl}_f(t)$, then $\sum_{i=1}^n (2t_i)^{-1}(s_i^*)^T s_i^* \leq \tau$, so that s_i^* form the required solution to (3.4.18).

It remains to prove that the compliance, regarded as a function of t, f, is CQr. But this is clear – the function $\sum_i (2t_i)^{-1} s_i^T s_i$ is CQr (as a sum of fractional-quadratic functions), so that the second inequality in (3.4.18) defines a CQr set. All remaining relations in (3.4.18) are linear inequalities and equations, and from our "calculus" of CQr sets we know that constraints of this type do not spoil CQ-representability. ■.

Remark 3.4.1 Note that after the CQr description (3.4.18) of the epigraph of compliance is *guessed*, it can be justified directly by a more or less simple reasoning (this, is, basically, how we handled the TTD problem in Lecture 1). The goal of our exposition was not merely to justify (3.4.18), but to demonstrate that one can *derive* this representation quite routinely from the Variational Principle (which is the only "methodologically correct" definition of compliance) and from the rule "when looking at a convex quadratic inequality, do not trust your eyes: what you see is a conic quadratic inequality".

Remarks and conclusions

Mechanical interpretation of Proposition 3.4.2 is very transparent. Consider the case of a "true" truss – the one where $B_i = b_i$ are just vectors. Here s_i are reals, and from the mechanical viewpoint, s_i represents the magnitude of the reaction force in bar i *divided by the length of the bar* – "reaction force per unit length". With this interpretation, $B_i s_i = s_i b_i$ is the reaction force in bar i, and the equation $\sum_{i=1}^m B_i s_i = f$ in (3.4.18) says that the sum of these reaction forces should compensate the external load. Now, given a real s_i, let us extend bar i until the reaction force per unit length will become s_i. With this deformation, the bar stores certain energy, and one can easily verify that this energy is exactly $\frac{s_i^2}{2t_i}$. Consequently, the expression $\sum_{i=1}^n \frac{s_i^2}{2t_i}$ is just the energy stored by the truss, the reaction force per unit length for bar i being s_i, $i=1,...,n$. Now, what is stated by Proposition 3.4.2 is that the compliance of a given truss with respect to a given load is the minimum, over all collections $\{s_i\}$ satisfying the equation (3.4.18.(a)), of the quantities $\sum_{i=1}^n \frac{s_i^2}{2t_i}$. In other words, we get another Variational Principle:

> *The actual reaction forces in a loaded truss minimize the total energy stored by the bars under the constraint that the sum of the reaction forces compensates the external load.*

Multi-Load TTD problem. The result of Proposition 3.4.2 – the fact that the compliance admits explicit CQ-representation – allows to process numerous versions of the TTD problem via the machinery of Conic Quadratic Programming. Consider, e.g., the *Multi-Load TTD problem* as follows:

Problem 3.4.1 [Multi-Load TTD problem] *Given a ground structure* $(n,m,b_1,...,b_n)$*, a finite set of "loading scenarios"* $f_1,...,f_k\in\mathbf{R}^m$ *and a material resource* $w>0$*, find truss* t *of total bar volume not exceeding* w *most stiff, in the worst case sense, with respect to the loads* $f_1,...,f_m$*, i.e., the one with the minimum worst-case compliance* $\max_{j=1,...,k}\mathrm{Compl}_{f_j}(t)$.

The origin of the problem is clear: in reality, a truss should withstand not merely to a single load of interest, but to numerous (nonsimultaneous) loads (e.g., think of a bridge: it should be able to carry "uniform multi-force load" corresponding to the situation in traffic hours, several "single-force" loads (a single car moving at night), a load coming from wind, etc.)

Equipped with Proposition 3.4.2, we can immediately pose the Multi-Load TTD problem as the following conic quadratic program:

$$\begin{array}{lrcl}
 & & \tau \to \min & \\
\text{s.t.} & & & \\
(a) & s_{ij}^2 & \leq & 2t_i\sigma_{ij},\ i=1,...,n, j=1,...,k; \\
(b) & \sum_{i=1}^n \sigma_{ij} & \leq & \tau,\ j=1,...,k; \\
(c) & t_i,\sigma_{ij} & \geq & 0,\ i=1,...,n, j=1,...,k; \\
(d) & \sum_{i=1}^n t_i & \leq & w; \\
(e) & \sum_{i=1}^n s_{ij}b_i & = & f_j,\ j=1,...,k,
\end{array} \tag{3.4.19}$$

the design variables being $\tau, t_i, s_{ij}, \sigma_{ij}$.

The structure of the constraints is clear: For every fixed $j=1,...,k$, the corresponding equations (e), (a), (b) and the inequalities $\sigma_{ij}\geq 0$ altogether express the fact that $\sum_{i=1}^n s_{ij}b_i = f_j$ and $\sum_{i=1}^n (2t_i)^{-1}s_{ij}^2 \leq \tau$, i.e., say that the compliance of truss t with respect to load f_j is $\leq \tau$ (Proposition 3.4.2). The remaining inequalities $t_i\geq 0$, $\sum_{i=1}^n t_i\leq w$ say that t is an admissible truss. Note also that the problem indeed is conic quadratic – relations (a) are nothing but c.q.i.'s: every one of them says that the triple $(s_{ij}, t_i, \sigma_{ij})$ should belong to the 3D ice-cream cone (more exactly, the image of this cone under a one-to-one linear transformation of $\mathbf{R}^3$, see Example 8 in our catalogue of CQr sets). A nice feature of our approach is that it allows to handle a lot of additional design constraints, e.g., various linear inequalities on t, like upper and lower bounds on bar volumes t_i; indeed, adding to (3.4.19) finitely many arbitrarily chosen linear constraints, we still get a conic quadratic problem.

Another advantage is that we can – completely routinely – apply to (3.4.19) the duality machinery (see Assignments to the lecture), thus coming to a basically equivalent form of the problem; as we shall see in the mean time, the problem in its dual form is incomparably better suited for numerical processing.

Where the LP form of the TTD problem comes from? We now can explain the miracle we have met with in Lecture 1 – the possibility to pose certain case of the TTD problem (the case which still is highly nonlinear!) as a usual LP program.

What we dealt with in Lecture 1 was the *single-load* TTD problem – problem of the form (3.4.19) with $k = 1$:

$$\begin{array}{rcl} & \tau \to \min & \\ \text{s.t.} & & \\ \sum_{i=1}^n \frac{s_i^2}{2t_i} & \leq & \tau \\ \sum_{i=1}^n s_i b_i & = & f \\ \sum_{i=1}^n t_i & \leq & w \\ t & \geq & 0. \end{array}$$

We can immediately eliminate the variable τ, thus coming to the problem

$$\begin{array}{rcl} \sum_{i=1}^n \frac{s_i^2}{2t_i} & \to & \min \\ \text{s.t.} & & \\ \sum_{i=1}^n s_i b_i & = & f \\ \sum_{i=1}^n t_i & \leq & w \\ t & \geq & 0 \end{array}$$

In the latter problem, we can explicitly carry out partial optimization with respect to t; to this end, given $\{s_i\}$, we should minimize our objective

$$\sum_{i=1}^n \frac{s_i^2}{2t_i}$$

over t varying in the simplex

$$t \geq 0, \sum_{i=1}^n t_i \leq w.$$

The minimization in question can be carried out analytically: it is clear that at the optimal solution the resource constraint is active, and the Lagrange rule yields, for some $\lambda > 0$ and all i,

$$t_i = \operatorname*{argmin}_{r>0} \left[\frac{s_i^2}{2r} - \lambda r\right] = (2\lambda)^{-1/2}|s_i|.$$

The sum of all t_i's should be w, which leads to

$$2\lambda = \left(\frac{\sum_{l=1}^n |s_l|}{w}\right)^2 \Rightarrow t_i = \frac{w|s_i|}{\sum_{l=1}^n |s_l|}, \ i = 1, ..., n.$$

Substituting the resulting t_i's into the objective, we get

$$\sum_{i=1}^n \frac{s_i^2}{2t_i} = \frac{1}{2w}\left(\sum_{i=1}^n |s_i|\right)^2,$$

and the remaining problem in the s-variables becomes

$$\min\left\{\frac{1}{2w}\left(\sum_{i=1}^n |s_i|\right)^2 : \sum_{i=1}^n s_i b_i = f\right\}.$$

And the latter problem is, of course, equivalent to an LP program

$$\sum_i |s_i| \to \min \mid \sum_i s_i b_i = f.$$

Note that the outlined reduction to LP does not work in the multi-load case, same as it does not work already in slightly modified single-load settings (i.e., when we impose additional linear constraints on t, like upper and lower bounds on t_i's).

3.5 Assignments to Lecture 3

3.5.1 Optimal control in linear discrete time dynamic system

Consider a discrete time Linear Dynamic System

$$\begin{array}{rcl} x(t) & = & A(t)x(t-1)+B(t)u(t),\ t=1,2,...,T; \\ x(0) & = & x_0 \end{array} \tag{S}$$

Here:

- t is the (discrete) time;
- $x(t) \in \mathbf{R}^l$ is the *state* vector: its value at instant t identifies the state of the controlled plant;
- $u(t) \in \mathbf{R}^k$ is the exogenous input at time instant t; $\{u(t)\}_{t=1}^T$ is the *control*;
- For every $t = 1, ..., T$, $A(t)$ is a given $l \times l$, and $B(t)$ – a given $l \times k$ matrices.

A typical problem of optimal control associated with (S) is to minimize a given functional of the trajectory $x(\cdot)$ under given restrictions on the control. As a simple problem of this type, consider the optimization model

$$c^T x(T) \to \min \mid \frac{1}{2}\sum_{t=1}^T u^T(t)Q(t)u(t) \le w, \tag{OC}$$

where $Q(t)$ are given *positive definite symmetric* matrices.

(OC) can be interpreted, e.g., as follows: $x(t)$ represents the position and the velocity of a rocket; $-c^T x$ is the height of the rocket at a state x (so that our goal is to maximize the height of the rocket at a given instant), equations in (S) represent the dynamics of the rocket, the control is responsible for the profile of the flight and and the left hand side of the constraint is the dissipated energy.

Exercise 3.1 [10] *1. Use* (S) *to express $x(T)$ via the control and convert* (OC) *in a quadratically constrained problem with linear objective w.r.t. the u-variables.*

2. Convert the resulting problem to a conic quadratic program

3. Pass to the resulting problem to its dual and find the optimal solution to the latter problem.

4. Assuming $w > 0$, prove that both the primal and the dual are strictly feasible. What are the consequences for the solvability status of the problems? Assuming, in addition, that $x_0 = 0$, what is the optimal value?

*5. *Assume that* (S), (OC) *form a finite-difference approximation to the continuous time Optimal Control problem*

$$c^T x(1) \to \min$$

s.t.

$$\begin{array}{rcl} \frac{d}{d\tau}x(\tau) & = & \alpha(\tau)x(\tau)+\beta(\tau)u(\tau), 0 \le \tau \le 1, x(0)=0 \\ \int_0^1 u^T(\tau)\gamma(\tau)u(\tau)d\tau & \le & w, \end{array}$$

$\gamma(\tau)$, for every $\tau \in [0,1]$, being a positive definite symmetric matrix.

Guess what should be the optimal value.

3.5.2 Around CQR's

Exercise 3.2 ^{5} *Let $\alpha_1, ..., \alpha_k$ be positive rational numbers.*

1. Assume that $\alpha_1 + ... + \alpha_k < 1$. Demonstrate that the set

$$\{(t, s_1, ..., s_k) \in \mathbf{R} \times \mathbf{R}^k_+ \mid t \leq f(s) \equiv s_1^{\alpha_1} s_2^{\alpha_2} ... s_k^{\alpha_k}\}$$

is CQr. What are the conclusions on convexity/concavity of the function $f(s)$, $\mathrm{Dom}\, f = \mathbf{R}^k_+$?

What happens with the CQ-representability when $\sum_i \alpha_i = 1$ and the denominators of α_i's are integral powers of 2? And what happens when the denominators are arbitrary positive integers[8]*?*

Hint: Modify properly the construction from Example 12

2. Demonstrate that the set

$$\{(t, s_1, ..., s_k) \in \mathbf{R}^n_{++} \mid t \geq g(s) \equiv s_1^{-\alpha_1} s_2^{-\alpha_2} ... s_k^{-\alpha_k}\}$$

is CQr. What are the conclusions on convexity/concavity of the function $g(s)$, $\mathrm{Dom}\, g = \mathbf{R}^k_{++}$?

Among important (convex) elementary functions, seemingly the only two which are *not* CQr are the exponent $\exp\{x\}$ and the minus logarithm $-\ln x$. In a sense, these are not two functions, but only one: CQ-representability deals with the geometry of the epigraph of a function, and the epigraphs of $-\ln x$ and $\exp\{x\}$, geometrically, are the same – we merely are looking at the same set from two different directions. Now, why the exponent is *not* CQr? The answer is intuitively clear: how could we represent a set given by a transcendental inequality by algebraic inequalities? A rigorous proof, however, requires highly nontrivial tools, namely, the Zaidenberg-Tarski Theorem:

> *Let B be a semialgebraic set in $\mathbf{R}^n \times \mathbf{R}^m$, i.e., a set given by a finite system of polynomial inequalities (strict as well as non-strict). Then the projection of B onto $\mathbf{R}^n$ also is semialgebraic.*

Now, by definition, a set $Q \subset \mathbf{R}^n$ is CQr if and only if it is the projection onto $\mathbf{R}^n$ of a specific semialgebraic set Q' of larger dimension – one given by a system of inequalities of the form $\{\| A_i x - b_i \|_2^2 \leq (p_i^T x - q_i)^2, p_i^T x - q_i \geq 0\}_{i=1}^N$. Therefore, assuming that the epigraph of the exponent is CQr, we would conclude that it is a semialgebraic set, which in fact is not the case.

Thus, the exponent, the minus logarithm (same as convex power functions with irrational exponentials, like x_+^{π}), are not captured by Conic Quadratic Programming. Let us, however, look at the funny construction as follows. As everybody knows,

$$\exp\{x\} = \lim_{k \to \infty} (1 + \frac{1}{k} x)^k.$$

Let us specify here k as an integral power of 2:

$$\exp\{x\} = \lim_{l \to \infty} f_l(x), \quad f_l(x) = (1 + 2^{-l} x)^{(2^l)}.$$

Note that every one of the functions $f(l)$ is CQr.

On the other hand, what is the exponent from the computational viewpoint? Something which does not exist! For a computer, the exponent is a "function" which is "well-defined" on a

[8] An open question

quite moderate segment – something like $-746 < x < 710$ (SUN does not understand numbers larger than `1.0e+309` and less than `1.0e-344`; MATLAB knows that $\exp\{709.7\} = 1.6550\mathrm{e}{+}308$ and $\exp\{-745\} = 4.9407\mathrm{e} - 324$, but believes that $\exp\{709.8\} =$ `Inf` and $\exp\{-746\} = 0$). And in this very limited range of values of x the "computer" exponent, of course, differs from the "actual one" – the former reproduces the latter with relative accuracy like 10^{-16}. Now the question:

Exercise 3.3 [5] *How large should be l in order for $f_l(\cdot)$ to be a "valid substitution" of the exponent in the computer, i.e. to approximate the latter in the segment $-746 < x < 710$ within relative inaccuracy 10^{-16}? What is the "length" of the CQR for such an f_l – how many additional variables and simple constraints of the type $s^2 \leq t$ do you need to get the CQR?*

Note that we can implement our idea in a smarter way. The approximation f_l of the exponent comes from the formula

$$\exp\{x\} = \left(\exp\{2^{-l}x\}\right)^{(2^l)} \approx \left(1 + 2^{-l}x\right)^{(2^l)},$$

and the quality of this approximation, l and the range $-a \leq x \leq b$ of values of x being given, depends on how well the exponent is approximated by its linearization in the segment $-2^{-l}A < x < 2^{-l}b$. What will happen when the linearization is replaced with a polynomial approximation of a higher order, i.e., when we use approximations

$$\exp\{x\} \approx g_l(x) = \left(1 + 2^{-l}x + \frac{1}{2}(2^{-l}x)^2\right)^{(2^l)}$$

or

$$\exp\{x\} \approx h_l(x) = \left(1 + 2^{-l}x + \frac{1}{2}(2^{-l}x)^2 + \frac{1}{6}(2^{-l}x)^3 + \frac{1}{24}(2^{-l}x)^4\right)^{(2^l)},$$

and so on? Of course, to make these approximation useful in our context, we should be sure that the approximations are CQr.

Exercise 3.4 [10]

1. Assume that $s(\cdot)$ is a nonnegative CQr function and you know its CQR (S). *Prove that for every rational $\alpha \geq 1$ the function $s^\alpha(\cdot)$ is CQr, the corresponding representation being readily given by* (S).

2. Prove that the polynomial $1 + t + t^2/2$ on the axis is nonnegative and is CQr. Find a CQR for the polynomial.

Conclude that the approximations $g_l(\cdot)$ are CQr. How large is a "sufficient" l (the one reproducing the exponent with the same quality as in the previous exercise) for these approximations? How many additional variables and constraints are needed?

3. Answer the same questions as in 2., but for the polynomial $1+t+t^2/2+t^3/6+t^4/24$ and the approximations $h_l(\cdot)$.

And now – the most difficult question. Here are the best numerical results we were able to obtain with the outlined scheme:

x	$\exp\{x\}$	Rel. error of f_{40}	Rel. error of g_{27}	Rel. error of h_{18}
-512	4.337e-223	4.e-6	5.e-9	5.e-11
-256	6.616e-112	2.e-6	3.e-9	8.e-12
-128	2.572e-56	1.e-6	1.e-9	1.e-11
-64	1.603e-28	5.e-7	6.e-10	9.e-12
-32	1.266e-14	2.e-7	1.e-10	1.e-11
-16	1.125e-07	1.e-7	2.e-10	1.e-11
-1	3.678e-01	7.e-9	7.e-9	1.e-11
1	2.718e+00	7.e-9	7.e-9	1.e-11
16	8.886e+06	1.e-7	2.e-10	2.e-11
32	7.896e+13	2.e-7	3.e-10	2.e-11
64	6.235e+27	5.e-7	6.e-10	2.e-11
128	3.888e+55	1.e-6	1.e-9	2.e-11
256	1.511e+111	2.e-6	2.e-9	3.e-11
512	2.284e+222	4.e-6	5.e-9	7.e-11

Exercise 3.5 [10] *Why the outlined scheme does not work on a computer, at least does not work as well as it is stated by the previous "analysis"?*

Around stable grasp

Recall that the Stable Grasp Analysis problem is to check whether the system of constraints

$$\begin{array}{rcl} \| F^i \|_2 & \leq & \mu(f^i)^T v^i, \ i=1,...,N \\ (v^i)^T F^i & = & 0, \ i=1,...,N \\ \sum_{i=1}^N (f^i + F^i) + F^{\rm ext} & = & 0 \\ \sum_{i=1}^N p^i \times (f^i + F^i) + T^{\rm ext} & = & 0 \end{array} \qquad \text{(SG)}$$

the variables being 3D vectors F^i, is or is not solvable. Here the data are given by a number of 3D vectors, namely,

- vectors v^i – unit inward normals to the surface of the body at the contact points;
- contact points p^i;
- vectors f^i – contact forces;
- vectors $F^{\rm ext}$ and $T^{\rm ext}$ of the external force and torque, respectively.

$\mu > 0$ is a given *friction coefficient*; we assume that $f_i^T v^i > 0$ for all i.

Exercise 3.6 [15] *1. Regarding* (SG) *as the system of constraints of a* <u>*maximization*</u> *program with trivial objective and applying the technique from Section 2.5, build the dual problem.*

2. Prove that the dual problem is strictly feasible. Derive from this observation that stable grasp is possible if and only if the dual objective is nonnegative on the dual feasible set.

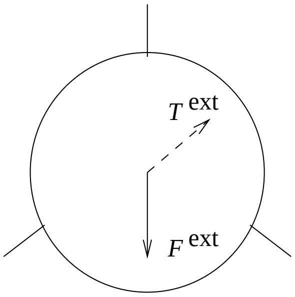
3. Assume that $\sum_{i=1}^N \mu[(f^i)^T v^i] <\| \sum_{i=1}^N f^i + F^{\rm ext}\}_2$. *Is a stable grasp possible?*

4. Let $T = \sum_{i=1}^N p^i \times f^i + T^{\rm ext}$, *and let* T^i *be the orthogonal projection of the vector* $p^i \times T$ *onto the plane orthogonal to* v^i. *Assume that*

$$\sum_{i=1}^N \mu[(f^i)^T v^i] \parallel T^i \parallel_2 < \parallel T \parallel_2^2 .$$

Is a stable grasp possible?

5. The data of the Stable Grasp problem are as follows:



(the "fingers" look at the center of the circle; the contact points are the vertices of the inscribed equilateral triangle). Magnitudes of all 3 contact forces are equal to each other, the friction coefficient is equal to 1, magnitudes of the external force and the external torque are equal to a; the torque is orthogonal to the plane of the picture. What is the smallest magnitude of contact forces which makes a stable grasp possible?

Around trusses

We are about to process the multi-load TTD problem 3.4.1, which we write down now as (cf. (3.4.19))

$$\begin{array}{rcl}\tau & \to & \min\\ s_{ij}^2 & \le & 4t_ir_{ij},\ i=1,...,n, j=1,...,k;\\ \sum_{i=1}^n r_{ij} & \le & \frac{1}{2}\tau,\ j=1,...,k;\\ \sum_{i=1}^n t_i & \le & w;\\ \sum_{i=1}^n s_{ij}b_i & = & f_j,\ j=1,...,k;\\ t_i, r_{ij} & \ge & 0,\ i=1,...,n, j=1,...,k,\end{array} \qquad (\mathrm{Pr})$$

the design variables being $s_{ij}, r_{ij}, t_i, \tau$; the variables σ_{ij} from (3.4.19) are twice the new variables r_{ij}.

Throughout this section we make the following two assumptions:

- The ground structure $(n, m, b_1, ..., b_n)$ is such that the matrix $\sum_{i=1}^n b_ib_i^T$ is positive definite;
- The loads of interest $f_1, ..., f_k$ are nonzero, and the material resource w is positive.

Exercise 3.7 [10] *1. Applying the technique from Section 2.5, build the dual to* (Pr) *problem* (Dl)*.*

Check that both (Pr) *and* (Dl) *are strictly feasible. What are the consequences for the solvability status of the problems and their optimal values?*

What is the design dimension of (Pr)*? The one of* (Dl) *?*

2. Convert problem (Dl) *into an equivalent problem of the design dimension* $mk + k + 1$*.*

Exercise 3.8 [10] *Let us fix a ground structure* $(n, m, b_1, ..., b_n)$ *and a material resource* w*, and let* $\mathcal{F}$ *be a finite set of loads.*

1. Assume that $\mathcal{F}_j \in \mathcal{F}$*,* $j = 1, ..., k$*, are subsets of* $\mathcal{F}$ *with* $\cup_{j=1}^k \mathcal{F}_j = \mathcal{F}$*. Let* μ_j *be the optimal value in the multi-load TTD problem with the set of loads* $\mathcal{F}_j$ *and* μ *be the optimal value in the multi-load TTD problem with the set of loads* $\mathcal{F}$*. Is it possible that* $\mu > \sum_{j=1}^k \mu_j$*?*

2. Assume that the ground structure includes $n = 1998$ tentative bars and that you are given a set $\mathcal{F}$ of $N = 1998$ loads. It is known that for every subset $\mathcal{F}'$ of $\mathcal{F}$ comprised of no more than 999 loads the optimal value in the multi-load TTD problem, the set of loading scenarios being $\mathcal{F}'$, does not exceed 1. What can be said about the optimal value in the multi-load TTD problem with the set of scenarios $\mathcal{F}$?

Answer similar question in the case when $\mathcal{F}$ is comprised of $N' = 19980$ loads.

Lecture 4

Semidefinite Programming

We are about to consider a generic conic program with really outstanding area of applications – the one of *Semidefinite Programming*.

4.1 Semidefinite cone and Semidefinite programs

4.1.1 Preliminaries

Let $\mathbf{S}^m$ be the space of symmetric $m\times m$ matrices, and $\mathbf{M}^{mn}$ be the space of rectangular $m\times n$ matrices with real entries. From the viewpoint of their linear structure (i.e., the operations of addition and multiplication by reals) $\mathbf{S}^m$ is just the arithmetic linear space $\mathbf{R}^{m(m+1)/2}$ of the dimension $\frac{m(m+1)}{2}$: by arranging the elements of a symmetric $m\times m$ matrix X in a single column, say, in the row-by-row order, you get a usual m^2-dimensional column vector; multiplication of a matrix by a real and addition of matrices correspond to the same operations with the "representing vector(s)". When A runs through $\mathbf{S}^m$, the vector representing A runs through $m(m+1)/2$-dimensional subspace of $\mathbf{R}^{m^2}$ comprised of vectors satisfying the "symmetry condition" – the coordinates coming from symmetric to each other pairs of entries in A are equal to each other. Similarly, $\mathbf{M}^{mn}$, as a linear space, is just $\mathbf{R}^{mn}$. It makes sense to equip the space $\mathbf{M}^{mn}$ with the inner product equal to the usual inner product of the vectors representing the matrices:

$$\langle X,Y\rangle = \sum_{i=1}^{m}\sum j = 1^n X_{ij}Y_{ij} = \mathrm{Tr}(X^TY),$$

where Tr stands for the trace – the sum of diagonal elements of a (square) matrix. Equipped with this inner product (called the *Frobenius inner product*, $\mathbf{M}^{mn}$ becomes a fully legitimate Euclidean space, and we may use in connection with this space different notions based upon the Euclidean structure – the (Frobenius) norm of a matrix

$$\| X \|_2 = \sqrt{\langle X,X\rangle} = \sqrt{\sum_{i,j=1}^{m} X_{ij}^2} = \sqrt{\mathrm{Tr}(X^TX)},$$

orthogonality, orthogonal complement of a linear subspace, etc. The space $\mathbf{S}^m$ equipped with the Frobenius inner product also becomes a Euclidean space; of course, the Frobenius inner product of symmetric matrices can be written down without any transposition sing:

$$\langle X,Y\rangle = \mathrm{Tr}(XY),\ X,Y\in\mathbf{S}^m.$$

Let us focus on the space $\mathbf{S}^m$. After it is equipped with the Frobenius inner product, we may speak about a cone dual to a given cone $\mathbf{K} \subset \mathbf{S}^m$:

$$\mathbf{K}_* = \{Y \in \mathbf{S}^m \mid \langle Y, x \rangle \geq 0 \quad \forall X \in \mathbf{K}\}.$$

Among the cones in $\mathbf{S}^m$, the one of especial interest is the *positive semidefinite cone* $\mathbf{S}^m_+$ – the one comprised of all symmetric positive semidefinite matrices[1]. It is easily seen (see Exercise 2.8) that $\mathbf{S}^m_+$ indeed is a cone and that this cone is self-dual:

$$(\mathbf{S}^m_+)_* = \mathbf{S}^m_+.$$

The interior $\mathbf{S}^m_{++}$ of the semidefinite cone $\mathbf{S}^m_+$ is comprised of *positive definite* symmetric $m \times m$ matrices – symmetric matrices A for which $x^T Ax > 0$ for all nonzero vectors x, or, which is the same, the symmetric matrices with positive eigenvalues.

As any other, the semidefinite cone gives rise to a family of conic programs "minimize a linear objective over the intersection of the semidefinite cone and an affine plane"; these are the *Semidefinite programs* we are about to investigate.

Before writing down a generic semidefinite program, we should resolve a small difficulty with notation. Normally we use lowercase Latin and Greek letters to denote vectors, and the uppercase letters – to denote matrices; e.g., in our usual notation a conic problem of the form (CP) looks like

$$c^T x \to \min \mid Ax - b \geq_{\mathbf{K}} 0. \tag{CP}$$

When trying to follow this notation in the case of Semidefinite programs, i.e., those with $\mathbf{K} = \mathbf{S}^m_+$, we get a collision with the notation related to the space where $\mathbf{S}^m_+$ lives. Look at (CP): without additional remarks it is unclear what is A – is it a $m \times m$ matrix from the space $\mathbf{S}^m$ or is it a linear mapping acting from the space of the design vectors – some $\mathbf{R}^n$ – to the space $\mathbf{S}^m$? When speaking about a conic problem on the cone $\mathbf{S}^m_+$, we should have in mind the second interpretation of A, while the standard notation in (CP) suggests the first – wrong! – interpretation. In other words, we meet with the necessity to distinguish between *linear mappings* acting to/from $\mathbf{S}^m$ and elements of $\mathbf{S}^m$ (which themselves are linear mappings from certain linear space to itself); in order to resolve the problem, we from now on make the following

Notational convention: In order to denote a linear mapping acting from a linear space to a space of matrices (or from a space of matrices to a linear space), we use uppercase script letters like $\mathcal{A}$, $\mathcal{B}$,... Elements of usual vector spaces $\mathbf{R}^n$ are, as always, denoted by lowercase Latin/Greek letters $a, b, ..., z, \alpha, ..., \zeta$, while elements of a space of matrices usually are denoted by uppercase Latin letters $A, B, ..., Z$. According to this convention, a semidefinite program of the form (CP) should be written down as

$$c^T x \to \min \mid \mathcal{A}x - B \geq_{\mathbf{S}^m_+} 0. \tag{$*$}$$

It also makes sense to simplify the sign $\geq_{\mathbf{S}^m_+}$ to $\succeq$ and the sign $>_{\mathbf{S}^m_+}$ to $\succ$ (same as we write $\geq$ instead of $\geq_{\mathbf{R}^m_+}$ and $>$ instead of $>_{\mathbf{R}^m_+}$). Thus, $A \succeq B$ ($\Leftrightarrow B \preceq A$) means that A and B are symmetric matrices of the same size and $A - B$ is positive semidefinite, while $A \succ B$ ($\Leftrightarrow B \prec A$) means that A, B are symmetric matrices of the same size with positive definite $A - B$.

[1] Recall that a symmetric $n \times n$ matrix A is called positive semidefinite if $x^T Ax \geq 0$ for all $x \in \mathbf{R}^m$; an equivalent definition is that all eigenvalues of A are nonnegative

We also need a special notation for the conjugate ("transposed") of a linear mapping $\mathcal{A}$ acting from/to a space of matrices. Recall that the conjugate to a linear mapping $\Xi : E \to F$ acting from a Euclidean space $(E, (\cdot,\cdot)_E)$ to a Euclidean space $(F, (\cdot,\cdot)_F)$ is the mapping $\Xi' : F \to E$ satisfying the identity

$$(\Xi e, f)_F = (e, \Xi' f)_E \quad \forall e \in E, f \in F;$$

when E and F are the usual coordinate spaces $\mathbf{R}^k$ and $\mathbf{R}^l$ equipped with the standard inner product $(x,y) = x^T y$, so that X and X' can be naturally identified with $k \times l$ and $l \times k$ matrices, respectively, these matrices are transposed to each other, and we can write X^T instead of X'. In the case when one of the spaces E, F (or both of them) are spaces of matrices, the notation Ξ^T for Ξ' comes into a collision with the notation for the transpose of an element from E or/and F. This is why, when speaking about a linear mapping $\mathcal{A}$ acting to/from a space of matrices, we denote it conjugate by $\mathcal{A}^*$.

Our last convention is how to write down expressions of the type $A\mathcal{A}xB$ ($\mathcal{A}$ is a linear mapping from some $\mathbf{R}^n$ to $\mathbf{S}^m$, $x \in \mathbf{R}^n$ $A, B \in \mathbf{S}^m$); what we are trying to denote is the result of the following operation: we first take the value $\mathcal{A}x$ of the mapping $\mathcal{A}$ at a vector x, thus getting a $m \times m$ matrix $\mathcal{A}x$, and then multiply this matrix from the left and from the right by the matrices A, B. In order to avoid misunderstandings, we write expressions of this type as

$$A[\mathcal{A}x]B$$

or as $A\mathcal{A}(x)B$, or as $A\mathcal{A}[x]B$.

How to specify a mapping $\mathcal{A} : \mathbf{R}^n \to \mathbf{S}^m$. A natural data specifying a linear mapping $A : \mathbf{R}^n \to \mathbf{R}^m$ is a collection of n elements of the "destination space" – n m-dimensional vectors $a_1, a_2, ..., a_n$ – such that

$$Ax = \sum_{j=1}^n x_j a_j, \;\; x = (x_1, ..., x_n)^T \in \mathbf{R}^n.$$

Similarly, a natural data specifying a linear mapping $\mathcal{A} : \mathbf{R}^n \to \mathbf{S}^m$ is a collection $A_1, ..., A_n$ of n $m \times m$ symmetric matrices such that

$$\mathcal{A}x = \sum_{j=1}^n x_j A_j, \;\; x = (x_1, ..., x_n)^T \in \mathbf{R}^n.$$

Via these data, a semidefinite program (*) can be written as

$$c^T x \to \min \mid x_1 A_1 + x_2 A_2 + ... + x_n A_n - B \succeq 0. \tag{SDPr}$$

Linear Matrix Inequality constraints and semidefinite programs. In the case of conic quadratic problems, we started with the simplest program of this type – the one with a single conic quadratic constraint $Ax - b \geq_{\mathbf{L}^m} 0$ – and then said that a conic quadratic program is a program with finitely many constraints of this type – it is a conic program on a *direct product* of the Lorentz cones. In contrast to this, when defining a semidefinite program, we impose on the design vector just one *Linear Matrix Inequality* (LMI) $\mathcal{A}x - B \succeq 0$. Now we indeed should not bother about more than a single LMI, due to the following simple fact:

A system of finitely many LMI's

$$\mathcal{A}_i x - B_i \succeq 0,\ i = 1, ..., k,$$

(of the same or of different sizes) is equivalent to the single *LMI*

$$\mathcal{A}x - B \succeq 0,$$

with

$$\mathcal{A}x = \mathrm{Diag}\left(\mathcal{A}_1 x, \mathcal{A}_2 x, ..., \mathcal{A}_k x\right), B = \mathrm{Diag}(B_1, ..., B_k);$$

here for a collection of symmetric matrices $Q_1, ..., Q_k$

$$\mathrm{Diag}(Q_1, ..., Q_k) = \begin{pmatrix} Q_1 & & \\ & \ddots & \\ & & Q_k \end{pmatrix}$$

is the block-diagonal matrix with the diagonal blocks $Q_1, ..., Q_k$.

Indeed, a block-diagonal symmetric matrix is positive (semi)definite if and only if all its diagonal blocks are so.

Dual to a semidefinite program (SDP). The dual to a general conic problem (CP) is, as we know, the problem

$$b^T\lambda \to \max \mid\ A^T\lambda = c,\ \lambda \geq_{\mathbf{K}^*} 0;$$

the matrix A^T is nothing but the linear mapping conjugate to the mapping A involved into the primal problem. When writing down the problem dual to (SDPr), we should

– replace $b^T\lambda$ – the usual inner product in $\mathbf{R}^n$ – with the Frobenius inner product (this is how the Euclidean structure on $\mathbf{S}^m$ is defined);

– to follow our notational convention and to write $\mathcal{A}^*$ instead of $\mathcal{A}^T$;

– to take into account that the semidefinite cone is self-dual.

Consequently, the problem dual to (SDPr) is the semidefinite program

$$\mathrm{Tr}(B\Lambda) \to \max \mid\ \mathcal{A}^*\Lambda = c,\ \Lambda \succeq 0.$$

Now, let $A_1, ..., A_n$ be the data specifying $\mathcal{A}$. How $\mathcal{A}^*$ acts? The answer is immediate:

$$\mathcal{A}^*\Lambda = (\mathrm{Tr}(A_1\Lambda), \mathrm{Tr}(A_2\Lambda), ..., \mathrm{Tr}(A_n\Lambda))^T.$$

Indeed, what we should verify that with the above definition we do have

$$[\mathrm{Tr}([\mathcal{A}x]\Lambda) =] \quad \langle \mathcal{A}x, \Lambda\rangle = (\mathcal{A}^*\Lambda)^T x \quad \forall \Lambda \in \mathbf{S}^m, x \in \mathbf{R}^n,$$

which is immediate:

$$\begin{array}{rcl} \mathrm{Tr}([\mathcal{A}x]\Lambda) & = & \mathrm{Tr}\left((\sum_{j=1}^n x_j A_j)\Lambda\right) \\ & = & \sum_{i=1}^n x_j\, \mathrm{Tr}(A_j\Lambda) \\ & = & (\mathrm{Tr}(A_1\Lambda), ..., \mathrm{Tr}(A_n\Lambda)) \begin{pmatrix} x_1 \\ ... \\ x_n \end{pmatrix}. \end{array}$$

We see that the "explicit" – without the sign * – form of the problem dual to (SDPr) is the semidefinite program

$$\mathrm{Tr}(B\Lambda) \to \max \mid\ \mathrm{Tr}(A_i\Lambda) = c_i,\ i = 1, ..., n; \Lambda \succeq 0. \tag{SDDl}$$

Conic Duality in the case of Semidefinite Programming. Same as it was done for conic quadratic programs, it makes sense to say explicitly when in the case of semidefinite programs we may use at "full power" the results of the Conic Duality Theorem. To this end let us note that our default (always in action!) assumption **A** on a conic program in the form of (CP) (Lecture 2) as applied to (SDPr) says that no nontrivial linear combination of the matrices $A_1, ..., A_n$ is 0. Strict feasibility of (SDPr) means that there exist x such that $\mathcal{A}x - B$ is positive definite, and strict feasibility of (SDDl) means that there exists a positive definite Λ satisfying $\mathcal{A}^*\Lambda = c$. According to the Conic Duality Theorem, if both primal and dual are strictly feasible, both are solvable, the optimal values being equal to each other, and the complementary slackness condition

$$[\mathrm{Tr}(\Lambda[\mathcal{A}x - B]) \equiv] \qquad \langle \Lambda, \mathcal{A}x - B\rangle = 0$$

is necessary and sufficient for a pair comprised of primal feasible solution x and dual feasible solution Λ to be comprised of optimal solutions to the corresponding problems.

It is easily seen (see Exercise 2.8) that for a pair X, Y of positive semidefinite symmetric matrices one has

$$\mathrm{Tr}(XY) = 0 \Leftrightarrow XY = YX = 0;$$

in particular, in the case of strictly feasible primal and dual problems the "primal slack" $S_* = \mathcal{A}x^* - B$ corresponding to a primal optimal solution commutates with (any) dual optimal solution Λ_*, and the product of these two matrices is 0. Besides this, S_* and Λ_*, as a pair of commutating symmetric matrices, share a common eigenbasis, and the fact that $S_*\Lambda_* = 0$ means that the eigenvalues of the matrices in this basis are "complementary": for every common eigenvector, either the eigenvalue of S_*, or the one of Λ_*, or both, are equal to 0 (cf. with complementary slackness in the LP case).

4.2 What can be expressed via LMI's?

Same as in the previous lecture, the first thing to be realized when speaking about "semidefinite programming universe" is how wide is it and how to recognize that a convex optimization program

$$c^T x \to \min \mid x \in X = \cap_{i=1}^m X_i \tag{P}$$

can be cast as a semidefinite program. And same as in the previous lecture, these questions actually ask when a given convex set/convex function are PDr – positive semidefinite representable. The definition of the latter notion is completely similar to the one of a CQr set/function:

> *We say that a convex set $X \subset \mathbf{R}^n$ is PDr, if there exists an affine mapping $(x, u) \to \mathcal{A}\begin{pmatrix} x \\ u \end{pmatrix} - B : \mathbf{R}^n_x \times \mathbf{R}^k_u \to \mathbf{S}^m$ such that*
>
> $$x \in X \Leftrightarrow \exists u : \mathcal{A}\begin{pmatrix} x \\ u \end{pmatrix} - B \succeq 0;$$
>
> *in other words, we say that X is PDr, if there exists LMI*
>
> $$\mathcal{A}\begin{pmatrix} x \\ u \end{pmatrix} - B \succeq 0,$$
>
> *the variables in the LMI being the original design vector x and a vector u of additional design variables such that X is a projection of the solution set of the LMI onto the*

x-space. An LMI with this property is called Semidefinite representation (SDR) *of the set X.*

A convex function $f : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ is called SDr, if its epigraph

$$\{(x,t) \mid t \geq f(x)\}$$

is a SDr set. A SDR of the epigraph of f is called semidefinite representation of f.

By exactly the same reasons as in the case of conic quadratic problems, one has:

1. *If f is a SDr function, then all its level sets $\{x \mid f(x) \leq a\}$ are SDr, SDR's of the level sets being explicitly given by (any) SDR of f;*
2. *If all the sets X_i in problem* (P) *are SDr with known SDR's, the problem can be explicitly converted to a semidefinite program.*

In order to understand which functions/sets are SDr, we may use the same approach as in Lecture 3. Moreover, "the calculus" – the list of basic rules preserving SD-representability, is exactly the same as in the case of conic quadratic problems – you just may repeat word by word the relevant reasoning from Lecture 3. Thus, the only issue to be addressed is the one of "raw materials" – of a catalogue of "simple" SDr functions/sets. Our first observation in this direction is as follows:

1-14. [2)] *If a function/set is CQr, it is also SDr, and any CQR of the function/set can be explicitly converted to its SDR.*

Indeed, the notion of a CQr/SDr function is a "derivative" of the notion of a CQr/SDr set: by definition, a function is CQr/SDr if and only if its epigraph is so. Now, CQr sets are exactly those sets which can be obtained as projections of the solution sets of systems of conic quadratic inequalities, i.e., as projections of inverse images, under affine mappings, of direct products of the Lorentz cones. Similarly, SDr sets are projections of the inverse images, under affine mappings, of positive semidefinite cones. Consequently,

(i) in order to verify that a CQr set is SDr as well, it suffices to show that an inverse image, under an affine mapping, of a direct product of the Lorentz cones – a set of the form

$$Z = \{z \mid Az - b \in \mathbf{K} = \prod_{i=1}^{l} \mathbf{L}^{k_i}\}$$

is the inverse image of a semidefinite cone under affine mapping. To this end, in turn, it suffices to demonstrate that

(ii) a direct product $\mathbf{K} = \prod_{i=1}^{l} \mathbf{L}^{k_i}$ of the Lorentz cones is an inverse image of a semidefinite cone under an affine mapping.

Indeed, representing $\mathbf{K}$ as $\{y \mid \mathcal{A}y - b \in \mathbf{S}^m_+\}$, we get

$$Z = \{z \mid Ax - b \in \mathbf{K}\} = \{z \mid \hat{A}z - \hat{B} \in \mathbf{S}^m_+\},$$

where $\hat{A}z - \hat{B} = \mathcal{A}(Az - b) - B$ is affine.

In turn, in order to prove (ii) it suffices to show that

(iii) A Lorentz cone $\mathbf{L}^k$ is an inverse image of a semidefinite cone under an affine mapping.

[2)] We refer to Examples 1-14 of CQ-representable functions/sets from Section 3.3

In fact the implication (iii) $\Rightarrow$ (ii) is given by our calculus – a direct product of SDr sets again is SDr. Just to recall where the calculus comes from, here is a direct verification:

Given a direct product $\mathbf{K} = \prod_{i=1}^{l} \mathbf{L}^{k_i}$ of Lorentz cones and given that every factor in the product is the inverse image of a semidefinite cone under an affine mapping:

$$\mathbf{L}^{k_i} = \{x_i \in \mathbf{R}^{k_i} \mid \mathcal{A}_i x_i - B_i \succeq 0\},$$

we can represent $\mathbf{K}$ as the inverse image of a semidefinite cone under an affine mapping, namely, as

$$\mathbf{K} = \{x = (x_1, ..., x_l) \in \mathbf{R}^{k_1} \times ... \times \mathbf{R}^{k_l} \mid \mathrm{Diag}(\mathcal{A}_1 x_i - B_1, ..., \mathcal{A}_l x_l - B_l) \succeq 0.\}$$

We have reached the point where no more reductions are necessary, and here is the demonstration of (iii). The case of $k = 1$ is trivial: the 1-dimensional Lorentz cone is just identical to the 1-dimensional semidefinite cones – both are the same nonnegative ray on the axis! In the case of $k > 1$ the issue is resolved by the following observation:

$$\begin{pmatrix} x \\ t \end{pmatrix} \in \mathbf{L}^k \Leftrightarrow A(x,t) = \begin{pmatrix} tI_{k-1} & x \\ x^T & t \end{pmatrix} \succeq 0 \qquad (4.2.1)$$

(x is $k-1$-dimensional, t is scalar, I_{k-1} is the $(k-1) \times (k-1)$ unit matrix). (4.2.1) would resolve the problem – the matrix $A(x,t)$ is linear in (x,t)!

It remains to verify (4.2.1), which is immediate. If $(x,t) \in \mathbf{L}^k$, i.e., if $\| x \|_2 \leq t$, then for every $y = \begin{pmatrix} \xi \\ \tau \end{pmatrix} \in \mathbf{R}^k$ (ξ is $(k-1)$-dimensional, τ is scalar) we have

$$\begin{array}{rcl} y^T A(t,x) y & = & \tau^2 t + 2\tau x^T \xi + t\xi^T \xi \geq \tau^2 t - 2|\tau| \| x \|_2 \| \xi \|_2 + t \| \xi \|_2^2 \\ & \geq & t\tau^2 - 2t|\tau| \| \xi \|_2 + t \| \xi \|_2^2 \\ & \geq & t(|\tau| - \| \xi \|_2)^2 \geq 0, \end{array}$$

so that $A(t,x) \succeq 0$. Vice versa, if $A(t,x) \succeq 0$, then of course $t \geq 0$. Assuming $t = 0$, we immediately obtain $x = 0$ (since otherwise for $y = \begin{pmatrix} x \\ 0 \end{pmatrix}$ we would have $0 \leq y^T A(t,x) y = -2 \| x \|_2^2$); thus, $A(t,x) \succeq 0$ implies $\| x \|_2 \leq t$ in the case of $t = 0$. To see that the same implication is valid in the case of $t > 0$ as well, let us set $y = \begin{pmatrix} -x \\ t \end{pmatrix}$ to get

$$0 \leq y^T A(t,x) y = tx^T x - 2tx^T x + t^3 = t(t^2 - x^T x),$$

i.e., to get $\| x \|_2 \leq t$, as claimed. ■

We see that the "expressive abilities" of semidefinite programming majorate those (already very high) of Conic Quadratic programming. In fact the gap here is quite significant. The first new possibility we get is to handle eigenvalues, and the importance of this possibility can hardly be overestimated.

SD-representability of functions of eigenvalues of symmetric matrices. Our first eigenvalue-related observation is as follows:

15. The largest eigenvalue $\lambda_{\max}(X)$ regarded as a function of $m \times m$ symmetric matrix X is SDr. Indeed, the epigraph of this function

$$\{(X,t) \in \mathbf{S}^m \times \mathbf{R} \mid \lambda_{\max}(X) \leq t\}$$

is given by the LMI

$$tI_m - X \succeq 0,$$

I_k being the unit $k \times k$ matrix.

Indeed, the eigenvalues of $tI_m - X$ are t minus the eigenvalues of X, so that the matrix $tI_m - X$ is positive semidefinite – all its eigenvalues are nonnegative – if and only if t majorates all eigenvalues of X.

The outlined example admits a natural generalization. Let M, A be two symmetric matrices of the same row size m, and let M be positive definite. The eigenvalues of the *pencil* $[M, A]$ are reals λ such that the matrix $\lambda M - A$ is singular. It is clear that the eigenvalues of the pencil are the usual eigenvalues of the matrix $M^{-1/2}AM^{-1/2}$:

$$\det(\lambda M - A) = 0 \Leftrightarrow \det(M^{1/2}(\lambda I_m - M^{-1/2}AM^{-1/2})M^{1/2}) = 0 \Leftrightarrow \det(\lambda I_m - M^{-1/2}AM^{-1/2}) = 0.$$

In particular, the usual eigenvalues of A are exactly the eigenvalues of the pencil $[I_m, A]$.

The announced extension of Example 15 is as follows:

15a. <u>[The maximum eigenvalue of a pencil]</u>: Let M be a positive definite symmetric $m \times m$ matrix, and let $\lambda_{\max}(X : M)$ be the largest eigenvalue of the pencil $[M, X]$, where X is symmetric $m \times m$ matrix. The inequality

$$\lambda_{\max}(X : M) \le t$$

is equivalent to the matrix inequality

$$tM - X \succeq 0.$$

In particular, $\lambda_{\max}(X : M)$, regarded as a function of X, is SDr.

15b. <u>The spectral norm $|X|$ of a symmetric $m \times m$ matrix X</u> – the maximum of <u>modulae</u> of the eigenvalues of X – is SDr. Indeed, a SDR of the epigraph

$$\{(X, t) \mid |X| \le t\} = \{(X, t) \mid \lambda_{\max}(X) \le t, \lambda_{\max}(-X) \le t\}$$

of the function is given by the pair of LMI's

$$tI_m - X \succeq 0,\ tI_m + X \succeq 0.$$

In spite of their simplicity, the indicated results are extremely useful. Let us give a more complicated example – a SDr for the sum of k *largest* eigenvalues of a symmetric matrix.

From now on, speaking about $m \times m$ symmetric matrix X, we denote by $\lambda_i(X)$, $i = 1, ..., m$, its eigenvalues *counted with their multiplicities* and *written down in the non-ascending order:*

$$\lambda_1(X) \ge \lambda_2(X) \ge ... \ge \lambda_m(X).$$

The vector comprised of the eigenvalues (in the indicated order) will be denoted $\lambda(X)$:

$$\lambda(X) = (\lambda_1(X), ..., \lambda_m(X))^T \in \mathbf{R}^m.$$

The question we are about to address is which functions of the eigenvalues are SDr. We already know that it is the case for the largest eigenvalue $\lambda_1(X)$. Another eigenvalues are <u>not</u> SDr by a very serious reason – they are not convex functions of X! And convexity, of course, is a necessary condition of SD-representability (cf. Lecture 3). It turns out, however, that the m functions

$$S_k(X) = \sum_{i=1}^{k} \lambda_i(X),\ k = 1, ..., m,$$

<u>are</u> convex and, moreover, are SDr:

15c. Sums of largest eigenvalues of a symmetric matrix. Let X be $m \times m$ symmetric matrix, and let $k \leq m$. Then the function $S_k(X)$ is SDr. Namely, the epigraph

$$\{(X,t) \mid S_k(x) \leq t\}$$

of the function admits the SDR

$$\begin{array}{lrcl} (a) & t - ks - \mathrm{Tr}(Z) & \geq & 0 \\ (b) & Z & \succeq & 0 \\ (c) & Z - X + sI_m & \succeq & 0 \end{array} \qquad (4.2.2)$$

where $Z \in \mathbf{S}^m$ and $s \in \mathbf{R}$ are additional variables.

> We should prove that
>
> (i) If a given pair X, t can be extended, by properly chosen s, Z, to a solution of the system of LMI's (4.2.2), then $S_k(X) \leq t$;
>
> (ii) Vice versa, if $S_k(X) \leq t$, then the pair X, t can be extended, by properly chosen s, Z, to a solution of (4.2.2).
>
> To prove (i), let us use the following basic fact (see Exercise 4.5.(i)):
>
> **(W)** *The vector $\lambda(X)$ is a monotone function of $X \in \mathbf{S}^m$, the space of symmetric matrices being equipped with the order $\succeq$:*
>
> $$X \succeq X' \Leftarrow \lambda(X) \geq \lambda(X').$$
>
> Assuming that (X, t, s, Z) is a solution to (4.2.2), we get $X \preceq Z + sI_m$, so that
>
> $$\lambda(X) \leq \lambda(Z + sI_m) = \lambda(Z) + s \begin{pmatrix} 1 \\ \dots \\ 1 \end{pmatrix},$$
>
> whence
>
> $$S_k(X) \leq S_k(Z) + sk.$$
>
> Since $Z \succeq 0$ (see (4.2.2.(b))), we have $S_k(Z) \leq \mathrm{Tr}(Z)$, and combining these inequalities we get
>
> $$S_k(X) \leq \mathrm{Tr}(Z) + sk.$$
>
> The latter inequality, in view of (4.2.2.(a))), implies $S_k(X) \leq t$. (i) is proved.
>
> To prove (ii), assume that we are given X, t with $S_k(X) \leq t$, and let us set $s = \lambda_k(X)$. Then the k largest eigenvalues of the matrix $X - sI_m$ are nonnegative, and the remaining are nonpositive. Let Z be a symmetric matrix with the same eigenbasis as X and such that the k largest eigenvalues of Z are the same as those of $X - sI_m$, and the remaining eigenvalues are zeros. The matrices Z and $Z - X + sI_m$ clearly are positive semidefinite (the first – by construction, and the second – since in the eigenbasis of X this matrix is diagonal with the first k diagonal entries being 0 and the remaining being the same as those of the matrix $sI_m - X$, i.e., being nonnegative). Thus, the matrix Z and the real s we have built satisfy (4.2.2.$(b), (c)$). In order to see that (4.2.2.(a)) is satisfied as well, note that by construction $\mathrm{Tr}(Z) = S_k(x) - sk$, whence $t - sk - Tr(Z) = t - S_k(x) \geq 0$.

In order to proceed, we need the following highly useful technical result:

Lemma 4.2.1 [Lemma on the Schur Complement] *Let*

$$A = \begin{pmatrix} B & C^T \\ C & D \end{pmatrix}$$

be a symmetric matrix with diagonal $k \times k$ block B and diagonal $l \times l$ block D. Assume that the North-Western block B is positive definite. Then A is positive (semi)definite if and only if the matrix

$$D - CB^{-1}C^T$$

is positive (semi)definite (this matrix is called the Schur complement of B in A).

Proof. The positive semidefiniteness of A is equivalent to the fact that

$$0 \le (x^T, y^T) \begin{pmatrix} B & C^T \\ C & D \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = x^T Bx + 2x^T C^T y + y^T Dy \quad \forall x \in \mathbf{R}^k, y \in \mathbf{R}^l,$$

or, which is the same, to the fact that

$$\inf_{x \in \mathbf{R}^k} \left[x^T Bx + 2x^T C^T y + y^T Dy \right] \ge 0 \quad \forall y \in \mathbf{R}^l.$$

Since B is positive definite, the infimum in x can be computed explicitly: the optimal x is $-B^{-1}C^T y$, and the optimal value is

$$y^T Dy - y^T CB^{-1}C^T y = y^T [D - CB^{-1}C^T] y.$$

As we have seen, the positive semidefiniteness of A is equivalent to the fact that the latter expression is nonnegative for every y, i.e., to the positive semidefiniteness of the Schur complement of B in A. The proof with evident modifications works also for the case when the question is when A is positive definite. ■

15d. "Determinant" of a symmetric positive semidefinite matrix. Let X be a symmetric positive semidefinite $m \times m$ matrix. Although its determinant

$$\mathrm{Det}\,(X) = \lambda_1(X)...\lambda_m(X)$$

is neither convex nor concave function of X (if $m > 2$), it turns out that the function $\mathrm{Det}^{1/p}(X)$ is concave in X whenever $p \ge m$. This function is important in many volume-related problems (see below); we are about to prove that

> *if $p > 2m$ is an integer (or if $p = 2m$ is an integral power of 2), the convex function*
>
> $$f_p(X) = \begin{cases} -\mathrm{Det}^{1/p}(X), & X \succeq 0 \\ +\infty, & \textit{otherwise} \end{cases}$$
>
> *is SDr.*
>
> Consider the following system of LMI's:
>
> $$\begin{pmatrix} I_m & \Delta^T \\ \Delta & X \end{pmatrix} \succeq 0, \mathrm{Dg}(\Delta) \succeq 0, \qquad \text{(D)}$$
>
> where Δ is $m \times m$ lower-triangular matrix comprised of additional variables and $\mathrm{Dg}(\Delta)$ denotes the diagonal matrix with the same diagonal entries as those of Δ.
>
> Now, as we know from Lecture 3 (see Exercise 3.2), the set
>
> $$\{(\delta, t) \in \mathbf{R}^m_+ \times \mathbf{R} \mid t \le (\delta_1, ..., \delta_m)^{2/p}\}$$
>
> admits an explicit CQ-representation. Consequently, this set admits an explicit SDR as well. The latter SDR is given by certain LMI $S(\delta, t; u) \succeq 0$, where u is the vector of additional variables of the SDR, and $S(\delta, t, u)$ is a matrix affinely depending on the arguments. We claim that

(!) *The system of LMI's* (D) & $S(\mathrm{Dg}(\Delta),t;u)\succeq 0$ *is a SDR for the set*

$$\{(X,t)\mid X\succeq 0, t\le \mathrm{Det}^{1/p}(X)\},$$

i.e., basically, for the epigraph of the function f_p (the epigraph is obtained from our set by reflection with respect to the plane $t=0$).

To support our claim, note that if $t\le \mathrm{Det}^{1/p}(X)$, then

(1) We can extend X by appropriately chosen lower triangular Δ with a nonnegative diagonal $\delta=\mathrm{Dg}(\Delta)$ to a solution of (D) in such a way that $\prod_{i=1}^m \delta_i=\mathrm{Det}^{1/2}(X)$;

> Indeed, it suffices to take as Δ the Choleski factor of X: [3] Since $X=\Delta\Delta^T$, Lemma on the Schur Complement says that (D) indeed is satisfied. And of course $X=\Delta\Delta^T\Rightarrow \mathrm{Det}(\Delta)=\mathrm{Det}^{1/2}(X)$.

(2) Since $\delta=\mathrm{Dg}(\Delta)\ge 0$ and $\prod_{i=1}^m\delta_i=\mathrm{Det}^{1/2}(X)$, we get $t\le \mathrm{Det}^{1/p}(X)=\left(\prod_{i=1}^m\delta_i\right)^{2/p}$, so that we can extend (t,δ) by a properly chosen u to a solution of the LMI $S(\mathrm{Dg}(\Delta),t;u)\succeq 0$.

We conclude that if $X\succeq 0$ and $t\le \mathrm{Det}^{1/p}(X)$, then one can extend the pair X,t by properly chosen Δ and u to a solution of the LMI (D) & $S(\mathrm{Dg}(\Delta),t;u)\succeq 0$, which is the first part of the proof of (!).

To complete the proof of (!), it suffices to demonstrate that if for a given pair X,t there exist Δ and u such that (D) and the LMI $S(\mathrm{Dg}(\Delta),t;u)\succeq 0$ are satisfied, then X is positive semidefinite and $t\le \mathrm{Det}^{1/p}(X)$. This is immediate: denoting $\delta=\mathrm{Dg}(\Delta)$ $[\ge 0]$ and applying Lemma on the Schur Complement, we conclude that $X\succeq \Delta\Delta^T$. Applying **(W)**, we get $\lambda(X)\ge\lambda(\Delta\Delta^T)$, whence of course $\mathrm{Det}(X)\ge \mathrm{Det}^2(\Delta)=\left(\prod_{i=1}^m\delta_i\right)^2$. Thus, $\prod_{i=1}^m\delta_i\le \mathrm{Det}^{1/2}(X)$. On the other hand, the LMI $S(\delta,t;u)\succeq 0$ takes place, which means that $t\le\left(\prod_{i=1}^m\delta_i\right)^{2/p}$. Combining the resulting inequalities, we come to $t\le \mathrm{Det}^{1/p}(X)$, as required. ■

SD-representability of functions of singular values. Consider the space $\mathbf{M}^{kl}$ of $k\times l$ rectangular matrices and assume that $k\le l$. Given a matrix $A\in\mathbf{M}^{kl}$, let us build the positive semidefinite $k\times k$ matrix $(AA^T)^{1/2}$; its eigenvalues are called *singular values* of A and are denoted by $\sigma_1(A),...\sigma_k(A)$: $\sigma_i(A)=\lambda_i((AA^T)^{1/2})$. According to convention of how we enumerate eigenvalues of a symmetric matrix, the singular values form a non-ascending sequence:

$$\sigma_1(A)\ge\sigma_2(A)\ge...\ge\sigma_k(A).$$

Importance of the singular values comes from the Singular Value Decomposition Theorem which states that a rectangular $k\times l$ matrix A $(k\le l)$ always can be represented as

$$A=\sum_{i=1}^k\sigma_i(A)e_if_i^T,$$

where $\{e_i\}_{i=1}^k$ and $\{f_i\}_{i=1}^k$ are orthonormal sequences in $\mathbf{R}^k$ and $\mathbf{R}^l$, respectively; this is a surrogate of the eigenvalue decomposition of a symmetric $k\times k$ matrix

$$A=\sum_{i=1}^k\lambda_i(A)e_ie_i^T,$$

[3] Recall that Linear Algebra says that a positive semidefinite symmetric matrix X can be factorized as $X=\Delta\Delta^T$ with lower triangular Δ, $\mathrm{Dg}(\Delta)\ge 0$; the resulting Δ is called the Choleski factor of X.

where $\{e_i\}_{i=1}^k$ form an orthonormal eigenbasis of A.

Among the singular values of a rectangular matrix, the most important is the largest – the first $\sigma_1(A)$. This is nothing but the *operator* (or *spectral*) *norm* of A – the quantity

$$|A| = \max\{\| Ax \|_2 | \| x \|_2 \leq 1\}.$$

For a symmetric matrix, the singular values are nothing but the modulae of the eigenvalues and our new definition of the norm coincides with the already given one.

It turns out that the sum of a given number of the largest singular values of A

$$\Sigma_p(A) = \sum_{i=1}^p \sigma_i(A)$$

is a convex and, moreover, a SDr function of A. In particular, the operator norm of A is SDr representable:

16. The sum of p largest singular values of a rectangular matrix $X \in \mathbf{M}^{kl}$ is SDr. In particular, the operator norm of a rectangular matrix is SDr:

$$|X| \leq t \Leftrightarrow \begin{pmatrix} tI_l & -X^T \\ -X & tI_k \end{pmatrix} \succeq 0.$$

Indeed, the result in question follows from the fact that the sums of p largest eigenvalues of a symmetric matrix are SDr (Example 15c) due to the following

Observation. *The singular values $\sigma_i(X)$ of a rectangular $k \times l$ matrix X $(k \leq l)$ for $i \leq k$ are equal to the eigenvalues $\lambda_i(\bar{X})$ of the $(k+l) \times (k+l)$ symmetric matrix*

$$\bar{X} = \begin{pmatrix} 0 & X^T \\ X & 0 \end{pmatrix}.$$

Since $\bar{X}$ linearly depends on X, the SDR's of the functions $S_p(\cdot)$ induce SDR's of the functions $\Sigma_p(X) = S_p(\bar{X})$ (Rule on affine substitution, Lecture 3; recall that all "calculus rules" established in Lecture 3 for CQR's work for SDR's as well).

Let us justify our observation. Let $X = \sum_{i=1}^k \sigma_i(X) e_i f_i^T$ be a singular value decomposition of X. We claim that the $2k$ $(k+l)$-dimensional vectors $g_i^+ = \begin{pmatrix} f_i \\ e_i \end{pmatrix}$ and $g_i^- = \begin{pmatrix} f_i \\ -e_i \end{pmatrix}$ are orthogonal to each other eigenvectors of $\bar{X}$ with the eigenvalues $\sigma_i(X)$ and $-\sigma_i(X)$, respectively, and that $\hat{X}$ vanishes on the orthogonal complement of the linear span of these vectors. In other words, we claim that the eigenvalues of $\bar{X}$, arranged in the non-ascending order, are as follows:

$$\sigma_1(X), \sigma_2(X), ..., \sigma_k(X), \underbrace{0, ..., 0}_{l-k}, -\sigma_k(X), -\sigma_{k-1}(X), ..., -\sigma_1(X);$$

this, of course, proves our Observation.

Now, the fact that the $2k$ vectors $g_i^{\pm}$, $i = 1, ..., k$, are mutually orthogonal and nonzero is evident. Furthermore (we write σ_i instead of $\sigma_i(X)$),

$$\begin{aligned} \begin{pmatrix} 0 & X^T \\ X & 0 \end{pmatrix} \begin{pmatrix} f_i \\ e_i \end{pmatrix} &= \begin{pmatrix} 0 & \sum_{j=1}^k \sigma_j f_j e_j^T \\ \sum_{j=1}^k \sigma_j e_j f_j^T & 0 \end{pmatrix} \begin{pmatrix} f_i \\ e_i \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=1}^k \sigma_j f_j (e_j^T e_i) \\ \sum_{j=1}^k \sigma_j e_j (f_j^T f_i) \end{pmatrix} \\ &= \sigma_i \begin{pmatrix} f_i \\ e_i \end{pmatrix} \end{aligned}$$

(we have used that both $\{f_j\}$ and $\{e_j\}$ are orthonormal systems). Thus, g_i^+ is an eigenvector of $\bar{X}$ with the eigenvalue $\sigma_j(X)$. Similar computation shows that g_i^- is an eigenvector of $\bar{X}$ with the eigenvalue $-\sigma_j(X)$.

It remains to verify that if $h = \begin{pmatrix} f \\ e \end{pmatrix}$ is orthogonal to all $g_i^\pm$ (f is l-dimensional, e is k-dimensional), then $\bar{X}h = 0$. Indeed, the orthogonality assumption means that $f^T f_i \pm e^T e_i = 0$ for all i, whence $e^T e_i$=0 and $f^T f_i = 0$ for all i. Consequently,

$$\begin{pmatrix} 0 & X^T \\ X & 0 \end{pmatrix} \begin{pmatrix} f \\ e \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^k f_j(e_j^T e) \\ \sum_{i=1}^k e_j(f_j^T f) \end{pmatrix} = 0. \qquad \blacksquare$$

"Nonlinear matrix inequalities". There are several cases when matrix inequalities $F(x) \succeq 0$, F being a nonlinear function of x taking values in the space of symmetric $m \times m$ matrices, can be "linearized" – expressed via LMI's.

17a. General quadratic matrix inequality. Let X be a rectangular $k \times l$ matrix and

$$F(X) = (AXB)(AXB)^T + CXD + (CXD)^T + E$$

be a "quadratic" matrix-valued function of of X; here $A, B, C, D, E = E^T$ are rectangular matrices such that the expression makes sense. Let m be the row size of the values of F. Consider the "epigraph" of the (matrix-valued!) function F – the set

$$\{(X, Y) \in \mathbf{M}^{kl} \times \mathbf{S}^m \mid F(X) \preceq Y\}.$$

We claim that the set is SDr with the SDR

$$\left(\begin{array}{c|c} I_r & (AXB)^T \\ \hline AXB & Y - E - CXD - (CXD)^T \end{array}\right) \succeq 0 \qquad [B : l \times r]$$

Indeed, by Lemma on the Schur Complement our LMI is satisfied if and only if the Schur complement of the North-Western block is positive semidefinite, which is exactly our original "quadratic" matrix inequality.

17b. General "fractional-quadratic" matrix inequality. Let X be a rectangular $k \times l$ matrix, and V be a positive definite symmetric $l \times l$ matrix. Then we can define the matrix-valued function

$$F(X, V) = XV^{-1}X^T$$

taking values in the space of $k \times k$ symmetric matrices. We claim that the closure of the "epigraph" of this (matrix-valued!) function – the set

$$E = \mathrm{cl}\{(X, V; Y) \in \mathbf{M}^{kl} \times \mathbf{S}^l_{++} \times \mathbf{S}^k \mid F(X, V) \equiv XV^{-1}X^T \preceq Y\}$$

is SDr, an SDR being given by the LMI

$$\begin{pmatrix} V & X^T \\ X & Y \end{pmatrix} \succeq 0. \qquad \text{(R)}$$

Indeed, by Lemma on the Schur Complement a triple (X, V, Y) with *positive definite* V belongs to the "epigraph of F" – satisfies the relation $F(X, V) \preceq Y$ – if and only if it satisfies (R). Now, if a triple (X, V, Y) belongs to E, i.e., is the limit of a sequence of triples from the epigraph of F, then the triple satisfies (R) (as a limit of triples satisfying (R)). Vice versa, if a triple (X, V, Y) satisfies (R), then V is positive semidefinite (as a diagonal block in a positive semidefinite matrix). The "regularized" triples $(X, V_\epsilon = V + \epsilon I_l, Y)$ associated with $\epsilon > 0$ satisfy (R) along with the triple (X, V, R); since, as we just have seen, $V \succeq 0$, we have $V_\epsilon \succ 0$, for $\epsilon > 0$. Consequently, the triples (X, V_ϵ, Y) belong to E (this was our very first observation); since the triple (X, V, Y) is the limit of the regularized triples which, as we have seen, all belong to the epigraph of F, the triple (X, Y, V) belongs to the closure E of this epigraph. ■

Nonnegative polynomials The material of this Section originates from [7]. Consider the problem of the best polynomial approximation – given a function f on certain segment, we are interested to find its best uniform (or the Least Squares,...) approximation by a polynomial of a given degree; this is a typical subproblem for all kinds of signal processing. Sometimes it makes sense to require the approximating polynomial to be nonnegative (think, e.g., of the case where the resulting polynomial is an estimate of an unknown probability density); how to express the nonnegativity restriction? As it was recently shown by Yu. Nesterov, it can be done via semidefinite programming:

The set of all nonnegative (on the entire axis, or on a given ray, or on a given segment) *polynomials of a given degree is SDr.*

In this statement (and everywhere below) we identify a polynomial $p(t)$ of degree (not exceeding) k with the $(k+1)$-dimensional vector $\pi = \mathrm{Coef}(p)$ of its coefficients:

$$p(t) = \sum_{i=0}^{k} p_i t^i \Rightarrow \mathrm{Coef}(p) = (p_0, p_1, ..., p_k)^T.$$

Consequently, a set of polynomials of the degree $\leq k$ becomes a set in $\mathbf{R}^{k+1}$, and we may ask whether this set is or is not SDr.

Let us look what are the SDR's of different sets of nonnegative polynomials. The key here is to get a SDR for the set $P^+_{2k}(\mathbf{R})$ of polynomials of (at most) a given degree $2k$ which are nonnegative on the entire axis[4)]

18a. Polynomials nonnegative an the entire axis: The set $P^+_{2k}(\mathbf{R})$ is SDr – it is the image of the semidefinite cone $\mathbf{S}^{k+1}_+$ under the affine mapping

$$X \mapsto \mathrm{Coef}(e^T(t) X e(t)) : \mathbf{S}^{k+1} \to \mathbf{R}^{2k+1}, \quad e(t) = \begin{pmatrix} 1 \\ t \\ t^2 \\ ... \\ t^k \end{pmatrix} \tag{C}$$

First note that the fact that $P^+ \equiv P^+_{2k}(\mathbf{R})$ is an affine image of the semidefinite cone indeed implies the SD-representability of P^+, see the "calculus" of conic representations in Lecture 3. Thus, all we need is to show that P^+ is exactly the same as the image, let it be called P, of $\mathbf{S}^{k+1}_+$ under the mapping (C).

(1) The fact that P *is contained* in P^+ is immediate. Indeed, let X be a $(k+1) \times (k+1)$ positive semidefinite matrix. Then X is a sum of dyadic matrices:

$$X = \sum_{i=1}^{k+1} p^i (p^i)^T, p^i = (p_{i0}, p_{i1}, ..., p_{ik})^T \in \mathbf{R}^{k+1}$$

[4)] It is clear why we have restricted the degree to be even: a polynomial of an odd degree cannot be positive on the entire axis!

(why?) But then

$$e^T(t)Xe(t) = \sum_{i=1}^{k+1} e^T(t)p^i[p^i]^T e(t) = \sum_{i=1}^{k+1}\left(\sum_{j=0}^{k} p_{ij}t^j\right)^2$$

is the sum of squares of other polynomials and therefore is nonnegative on the axis. Thus, the image of X under the mapping (C) belongs to P^+.

Note that reversing our reasoning, we get the following result:

> (!) *If a polynomial $p(t)$ of degree $\leq 2k$ can be represented as a sum of squares of other polynomials, then the vector* Coef(p) *of the coefficients of p belongs to the image of $\mathbf{S}^{k+1}_+$ under the mapping* (C).

With (!), the remaining part of the proof – the demonstration that the image of $\mathbf{S}^{k+1}_+$ *contains* P^+, is readily given by the following well-known algebraic fact:

> (!!) *A polynomial is nonnegative on the axis if and only if it is a sum of squares of polynomials.*
>
> The proof of (!!) is that nice that it makes sense to present it here. The "if" part is evident. To prove the "only if" one, assume that $p(t)$ is nonnegative on the axis, and let the degree of p (it must be even) be $2k$. Now let us look at the roots of p. The real roots $\lambda_1, ..., \lambda_r$ must be of even multiplicities $2m_1, 2m_2, ...2m_r$ each (otherwise p would alter its sign in a neighbourhood of a root, which is impossible - p is nonnegative!) The complex roots of p can be arranged in conjugate pairs $(\mu_1, \mu_1^*), (\mu_2, \mu_2^*), ..., (\mu_s, \mu_s^*)$, and the factor of p
>
> $$(t-\mu_i)(t-\mu_i^*) = (t - \Re\mu_i)^2 + (\Im\mu_i)^2$$
>
> corresponding to such a pair is a sum of two squares. Finally, the leading coefficient of p is positive. Consequently, we have
>
> $$p(t) = \omega^2[(t-\lambda_1)^2]^{m_1}...[(t-\lambda_r)^2]^{m_r}[(t-\mu_1)(t-\mu_1^*)]...[(t-\mu_s)(t-\mu_s^*)]$$
>
> *is a product of sums of squares.* But such a product is itself a sum of squares (open the parentheses)!
>
> In fact we may say more: a nonnegative polynomial p is a sum of just two squares! To see this, note that, as we have seen, p is a product of sums of *two* squares and take into account the following fact (Louville):
>
> *The product of sums of two squares is again a sum of two squares:*
>
> $$(a^2+b^2)(c^2+d^2) = (ac-bd)^2 + (ad+bc)^2$$
>
> (cf. with: "the modulus of a product of two complex numbers is the product of their modulae").

Equipped with the SDR of the set $P^+_{2k}(\mathbf{R})$ of polynomials nonnegative on the entire axis, we can immediately obtain SDR's for the polynomials nonnegative on a given ray/segment:

18b. Polynomials nonnegative on a ray/segment.

1) *The set $P^+_k(\mathbf{R}_+)$ of (coefficients of) those polynomials of degree $\leq k$ which are nonnegative on the nonnegative ray, is SDr.*

Indeed, this set is the inverse image of the SDr set $P^+_{2k}(\mathbf{R})$ under the linear mapping of the spaces of (coefficients of) polynomials given by

$$p(t) \mapsto p^+(t) \equiv p(t^2)$$

(recall that the inverse image of an SDr set is SDr).

2) *The set $P^+_k([0,1])$ of (coefficients of) those polynomials of degree $\leq k$ which are nonnegative on the segment $[0,1]$, is SDr.*

Indeed, a polynomial $p(t)$ of degree $\leq k$ is nonnegative on $[0,1]$ if and only if the rational function

$$g(t) = p\left(\frac{t^2}{1+t^2}\right)$$

is nonnegative on the entire axis, or, which is the same, when the polynomial

$$p^+(t) = (1+t^2)^k g(t)$$

of degree $\leq 2k$ is nonnegative on the entire axis. The coefficients of p^+ depend linearly on the coefficients of p, and we conclude that $P_k^+([0,1])$ is the inverse image of the SDr set $P_{2k}^+(\mathbf{R})$ under certain linear mapping.

Our last example in this series deals with trigonometric polynomials

$$p(\phi) = a_0 + \sum_{l=1}^{k}[a_l \cos(l\phi) + b_l \sin(l\phi)]$$

Identifying such a polynomial with its vector of coefficients Coef$(p) \in \mathbf{R}^{2k+1}$, we may ask how to express the set $S_k^+(\Delta)$ of those trigonometric polynomials of degree $\leq k$ which are nonnegative on a segment $\Delta \subset [0, 2\pi]$.

18c. Trigonometric polynomials nonnegative on a segment. The set $S_k^+(\Delta)$ is SDr.

Indeed, it is known from school that $\sin(l\phi)$ and $\cos(l\phi)$ are polynomials of $\sin(\phi)$ and $\cos(\phi)$, and the latter functions, in turn, in turn, are rational functions of $\zeta = \tan(\phi/2)$:

$$\cos(\phi) = \frac{1-\zeta^2}{1+\zeta^2}, \sin(\phi) = \frac{2\zeta}{1+\zeta^2} \quad [\zeta = \tan(\phi/2)].$$

Consequently, a trigonometric polynomial $p(\phi)$ of degree $\leq k$ can be represented as a rational function of $\zeta = \tan(\phi/2)$:

$$p(\phi) = \frac{p^+(\zeta))}{(1+\zeta^2)^k} \quad [\zeta = \tan(\phi/2)],$$

where the coefficients of the algebraic polynomial p^+ of degree $\leq 2k$ are linear functions of the coefficients of p. Now, the requirement for p to be nonnegative on a given segment $\Delta \subset [0, 2\pi]$ is equivalent to the requirement for p^+ to be nonnegative on a "segment" Δ^+ (which, depending on Δ, may be either the usual finite segment, or a ray, or the entire axis). We see that $S_k^+(\Delta)$ is inverse image, under certain linear mapping, of the the SDr set $P_{2k}^+(\Delta^+)$, so that $S_k^+(\Delta)$ itself is SDr.

Finally, we may ask which part of the outlined results can be saved when we pass from nonnegative polynomials of one variable to those of two or more variables. Unfortunately, not too much. E.g., when interested in polynomials of a given degree with $r > 1$ variables, we still may get those of them *who are sums of squares* as the image of a positive semidefinite cone under certain linear mapping similar to (D). The difficulty is that in the multi-dimensional case nonnegativity of a polynomial and its representability as a sum of squares are different things; consequently, what comes from the semidefinite cone is only a part of what we are interested to get.

4.3 Applications, I: Combinatorics

Due to its tremendous "expressive abilities", semidefinite programming has an extremely wide variety of applications. What we are about to do is to overview the most important of these applications. We start with brief presentation of what comes "from inside" Mathematics and then will focus, in more details, on the applications in Engineering. The most important "inner" applications of semidefinite programming are in building *relaxations of combinatorial problems.*

Combinatorial problems and their relaxations. Numerous problems of planning, placement, routing, etc., can be posed as optimization programs with discrete design variables – integer or zero-one – as *combinatorial optimization problems.* There are several "universal forms" of combinatorial problems, among them Linear Programming programs with integer variables and Linear Programming problems with 0-1 variables; a problem given in one of these forms always can be converted to any other universal form, so that in principle it it does not matter which form to use. Now, the majority of combinatorial problems are difficult – we do not know theoretically efficient, in certain precise meaning of the notion, algorithms for solving these problems; what we know is that nearly all these problems are, in a sense, equivalent to each other and are *NP-complete*; what does the latter notion mean exactly, this will be explained in Lecture 5; for the time being it suffices to say that NP-completeness of a problem P means that the problem is "as difficult as a combinatorial problem can be" – if we knew an efficient algorithm for P, we would be able to convert it to an efficient algorithm for *any other* combinatorial problem. NP-complete problems may look extremely "simple", as it is demonstrated by the following example:

> (Stones) *Given n stones of positive integer weights (i.e., given n positive integers $a_1, ..., a_n$), check whether you can partition these stones into two groups of equal weight, i.e., check whether a linear equation*
>
> $$\sum_{i=1}^{n} a_i x_i = 0$$
>
> *has a solution with $x_i = \pm 1$.*

Theoretically difficult combinatorial problems are difficult to solve in practice as well. An important ingredient in basically all algorithms for combinatorial optimization is a technique for building bounds for the unknown optimal value of a given (sub)problem. A typical way to estimate the optimal value of an optimization program

$$f(x) \to \min \mid x \in X$$

from above is to present a feasible solution $\bar{x}$ and to say that the optimal value is $\leq f(\bar{x})$. And a typical way to bound the optimal value from below is to pass from the problem to its *relaxation* – to a problem of the form

$$f(x) \to \min \mid x \in X'$$

with a *larger* feasible set X' ($X' \supset X$). The optimal value of the relaxation clearly is a lower bound for the actual optimal value, so that whenever the relaxation is efficiently solvable (to ensure this, we should take care of how we choose X'), it provides us with a "computable" lower bound on the actual optimal value.

When building a relaxation, one should take care of two issues: the relaxation, as it was already explained, should be "efficiently solvable"; at the same time, we are interested in "tight" relaxations, otherwise the bounds we get may be by far "too optimistic" and therefore be of no actual use. For a long time, the only practical ways of building relaxations were aimed at obtaining LP relaxations, since these were the only problems we could solve efficiently in practice. With progress in optimization techniques, nonlinear relaxations become more and more "practical"; as a result, we are witnessing a growing theoretical and computational activity in the area of nonlinear relaxations of combinatorial problems. Among the related developments, most, if not all, deal with *semidefinite* relaxations. Let us look at where they come from.

4.3.1 Shor's Semidefinite Relaxation scheme

As it was already mentioned, there are numerous "universal forms" of combinatorial problems – such that every combinatorial problem can be written down in this form. E.g., a combinatorial problem can be posed as the one of minimizing quadratic objective under quadratic *equality* constraints:

$$\begin{array}{lrcl} & f_0(x) = x^T A_0 x + 2b_0^T x + c_0 & \to & \min \\ \text{s.t.} & & & \\ & f_i(x) = x^T A_i x + 2b_i^T x + c_i & = & 0,\ i=1,...,m \\ & & & [x \in \mathbf{R}^n]. \end{array} \tag{4.3.1}$$

To see that this form is "universal", note that one of universal forms of a combinatorial problem is an LP program with Boolean (0-1) variables:

$$c^T x \to \min \mid a_i^T x - b_i \geq 0,\ \ i=1,...,m; x_j \in \{0,1\},\ j=1,...,n. \tag{B}$$

The fact that a variable x_j must be Boolean can be expressed by the quadratic equality

$$x_j^2 - x_j = 0,$$

and a linear inequality $a_i^T x - b_i \geq 0$ can be expressed by the quadratic equality $a_i^T x - b_i - s_i^2 = 0$, s_i being an additional variable. Thus, (B) is equivalent to the problem

$$c^T x \to \min \mid a_i^T x - b_i - s_i^2 = 0,\ i=1,...,m; x_j^2 - x_j = 0,\ j=1,...,n$$

of the form (4.3.1).

Now, in order to bound from below the optimal value in (4.3.1) we may use the same arguments as when building the dual problem. Namely, let us choose somehow "weights" λ_i, $i=1,...,m$ of arbitrary signs, and let us add the constraints of (4.3.1) with these weights to the objective, thus coming to the function

$$\begin{array}{rcl} f_\lambda(x) & = & f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) \\ & = & x^T A(\lambda) x + 2b^T(\lambda) x + c(\lambda), \\ & & \begin{array}{rcl} A(\lambda) & = & A_0 + \sum_{i=1}^m \lambda_i A_i \\ b(\lambda) & = & b_0 + \sum_{i=1}^m \lambda_i b_i \\ c(\lambda) & = & c_0 + \sum_{i=1}^m \lambda_i c_i \end{array} \end{array} \tag{4.3.2}$$

Due to its origin, the function $f_\lambda(x)$ is equal to the actual objective $f_0(x)$ on the feasible set of the problem (4.3.1). Consequently, the unconstrained – over the entire $\mathbf{R}^n$ – infimum of this function

$$a(\lambda) = \inf_{x \in \mathbf{R}^n} f_\lambda(x)$$

is a lower bound for the optimal value in (4.3.1). We come to the following simple result (cf. the Weak Duality Theorem:)

> (*) *Assume that $\lambda \in \mathbf{R}^m$ and $\zeta \in \mathbf{R}$ are such that*
>
> $$f_\lambda(x) - \zeta \geq 0 \quad \forall x \in \mathbf{R}^n \tag{4.3.3}$$
>
> *(i.e., that $\zeta \leq a(\lambda)$). Then ζ is a lower bound for the optimal value in* (4.3.1).

It remains to understand what does it mean that (4.3.3) holds. It is easy: recalling the structure of f_λ, we see that this relation says exactly that the inhomogeneous quadratic form

$$g_\lambda(x) = x^T A(\lambda)x + 2b^T(\lambda)x + c(\lambda) - \zeta$$

is nonnegative on the entire space. Now, the fact that an inhomogeneous quadratic form

$$g(x) = x^T Ax + 2b^T x + c$$

always is nonnegative is equivalent to the fact that some *homogeneous* quadratic form is nonnegative. Indeed, given $t \neq 0$ and $x \in \mathbf{R}^n$, let us look what does it mean that $g(t^{-1}x)$ is nonnegative; this means nothing but the nonnegativity of the form

$$G(x,t) = x^T Ax + 2tb^T x + ct^2.$$

We see that if g is nonnegative, then the *homogeneous* quadratic form $G(x,t)$ with $(n+1)$ variables is nonnegative whenever $t \neq 0$; by continuity, G is nonnegative everywhere. Thus, if g is nonnegative, then G is, and of course vice versa (since $g(x) = G(x,1)$). Now, to say that G is nonnegative everywhere is literally the same as to say that the matrix

$$\begin{pmatrix} c & b^T \\ b & A \end{pmatrix} \tag{4.3.4}$$

is positive semidefinite.

It is worthy to catalogue our simple observation:

> **Simple Lemma.** *A quadratic inequality with a (symmetric) $n \times n$ matrix A*
>
> $$x^T Ax + 2b^T x + c \geq 0$$
>
> *is trivially true – is valid for all $x \in \mathbf{R}^n$ – if only if the matrix* (4.3.4) *is positive semidefinite.*

Applying this observation to $g_\lambda(x)$, we get the following equivalent reformulation of (*):

> *Whenever $(\lambda, \zeta) \in \mathbf{R}^m \times \mathbf{R}$ satisfy the LMI*
>
> $$\begin{pmatrix} \sum_{i=1}^m \lambda_i c_i - \zeta & b_0^T + \sum_{i=1}^m \lambda_i b_i^T \\ b_0 + \sum_{i=1}^m \lambda_i b_i & A_0 + \sum_{i=1}^m \lambda_i A_i \end{pmatrix} \succeq 0,$$
>
> *ζ is a lower bound for the optimal value in* (4.3.1).

Now, what is the best lower bound we can get with this scheme? Of course, it is the optimal value in the semidefinite program

$$\zeta \to \max \mid \begin{pmatrix} c_0 + \sum_{i=1}^m \lambda_i c_i - \zeta & b_0^T + \sum_{i=1}^m \lambda_i b_i^T \\ b_0 + \sum_{i=1}^m \lambda_i b_i & A_0 + \sum_{i=1}^m \lambda_i A_i \end{pmatrix} \succeq 0 \tag{4.3.5}$$

with design variables λ_i, ζ.

We have proved the following simple

Proposition 4.3.1 *The optimal value in* (4.3.5) *is a lower bound for the optimal value in* (4.3.1).

The outlined scheme is extremely transparent, but it does not look as a relaxation scheme as explained above – where is the extension of the feasible set of the original problem? In fact the scheme is of this type. To see it, let us note that the value of a quadratic form at a point $x \in \mathbf{R}^n$ can be written down as the Frobenius inner product of certain matrix and the dyadic matrix $X(x) = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^T$:

$$x^T A x + 2b^T x + c = \begin{pmatrix} 1 \\ x \end{pmatrix}^T \begin{pmatrix} c & b^T \\ b & A \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix} = \mathrm{Tr}\left(\begin{pmatrix} c & b^T \\ b & A \end{pmatrix} X(x) \right).$$

Consequently, (4.3.1) can be written down as the problem

$$\mathrm{Tr}\left(\begin{pmatrix} c_0 & b_0^T \\ b_0 & A_0 \end{pmatrix} X(x) \right) \to \min \mid \mathrm{Tr}\left(\begin{pmatrix} c_i & b_i^T \\ b_i & A_i \end{pmatrix} X(x) \right) = 0, \ i = 1, ..., m. \tag{4.3.6}$$

Thus, we may think of (4.3.2) as of a problem with *linear* objective and *linear* equality constraints with the design vector X being a symmetric $(n+1) \times (n+1)$ matrix *running through the nonlinear manifold $\mathcal{X}$ comprised of dyadic matrices $X(x)$, $x \in \mathbf{R}^n$*. Now, what we for sure know about $\mathcal{X}$ is that it is comprised of positive semidefinite matrices with the North-Western entry equal to 1. Denoting by $\mathcal{X}$ the latter set and replacing $\mathcal{X}$ by $\mathcal{X}$, we get a relaxation of (4.3.6), i.e., essentially, of our original problem (4.3.1). This relaxation is the semidefinite program

$$\begin{array}{c} \mathrm{Tr}(\bar{A}_0 X) \to \min \mid \mathrm{Tr}(\bar{A}_i X) = 0, i = 1, ..., m; X \succeq 0; X_{11} = 1 \\ \left[A_i = \begin{pmatrix} c_i & b_i^T \\ b_i & A_i \end{pmatrix}, \ i = 1, ..., m \right] \end{array} \tag{4.3.7}$$

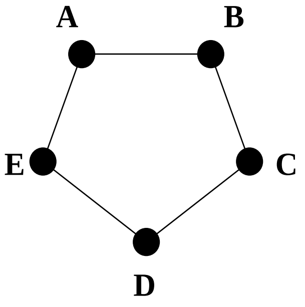
and we get the following

Proposition 4.3.2 *The optimal value in the semidefinite program* (4.3.7) *is a lower bound for the optimal value in* (4.3.1).

You can easily verify that problem (4.3.5) is just the semidefinite dual of (4.3.7); thus, when deriving (4.3.5), we in fact were implementing the idea of relaxation. This is why in the sequel we call both (4.3.7) and (4.3.5) *semidefinite relaxations* of (4.3.1). Let us look at several interesting examples.

4.3.2 Stability number, Shannon capacity and Lovasz capacity of a graph

Stability number of a graph. Consider a (non-oriented) graph – a finite set of *nodes* somehow linked by arcs[5)], like the simple 5-node graph C_5:



Graph C_5

One of fundamental characteristics of a graph Γ is its *stability number* $\alpha(\Gamma)$ defined as the maximum cardinality of an *independent* subset of nodes – a subset such that no two nodes from it are linked by an arc. E.g., the stability number for the above graph is 2.

The problem of computing the stability number of a given graph is NP-complete, this is why it is important to know how to bound this number.

Shannon capacity of a graph. An interesting by its own right upper bound on the stability number of a graph is the *Shannon capacity* $\Theta(\Gamma)$ defined as follows.

Let us treat nodes of Γ as letters of certain alphabet, and the arcs as possible errors in certain communication channel: you can send trough the channel one letter per unit time, and what will be obtained on the other end of the channel, can be either the letter you have sent, or any letter adjacent to it. Now assume that you are planning to communicate with an addressee through the channel by sending n-letter words (n is fixed). You fix in advance a dictionary D_n of words to be used and make this dictionary known to the addressee. What you are interested in when building the dictionary is to get a *good* one: such that no word from it could be transformed by the channel into another word from the dictionary. If your dictionary satisfies this requirement, that you may be sure that the addressee will never misunderstand you: whatever word from the dictionary you send and whatever possible transmission errors will occur, the addressee is able either to get the correct message, or to realize that the message was corrupted during transmission, but there is no risk that your "yes" will be read as "no!". Now, in order to utilize the channel "at full capacity", you are interested to get as large dictionary as possible. How many words it can include? The answer is clear: this is nothing but the stability number of the graph Γ^n defined as follows. The nodes of Γ^n are ordered n-element collections of the nodes of Γ – all possible n-letter words in your alphabet, and two distinct nodes $(i_1, ..., i_n)$ $(j_1, ..., j_n)$ are linked by an arc if and only if for every l l-th letters i_l and j_l in the two words either coincide, or are linked by arcs in Γ. (i.e., two distinct n-letter words are adjacent, if one of them, as a result of transmission, can be transformed into another). Let us denote the maximum number of words in a "good" dictionary D_n (i.e., the stability number of Γ^n) by $f(n)$, The function $f(n)$

[5)] One of formal definitions of a (non-oriented) graph is as follows: a n-node graph is just a $n \times n$ symmetric matrix A with entries $0, 1$ and zero diagonal. The rows (and the columns) of the matrix are identified with the nodes $1, 2, ..., n$ of the graph, and the nodes i, j are adjacent (i.e., linked by an arc) exactly for those i, j with $A_{ij} = 1$.

possesses the following nice property:

$$f(k)f(l) \leq f(k+l),\ k,l = 1,2,... \tag{$*$}$$

> Indeed, given the best (of cardinality $f(k)$) good dictionary D_k and the best good dictionary D_l, let us build a dictionary comprised of all $(k+l)$-letter words as follows: the initial k-letter fragment of a word belongs to D_k, and the remaining l-letter fragment belongs to D_l. The resulting dictionary clearly is good and contains $f(k)f(l)$ words, and (*) follows.

Now, this is a simple exercise in analysis to see that for a nonnegative function f with property (*) one has

$$\lim_{k\to\infty}(f(k))^{1/k} = \sup_{k\geq 1}(f(k))^{1/k} \in [0,+\infty].$$

In our situation $\sup_{k\geq 1}(f(k))^{1/k} < \infty$, since clearly $f(k) \leq n^k$, n being the number of letters (the number of nodes in Γ). Consequently, the quantity

$$\Theta(\Gamma) = \lim_{k\to\infty}(f(k))^{1/k}$$

is well-defined; moreover, for every k the quantity $(f(k))^{1/k}$ is a lower bound for $\Theta(\Gamma)$. The number $\Theta(\Gamma)$ is called the *Shannon capacity* of Γ. Our immediate observation is that

(!) *The Shannon capacity number $\Theta(\Gamma)$ majorates the stability number of Γ:*

$$\alpha(\Gamma) \leq \Theta(\Gamma).$$

> Indeed, as we remember, $(f(k))^{1/k}$ is a lower bound for $\Theta(\Gamma)$ for every $k = 1,2,...$; setting $k = 1$ and taking into account that $f(1) = \alpha(\Gamma)$, we get the desired result.

We see that the Shannon capacity number is an upper bound on the stability number; and this bound has a nice interpretation in terms of the Information Theory. A bad news is that we do not know how to compute the Shannon capacity. E.g., what is it for the toy graph C_5?

The stability number of C_5 clearly is 2, so that our first observation is that

$$\Theta(C_5) \geq \alpha(C_5) = 2.$$

To get a better estimate, let us look the graph $(C_5)^2$ (as we remember, $\Theta(\Gamma) \geq (f(k))^{1/k} = (\alpha(\Gamma^k))^{1/k}$ for every k). The graph $(C_5)^2$ has 25 nodes, so that we do not draw it; it, however, is not that difficult to find its stability number, which turns out to be 5. A good 5-element dictionary ($\equiv$ a 5-node independent set in $(C_5)^2$) is, e.g.,

$$AA, AC, BE, CB, DE.$$

Thus, we get

$$\Theta(C_5) \geq \sqrt{\alpha((C_5)^2)} = \sqrt{5}.$$

Attempts to compute the subsequent lower bounds $(f(k))^{1/k}$, as long as they are implementable (think how many vertices there are in $(C_5)^4$!), do not yield any improvements, and for more than 20 years it remained unknown whether $\Theta(C_5) = \sqrt{5}$ or is $> \sqrt{5}$. And this is for a toy graph! The breakthrough in the area of upper bounds for the stability number is due to L. Lovasz who in early 70's found a new – computable! – bound of this type.

Lovasz capacity number. Given a n-node graph Γ, let us associate with it an affine matrix-valued function $\mathcal{L}(x)$ taking values in the space of $n \times n$ symmetric matrices, namely, as follows:

For every pair i, j of indices $(1 \leq i, j \leq n)$ such that the nodes i and j are <u>not</u> linked by an arc, the ij-th entry of $\mathcal{L}$ is identically equal to 1;

For a pair $i < j$ of indices such that the nodes i, j are linked by an arc, the ij-th and the ji-th entries in $\mathcal{L}$ are equal to x_{ij} – to the variable responsible for the arc (i, j).

Thus, $\mathcal{L}(x)$ indeed is an affine function of N design variables x_{ij}, N being the number of arcs in the graph. E.g., for graph C_5 the function $\mathcal{L}$ is as follows:

$$\mathcal{L} = \begin{pmatrix} 1 & x_{AB} & 1 & 1 & x_{EA} \\ x_{AB} & 1 & x_{BC} & 1 & 1 \\ 1 & x_{BC} & 1 & x_{CD} & 1 \\ 1 & 1 & x_{CD} & 1 & x_{DE} \\ x_{EA} & 1 & 1 & x_{DE} & 1 \end{pmatrix}.$$

Now, the Lovasz capacity number, $\vartheta(\Gamma)$, by definition, is the optimal value in the optimization program

$$\lambda_{\max}(\mathcal{L}(x)) \to \min,$$

i.e., the optimal value in the semidefinite program

$$\lambda \to \min \mid \lambda I_n - \mathcal{L}(x) \succeq 0. \tag{L}$$

Proposition 4.3.3 *[Lovasz] One has*

$$\vartheta(\Gamma) \geq \Theta(\Gamma).$$

Thus, the Lovasz capacity number is an upper bound for the Shannon capacity one and, consequently, for the stability number:

$$\vartheta(\Gamma) \geq \Theta(\Gamma) \geq \alpha(\Gamma).$$

For graph C_5, the Lovasz capacity number can be easily computed analytically and turns out to be exactly $\sqrt{5}$. Thus, a small byproduct of Lovasz's result is a solution to the problem which remained open for two decades.

Let us look how the Lovasz bound on the stability number can be obtained from the general relaxation scheme. To this end note that the stability number of an n-node graph Γ is the optimal value in the following optimization problem with 0-1 variables:

$$e^T x \to \max \mid x_i x_j = 0 \text{ whenever } i, j \text{ are adjacent nodes }, x_i \in \{0, 1\},\ i = 1, ..., n,$$
$$e = (1, ..., 1)^T \in \mathbf{R}^n.$$

Indeed, 0-1 n-dimensional vectors can be identified with sets of nodes of Γ: the coordinates x_i of the vector x representing a set A of nodes are ones for $i \in A$ and zeros otherwise. The quadratic equality constraints $x_i x_j = 0$ for such a vector express equivalently the fact that the corresponding set of nodes is independent, and the objective $e^T x$ counts the cardinality of this set.

As we remember, the 0-1 restrictions on the variables can be represented equivalently by quadratic equality constraints, so that the stability number of Γ is the optimal value in the following problem with quadratic (in fact linear) objective and quadratic equality constraints:

$$\begin{array}{rrcl} & e^T x & \to & \max \\ \text{s.t.} & & & \\ & x_i x_j & = & 0, \ (i,j) \text{ is an arc} \\ & x_i^2 - x_i & = & 0, \ i = 1, ..., n. \end{array} \tag{4.3.8}$$

The latter problem is in the form of (4.3.1), with the only difference that the objective should be maximized rather than minimized. Switching from maximization of $e^T x$ to minimization of $(-e)^T x$ and passing to (4.3.5), we get the problem

$$\zeta \to \max \mid \begin{pmatrix} -\zeta & -\frac{1}{2}(e+\mu)^T \\ -\frac{1}{2}(e+\mu) & A(\mu,\lambda) \end{pmatrix} \succeq 0,$$

where μ is n-dimensional and $A(\mu,\lambda)$ is as follows:

- The diagonal entries of $A(\mu,\lambda)$ are $\mu_1, ..., \mu_n$;

- The off-diagonal cells ij corresponding to non-adjacent nodes i,j ("empty cells") are zeros;

- The off-diagonal cells ij, $i < j$, and the symmetric cells ji corresponding to adjacent nodes i,j ("arc cells") are filled with free variables λ_{ij}.

Note that the optimal value in the resulting problem is a *lower* bound for *minus* the optimal value of (4.3.8), i.e., for minus the stability number of Γ.

Passing in the resulting problem from the variable ζ to a new variable $\xi = -\zeta$ and again switching from maximization of $\zeta = -\xi$ to minimization of ξ, we end up with the semidefinite program

$$\xi \to \min \mid \begin{pmatrix} \xi & -\frac{1}{2}(e+\mu)^T \\ -\frac{1}{2}(e+\mu) & A(\mu,\lambda) \end{pmatrix} \succeq 0. \tag{4.3.9}$$

The optimal value in this problem is the minus optimal value in the previous one, which, in turn, is a lower bound on the minus stability number of Γ; consequently, the optimal value in (4.3.9) is an *upper* bound on the stability number of Γ.

We have built a semidefinite relaxation (4.3.9) of the problem of computing the stability number of Γ; the optimal value in the relaxation is an upper bound on the stability number. To get the Lovasz relaxation, let us further fix the μ-variables at the level 1 (this may only increase the optimal value in the problem, so that it still will be an upper bound for the stability number). With this modification, we come to the problem

$$\xi \to \min \mid \begin{pmatrix} \xi & -e^T \\ -e & A(e,\lambda) \end{pmatrix} \succeq 0.$$

In a feasible solution to the problem, ξ should be ≥ 1 (it is an upper bound for $\alpha(\Gamma) \geq 1$). When $\xi \geq 1$, the LMI

$$\begin{pmatrix} \xi & -e^T \\ e & A(e,\lambda) \end{pmatrix} \succeq 0$$

by Lemma on Schur complement is equivalent to the LMI

$$A(e,\lambda) \succeq (-e)\xi^{-1}(-e)^T,$$

or, which is the same, to the LMI

$$\xi A(e,\lambda) - ee^T \succeq 0.$$

The left hand side matrix in the latter LMI is equal to $\xi I_n - B(\xi,\lambda)$, where the matrix $B(\xi,\lambda)$ is as follows:

- The diagonal entries of $B(\xi,\lambda)$ are equal to 1;
- The off-diagonal "empty cells" are filled with ones;
- The "arc cells" from a symmetric pair off-diagonal pair ij and ji $(i<j)$ are filled with $\xi\lambda_{ij}$.

Passing from the design variables λ to the new ones $x_{ij} = \xi\lambda_{ij}$, we conclude that problem (4.3.9) with μ's set to ones is equivalent to the problem

$$\xi \to \min \mid \xi I_n - \mathcal{L}(x) \succeq 0,$$

which is exactly the problem defining, via its optimal value, the Lovasz capacity number of Γ.

As a byproduct of our derivation, we get the easy part of the Lovasz Theorem – the inequality $\vartheta(\Gamma) \geq \alpha(\Gamma)$; this inequality, however, could be easily obtained directly from the definition of $\vartheta(\Gamma)$. The advantage of our derivation is that it demonstrates what is the origin of $\vartheta(\Gamma)$.

How good is Lovasz capacity number? The Lovasz capacity number plays a very important role in numerous graph-related problems; there is an important sub-family of graphs – *perfect graphs* – for which this number coincides with the stability number. However, for a general-type graph Γ $\vartheta(\Gamma)$ may be a fairly poor bound for $\alpha(\Gamma)$. Lovasz has proved that for any graph Γ with n nodes, $\vartheta(\Gamma)\vartheta(\hat{\Gamma}) \geq n$, where $\hat{\Gamma}$ is the *complement* to Γ (i.e., two distinct nodes are adjacent in $\hat{\Gamma}$ if and only if they are *not* adjacent in Γ). It follows that for n-node graph Γ one always have $\max[\vartheta(\Gamma), \vartheta(\hat{\Gamma})] \geq \sqrt{n}$. On the other hand, it turns out that for a random n-node graph Γ (the arcs are drawn at random and independently of each other, with probability 0.5 to draw an arc linking two given distinct nodes) $\max[\alpha(\Gamma), \alpha(\hat{\Gamma})]$ "typically" (with probability approaching 1 as n grows) is of order of $\ln n$. It follows that for random n-node graphs a typical value of the ratio $\vartheta(\Gamma)/\alpha(\Gamma)$ is at least of order of $n^{1/2}/\ln n$; as n grows, this ratio blows up to ∞.

A natural question arises: are there "difficult" (NP-complete) combinatorial problems admitting "good" semidefinite relaxations – those with the quality of approximation not deteriorating as the sizes of instances grow? Let us look at two recent (and breakthrough) results in this direction.

4.3.3 The MAXCUT problem

The *maximum cut* problem is as follows:

Problem 4.3.1 [MAXCUT] *Let Γ be a n-node graph, and let the arcs (i,j) of the graph be associated with nonnegative "weights" a_{ij}. The problem is to find a cut of the largest possible weight – to partition the set of nodes in two parts S, S' in such a way that the total weight of arcs "linking S and S'" – with one of the two incident nodes in S and the other one in S' – is as large as possible.*

When speaking about the MAXCUT problem we always may assume that the weights $a_{ij} = a_{ji} \geq 0$ are defined for every pair i, j of indices; it suffices to set $a_{ij} = 0$ for pairs i, j of non-adjacent nodes.

In contrast to the *minimum cut* problem (where we should minimize the weight of a cut instead of maximizing it), which is, basically, a nice LP program of finding the maximum flow in a net and is therefore efficiently solvable, the MAXCUT problem is as difficult as a combinatorial problem might be – it is NP-complete. It, however, is easy to build a semidefinite relaxation of MAXCUT. To this end let us pose MAXCUT as a quadratic problem with quadratic equality constraints. Let Γ be a n-node graph. A cut (S, S') – a partitioning of the set of nodes in two non-overlapping parts S, S' – may be identified with a n-dimensional vector x with coordinates ± 1 – $x_i = 1$ for $i \in S$ and $x_i = -1$ for $i \in S'$. It is immediately seen that the quantity $\frac{1}{2}\sum_{i,j=1}^{n} a_{ij}x_i x_j$ is the total weight of arcs with both ends either in S or in S' minus the weight of the cut (S, S'); consequently, the quantity

$$\frac{1}{2}\left[\frac{1}{2}\sum_{i,j=1}^{n} a_{ij} - \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}x_i x_j\right] = \frac{1}{4}\sum_{i,j=1}^{n} a_{ij}(1 - x_i x_j)$$

is exactly the weight of the cut (S, S').

We conclude that the MAXCUT problem can be posed as the following quadratic problem with quadratic equality constraints:

$$\frac{1}{4}\sum_{i,j=1}^{n} a_{ij}(1 - x_i x_j) \to \max \mid x_i^2 = 1, \ i = 1, ..., n. \tag{4.3.10}$$

It is immediately seen that for the problem in question the semidefinite relaxation (4.3.7) after evident simplifications becomes the semidefinite program

$$\begin{array}{rrcl} & \frac{1}{4}\sum_{i,j=1}^{n} a_{ij}(1 - x_{ij}) & \to & \max \\ \text{s.t.} & & & \\ & X = [X_{ij}]_{i,j=1}^{n} = X^T & \succeq & 0 \\ & X_{ii} & = & 1, \ i = 1, ..., n; \end{array} \tag{4.3.11}$$

the optimal value in the latter problem is an <u>upper</u> bound for the optimal value in MAXCUT.

> The fact that (4.3.11) is a relaxation of (4.3.10) can be established immediately, independently of any "general theory": (4.3.10) is the problem of maximizing the objective
>
> $$\frac{1}{4}\sum_{i,j=1}^{n} a_{ij} - \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}x_i x_j \equiv \frac{1}{4}\sum_{i,j=1}^{n} a_{ij} - \frac{1}{4}\mathrm{Tr}(AX(x)), \quad X(x) = xx^T$$
>
> over all rank 1 matrices $X(x) = xx^T$ coming from n-dimensional vectors x with entries ± 1. All these matrices are symmetric positive semidefinite with unit entries on the diagonal, i.e., they belong the feasible set of (4.3.11). Thus, (4.3.11) indeed is a relaxation of (4.3.10).

The quality of the semidefinite relaxation (4.3.11) is given by the following brilliant result of Goemans and Williamson:

Theorem 4.3.1 *Let OPT be the optimal value in the MAXCUT problem* (4.3.10)*, and SDP be the optimal value in the semidefinite relaxation* (4.3.11)*. Then*

$$1 \geq \frac{OPT}{SDP} \geq 0.87856... \tag{4.3.12}$$

The reasoning used by Goemans and Williamson is so beautiful that it is impossible not to reproduce it here.

The left inequality in (4.3.12) is evident – it simply says that our semidefinite program (4.3.11) is a relaxation of MAXCUT. To get the right inequality, Goemans and Williamson act as follows. Let X be a feasible solution to the semidefinite relaxation. Since X is positive semidefinite, it is the Gram matrix of a collection of n vectors $v_1, ..., v_n$:

$$X_{ij} = v_i^T v_j.$$

And since all X_{ii} are equal to 1, the vectors v_i are of unit Euclidean norm. Given X, we can easily find v_i's (e.g., via the Choleski decomposition of X). Now let us look at the following procedure for generating *random* cuts of the graph: we choose at random, according to the uniform distribution on the unit sphere in $\mathbf{R}^n$, a unit vector v and build the cut

$$S = \{i \mid v^T v_i \geq 0\}.$$

What is the *expected value* of the weight of this random cut? The expected contribution of a particular pair i, j to the expected weight of our random cut is $\frac{1}{4}a_{ij}$ times twice the probability of the event that the nodes i and j will be "separated" by v, i.e., that the products $v^T v_i$ and $v^T v_j$ will have opposite signs. By elementary arguments, the probability of this event is just twice the ratio of the angle between the vectors v_i and v_j to 2π, as is seen from the following picture:

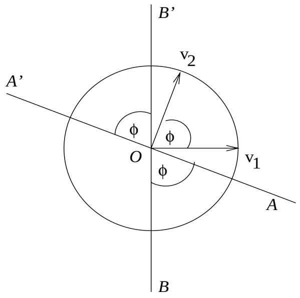


Figure 4.1

w is the projection of v on the 2D plane spanned by v_i and v_j; the direction of w is uniformly distributed in $[0, 2\pi]$, and v separates v_i and v_j exactly when w belongs to one of the angles AOB, A$'$OB$'$

Thus, the expected contribution of a pair i, j to the expected weight of our random cut is $\frac{1}{2}a_{ij}\frac{\arccos(v_i^T v_j)}{\pi}$, and the expected weight of the random cut is

$$W[X] = \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}\frac{\arccos(X_{ij})}{\pi}.$$

Comparing the right hand side term by term with the value

$$f(X) = \frac{1}{4}\sum_{i,j=1}^{n} a_{ij}(1 - X_{ij})$$

at X of the objective in the relaxed problem (4.3.11) and taking into account that $a_{ij} \geq 0$ and that for all $x \in [-1,1]$ one has[6)]

$$\frac{\arccos(x)}{\pi} \geq \alpha\frac{1}{2}(1-x), \quad \alpha = 0.87856...,$$

we come to

$$W[X] \geq \alpha f(X).$$

This inequality is valid for every feasible solution X of the semidefinite relaxation, in particular, for the optimal solution X^*. We conclude that already the *expectation* of the weight of random cut generated "from X^*" is at least $\alpha\, SDP$; and the maximum possible weight OPT of a cut may be only larger than this expectation, so that $OPT/SDP \geq \alpha = 0.87856...$

Note that the above construction not only provides a proof of Theorem 4.3.1, but offers a randomized algorithm for constructing a random cut which, at average, has weight at least 0.87856 of OPT. Indeed, it suffices to solve the semidefinite relaxation (4.3.11) (which can be done efficiently, provided that we will be satisfied with an ϵ-solution – a feasible X such that the value of the objective of (4.3.11) at X is at least $(1-\epsilon)\cdot SDP$ – with any once for ever fixed $\epsilon > 0$, say, with $\epsilon = 1.e-6$). After a (nearly) optimal solution X to (4.3.11) is found, we use it to generate random cuts, as explained in the above construction.

4.3.4 Extensions

In the MAXCUT problem, we in fact are maximizing the homogeneous quadratic form

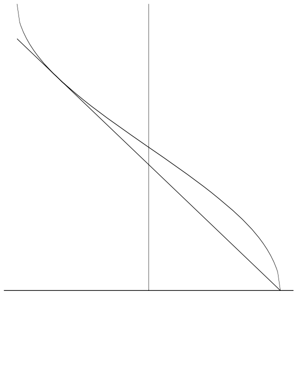
$$x^T Ax \equiv \sum_{i=1}^n \left(\sum_{j=1}^n a_{ij}\right) x_i^2 - \sum_{i,j=1}^n a_{ij}x_ix_j$$

over the set S_n of n-dimensional vectors x with coordinates ± 1. The matrix A of this form is positive semidefinite (Exercise 4.2) and possesses a specific feature that the off-diagonal entries are nonpositive, while the sum of the entries in every row is 0. What happens when we are maximizing over S_n a quadratic form $x^T Ax$ with a general-type (symmetric) matrix A? An extremely nice result in this direction was recently obtained by Yu. Nesterov. The cornerstone of Nesterov's construction relates to the case when A is positive semidefinite, and this is the case we shall start with. Note that the problem of maximizing a quadratic form $x^T Ax$ with positive semidefinite (and, say, integer) matrix A over S_n, same as MAXCUT, is NP-complete.

The semidefinite relaxation of the problem

$$x^T Ax \to \max \mid x \in S_n \quad [\Leftrightarrow x_i \in \{-1,1\},\, i = 1, ..., n] \tag{4.3.13}$$

[6)] Look at the picture:



can be built exactly in the same way as (4.3.11) and turns out to be the semidefinite program

$$\begin{array}{rrcl} & \mathrm{Tr}(AX) & \to & \max \\ \text{s.t.} & & & \\ & X = X^T = [X_{ij}]_{i,j=1}^n & \succeq & 0 \\ & X_{ii} & = & 1,\ i = 1, ..., n. \end{array} \tag{4.3.14}$$

The optimal value in this problem, let it again be called SDP, is $\geq$ the optimal value OPT in the original problem (4.3.13). The ratio OPT/SDP, however, cannot be too large:

Theorem 4.3.2 [Nesterov's Theorem] *Let A be positive semidefinite. Then*

$$SDP \geq OPT \geq \frac{2}{\pi} SDP \quad [2/\pi = 0.6366...]$$

The proof utilizes the central idea of Goemans and Williamson in the following brilliant reasoning:

The fact that $SDP \geq OPT$ was already mentioned (it comes from the origin of (4.3.14) as a relaxation of (4.3.13)). Let X be a feasible solution to the relaxed problem; then $X_{ij} = v_i^T v_j$ for a system of unit vectors $v_1, ..., v_n$. Similarly to the MAXCUT construction, let us associate with this representation of X a random generator of vectors from S_n: choosing a direction v uniformly on the unit sphere, we build vector x with the ±1-coordinates

$$x_i = \mathrm{sign}(v^T v_i),$$

where $\mathrm{sign}(a)$ is $+1$ for $a \geq 0$ and is -1 for $a < 0$. What is the expected value of the objective $x^T A x$ of (4.3.13) over the generated points x? It is

$$V = \sum_{i,j=1}^n a_{ij} E_v \{\mathrm{sign}(v^T v_i)\mathrm{sign}(v^T v_j)\},$$

where E_v denotes expectation with respect to v uniformly distributed over the unit sphere. An expectation $E_v\{\mathrm{sign}(v^T v_i)\mathrm{sign}(v^T v_j)\}$ can be easily computed: when projecting v on the 2D plane spanned by v_i, v_j, we get a vector w with the direction uniformly distributed in $[0, 2\pi]$, and the expectation in question is the probability for w to have inner products with v_i and v_j of the same sign (i.e., to belong to the union of angles AOA′ and BOB′ on Fig. 4.1) minus probability to have inner products with v_i and v_j of opposite signs (i.e., to belong to the union of the angles AOB and A′OB′). The indicated difference clearly is

$$\frac{1}{2\pi}\left[2\pi - 4\arccos(v_i^T v_j)\right] = \frac{2}{\pi}\arcsin(v_i^T v_j) = \frac{2}{\pi}\arcsin(X_{ij}).$$

Thus,

$$V = \frac{2}{\pi}\sum_{i,j=1}^n a_{ij}\arcsin(X_{ij}).$$

recalling that V is the expected value of the objective in (4.3.13) with respect to certain probability distribution on the feasible set S_n of the problem, we get $V \leq OPT$. Following Nesterov, we have proved

Lemma 4.3.1 *Let X be a feasible solution to* (4.3.14)*. Then the optimal value OPT in* (4.3.13) *satisfies the relation*

$$OPT \geq \frac{2}{\pi}\sum_{i,j=1}^n a_{ij}\arcsin(X_{ij}).$$

Consequently,

$$OPT \geq \frac{2}{\pi}\max\left\{\mathrm{Tr}(A\arcsin[X]) \mid X \succeq 0, X_{ii} = 1,\ i = 1, ..., n\right\}, \tag{4.3.15}$$

where $\arcsin[X]$ *is the matrix with the elements* $\arcsin(X_{ij})$.

Nesterov completes the proof by the following unexpected, although simple,

> Observation. *For a positive semidefinite symmetric matrix X with diagonal entries ± 1 (in fact – for any positive semidefinite X with $|X_{ij}| \leq 1$) one has*
>
> $$\arcsin[X] \succeq X.$$
>
> The proof of Observation is immediate: denoting by $[X]^k$ the "entry-wise k-th power of X" – the matrix with the entries X_{ij}^k – and making use of the Taylor series for the arcsin (this series converges uniformly on $[-1,1]$), for a matrix X with all entries belonging to $[-1,1]$ we get
>
> $$\arcsin[X] - X = \sum_{k=1}^{\infty} \frac{1 \times 3 \times 5 \times \ldots \times (2k-1)}{2^k k!(2k+1)} [X]^{2k+1},$$
>
> and all we need is to note that the entry-wise product of two symmetric positive semidefinite matrices is positive semidefinite[7]

Since A is positive semidefinite, Observation implies that $\sum_{i,j=1}^{n} a_{ij} \arcsin(X_{ij}) = \mathrm{Tr}(A \arcsin[X]) \geq \mathrm{Tr}(AX)$ for every feasible solution X of the semidefinite relaxation (4.3.14), i.e., that the expression in the right hand side of (4.3.15) is at least $\frac{2}{\pi} SDP$. ■

Note that in fact the inequality in (4.3.15) is equality (see Exercise 4.23).

4.4 Applications, II: Stability Analysis

Semidefinite programming is a natural language to pose and to process numerous stability-related problems arising in engineering. Let us look at several examples.

4.4.1 Dynamic stability in Mechanics

"Free motions" of numerous "linear elastic" mechanical systems, i.e., their behaviour in absence of external loads, are governed by systems of differential equations of the type

$$M \frac{d^2}{dt^2} x(t) = -Ax(t), \tag{N}$$

where $x(t) \in \mathbf{R}^n$ is the state vector of the system at instant t, M is the (generalized) "mass matrix", and A is the "stiffness matrix" of the system. (N), basically, is the Newton law for a system with the potential energy $\frac{1}{2} x^T A x$.

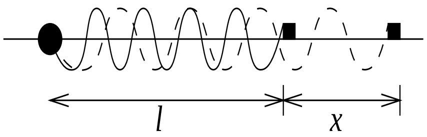
> As a simple example, consider a system of m points of masses $\mu_1, \ldots, \mu_k$ linked by springs with given elasticity coefficients; here x is comprised of the displacements $x_i \in \mathbf{R}^d$ ($d = 1/2/3$) of the points from their equilibrium positions e_i, and the Newton equations become
>
> $$\mu_i \frac{d^2}{dt^2} x_i(t) = -\sum_{j \neq i} \nu_{ij} (e_i - e_j)(e_i^T x_i - e_j^T x_j), i = 1, \ldots, k,$$

[7] This is the standard fact of Linear Algebra. The easiest way to understand it is to note that if P, Q are positive semidefinite symmetric matrices of the same size, then they are Gram matrices: $P_{ij} = p_i^T p_j$ for certain system of vectors p_i from certain (no matter from which exactly) $\mathbf{R}^N$ and $Q_{ij} = q_i^T q_j$ for a system of vectors q_i from certain $\mathbf{R}^M$. But then the entry-wise product of P and Q – the matrix with the entries $P_{ij} Q_{ij} = (p_i^T p_j)(q_i^T q_j)$ – also is a Gram matrix, namely, the Gram matrix of matrices $p_i q_i^T \in \mathbf{M}^{NM} = \mathbf{R}^{NM}$. Since every Gram matrix is positive semidefinite, the entry-wise product of P and Q is positive semidefinite.

where ν_{ij} are given by

$$\nu_{ij} = \frac{\kappa_{ij}}{\| e_i - e_j \|_2^2},$$

$\kappa_{ij} > 0$ being the elasticity coefficients of the springs. The resulting system is of the form (N) with a diagonal M and positive semidefinite symmetric A. The simplest system of this type we know from school – this is a *pendulum* (a single point capable to slide along a given axis and linked by a spring to a fixed point on the axis):



$$\frac{d^2}{dt^2}x(t) = -\nu x(t), \quad \nu = \frac{\kappa}{l}.$$

Another example is given by a truss (see Section 1.3.5); here A is the bar-stiffness matrix $\sum_i t_i b_i b_i^T$, and the mass matrix is

$$M = \sum_i t_i \beta_i \beta_i^T, \quad \beta_i = \sqrt{\frac{\mu}{\kappa}} l_i b_i,$$

where μ is the material density, κ is the Young modulus, and l_i is the length of bar i.

Note that in the indicated examples both the mass matrix M and the stiffness matrix A are symmetric positive semidefinite; in "nondegenerate" cases they are even positive definite, and this is what we assume from now on. Under this assumption, we can pass in (N) from the variables $x(t)$ to the variables $y(t) = M^{-1/2}x(t)$; with respect to these variables, the system becomes

$$\frac{d^2}{dt^2}y(t) = -\hat{A}y(t), \ \hat{A} = M^{-1/2}AM^{-1/2}. \tag{N$'$}$$

It is known that the space of solutions of a system of the latter type (where $\hat{A}$ is symmetric positive definite) is spanned by fundamental (perhaps complex-valued) solutions of the type $\exp\{\mu t\}f$. A nontrivial (with $f \neq 0$) function of this type indeed is a solution to (N$'$) if and only if

$$(\mu^2 I + \hat{A})f = 0,$$

so that the allowed values of μ^2 are exactly the minus eigenvalues of the matrix $\hat{A}$, f being the corresponding eigenvectors of $\hat{A}$. But the matrix $\hat{A}$ is symmetric positive definite, so that the only allowed values of μ are purely imaginary, with the imaginary parts $\pm\sqrt{\lambda_j(\hat{A})}$. Recalling that the eigenvalues/eigenvectors of $\hat{A}$ are exactly the eigenvalues/eigenvectors of the pencil$[M, A]$, we come to the following result:

> (!) *In the case of positive definite symmetric M, A, the solutions to* (N) *– the "free motions" of the corresponding mechanical system $\mathcal{S}$ – are exactly the solutions of the form*
>
> $$x(t) = \sum_{j=1}^{n} [a_j \cos(\omega_j t) + b_j \sin(\omega_j t)]e_j,$$
>
> *where a_j, b_j are free real parameters, $e_j 0$ are the eigenvectors of the pencil $[M, A]$:*
>
> $$(\lambda_j M - A)e_j = 0$$

and $\omega_j = \sqrt{\lambda_j}$. Thus, the "free motions" of the system $\mathcal{S}$ are comprised of harmonic oscillations along the eigenvectors of the pencil $[M, A]$, the frequencies of the oscillations ("the eigenfrequencies of the system") being the square roots of the corresponding eigenvalues of the pencil.

From the engineering viewpoint, the "dynamic behaviour" of mechanical constructions like buildings, masts, bridges, etc., is the better the larger are the eigenfrequencies of the system[8]. This is why a typical design restriction in mechanical engineering is a lower bound

$$\lambda_{\min}(A : M) \geq \lambda_* \quad [\lambda_* > 0] \tag{4.4.1}$$

on the *smallest* eigenvalue $\lambda_{\min}(A : M)$ of the pencil $[M, A]$ comprised of the mass and the stiffness matrices of the system to be designed. In the case of positive definite symmetric mass matrices such a bound is equivalent to the matrix inequality

$$A - \lambda_* M \succeq 0 \tag{4.4.2}$$

(why?). If M and A are affine functions of the design variables (as it is, e.g., the case in Truss Design), the matrix inequality (4.4.2) is a *linear* matrix inequality on the design variables, and therefore it can be processed via the machinery of semidefinite programming. E.g., when adding to the Truss Topology Design problem (Lecture 3) a lower bound on the minimum eigenfrequency of the truss to be designed, we end up with a semidefinite program. Moreover, in the cases when A is affine in the design variables, and M is constant, (4.4.2) is an LMI on the design variables and λ_*, and we may play with λ_*: e.g., a problem of the type "given the mass matrix of the system to be designed and a number of (SDr) constraints on the design variables, build a system with as large minimum eigenfrequency as possible" is a semidefinite program (provided, of course, that the stiffness matrix of the system is affine in the design variables).

4.4.2 Lyapunov stability analysis/synthesis

The material of this Section originates from [3]. The topic we are coming to now was already touched in Lecture 2 (it provided us with an important example of a non-polyhedral conic problem); this is the topic of Lyapunov stability analysis. Consider a time-varying *uncertain* linear dynamic system

$$\frac{d}{dt}x(t) = A(t)x(t), \; x(0) = x_0 \tag{ULS}$$

Here $x(t) \in \mathbf{R}^n$ represents the state of a "plant" at time instant t, x_0 is the initial state and $A(t)$ is a time-varying $n \times n$ matrix. We assume that the system is *uncertain* in the sense that we have no idea of what is x_0, and all we know about $A(t)$ is that this matrix, at any time instant t, belongs to a known in advance uncertainty set $\mathcal{U}$. Thus, we are speaking about a wide family of linear dynamic systems rather than on a single system of this type; and it makes sense to call a *trajectory* of the uncertain linear system (ULS) every function $x(t)$ which is an "actual trajectory" of a system from the family, i.e., is such that

$$\frac{d}{dt}x(t) = A(t)x(t)$$

[8] Think about a building and an earthquake or about sea waves and a light house: in this case the external load acting at the system is time-varying and can be represented as a sum of harmonic oscillations of different (and low) frequencies; if some of these frequencies are close to the eigenfrequencies of the system, the system can be crushed by resonance. In order to avoid this risk, one is interested to move the eigenfrequencies of the system away from 0 as far as possible.

for all $t \geq 0$ and certain matrix-valued function $A(t)$ taking all its values in $\mathcal{U}$.

Note that we may model a *nonlinear* dynamic system

$$\frac{d}{dt}x(t) = f(t, x(t)) \quad [x \in \mathbf{R}^n] \tag{NLS}$$

with a given right hand side $f(t, x)$ and a given equilibrium $x(t) \equiv 0$ (i.e., $f(t, 0) = 0 \quad \forall t$) as an uncertain *linear* system. Indeed, let us define the set $\mathcal{U}_f$ as the closed convex hull of all $n \times n$ matrices $\frac{\partial}{\partial x} f(t, x)$ coming from all $t \geq 0$ and all $x \in \mathbf{R}^n$. Then for every point $x \in \mathbf{R}^n$ we have

$$\begin{array}{c} f(t,x) = f(t,0) + \int_0^t \left[\frac{\partial}{\partial x} f(t,sx)\right] x ds = A_x(t)x, \\ A_x(t) = \int_0^1 \frac{\partial}{\partial x} f(t,sx) ds \in \mathcal{U}. \end{array}$$

We see that every trajectory of the original nonlinear system (NLS) is also a trajectory of the uncertain linear system (ULS) associated with the uncertainty set $\mathcal{U} = \mathcal{U}_f$ (this trick is called “global linearization”). Of course, the set of trajectories of the resulting uncertain linear system may be much wider than the set of trajectories of (NLS); however, all “good news” about the uncertain system (like “all trajectories of (ULS) share such and such property”) are automatically valid for the trajectories of the “nonlinear system of interest” (NLS), and only “bad news” about (ULS) (“such and such property is not shared by some trajectories of (ULS)”) may say nothing about the system of interest (NLS).

The basic question about a dynamic system is the one of its stability. For (ULS), this question sounds as follows:

(?) *Is it true that* (ULS) *is stable, i.e., that*

$$x(t) \to 0, \ t \to \infty$$

for every trajectory of the system?

A sufficient condition for the stability of (ULS) is the existence of a *quadratic Lyapunov function* – a quadratic form $\mathcal{L}(x) = x^T X x$, X being *positive definite* symmetric matrix, such that

$$\frac{d}{dt}\mathcal{L}(x(t)) \leq -\alpha \mathcal{L}(x(t)) \tag{4.4.3}$$

for certain $\alpha > 0$ and all trajectories of (ULS):

Lemma 4.4.1 [Quadratic Stability Certificate] *Assume* (ULS) *admits a quadratic Lyapunov function* $\mathcal{L}$. *Then* (ULS) *is stable.*

Proof. Indeed, if (4.4.3) is valid for all trajectories of (ULS), then, integrating this differential inequality, we get

$$\mathcal{L}(x(t)) \leq \exp\{-\alpha \mathcal{L}(x(0))\} \to 0, \ t \to \infty.$$

Since $\mathcal{L}(\cdot)$ is a positive definite quadratic form, $\mathcal{L}(x(t)) \to 0$ implies that $x(t) \to 0$. ■

Of course, Lemma 4.4.1 is applicable to non-quadratic Lyapunov functions as well as to quadratic ones: all we need is (4.4.3) plus the assumption that $\mathcal{L}(x)$ is smooth, nonnegative and is bounded away from 0 outside every neighbourhood of the origin. The advantage of a *quadratic* Lyapunov function is that we more or less understand how to find such a function, if it exists:

Proposition 4.4.1 [Existence of Quadratic Stability Certificate] *Let $\mathcal{U}$ be the uncertainty set associated with uncertain linear system* (ULS)*. The system admits quadratic Lyapunov function if and only if the optimal value in the "semi-infinite*[9] *semidefinite program"*

$$\begin{array}{rrcl} & s & \to & \min \\ s.t. & & & \\ & sI_n - A^T X - XA & \succeq & 0, \quad \forall A \in \mathcal{U} \\ & X & \succeq & I_n \end{array} \tag{Ly}$$

the design variables in the problem being $s \in \mathbf{R}$ and $X \in \mathbf{S}^n$, is strictly negative. Moreover, every feasible solution to the problem with negative value of the objective provides a quadratic Lyapunov stability certificate for (ULS)*.*

Proof is immediate. The derivative $\frac{d}{dt}\left[x^T(t)Xx(t)\right]$ of the value of a quadratic function x^TXx on a trajectory of (ULS) is equal to

$$\left[\frac{d}{dt}x(t)\right]^T Xx(t) + x^T(t)X\left[\frac{d}{dt}x(t)\right] = x^T(t)[A^T(t)X + XA(t)]x(t).$$

If x^TXx is a Lyapunov function, the resulting quantity should be at most $-\alpha x^T(t)Xx(t)$, i.e., we should have

$$x^T(t)\left[-\alpha X - A^T(t)X - XA(t)\right]x(t) \geq 0.$$

The resulting inequality should be valid for every possible value of $A(t)$ at the time instant t and every possible value $x(t)$ of a trajectory of the system at this instant. Since possible values of $x(t)$ fill the entire $\mathbf{R}^n$ and possible values of $A(t)$ fill the entire $\mathcal{U}$, we come to

$$-\alpha X - A^T X - XA \succeq 0 \quad \forall A \in \mathcal{U}.$$

By definition of a quadratic Lyapunov function, $X \succ 0$ and $\alpha > 0$; by normalization (dividing both X and α by the smallest eigenvalue of X), we get a pair $\hat{s} > 0, \hat{X} \geq I$ such that

$$-\hat{s}\hat{X} - A^T\hat{X} - \hat{X}A \succeq 0 \quad \forall A \in \mathcal{U}; \; \hat{X} \succeq I_n.$$

Since $\hat{X}$ is positive definite, we have $-\hat{X} \preceq -\gamma I_n$ for certain positive γ; but then $(s = -\gamma\hat{s}, \hat{X})$ is a feasible solution to (Ly) with negative value of the objective. Thus, if (ULS) admits a quadratic Lyapunov function, then (Ly) has a feasible solution with negative value of the objective. Reversing the reasoning, we see that the inverse implication also is true. ∎

Lyapunov Stability Analysis

We see that in order to certify stability of an uncertain linear system it suffices to point out a feasible solution to (Ly) with a negative value of the objective. It should be stressed that such a possibility is no more than a sufficient condition for stability: if it exists, the system is for sure stable. However, if the possibility in question is absent – the optimal value in (Ly) is nonnegative – all we can say is that *the stability of* (ULS) *cannot be certified by a quadratic*

[9] i.e., with infinitely many LMI constraints

Lyapunov function, although (ULS) still may be stable.[10] In this sense, stability analysis based on quadratic Lyapunov functions – "good systems are those admitting quadratic Lyapunov functions" – is conservative. This drawback, however, is in a sense compensated by the fact that this kind of stability analysis is "implementable" – in many cases we can efficiently solve (Ly), thus getting a quadratic "stability certificate", provided that it exists. Let us look at two cases of this type.

Polytopic uncertainty set. The first "tractable case" of (Ly) is the one where $\mathcal{U}$ is a polytope *given as a convex hull of a number of points*:

$$\mathcal{U} = \mathrm{Conv}\{A_1, ..., A_N\}.$$

In this case (Ly) clearly is equivalent to the usual semidefinite program

$$s \to \min \mid sI_n - A_i^T X - XA_i \succeq 0, \ i = 1, ..., N; X \succeq I_n$$

(why?)

The assumption that $\mathcal{U}$ is a polytope *given as a convex combination of a finite set* is crucial for a possibility to get a "computationally tractable" equivalent reformulation of (Ly). If $\mathcal{U}$ is, say, a polytope *given by a list of linear inequalities* (e.g., all we know about the entries of $A(t)$ are certain intervals where the entries take their values; it is called an "interval" uncertainty), (Ly) may become as hard as a problem can be; it may even happen that to check whether a *given* pair (s, X) is feasible for (Ly) already is a "computationally intractable" problem. The same bad things may occur when $\mathcal{U}$ is a general-type ellipsoid in the space of $n \times n$ matrices. There exists, however, a specific type of "uncertainty ellipsoids" $\mathcal{U}$ for which (Ly) is "easy". Let us look at this case.

Norm-bounded perturbations. In numerous applications the $n \times n$ matrices A forming the uncertainty set $\mathcal{U}$ are obtained from a fixed matrix A_* by adding perturbations of the form $B\Delta C$, $B \in \mathbf{M}^{nk}$ and $C \in \mathbf{M}^{ln}$ being given rectangular matrices and $\Delta \in \mathbf{M}^{kl}$ being the "source of perturbations" varying in a "simple" set $\mathcal{D}$:

$$\mathcal{U} = \{A = A_* + B\Delta C \mid \Delta \in \mathcal{D} \subset \mathbf{M}^{kl}\} \quad \left[B \in \mathbf{M}^{nk}, 0 \neq C \in \mathbf{M}^{ln}\right] \tag{4.4.4}$$

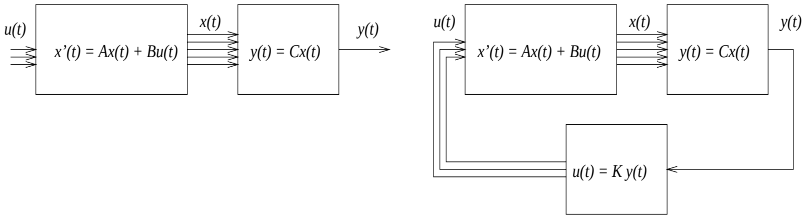
To give an instructive example, consider a *controlled* linear time-invariant dynamic system

$$\begin{array}{rcl} \frac{d}{dt}x(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{array} \tag{4.4.5}$$

(x is the state, u is the control and y is the output we can observe) "closed" by a feedback

$$u(t) = Ky(t).$$

[10] The only case when the existence of a quadratic Lyapunov function is a criterion – a *necessary and sufficient* condition – for stability is the simplest case of certain time-invariant linear system $\frac{d}{dt}x(t) = Ax(t)$ ($\mathcal{U} = \{A\}$). This is the case which led Lyapunov to the general concept of what is now called "a Lyapunov function" and what is the basic approach to establishing convergence of different time-dependent processes to their equilibria. Note also that in the case of time-invariant linear system there exists a straightforward stability criterion – all eigenvalues of A should have negative real parts. The advantage of the Lyapunov approach is that it can be extended to more general situations, which is not the case for the eigenvalue criterion.



Open loop (left) and closed loop (right) controlled systems

The resulting "closed loop system" is given by

$$\frac{d}{dt}x(t) = \hat{A}x(t), \quad \hat{A} = A + BKC. \tag{4.4.6}$$

Now assume that A, B and C are constant and known, but the feedback K is drifting around certain nominal feedback K^*: $K = K^* + \Delta$. As a result, the matrix $\hat{A}$ of the closed loop system also becomes drifting around its nominal value $A^* = A + BK^*C$, and the perturbations in $\hat{A}$ are exactly of the form $B\Delta C$.

Note that we could get essentially the same kind of drift in $\hat{A}$ assuming, instead of additive perturbations, multiplicative perturbations $C = (I_l + \Delta)C^*$ in the observer (or multiplicative disturbances in the actuator B).

Now assume that input perturbations Δ are of spectral norm $|\Delta|$ not exceeding a given ρ (*norm-bounded perturbations*):

$$\mathcal{D} = \{\Delta \in \mathbf{M}^{kl} \mid |\Delta| \le \rho\} \tag{4.4.7}$$

Proposition 4.4.2 [3] *In the case of uncertainty set* (4.4.4), (4.4.7) *the "semi-infinite" semidefinite program* (Ly) *is equivalent to the usual semidefinite program*

$$\begin{array}{rcl} \alpha & \to & \min \\ \begin{pmatrix} \alpha I_n - A_*^T X - XA_* - \lambda C^T C & XB \\ B^T X & \lambda I_k \end{pmatrix} & \succeq & 0 \\ X & \succeq & I_n \end{array} \tag{4.4.8}$$

with design variables α, λ, X.

Besides this, when increasing the set of perturbations (4.4.7) *to the ellipsoid*

$$\mathcal{E} = \{\Delta \in \mathbf{M}^{kl} \mid \| \Delta \|_2 \equiv \sqrt{\sum_{i=1}^{k} \sum_{j=1}^{l} \Delta_{ij}^2} \le \rho\}, \tag{4.4.9}$$

we basically do not vary (Ly) *– the latter problem in the case of the uncertainty set* (4.4.4), (4.4.9) *still is equivalent to* (4.4.8).

Proof. It suffices to verify the following general statement:

Lemma 4.4.2 *Consider the matrix inequality*

$$Y - Q^T \Delta^T P^T Z^T R - R^T Z P \Delta Q \succeq 0 \tag{4.4.10}$$

where Y *is symmetric* $n \times n$ *matrix,* Δ *is a* $k \times l$ *matrix and* P, Q, Z, R *are rectangular matrices of appropriate sizes (i.e.,* $q \times k$, $l \times n$, $p \times q$ *and* $p \times n$*, respectively). Given*

Y, P, Q, Z, R, with $Q \neq 0$ (this is the only nontrivial case), this matrix inequality is satisfied for all Δ with $|\Delta| \leq \rho$ if and only if it is satisfied for all Δ with $\| \Delta \|_2 \leq \rho$, and this is the case if and only if

$$\begin{pmatrix} Y - \lambda Q^T Q & -\rho R^T Z P \\ -\rho P^T Z^T R & \lambda I_k \end{pmatrix} \succeq 0$$

for a properly chosen real λ.

The statement of Proposition 4.4.7 is just a particular case of Lemma 4.4.2: e.g., in the case of uncertainty set (4.4.4), (4.4.7) a pair (α, X) is a feasible solution to (Ly) if and only if $X \succeq I_n$ and (4.4.10) is valid for $Y = \alpha X - A_*^T X - X A_*$, $P = B$, $Q = C$, $Z = X$, $R = I_n$; Lemma 4.4.2 provides us with an LMI reformulation of the latter property, and this LMI is exactly what we see in the statement of Proposition.

Proof of Lemma. (4.4.10) is valid for all Δ with $|\Delta| \leq \rho$ (let us call this property of (Y, P, Q, Z, R) "Property 1") if and only if

$$\begin{array}{c} \forall \xi \in \mathbf{R}^n \quad \forall (\Delta : |\Delta| \leq \rho) : \\ 2[\xi^T R^T Z P] \Delta [Q \xi] \leq \xi^T Y \xi, \end{array}$$

or, which is the same, if and only if

$$\max_{\Delta : |\Delta| \leq \rho} 2 \left[[P^T Z^T R \xi]^T \Delta [Q \xi] \right] \leq \xi^T Y \xi \quad \forall \xi$$

(Property 2). The maximum over Δ, $|\Delta| \leq \rho$, of the quantity $\eta^T \Delta \zeta$, clearly is equal to ρ times the product of the Euclidean norms of the vectors η and ζ (why?). Thus, Property 2 is equivalent to

$$\xi^T Y \xi - 2\rho \| Q\xi \|_2 \| P^T Z^T R \xi \|_2 \geq 0 \quad \forall \xi \tag{I}$$

(Property 3). Now is the trick: Property 3 is clearly equivalent to the following fact

Property 4: *Every pair $\zeta = (\xi, \eta)$ comprised of a n-dimensional vector ξ and k-dimensional vector η which satisfies the quadratic inequality*

$$\xi^T Q^T Q \xi - \eta^T \eta \geq 0 \tag{II}$$

satisfies also the quadratic inequality

$$\xi^T Y \xi - 2\rho \eta^T P^T Z^T R \xi \geq 0. \tag{III}$$

Indeed, for a fixed ξ the minimum, over η satisfying (II), value of the left hand side in (III) is nothing but the left hand side of (I).

It remains to use the following fundamental fact:

$\mathcal{S}$-Lemma. *Let $x^T A x$ and $x^T B x$ be two quadratic forms. Assume that the quadratic inequality*

$$x^T A x \geq 0 \tag{A}$$

is strictly feasible (i.e., is satisfied as strict at certain point). Then the quadratic inequality

$$x^T B x \geq 0 \tag{B}$$

is a consequence of (A) – (B) is satisfied at every solution to (A) – if and only if there exists a nonnegative λ such that

$$B \succeq \lambda A.$$

The $\mathcal{S}$-Lemma is exactly what we need to complete the proof. Indeed, Property 4 says that the quadratic inequality (III) with variables ξ, η is a consequence of (II); by $\mathcal{S}$-Lemma (recall that $Q \neq 0$, so that (II) is strictly feasible!) this is equivalent to the existence of a nonnegative λ such that

$$\begin{pmatrix} Y & -\rho R^T Z P \\ -\rho P^T Z^T R & \end{pmatrix} - \lambda \begin{pmatrix} Q^T Q & \\ & -I_k \end{pmatrix} \succeq 0,$$

which is exactly what is said by Lemma 4.4.2, as far as the perturbations Δ with $|\Delta| \leq \rho$ are concerned. The case of perturbations with $\|\cdot\|_2$-norm not exceeding ρ is completely similar, since the equivalence between Properties 2 and 3 is valid independently of which norm of perturbations – $|\cdot|$ or $\|\cdot\|_2$ – is used.

Lyapunov Stability Synthesis

We have seen that the question of whether a given uncertain linear system (ULS) admits a quadratic Lyapunov function, under reasonable assumptions on the underlying uncertainty set, can be reduced to a semidefinite program. Now let us switch from the *analysis* question – whether a stability of an uncertain linear system may be certified by a quadratic Lyapunov function – to the *synthesis* question as follows. Assume that we are given an *uncertain open loop controlled system*

$$\begin{array}{rcl} \frac{d}{dt}x(t) & = & A(t)x(t) + B(t)u(t) \\ y(t) & = & C(t)x(t); \end{array} \tag{UOS}$$

all we know about the collection $(A(t), B(t), C(t))$ comprised of time-varying $n \times n$ matrix $A(t)$, $n \times k$ matrix $B(t)$ and $l \times n$ matrix $C(t)$ is that this collection, at every time instant t, belongs to a given uncertainty set $\mathcal{U}$. The question is whether we can equip our uncertain "open loop" system (UOS) with a linear feedback

$$u(t) = Ky(t)$$

in such a way that the resulting uncertain closed loop system

$$\frac{d}{dt}x(t) = [A(t) + B(t)KC(t)]\,x(t) \tag{UCS}$$

will be stable and, moreover, its stability could be certified by a quadratic Lyapunov function. In other words, now we are simultaneously seeking for a "stabilizing controller" and a quadratic Lyapunov certificate of its stabilizing ability.

With already known to us "global linearization" trick we may use the results on uncertain controlled linear systems to build stabilizing linear controllers for *nonlinear* controlled systems

$$\begin{array}{rcl} \frac{d}{dt}x(t) & = & f(t, x(t), u(t)) \\ y(t) & = & g(t, x(t)) \end{array}$$

Assuming $f(t, 0, 0) = 0$, $g(t, 0) = 0$ and denoting by $\mathcal{U}$ the closed convex hull of triples of matrices

$$\left(\frac{\partial}{\partial x} f(t, x, u), \frac{\partial}{\partial u} f(t, x, u), \frac{\partial}{\partial x} g(t, x)\right)$$

coming from all possible t, x, u, we see that every trajectory of the original nonlinear system is a trajectory of the uncertain linear system (UOS) associated with the set $\mathcal{U}$. Consequently, if we are able to find a stabilizing controller for (UOS) and certify its stabilizing property by a quadratic Lyapunov function, the resulting controller/Lyapunov function will stabilize the nonlinear system and certify the stability of the closed loop system, respectively.

Exactly the same reasoning as in the previous section leads us to the following

Proposition 4.4.3 *Let $\mathcal{U}$ be the uncertainty set associated with an uncertain open loop controlled system* (UOS). *The system admits a stabilizing controller along with a quadratic Lyapunov stability certificate for the stability of the corresponding closed loop system if and only if the optimal value in the optimization problem*

$$\begin{array}{rrcl} & s & \to & \min \\ s.t. & & & \\ & [A+BKC]^T X + X[A+BCK] & \preceq & sI_n \quad \forall (A,B,C)\in\mathcal{U} \\ & X & \succeq & I_n, \end{array} \tag{LyS}$$

design variables being s, X, K, is negative. Moreover, every feasible solution to the problem with negative value of the objective provides stabilizing controller along with quadratic Lyapunov stability certificate for the closed loop system.

A bad news about (LyS) is that now it is much more difficult to rewrite this problem as a semidefinite program than in the analysis case (i.e., the case of $K=0$) – (LyS) is semi-infinite system of *nonlinear* matrix inequalities. There is, however, an important particular case where this difficulty can be eliminated. This is the case of a feedback via the *full* state vector – the case when $y(t)=x(t)$ (i.e., the matrix $C(t)$ always is unit). Assuming that it is the case, note that all we need in order to get a stabilizing controller along with a quadratic Lyapunov certificate of its stabilizing ability is to solve a system of *strict* matrix inequalities

$$\begin{array}{rcl} [A+BK]^T X + X[A+BK] & \preceq & Z \prec 0 \quad \forall (A,B)\in\mathcal{U} \\ X & \succ & 0 \end{array} \tag{$*$}$$

– given a solution (X,K,Z) to this system, we always may convert it, just by normalization of X, to a solution of (LyS). Now let us pass in (*) to new variables

$$Y = X^{-1}, L = KX^{-1}, W = X^{-1}ZX^{-1} \quad \left[\Leftrightarrow X = Y^{-1}, K = LY^{-1}, Z = Y^{-1}WY^{-1}\right].$$

With respect to the new variables the system (*) becomes

$$\begin{array}{c} \left\{ \begin{array}{rcl} [A+BLY^{-1}]^T Y^{-1} + Y^{-1}[A+BLY^{-1}] & \preceq & Y^{-1}WY^{-1} \prec 0 \\ Y^{-1} & \succ & 0 \end{array} \right. \\ \Updownarrow \\ \left\{ \begin{array}{rcl} L^T B^T + YA^T + BL + AY & \preceq & W \prec 0, \quad \forall (A,B)\in\mathcal{U} \\ Y & \succ & 0 \end{array} \right. \end{array}$$

(we have multiplied all original matrix inequalities from the left and from the right by Y). What we end up with, is a system of strict *linear* matrix inequalities with respect to our new design variables L, Y, W; the question of whether this system is solvable again can be converted to the question of whether the optimal value in a problem of the type (LyS) is negative, and we come to the following

Proposition 4.4.4 *Consider an uncertain controlled linear system with complete observer:*

$$\begin{array}{rcl} \frac{d}{dt}x(t) & = & A(t)x(t) + B(t)u(t) \\ y(t) & = & x(t) \end{array}$$

and let $\mathcal{U}$ be the corresponding uncertainty set (which now is comprised of pairs (A,B) of possible values of $(A(t),B(t))$).

The system can be stabilized by a linear controller

$$u(t) = Ky(t) \quad [\equiv Kx(t)]$$

in such a way that the resulting closed loop uncertain system

$$\frac{d}{dt}x(t) = [A(t) + B(t)K]x(t)$$

admits a quadratic Lyapunov stability certificate if and only if the optimal value in the optimization problem

$$\begin{array}{rrcl} & s & \to & \min \\ s.t. & & & \\ & BL + AY + L^TB^T + YA^T & \preceq & sI_n \quad \forall (A,B) \in \mathcal{U} \\ & Y & \succeq & 0 \end{array} \tag{Ly*}$$

the design variables being $s \in \mathbf{R}$, $Y \in \mathbf{S}^n$, $L \in \mathbf{M}^{kn}$, is negative. Moreover, every feasible solution to (Ly*) *with negative value of the objective provides a stabilizing linear controller along with related quadratic Lyapunov stability certificate.*

In particular, in the polytopic case:

$$\mathcal{U} = \mathrm{Conv}\{(A_1,B_1),...,(A_N,B_N)\}$$

the Quadratic Lyapunov Stability Synthesis reduces to solving the semidefinite program

$$s \to \min \mid B_iL + A_iY + YA_i^T + L^Tb_i^T \preceq sI_n, \; i = 1,...,N; Y \succeq I_n.$$

4.5 Applications, III: Robust Quadratic Programming

The concept of robust counterpart of an optimization problem with uncertain data (see Section 3.4.2) in no sense is restricted to Linear Programming. Whenever we have an optimization problem and understand what are its data, we may ask what happens when the data are uncertain and all we know is an uncertainty set the data belong to. Given an uncertainty set, we may impose on candidate solutions the restriction to be robust feasible – to satisfy all realizations of the constraints, the data running through the uncertainty set; and the robust counterpart of an uncertain problem is the problem of minimizing the objective[11)] over the set of robust feasible solutions.

Now, we have seen in Section 3.4.2 that the "robust form" of an uncertain linear inequality, the coefficients of the inequality varying in an ellipsoid, is a conic quadratic inequality; as a result, the robust counterpart of an uncertain LP problem with ellipsoidal uncertainty is a conic quadratic problem. What is the "robust form" of a conic quadratic inequality

$$\| Ax + b \|_2 \leq c^Tx + d \qquad [A \in \mathbf{M}^{mn}, b \in \mathbf{R}^m, c \in \mathbf{R}^n, d \in \mathbf{R}] \tag{4.5.1}$$

[11)] For the sake of simplicity we assume that the objective is "certain" – is not affected by the data uncertainty. We always may come to this situation by passing to an equivalent problem with linear (and standard) objective:

$$f(x) \to \min \mid x \in ... \mapsto t \to \min \mid f(x) - t \leq 0, x \in ...$$

with uncertain data $(A,b,c,d)\in\mathcal{U}$? How to describe the set of those x's which satisfy all realizations of this inequality, i.e., are such that

$$\| Ax+b \|_2 \le c^T x + d \quad \forall (A,b,c,d)\in\mathcal{U} \tag{4.5.2}$$

?

We are about to demonstrate that in the case when the data (P,p) of the left hand side and the data (q,r) of the right hand side of the inequality *independently of each other run through respective ellipsoids*:

$$\begin{array}{rcl}\mathcal{U} & = & \{(A,b,c,d) \mid \exists (u\in\mathbf{R}^l, u^Tu\le 1, v\in\mathbf{R}^r, v^Tv\le 1): \\ & & [A;b]=[A^0;b^0]+\sum_{i=1}^l u_i[A^i;b^i], (c,d)=(c^0,d^0)+\sum_{i=1}^r v_i(c^i,d^i)\}\end{array} \tag{4.5.3}$$

the robust version (4.5.2) of the uncertain inequality (4.5.1) can be expressed via LMI's:

Proposition 4.5.1 [Robust counterpart of a conic quadratic inequality with simple ellipsoidal uncertainty] *In the case of uncertainty* (4.5.3), *the set given by* (4.5.2) *is SD-representable with the following SD-representation: x satisfies* (4.5.2) *if and only if there exist real s, μ such that the triple (x,s,μ) satisfies the following LMI's:*

$$\begin{array}{ll}
(a) & \left(\begin{array}{c|cccc}
(c^0)^Tx+d^0-s & (c^1)^Tx+d^1 & (c^2)^Tx+d^2 & \dots & (c^r)^Tx+d^r \\ \hline
(c^1)^Tx+d^1 & (c^0)^Tx+d^0-s & & & \\
(c^2)^Tx+d^2 & & (c^0)^Tx+d^0-s & & \\
\dots & & & \ddots & \\
(c^r)^Tx+d^r & & & & (c^0)^Tx+d^0-s
\end{array}\right) \succeq 0 \\
(b) & \left(\begin{array}{c|c|ccc}
sI_m & A^0x+b^0 & A^1x+b^1 & \dots & A^lx+b^l \\ \hline
[A^0x+b^0]^T & s-\mu & & & \\ \hline
[A^1x+b^1]^T & & \mu & & \\
\vdots & & & \ddots & \\
[A^lx+b^l]^T & & & & \mu
\end{array}\right) \succeq 0
\end{array} \tag{4.5.4}$$

Proof. Since the uncertain data of the left and of the right hand side in (4.5.1) independently of each other run through their respective ellipsoids, x satisfies (4.5.2) (Property 0) if and only if

(Property 1) There exists $s\in\mathbf{R}$ such that

$$\begin{array}{llll}(a) & s & \le & [c^0+\sum_{i=1}^r v_ic^i]^Tx+[d^0+\sum_{i=1}^r v_id^i] \quad \forall v: v^Tv\le 1 \\ (b) & s & \ge & \left\| [A^0+\sum_{i=1}^l u_iA^i]x+[b^0+\sum_{i=1}^l u_ib^i] \right\|_2 \quad \forall u: u^Tu\le 1.\end{array} \tag{4.5.5}$$

Now, the relation (4.5.5.(a)) is equivalent to the conic quadratic inequality

$$[c^0]^Tx+d^0-\left\| \begin{pmatrix}[c^1]^Tx+d^1 \\ [c^2]^Tx+d^2 \\ \dots \\ [c^r]^Tx+d^r\end{pmatrix} \right\|_2 \ge s$$

(why?), or, which is the same (see (4.2.1)), to the LMI (4.5.4.(a)).

Now, let us set

$$\begin{array}{rcl} p(x) & = & A^0x+b^0 \in \mathbf{R}^m; \\ P(x) & = & [A^1x+b^1; A^2x+b^2; ...; A^lx+b^l] \in \mathbf{M}^{ml}. \end{array}$$

The relation (4.5.5.(b)) is nothing but

$$s \geq \parallel P(x)u+p(x) \parallel_2 \quad \forall u: u^Tu \leq 1;$$

thus, it is equivalent to the fact that $s \geq 0$ and the quadratic form of u

$$s^2 - p^T(x)p(x) - 2p^T(x)P(x)u - u^TP^T(x)P(x)u$$

is nonnegative whenever $u^Tu \leq 1$. This, in turn, clearly is equivalent to the fact that the *homogeneous* quadratic form

$$(s^2 - p^T(x)p(x))t^2 - 2tp^T(x)P(x)u - u^TP^T(x)P(x)u$$

of u, t ($t \in \mathbf{R}$) is nonnegative whenever $u^Tu \leq t^2$. Applying $\mathcal{S}$-Lemma, we come to the following conclusion:

> (!) *Relation* (4.5.5.(b)) *is equivalent to the facts that* $s \geq 0$ *and that there exists* $\nu \geq 0$ *such that*
>
> $$(s^2 - p^T(x)p(x))t^2 - 2tp^T(x)P(x)u - u^TP^T(x)P(x)u - \nu[t^2 - u^Tu] \geq 0 \quad \forall (u,t) \in \mathbf{R}^l \times \mathbf{R}. \tag{4.5.6}$$

We now claim that the quantity ν in (!) can be represented as μs with some nonnegative μ. There is nothing to prove when we are in the case of $s > 0$. Now assume that $s = 0$ and that (4.5.6) is satisfied by some $\nu \geq 0$. Then $\nu = 0$ (look what happens when $t = 1, u = 0$), and it again can be represented as μs with, say, $\mu = 0$. Thus, we have demonstrated that

> (!!) *Relation* (4.5.5.(b)) *is equivalent to the facts that* $s \geq 0$ *and that there exists* $\mu \geq 0$ *such that*
>
> $$s\left[(s-\mu)t^2 + \mu u^Tu\right] - \left[p^T(x)p(x)t^2 + 2tp^T(x)P(x)u + u^TP^T(x)P(x)u\right] \geq 0 \quad \forall (u,t),$$
>
> *or, which is the same, such that*
>
> $$s\begin{pmatrix} s-\mu & \\ & \mu I_l \end{pmatrix} - ([p(x); P(x)]]^T[p(x); P(x)] \succeq 0 \tag{4.5.7}$$

Now note that when $s > 0$, (4.5.7) says exactly that the Schur complement to the North-Western block in the matrix

$$\left(\begin{array}{c|c} sI_m & [p(x); P(x)] \\ \hline [p(x); P(x)]^T & \begin{matrix} s-\mu & \\ & \mu I_l \end{matrix} \end{array}\right) \tag{*}$$

is positive semidefinite. By Lemma on the Schur Complement, this is exactly the same as to say that $s > 0$ and the matrix (*) is positive semidefinite. Thus, in the case of $s > 0$ relation (4.5.5) says precisely that there exists μ such that (*) is positive semidefinite. In the case of $s = 0$ relation (4.5.5) can be satisfied if and only if $p(x) = 0$ and $P(x) = 0$ (see (!!)); and this again is exactly the case when there exists a μ such that (*) is positive semidefinite. Since (*) may be positive semidefinite only when $s \geq 0$, we come to the following conclusion:

Relation (4.5.5.(b)) is satisfied if and only if there exists μ such that the matrix (*) is positive semidefinite.

It remains to note that (*) is exactly the matrix in (4.5.4.(b)). ■

Remark 4.5.1 We have built explicit semidefinite representation of the robust version of a conic quadratic inequality in the case of "simple" ellipsoidal uncertainty. In more complicated cases (e.g., when b, c, d, in (4.5.1) are not affected by uncertainty and the matrix A is "interval": all its entries, independently of each other, vary in given intervals) it may be "computationally intractable" already to check whether a given x is robust feasible.

Example: Robust synthesis of antenna array. We have already considered the problem of antenna array synthesis in the Tschebyshev setting, i.e., when the discrepancy between the target diagram and the designed one is measured in the uniform norm (Section 1.2.4). We have also seen that the solution to the resulting LP problem may be extremely unstable with respect to small "implementation errors" and have shown how to overcome this difficulty by switching from the "nominal design" to the one given by the Robust Counterpart methodology (Section 3.4.2). Now, what happens if we switch from the uniform norm of the discrepancy to the $\|\cdot\|_2$-norm, i.e., specify the optimal design as the optimal solution x_j^* to the usual Least Squares problem

$$\| Z_* - \sum_{j=1}^{10} x_j Z_j \|_2 \equiv \sqrt{\frac{1}{N}\sum_{\theta\in T}(Z_*(\theta) - \sum_{j=1}^{10} x_j Z_j(\theta))^2} \to \min \qquad \text{(LS)}$$

N being the cardinality of the grid T? (Note that the factor $\frac{1}{N}$ under the square root does not influence the Least Squares solution. The only purpose of it is to make the figures comparable with those related to the case of the best uniform approximation: with our normalization, we have

$$\| Z_* - \sum_{j=1}^{10} x_j Z_j \|_2 \le \| Z_* - \sum_{j=1}^{10} x_j Z_j \|_\infty$$

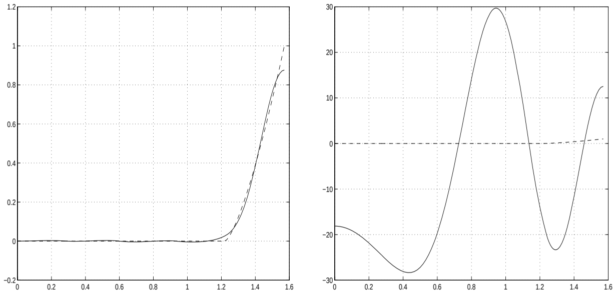
for every x.)

(LS) is just a Linear Algebra problem: its optimal solution x^* is exactly the solution to the "normal system of equations"

$$(A^T A)x = A^T b,$$

where $b = \frac{1}{\sqrt{N}}(Z_*(\theta_1), ...Z_*(\theta_N))^T$ and A is the $N \times 10$ matrix with the columns $\frac{1}{\sqrt{N}}(Z_j(\theta_1), ..., Z_j(\theta_N))^T$.

Now let us check what are the stability properties of the Least Squares solution. Same as in Section 3.4.2, assume that the actual amplification coefficients x_j are obtained from their nominal values x_j^* by random perturbations $x_j^* \mapsto x_j = p_j x_j^*$, where p_j are independent random factors with expectations 1 taking values in the segment $[0.999, 1.001]$. With experience given by Section 3.4.2 we shall not be too surprised by the fact that these stability properties are extremely poor:



"Dream and reality": The nominal diagram (left, solid) and an actual diagram (right, solid) [dashed: the target diagram]

> The target diagram varies from 0 to 1, and the nominal diagram – the one corresponding to $x_j = x_j^*$ – is at the $\| \cdot \|_2$-distance 0.0178 from the target diagram. An actual diagram varies from ≈ -30 to ≈ 30 and is at the $\| \cdot \|_2$-distance 20.0 (1124 times larger!) from the target.

The reason for instability of the nominal Least Squares solution is, basically, the same as in the case of the nominal Tschebyshev solution. The system of "basic functions" Z_j is nearly linearly dependent (this unpleasant phenomenon is met in the majority of approximation problems arising in applications). As a result, the normal system of equations becomes ill conditioned, and its solution becomes large. And of course even small relative perturbations of very large nominal amplification coefficients may cause (and in fact cause) huge perturbations in the actual diagram...

In order to resolve the difficulty, let us use the Robust Counterpart methodology. What we are solving now is a conic quadratic inequality, so that we may use the results of this section. The question, however, is how to define a reasonable uncertainty set. Let us look at this question for a general conic quadratic inequality

$$\| Ax + b \|_2 \leq c^T x + d, \tag{CQI}$$

assuming, as it is the case in our Antenna example, that the uncertainty comes from the fact that the entries x_j of a candidate solution are affected by "noise":

$$x_j \mapsto x_j(1 + \kappa_j \epsilon_j), \tag{4.5.8}$$

where $\epsilon_1, ..., \epsilon_n$ are independent random variables with zero means and unit standard deviations and $\kappa_j \geq 0$ are (deterministic) "relative implementation errors".

Our question is: what is a "reliable" version of the inequality (CQI) in the case of random perturbations (4.5.8) in x? Note that it is the same – to think that the data A, b, c, d in (CQI) are fixed and there are perturbations in x and to think that there are no perturbations in x, but the data in (CQI) are "perturbed equivalently". With this latter viewpoint, how could we define an uncertainty set $\mathcal{U}$ in such a way that the robust counterpart of the uncertain conic inequality

$$\| A'x + b' \|_2 \leq (c')^T x + d' \quad \forall (A', b', c', d') \in \mathcal{U}$$

would be a "reliable version" of (CQI)?

A nice feature of the question we have posed is that it is not a purely mathematical question: it has to do with modeling, and modeling – description of a real world situation in mathematical terms – always is beyond the scope of the mathematics itself. It follows that in our current situation we are free to use whatever arguments we want – detailed (and time-consuming) mathematical analysis of the random variables we are dealing with, common sense, spiritualism,...; proof of our pudding will be eating – testing the resulting robust solution.

Since we have no that much experience with spiritualism and the detailed mathematical analysis of the situation seems to be too involving, we prefer to rely upon common sense, namely, to choose somehow a "safety parameter" ω of order of 1 and to make use of the following "principle":

A nonnegative random variable "never" is larger than ω times its expected value.

In less extreme form, we are not going to take care of "rare" events – those where the above "principle" is violated.

Equipped with our "principle", let us build a "reliable" version of (CQI) as follows. First, let us "separate" the influence of the perturbations on the left and on the right hand sides of our inequality. Namely, the value of the right hand side $c^T y + d$ at a randomly perturbed x – i.e., at a random vector y with the coordinates

$$y_i = x_i + \kappa_i x_i \epsilon_i$$

– is a random variable of the form

$$c^T x + d + \eta,$$

η being a zero mean random variable with the standard deviation $V^{1/2}(x)$, where

$$V(x) = \sum_{i=1}^{n} c_i^2 \kappa_i^2 x_i^2. \tag{4.5.9}$$

According to our "principle", the value of the right hand side in (CQI) is "never" less than the quantity

$$R(x) = c^T x + d - \omega V^{1/2}(x). \tag{4.5.10}$$

It follows that if we shall ensure that the value of the left hand side in (CQI) will "never" be larger than $R(x)$, the perturbations in x will "never" result in violating (CQI). This scheme is a bit conservative (it may happen that a perturbation which increases the left hand side of (CQI) increases the right hand side as well), but this is life – we are aimed at getting something "tractable" and thus may afford ourselves to be conservative; recall that at the moment we are not obliged to be rigorous!

Now we came to the situation as follows. We would like $R(x)$ to be a "reliable upper bound" on the values of the left hand side in (CQI), and this requirement on x will be the "reliable version" of (CQI) we are building. And what are "typical" values of the left hand side of (CQI)? These are Euclidean norms of the random vector

$$z \equiv z(x) + \zeta = Ax + b + \zeta,$$

where

$$\zeta = \sum_{j=1}^{n} \kappa_j x_j \epsilon_j A_j$$

(A_j are the columns of A); note that the vector ζ is a random vector with zero mean. For a given x, let e be the unit vector collinear to the vector $Ax + b$. We have

$$\| z \|_2^2 = l_x^2 + \| \zeta_x \|_2^2,$$

where l_x is the length of the projection of z on the line spanned by e and ζ_x is the projection of z (or, which is the same, of ζ) onto the orthogonal complement to this line.

We have $\| \zeta_x \|_2^2 \leq \| \zeta \|_2^2$, and the expected value of $\| \zeta \|_2^2$ is

$$S(x) = \sum_{j=1}^{n} S_{jj} x_j^2, \quad S_{jj} = \kappa_j^2 A_j^T A_j. \tag{4.5.11}$$

According to our "principle", $\| \zeta_x \|_2$ is "never" greater than $\omega S^{1/2}(x)$.

Now let us find a "never"-type upper bound for l_x – for the length of the projection of z onto the line spanned by the vector $Ax + b$ (or, which is the same, by the unit vector e). We have, of course,

$$|l_x| \leq \| Ax + b \|_2 + |e^T \zeta|.$$

Now, $e^T \zeta$ is the random variable

$$e^T \zeta = \sum_{i,j=1}^{n} a_{ij} e_i \kappa_j x_j \epsilon_j = \sum_{j=1}^{n} \left[\sum_{i=1}^{n} a_{ij} e_i \right] \kappa_j x_j \epsilon_j$$

with the zero mean and the variance

$$v = \sum_{j=1}^{n} \left[\kappa_j \sum_{i=1}^{n} a_{ij} e_i \right]^2 x_j^2. \tag{4.5.12}$$

In order to end up with "tractable" formulae, it makes sense to bound from above the latter quantity by a simple function of x (this is not the case for the quantity itself: e depends on x in a not that pleasant way!). A natural bound of this type is

$$v \leq \sigma^2 \| x \|_\infty^2$$

where $\| x \|_\infty = \max_i |x_i|$

$$\sigma = |A \, \mathrm{Diag}(\kappa_1, ..., \kappa_n)| \tag{4.5.13}$$

$|\cdot|$ being the operator norm. Indeed, the coefficients at x_j^2 in (4.5.12) are the squared entries of the vector $\mathrm{Diag}(\kappa_1, ..., \kappa_n)A^T e$, and since e is unit, the sum of these squared entries does not exceed σ^2.

According to our "principle", the absolute value of $e^T\zeta$ "never" exceeds the quantity

$$\omega\sigma \parallel x \parallel_\infty .$$

Combining all our observations together, we conclude that $\parallel z \parallel_2$ "never" exceeds the quantity

$$L(x) = \sqrt{[\parallel Ax+b \parallel_2 + \omega\sigma \parallel x \parallel_\infty]^2 + \omega^2 S(x)}.$$

Consequently, the "reliable" version of (CQI) is the inequality $L(x) \le R(x)$, i.e., the inequality

$$\begin{array}{c}\sqrt{[\parallel Ax+b \parallel_2 + \omega\sigma \parallel x \parallel_\infty]^2 + \omega^2 S(x)} \le c^T x + d - \omega V^{1/2}(x) \\ \left[\begin{array}{rcl} \sigma & = & |A\,\mathrm{Diag}(\kappa_1, ..., \kappa_n)| \\ S(x) & = & \sum_{j=1}^n [\kappa_j^2 A_j^T A_j] x_j^2 \\ V(x) & = & \sum_{j=1}^n \kappa_j^2 c_j^2 x_j^2 \end{array}\right]\end{array} \tag{4.5.14}$$

The resulting inequality is "quite tractable" – it can be represented by the following system of linear and conic quadratic inequalities:

$$\begin{array}{rcl} t_1 + t_2 & \le & c^T x + d; \\ \omega \parallel Wx \parallel_2 & \le & t_1, \\ & & W = \mathrm{Diag}(\kappa_1 c_1, \kappa_2 c_2, ..., \kappa_n c_n); \\ \parallel Ax+b \parallel_2 & \le & s_1; \\ |x_i| & \le & s_2,\ i = 1, ..., n; \\ \left\| \begin{pmatrix} s_1 + \omega\sigma s_2 \\ \omega D x \end{pmatrix} \right\|_2 & \le & t_2, \\ & & D = \mathrm{Diag}(|\kappa_1| \parallel A_1 \parallel_2, |\kappa_2| \parallel A_2 \parallel_2, ..., |\kappa_n| \parallel A_n \parallel_2), \end{array} \tag{4.5.15}$$

t_1, t_2, s_1, s_2 being additional design variables.

It is worthy to mention – just for fun! – that problem (4.5.15) is, in a sense, the robust counterpart of (CQI) associated with a specific ellipsoidal uncertainty. Indeed, let us rewrite (CQI) equivalently as the following crazy system of inequalities with respect to x and additional variables t_1, t_2, s_1, s_2:

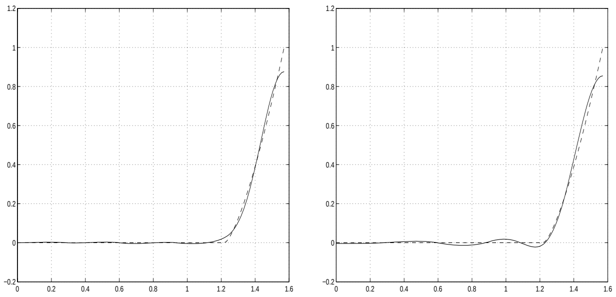
$$\begin{array}{rcl} t_1 + t_2 & \le & c^T x + d; \\ \parallel \mathrm{Diag}(\alpha_1, ..., \alpha_n)x \parallel_2 & \le & t_1 \\ & & \alpha_1 = 0, \alpha_2 = 0, ..., \alpha_n = 0; \\ \parallel Ax+b \parallel_2 & \le & s_1; \\ |x_i| & \le & s_2,\ i = 1, ..., n; \\ \left\| \begin{pmatrix} s_1 + \beta_0 s_2 \\ \mathrm{Diag}(\beta_1, \beta_2, ..., \beta_n)x \end{pmatrix} \right\|_2 & \le & t_2, \\ & & \beta_0 = 0, \beta_1 = 0, ..., \beta_n = 0. \end{array} \tag{4.5.16}$$

Now assume that the data α_i, β_i in this system are uncertain, namely, linearly depend on "perturbation" u varying in the segment $[-1, 1]$ (ellipsoidal uncertainty!):

$$\begin{array}{rcl} \alpha_i & = & [\omega\kappa_i c_i]u,\ i = 1, ..., n; \\ \beta_0 & = & [\omega\sigma]u; \\ \beta_i & = & [\omega|\kappa_i| \parallel A_i \parallel_2]u,\ i = 1, ..., n. \end{array}$$

It is easily seen that the robust counterpart of (4.5.16) – which is, basically, our original conic inequality (CQI) – is exactly (4.5.15). Thus, (4.5.15) is the robust counterpart of (CQI) corresponding to an ellipsoidal uncertainty, *and this uncertainty affects the data which are not seen in* (CQI) *at all!*

What about the pudding we have cooked? How this approach works in our Antenna Synthesis problem? It works fine! Here is the picture (safety parameter $\omega = 1$):



"Dream and reality": The nominal Least Squares diagram (left, solid) and an actual diagram yielded by Robust Least Squares (right, solid) [dashed: the target diagram]

The robust optimal value in our uncertain Least Squares problem is 0.0236 (approximately by 30% larger than the nominal optimal value 0.0178 – the one corresponding to the usual Least Squares with no "implementation errors"). The $\|\cdot\|_2$-distance between the target diagram and the actual diagram shown on the picture is the same 0.0236. When generating a sample of random diagrams yielded by our Robust Least Squares design, this distance varies in the third digit after the decimal dot only: in a sample of 40 diagrams, the distances to the target were varying from 0.0236 to 0.0237. And what happens when in course of our design we thought that the "implementation errors" will be 0.1%, while in reality they are 1% – 10 times larger? Nothing that bad: now the $\|\cdot\|_2$-distances from the target in a sample of 40 diagrams vary from 0.0239 to 0.0384.

4.6 Applications, IV: Synthesis of filters and antennae arrays

The models to be presented in this section originate from [9]. Consider a discrete time linear time invariant SISO ("single input – single output") dynamic system (cf. Section 1.2.3). Such a system $\mathcal{H}$ takes on input a two-side sequence of reals $u(\cdot) = \{u(k)\}_{k=-\infty}^{\infty}$ and converts it into the output sequence $\mathcal{H}u(\cdot)$ according to

$$\mathcal{H}u(k) = \sum_{l=-\infty}^{\infty} u(l)h(k-l),$$

where $h = \{h(k)\}_{k=-\infty}^{\infty}$ is a real sequence characteristic for $\mathcal{H}$ – the *impulse response* of $\mathcal{H}$. Let us focus on the case of a *filter* – a *casual* system with *finite memory*. Causality means that $h(k) = 0$ for $k < 0$, so that the output $\mathcal{H}u$ at every time instant k is independent of the input after this instant, while the property to have memory n means that $h(k) = 0$ for $k \geq n$, so that $\mathcal{S}u(k)$, for every k, depends on n inputs $u(k), u(k-1), ..., u(k-n+1)$ only. Thus, a *filter of order* n is just a sequence $h = \{h(k)\}_{k=-\infty}^{\infty}$ with $h(k) = 0$ for all negative k and all $k \geq n$. Of course, a filter $\{h(k)\}_{k=-\infty}^{\infty}$ of order n can be identified with the vector $h = (h(0), ..., h(n-1))^T \in \mathbf{R}^n$.

A natural way to look at a filter $h(\cdot)$ of order n is to associate with it the polynomial

$$\hat{h}(z) = \sum_{l=0}^{n-1} h(l)z^l.$$

As any other polynomial on the complex plane, $\hat{h}$ is completely determined by its restriction on the unit circumference $|z| = 1$. This restriction, regarded as 2π-periodic function of real variable

ω,

$$H(\omega) = \hat{h}(\exp\{\mathrm{i}\omega\}) = \sum_{l=0}^{n-1} h(l)\exp\{\mathrm{i}l\omega\}$$

is called the *frequency response* of the filter $h(\cdot)$. The frequency response is just a trigonometric polynomial (with complex coefficients) of ϕ of degree $\leq n-1$.

The meaning of the frequency response is quite transparent: if the input to the filter is a harmonic oscillation

$$u(k) = \Re(a\exp\{\mathrm{i}\omega k\}) = |a|\cos(\omega k + \arg(a)) \quad [a \in \mathbf{C} - \text{“complex amplitude”}]$$

then the output is

$$\begin{array}{rcl} \mathcal{H}u(k) & = & \sum_{l=-\infty}^{\infty} u(l)h(k-l) \\ & = & \sum_{l=0}^{n-1} h(l)u(k-l) \\ & = & \Re\sum_{l=0}^{n-1} h(l)a\exp\{\mathrm{i}\omega(k-l)\} \\ & = & \Re H(-\omega)a\exp\{i\omega k\}. \end{array}$$

Thus, the output is a harmonic oscillation of the same frequency as the input, and the complex amplitude of the output is $H(-\omega)$ times the complex amplitude of the input. Thus, the frequency response says how the filter affects a harmonic oscillation of certain frequency ω: the filter multiplies the real amplitude of the oscillation by $|H(-\omega)|$ and shifts the initial phase of the oscillation by $\arg(H(-\omega))$. Since typical “inputs of interest” can be decomposed in sums of harmonic oscillations, typical design specifications in filter synthesis problems have to do with the frequency response – they prescribe its behaviour on a segment $\Delta \in [-\pi, \pi]$. Note that the coefficients $h(l)$ of filter are real, so that the frequency response possesses an evident symmetry:

$$H(-\omega) = H^*(\omega),$$

where z^* denotes the complex conjugate of a complex number z. Consequently, it makes sense to specify the behaviour of a frequency response on the segment $[0, \pi]$ only.

The simplest type of design specifications would be to fix a “target function” $F(\omega)$, $0 \leq \omega \leq \pi$, and to require from H to be as close as possible (e.g., in the uniform metric) to the target. This would result in a Tschebyshev-type problem

$$\max_{0\leq\omega\leq\pi} |F(\omega) - \sum_{l=0}^{n-1} h(l)\exp\{\mathrm{i}l\omega)| \to \min,$$

the (real) design variables being $h(0), ..., h(n-1)$. After discretization in ω, we would end up with a simple conic quadratic (or even a Linear Programming) program.

The outlined design specifications are aimed at prescribing both what a filter does with the real amplitudes of harmonic oscillations and their initial phases. However, in most of applications the only issue of interest is how the filter affects the real amplitudes of harmonic oscillations of different frequencies, and not how it shifts the phases. Consequently, typical design specifications prescribe the behaviour of $|H(\omega)|$ only, e.g., require from this function to be between two given bounds:

$$L(\omega) \leq |H(\omega)| \leq U(\omega),\ 0 \leq \omega \leq \pi. \tag{B}$$

E.g., when designing a low-pass filter, we are interested to reproduce exactly the amplitudes of oscillations with frequencies below certain level and to suppress oscillations with frequencies higher than another prescribed level, i.e., the specifications are like

$$1-\epsilon \leq |H(\omega)| \leq 1+\epsilon, 0 \leq \omega \leq \underline{\omega}, \quad |H(\omega)| \leq \epsilon, \overline{\omega} \leq \omega \leq \pi.$$

When trying to process the constraint of the latter type, we meet with a severe difficulty: $|H(\omega)|$ is *not* a convex function of our natural design parameters $h(0),...,h(n-1)$. There is, however, a way to overcome the difficulty: it turns out that the function $|H(\omega)|^2$ can be "linearly parameterized" by properly chosen new design parameters, so that lower and upper bounds on $|H(\omega)|^2$ become linear – and thus tractable – constraints on the new design parameters. And it is, of course, the same – to impose bounds on $|H(\omega)|$ and on $|H(\omega)|^2$.

A "proper parameterization" of the function $R(\omega) \equiv |H(\omega)|^2$ is very simple. We have

$$\begin{gathered} H(\omega) = \sum_{l=0}^{n-1} h(l)\exp\{\mathrm{i}l\omega\} \\ \Downarrow \\ R(\omega) = \left(\sum_{l=0}^{n-1} h(l)\exp\{\mathrm{i}l\omega\}\right)\left(\sum_{l=0}^{n-1} h(l)\exp\{-\mathrm{i}l\omega\}\right) = \sum_{l=-(n-1)}^{n-1} r(l)\exp\{\mathrm{i}l\omega\}, \\ r(l) = \sum_p h(p)h(l+p). \end{gathered}$$

The reals $\{r(l)\}_{l=-(n-1)}^{n-1}$ – they are called the *autocorrelation coefficients* of the filter h – are exactly the parameters we need. Note that $r(l) = r(-l)$:

$$r(-l) = \sum_p h(p)h(\underbrace{-l+p}_{q}) = \sum_q h(l+q)h(q) = r(l),$$

so that

$$R(\omega) = \sum_{l=-(n-1)}^{n-1} r(l)\exp\{\mathrm{i}l\omega)\} = r(0) + 2\sum_{l=1}^{n-1} r(l)\cos(l\omega).$$

Thus, $R(\cdot)$ is just an even trigonometric polynomial of degree $\leq n-1$ with real coefficients, and $r(\cdot)$ are, essentially, the coefficients of this trigonometric polynomial.

The function $R(\omega) = |H(\omega)|^2$ is linearly parameterized by the coefficients $r(\cdot)$, which is fine. These coefficients, however, cannot be arbitrary: not every even trigonometric polynomial of a degree $\leq n-1$ is $|H(\omega)|^2$ for certain $H(\omega) = \sum_{l=0}^{n-1} h(l)\exp\{\mathrm{i}l\omega\}$! The coefficients $r = (r(0), 2r(1), 2r(2), ..., 2r(n-1))^T$ of "proper" even trigonometric polynomials $R(\cdot)$ – those which are squares of modulae of frequency responses – form a proper subset $\mathcal{R}$ in $\mathbf{R}^n$, and in order to be able to handle constraints of the type (B), we need a "tractable" representation of $\mathcal{R}$. Such a representation does exist, due to the following fundamental fact:

Proposition 4.6.1 [Spectral Factorization Theorem] *A trigonometric polynomial*

$$R(\omega) = a_0 + \sum_{l=1}^{n-1}(a_l\cos(l\omega) + b_l\sin(l\omega))$$

with real coefficients $a_0,...,a_{n-1},b_1,...,b_{n-1}$ *can be represented as*

$$\left|\sum_{l=0}^{n-1} h(l)\exp\{\mathrm{i}l\omega\}\right|^2 \qquad (*)$$

for properly chosen complex $h(0),h(1),...,h(n-1)$ *if and only if* $R(\omega)$ *is nonnegative on* $[-\pi,\pi]$. *An even trigonometric polynomial* $R(\omega)$ *of degree* $\leq n-1$ *can be represented in the form* (*) *with real* $h(0),...,h(n-1)$ *if and only if it is nonnegative on* $[-\pi,\pi]$ *(or, which is the same, on* $[0,\pi]$*).*

Postponing the proof of Proposition till the end of this section, let us look what are the consequences. The Proposition says that the set $\mathcal{R} \in \mathbf{R}^n$ of the coefficients of those even trigonometric polynomials of degree $\leq n-1$ which are squares of modulae of frequency responses of filters of order n is exactly the set of coefficients of those even trigonometric polynomials of degree $\leq n-1$ which are nonnegative on $[-\pi,\pi]$. Consequently, this set is SDr with an explicit semidefinite representation (Example 18c, Section 4.2). Thus, passing from our original design variables $h(0),...,h(n-1)$ to the new design variables $r \in \mathcal{R}$, we make the design specifications of the form (B) a (semi-infinite) system of *linear* constraints on the design variables varying in a SD-representable set. As a result, we get a possibility to handle numerous Filter Synthesis problems with design specifications of the type (B) via Semidefinite Programming. Let us look at a couple of examples (for more examples and more details, see [9]).

Example 1: "Low-pass filter". Assume we are given a number of (possibly, overlapping) segments $\Delta_k \subset [0,\pi]$, $k=1,...,K$, along with nonnegative continuous functions $S_k(\omega) \leq T_k(\omega)$ defined on these segments, and our goal is to design a filter of a given order n with $|H(\omega)|^2$ being at every segment Δ_k as close as possible to the "strip" between S_k and T_k. Taking into account that a natural measure of closeness in Filter Synthesis problems is the "relative closeness", we can pose the problem as

$$\epsilon \to \min \mid \frac{1}{(1+\epsilon)} S_k(\omega) \leq |H(\omega)|^2 \leq (1+\epsilon)T_k(\omega) \quad \forall \omega \in \Delta_k \quad \forall k=1,...,K. \tag{P}$$

E.g., when dealing with two non-overlapping segments $\Delta_1 = [0,\underline{\omega}]$ and $\Delta_2 = [\overline{omega},\pi]$ and setting $S_1 \equiv T_1 \equiv 1$, $S_2 \equiv 0$, $T_2 \equiv \beta$ with small positive β, we come to the problem of designing a low-pass filter: $|H(\omega)|$ should be as close to 1 as possible in Δ_1 and should be small in Δ_2.

Problem (P) written down via the autocorrelation coefficients becomes

$$\begin{array}{llrcll}
 & & \epsilon & \to & \min & \\
\text{s.t.} & & & & & \\
(a) & \delta S_k(\omega) \leq R(\omega) \equiv r(0) + 2\sum_{l=1}^{n-1} r(l)\cos(l\omega) & & \leq & (1+\epsilon)T_k(\omega) \quad \forall \omega \in \Delta_k, & \\
 & & & & k=1,...,K; & (\mathrm{P}') \\
(b) & & \delta(1+\epsilon) & \geq & 1; & \\
(c) & & \delta, \epsilon & \geq & 0; & \\
(d) & & r & \in & \mathcal{R}. &
\end{array}$$

Indeed, $(a)--(c)$ say that

$$\frac{1}{1+\epsilon} S_k(\omega) \leq R(\omega) \leq (1+\epsilon)T_k(\omega), \ \omega \in \Delta_k, \ k=1,...,K,$$

while the role of the constraint (d) is to express the fact that $R(\cdot)$ comes from certain filter of order n.

Problem (P$'$) is not exactly a semidefinite program – the only obstacle is that the constraints (a) are "semi-infinite". In order to overcome this difficulty, we can use discretization in ω (i.e., can replace segments Δ_k by fine finite grids), thus approximating (P) by a semidefinite program. In many cases we can even avoid approximation. Indeed, assume that all S_k and T_k are trigonometric polynomials. As we know from Example 18c, Section 4.2, the restriction that a trigonometric polynomial $R(\omega)$ majorates (or is majorated by) another trigonometric polynomial is a SDr constraint on the coefficients of the polynomials, so that in the case in question the constraints (a) are SDr restrictions on r,δ,ϵ.

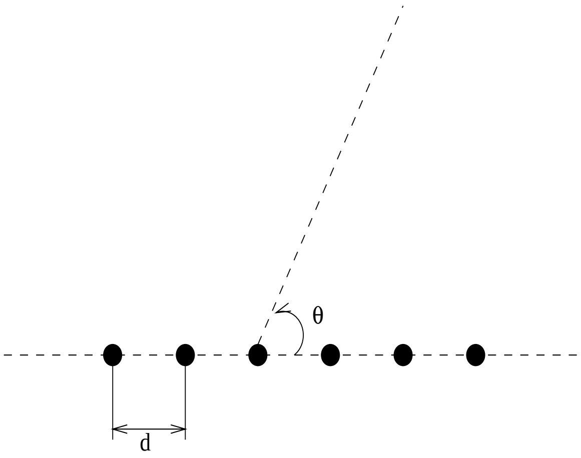
Note that instead of minimizing the "relative uniform distance" between $|H(\omega)|^2$ and given targets we can minimize the "relative $\|\cdot\|_2$-distance". A natural form of the latter problem is

$$\begin{array}{rcl} \epsilon & \to & \min \\ \text{s.t.} & & \\ \frac{1}{1+\epsilon_k(\omega)}S_k(\omega) \leq |H(\omega)|^2 & \leq & (1+\epsilon_k(\omega))T_k(\omega),\ \omega\in\Delta_k, \\ & & k=1,...,K; \\ \sqrt{\frac{1}{|\Delta_k|}\int\limits_{\Delta_k}\epsilon_k^2(\omega)d\omega} & \leq & \epsilon,\ k=1,...,K. \end{array}$$

After discretization in ω – replacing Δ_k by a finite set $\Omega_k\subset\Delta_k$ – we can pose the problem as the semidefinite program

$$\begin{array}{rcl} \epsilon & \to & \min \\ \text{s.t.} & & \\ \delta_k(\omega)S_k(\omega) \leq R(\omega) \equiv r(0)+2\sum_{l=1}^{n-1} r(l)\cos(l\omega) & \leq & (1+\epsilon_k(\omega))T_k(\omega),\ \forall\omega\in\Omega_k, \\ & & k=1,...,K; \\ \delta_k(\omega)(1+\epsilon_k(\omega)) & \geq & 1,\ \forall\omega\in\Omega_k, \\ & & k=1,...,K; \\ \sqrt{\frac{1}{\text{Card}(\Omega_k)}\sum_{\omega\in\Omega_k}\epsilon_k(\omega)^2} & \leq & \epsilon,\ k=1,...,K; \\ r & \in & \mathcal{R}. \end{array}$$

Example 2. Synthesis of array of antennae. Consider a linear array of antennae (see Section 1.2.4) comprised of n equidistantly placed harmonic oscillators in the plane XY:



Restricting the diagram of the array to directions from the plane XY only, we can easily see that the diagram of the array depends on the angle θ between the direction in question and the line where the oscillators are placed and is nothing but

$$Z(\theta)=\sum_{l=0}^{n-1} z_l\exp\{-\mathrm{i}l\Omega(\theta)\},\quad \Omega(\theta)=-\frac{2\pi d}{\lambda}\cos\theta,$$

where $z_0,z_1,...,z_{n-1}$ are the (complex) amplification coefficients of the oscillators and λ is the wavelength.

In our previous Antenna Synthesis considerations, we were interested in the case when the design specifications are aimed to get a diagram as close as possible to a given target diagram $Z_*(\theta)$. At the same time, what is of interest in many Antenna Synthesis problems is only the modulus $|Z(\theta)|$ of the resulting diagram (this modulus is responsible for the energy sent by antenna in a direction θ). Thus, same as in the Filter Synthesis, in many Antenna Synthesis problems we are interested in a "prescribed behaviour" of the function $|Z(\theta)|$. And here again Proposition 4.6.1 is the key for handling the problem via Convex Optimization. Indeed, defining the function

$$H(\omega) = \sum_{l=0}^{n-1} z_l \exp\{\mathrm{i}l\omega\},$$

we get a frequency response of a *complex* filter $h = \{h(l) = z_l\}_{l=0}^{n-1}$ such that

$$Z(\theta) = H(\Omega(\theta)).$$

It follows that to impose restrictions, like upper and lower bounds, on the function $|Z(\theta)|$, $0 \le \theta \le \pi$, is the same as to impose bounds of the same type on the function $|H(\omega)|$ in the segment Δ of values taken by $\Omega(\theta)$ when θ varies from 0 to π. Assuming (which normally indeed is the case) that $\lambda > 2d$, we observe that the mapping $\theta \mapsto \Omega(\theta)$ is a one-to-one mapping of the segment $[0, \pi]$ on certain segment $\Delta \subset [-\pi, \pi]$, so that design specifications on $|Z(\theta)|$ can be easily converted to design specifications on $|H(\theta)|$. E.g., to build a diagram with $|Z(\theta)|$ as close as possible to given "stripes" becomes the problem

$$\begin{array}{c} \epsilon \to \min \mid \frac{1}{1+\epsilon} S_k(\omega) \le R(\omega) \le (1+\epsilon)T_k(\omega) \quad \forall \omega \in \Delta_k,\ k = 1, ..., K. \\ R(\omega) = |H(\omega)|^2 = \sum_{l=-(n-1)}^{n-1} r(l) \exp\{\mathrm{i}l\omega\} \end{array} \qquad (\mathrm{P}_{\mathbf{C}})$$

The only differences between ($\mathrm{P}_{\mathbf{C}}$) and the problem (P) we have investigated in Example 1 is that now the autocorrelation coefficients come from *complex* amplification coefficients z_l – the "actual" design variables:

$$r(k) = \sum_{l=0}^{n-1} z_l^* z_{l+k}$$

and are therefore complex; as it is immediately seen, these complex coefficients possess the symmetry

$$r(-k) = r^*(k),\ |k| \le n-1,$$

the symmetry reflecting the fact that the function $R(\omega)$ is real-valued. This function, as it is immediately seen, is just a real trigonometric polynomial (now not necessary even) of degree $\le n-1$:

$$R(\omega) = \rho(0) + \sum_{l=1}^{n-1} (\rho(2l-1)\cos(l\omega) + \rho(2l)\sin(l\omega))$$

with real vector of coefficients $\rho = (\rho(0), ..., \rho(2n-2))^T \in \mathbf{R}^{2n-1}$. The vector of these coefficients can be treated as our new design vector, and Proposition 4.6.1 says when such a vector gives rise to a function $R(\omega)$ which indeed is of the form $|H(\omega)|^2$ with $H(\omega) = \sum_{l=0}^{n-1} r_l \exp\{\mathrm{i}l\omega\}$: this is the case if and only if the trigonometric polynomial $R(\cdot)$ is nonnegative on $[-\pi, \pi]$. As we remember from Example 18c, Section 4.2, the set $\mathcal{C}$ of the vectors of coefficients r of this type is SD-representable.

In view of the outlined observations, problem ($\mathrm{P}_{\mathbf{C}}$) can be posed as a "semi-infinite" semidefinite program in exactly the same way as problem (P), and this semi-infinite program can be approximated by (or sometimes is equivalent to) a usual semidefinite program. E.g., when approximating segments Δ_k by fine finite grids Ω_k, we approximate $(\mathrm{P}_{\mathbf{C}})$ by the semidefinite program

$$\begin{array}{rcl} & \epsilon \rightarrow & \min \\ \text{s.t.} & & \\ \delta S_k(\omega) \leq R(\omega) \equiv \rho(0) + \sum_{l=1}^{n-1}(\rho(2l-1)\cos(l\omega) + \rho(2l)\sin(l\omega)) & \leq & (1+\epsilon)T_k(\omega) \\ & & \forall \omega \in \Omega_k, k = 1, ..., K; \\ \delta(1+\epsilon) & \geq & 1; \\ \delta, \epsilon & \geq & 0; \\ \rho & \in & \mathcal{C}, \end{array}$$

the design variables in the problem being δ, ϵ and $\rho = (\rho(0), ..., \rho(2n-2))^T \in \mathbf{R}^{2n-1}$.

Proof of Proposition 4.6.1. In fact the Proposition is a particular case of a fundamental result of Functional Analysis – the Theorem of Spectral Decomposition of a unitary operator on Hilbert space. The particular case we are interested in admits a quite elementary and straightforward proof.

Let us first prove that a real trigonometric polynomial

$$R(\omega) = c_0 + \sum_{l=1}^{n-1}(a_0 \cos(l\omega) + b_0 \sin(l\omega))$$

can be represented as $\left|\sum_{l=0}^{n-1} h(l) \exp\{\mathrm{i}l\omega\}\right|^2$ for some complex coefficients $h(l)$ if and only if $R(\cdot)$ is nonnegative on $[-\pi, \pi]$. The necessity is evident, so let us focus on the sufficiency. Thus, assume that R is nonnegative, and let us prove that then R admits the required decomposition.

1^0. It suffices to prove the announced statement in the case when $R(\omega)$ is strictly positive on $[-\pi, \pi]$ rather than merely nonnegative. Indeed, assume that our decomposition is possible for positive trigonometric polynomials. Given a nonnegative polynomial R, let us apply our assumption to the positive trigonometric polynomial $R(\omega) + \epsilon$, $\epsilon > 0$:

$$R(\omega) + \epsilon = \left|\sum_{l=0}^{n-1} h_\epsilon(l) \exp\{\mathrm{i}l\omega)\right|^2.$$

From this representation it follows that

$$c_0 + \epsilon = \sum_{l=0}^{n-1} |h_\epsilon(l)|^2,$$

whence the coefficients $h_\epsilon(l)$ remain bounded as $\epsilon \rightarrow +0$. Taking as h an accumulation point of the vectors h_ϵ as $\epsilon \rightarrow +0$, we get

$$R(\omega) = \left|\sum_{l=0}^{n-1} h(l) \exp\{\mathrm{i}l\omega\}\right|^2,$$

as required.

2^0. Thus, it suffices to consider the case when R is a positive trigonometric polynomial. And of course we may assume that the degree of R is exactly $n-1$, i.e., that $a_{n-1}^2 + b_{n-1}^2 > 0$.

We can rewrite R in the form

$$R(\omega) = \sum_{l=-(n-1)}^{n-1} r(l) \exp\{\mathrm{i}l\omega\}; \tag{4.6.1}$$

since R is real-valued, we have

$$r(l) = r^*(-l).|l| \leq n-1. \tag{4.6.2}$$

Now consider the polynomial

$$P(z) = z^(n-1) \left(\sum_{l=-(n-1)}^{n-1} r(l) z^l \right).$$

This polynomial is of degree exactly $2(n-1)$, is nonzero at $z = 0$ and it has no zeros on the unit circumference (since $|P(\exp\{\mathrm{i}\omega\})| = R(\omega)$). Moreover, from (4.6.2) it immediately follows that if λ is a root of $P(z)$, then also $(\lambda^*)^{-1}$ is a root of the polynomial of exactly the same multiplicity as λ. It follows that the roots of the polynomial P can be separated in two non-intersecting groups: $(n-1)$ roots λ_l, $l = 1, ..., n-1$, inside the unit circle and $(n-1)$ roots $1/\lambda_l^*$ outside the circle. Thus,

$$P(z) = \alpha \left[\prod_{l=1}^{n-1} (z - \lambda_l) \right] \left[\prod_{l=1}^{n-1} (z - 1/\lambda_l^*) \right].$$

Moreover, we have

$$\begin{array}{rcl} R(0) & = & P(1) \\ & = & \alpha \prod_{l=1}^{n-1} [(1-\lambda_l)(1-1/\lambda_l^*)] \\ & = & \alpha(-1)^{n-1} \left[\prod_{l=1}^{n-1} |1-\lambda_l|^2 \right] \left[\prod_{l=1}^{n-1} \lambda_l^* \right]^{-1}, \end{array}$$

and since $R(0) > 0$, the number

$$\alpha(-1)^{n-1} \left[\prod_{l=1}^{n-1} \lambda_l^* \right]^{-1}$$

is positive. Denoting this number by β^{-2} ($beta > 0$), let us set

$$H(\omega) = \beta \prod_{l=1}^{n-1} (\exp\{\mathrm{i}\omega\} - \lambda_l) \equiv \sum_{l=0}^{n-1} h(l) \exp\{\mathrm{i}l\omega\}.$$

Then

$$\begin{array}{rcl} |H(\omega)|^2 & = & \beta^2 \left[\prod_{l=1}^{n-1} (\exp(\mathrm{i}\omega) - \lambda_l)(\exp\{-\mathrm{i}\omega\} - \lambda_l^*) \right] \\ & = & \beta^2 \exp\{-\mathrm{i}(n-1)\omega\}(-1)^{n-1} \left[\prod_{l=1}^{n-1} \lambda_l^* \right] \left[\prod_{l=1}^{n-1} [(\exp\{\mathrm{i}\omega\} - \lambda_l)(\exp\{\mathrm{i}\omega\} - 1/\lambda_l^*)] \right] \\ & = & \beta^2 \exp\{-\mathrm{i}(n-1)\omega\}(-1)^{n-1}\alpha^{-1} \left[\prod_{l=1}^{n-1} \lambda_l^* \right] P(\exp\{\mathrm{i}\omega\}) \\ & = & \exp\{-\mathrm{i}(n-1)\omega\} P(\exp\{\mathrm{i}\omega\}) \\ & = & R(\omega), \end{array}$$

as required.

3^0. To complete the proof of Proposition 4.6.1, it suffices to verify that if $R(\omega)$ is an *even* nonnegative trigonometric polynomial, then the coefficients $h(l)$ in the representation $R(\omega) = |\sum_{l=1}^{n-1} h(l) \exp\{\mathrm{i}l\omega\}|^2$ can be chosen real. But this is immediate: if $R(\cdot)$ is even, the coefficients $\rho(l)$ in (4.6.1) are real, so that $P(z)$ is a polynomial with real coefficients. Consequently, the complex numbers met among the roots $\lambda_1, ..., \lambda_{n-1}$ are met only in conjugate pairs, both members of a pair being roots of the same multiplicity. Consequently, the function $H(\omega)$ is $\hat{h}(\exp\{i\omega\})$, where $\hat{h}(\cdot)$ is a real polynomial, as claimed. ■

4.7 Applications, V: Design of chips

The model to be presented in this Section originates from [4]. Consider an RC-electric circuit, i.e., a circuit comprised of three types of elements: (1) resistors; (2) capacitors; (3) resistors in series combination with outer sources of voltage:

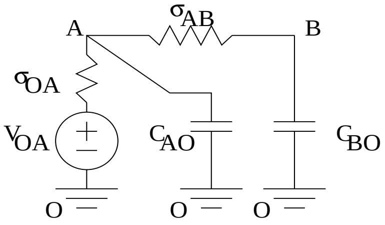


Figure 4.2. A simple circuit

Element OA:	outer supply of voltage V_{OA} and resistor with conductance σ_{OA}
Element AO:	capacitor with capacitance C_{AO}
Element AB:	resistor with conductance σ_{AB}
Element BO:	capacitor with capacitance C_{BO}

E.g., a chip is, electrically a very complicated circuit comprised of elements of the indicated type. When designing chips, the following characteristics of it are of primary importance:

- *Speed.* In a chip, the outer voltages are switching at certain frequency from one constant value to another. Every switch is accompanied by a "transition period"; during this period, the potentials/currents in the elements are moving from their previous values (corresponding to the static steady state for the "old" outer voltages to the values corresponding to the new static steady state. Since there are elements with "inertia" – capacitors – this transition period takes some time[12]. In order to ensure stable performance of the chip, the transition period should be essentially less than the time between subsequent switches in the outer voltages. Thus, the duration of the transition period is responsible for the speed at which the chip can be used.

- *Dissipated heat.* Resistors comprising the chip dissipate heat, and this heat should be taken away, otherwise the chip will be destroyed; this requirement is very serious for modern "high-density" chips. Thus, a characteristic of vital importance is the dissipated heat power.

The two desires – to get a chip with high speed (i.e., small transition period) and to get a chip with small dissipated heat – usually are not "coherent" to each other. As a result, a chip designer faces the tradeoff problem like "to get a chip with a given speed and as small dissipated heat as possible". We are about to demonstrate that the arising optimization problem belongs to the "semidefinite universe".

4.7.1 Building the model

A circuit

Mathematically, a circuit can be represented as a graph; the nodes of the graph correspond to the points where elements of the circuit are linked to each other, and the arcs correspond to the elements themselves. We may assume that the nodes are enumerated: 1,2,...,N, and that the arcs are (somehow) oriented, so that for every arc γ there exists its starting node $s(\gamma)$ and

[12] From purely mathematical viewpoint, the transition period takes infinite time – the currents/voltages approach asymptotically the new steady state, but never actually reach it. From the engineering viewpoint, however, we may think that the transition period is over when the currents/voltages become close enough to the new static steady state.

its destination node $d(\gamma)$. Note that we do not forbid "parallel arcs" – distinct arcs linking the same pairs of nodes (e.g., for circuit depicted on Fig. 4.2 we could orient the two arcs linking the "ground" O and A – the one with resistor and the one with capacitor – in the same way, which would give us two distinct parallel arcs). Let us denote by Γ the set of all arcs of our graph – all elements of our circuit, and let us equip an arc $\gamma \in \Gamma$ by a triple of parameters $v_\gamma, c_\gamma, \sigma_\gamma$ ("outer voltage, capacitance, conductance") as follows:

- For an arc γ representing a resistor: σ_γ is the conductance of the resistor, $c_\gamma = v_\gamma = 0$;
- For an arc γ representing a capacitor: c_γ is the capacitance of the capacitor, $v_\gamma = \sigma_\gamma = 0$;
- For an arc γ of the type "outer source of voltage – resistor" σ_γ is the conductance of the resistor, v_γ is the outer voltage, and $c_\gamma = 0$.

Transition period

Let us build a model for the duration of a transition period. The question we are addressing is: assume that before instant 0 the outer voltages were certain constants and the circuit was in the corresponding static steady state. At the instant $t = 0$ the outer voltages jump to new values v_γ and remain at these values. What will happen with the circuit? The answer is given by the Kirchoff laws and is as follows. Let $u_i(t)$, $t \geq 0$, be the potentials at the nodes $i = 1, ..., N$ at time instant t, and let $I_\gamma(t)$ be the currents in arcs $\gamma \in \Gamma$ an the instant.

The first law of Kirchoff says that

$$\begin{array}{rcll} I_\gamma(t) & = & \sigma_\gamma[u_{s(\gamma)}(t) - u_{d(\gamma)}(t)], & \text{if } \gamma \text{ is a resistor;} \\ I_\gamma(t) & = & c_\gamma \frac{d}{dt}[u_{s(\gamma)}(t) - u_{d(\gamma)}(t)], & \text{if } \gamma \text{ is a capacitor;} \\ I_\gamma(t) & = & \sigma_\gamma[u_{s(\gamma)}(t) - u_{d(\gamma)}(t) - v_\gamma], & \text{if } \gamma \text{ is an outer voltage followed by a resistor.} \end{array}$$

With our rule for assigning parameters to the arcs, we can write these relations in the unified form

$$I_\gamma(t) = \sigma_\gamma[u_{s(\gamma)}(t) - u_{d(\gamma)}(t) - v_\gamma] - c_\gamma \frac{d}{dt}[u_{s(\gamma)}(t) - u_{d(\gamma)}(t)]. \tag{4.7.1}$$

The second law of Kirchoff says that for every node i, the sum of currents in the arcs entering the node should be equal to the sum of currents in the arcs leaving the node. To represent this law conveniently, let us introduce the *incidence matrix* $\mathcal{P}$ of our circuit as follows:

> (Incidence matrix): The columns of the matrix $\mathcal{P}$ are indiced by nodes $1, ..., N$, and the rows are indiced by the arcs $\gamma \in \Gamma$. The row P_γ corresponding to an arc γ is comprised of zeros, except two entries: the one in the column $s(\gamma)$ – this entry is $+1$, and the one in the column $d(\gamma)$, which is -1.

With this formalism, the second law of Kirchoff is

$$P^T I(t) = 0, \tag{4.7.2}$$

$I(t)$ being the vector with the entries $I_\gamma(t)$, $\gamma \in \Gamma$. Note that with the same formalism, (4.7.1) can be written down as

$$I(t) = \Xi P u(t) + \Theta P \frac{d}{dt} u(t) - \Xi v, \tag{4.7.3}$$

where

- $u(t)$ is the N-dimensional vector comprised of the potentials of the nodes $u_i(t)$, $i = 1, ..., N$;
- Ξ is the diagonal $M \times M$ matrix (M is the number of arcs) with diagonal entries σ_γ, $\gamma \in \Gamma$;
- v is the M-dimensional vector comprised of outer voltages v_γ, $\gamma \in \Gamma$;
- Θ is the diagonal $M \times M$ matrix with the diagonal entries c_γ, $\gamma \in \Gamma$.

Multiplying the latter equality by P^T and taking into account (4.7.2), we get

$$\left[P^T \Theta P\right] \frac{d}{dt} u(t) = -P^T \Xi P u(t) + P^T \Xi v. \tag{4.7.4}$$

Now, the potentials are the quantities defined "up to adding an additive constant": what makes physical sense, are not the potentials themselves, but their differences. In order to avoid the resulting non-uniqueness of our solutions, we may enforce one of the potentials, say, $u_N(t)$, to be identically zero (the node N is "the ground"). Let

$$\begin{array}{rcl} C & = & ([P^T \Theta P]_{ij})_{i,j \le N-1}, \\ S & = & ([P^T \Xi P]_{ij})_{i,j \le N-1}, \\ R & = & ([P^T \Xi])_{i\gamma})_{i \le N-1, \gamma \in \Gamma} \end{array} \tag{4.7.5}$$

be the corresponding sub-matrices of the matrices participating in (4.7.4). Denoting by $w(t)$ the $(N-1)$-dimensional vector comprised of the first $N-1$ entries of the vector of potentials $u(t)$ (the latter vector is normalized by $u_N(t) \equiv 0$), we can rewrite (4.7.5) as

$$C \frac{d}{dt} w(t) = -S w(t) + R v. \tag{4.7.6}$$

Note that due to their origin the matrices Ξ and Θ are diagonal with nonnegative diagonal entries, i.e., they are symmetric positive semidefinite; consequently, the matrices C, S also are symmetric positive semidefinite. In the sequel, we make the following

Assumption (A): *The matrices C and S are positive definite.*

In fact (A) is a quite reasonable restriction on the topology of the circuit: when deleting all capacitors, the resulting net comprised of resistors should be connected, and similarly for the net of capacitors obtained after the resistors are deleted. With this assumption, (4.7.6) is equivalent to a system of linear ordinary differential equations with constant coefficients and a constant right hand side:

$$\frac{d}{dt} w(t) = -C^{-1} S w(t) + C^{-1} R v. \tag{4.7.7}$$

Now, the matrix of the system is similar to the negative definite matrix:

$$-C^{-1} S = C^{-1/2} [-C^{-1/2} S C^{-1/2}] C^{1/2}.$$

Consequently, the eigenvalues $(-\lambda_i)$, $i = 1, ..., N$, of the matrix $C^{-1}S$ of the system (4.7.7) are negative, there exists a system $e_1, ..., e_N$ of linearly independent eigenvectors associated with these eigenvectors and the solution to (4.7.7) is of the form

$$w(t) = w_*(v) + \sum_{i=1}^{N} \kappa_i \exp\{-\lambda_i t\} e_i,$$

where

$$w_*(v) = S^{-1}Rv \tag{4.7.8}$$

is the vector comprised of the static steady-state potentials associated with the outer voltages v and κ_i are certain constants coming from the initial state $w(0)$ of (4.7.7). From (4.7.3) we get a representation of the same structure also for the currents:

$$I(t) = I_*(v) + \sum_{i=1}^{N} \chi_i \exp\{-\lambda_i\} f_i \tag{4.7.9}$$

with certain M-dimensional vectors f_i.

We see that during the transition period, the potentials and the currents approach the steady state exponentially fast, the rate of convergence being governed by the quantities λ_i. The "most unfavourable", with respect to the initial conditions, rate of convergence to the steady state is given by the *smallest* of the λ_i's. Thus, the quantity

$$\hat{\lambda} = \min_i \lambda_i(C^{-1/2}SC^{-1/2}) \tag{4.7.10}$$

can be treated as a (perhaps rough) measure of speed of the circuit. The larger is the quantity, the shorter is a transition period; for all not that big initial conditions, during a moderate constant times the quantity $1/\hat{\lambda}$, the potential and the currents in the circuit will be "very close" to their steady-state values. It was proposed by S. Boyd to treat $1/\hat{\lambda}$ as the characteristic "time constant" of the underlying circuit and to formulate restrictions of the type "the duration of a transition period in a circuit should be at most this and this" as "the time constant of the circuit should be at most that and that" [13]. Now, it is easy to understand what is, mathematically, $\hat{\lambda}$ – it is nothing but the smallest eigenvalue $\lambda_{\min}(S : C)$ of the pencil $[C, S]$ (cf. Section 4.4.1). Consequently, the requirement "speed of the circuit to be designed should be not less than..." in Boyd's methodology becomes the restriction $1/\lambda_{\max}(S : C) \leq T$ with a prescribed T, or, which is the same, it becomes the matrix inequality

$$S - \kappa C \succeq 0, \qquad [\kappa = T^{-1}]. \tag{4.7.11}$$

As we shall see in a while, S and C in typical chip design problems are *affine* functions of the design variables, so that the *care of the speed of the chip to be designed can be expressed by an LMI.*

Dissipated heat

When speaking about the dissipated heat, we may be interested in

(1) the heat power dissipated in the steady state corresponding to given outer voltages;

(2) the heat dissipated during a transition.

As we shall see in a while, imposing restrictions on the "steady-state" dissipated heat power leads to an "intractable" computational problems, while restrictions on the heat dissipated in a transition period in some meaningful cases (although not always) lead to semidefinite programs.

[13] For many years, engineers were (and still are) using a more "precise" measure of speed – the *Elmore* constant. A disadvantage of the Elmore constant is that it can be efficiently computed only for a restricted family of circuits. In contrast to this, Boyd's "time constant" is "computationally tractable".

Bad news on steady-state dissipated heat power. Physics says that the dissipated heat power in the steady state corresponding to outer voltages v is

$$H = \sum_{\gamma\in\Gamma} (I_*(v))_\gamma[(u_*(v))_{s(\gamma)} - (u_*(v))_{d(\gamma)} - v_\gamma] = [I_*(v)]^T(Pu_*(v) - v),$$

where $I_*(v)$ and $u_*(v)$ are the steady-state currents and potentials associated with v[14]. Our formula expresses the rule known to everybody from school: "the heat power dissipated by a resistor is the product of the current and the voltage applied to the resistor".

In fact H is given by the following

> **Variational Principle:** *Given a circuit satisfying assumption* (A) *and a vector of outer voltages v, consider the quadratic form*
>
> $$G(u) = (Pu - v)^T \Xi (Pu - v)$$
>
> *of N-dimensional vector u. The heat power dissipated by the circuit at the static steady state associated with v is the minimum value of this quadratic form over $u \in \mathbf{R}^N$.*

Indeed, G depends on the differences of coordinates of u only, so that its minimum over all u is the same as its minimum over u of the form $u = \begin{pmatrix} w \\ 0 \end{pmatrix}$. Regarded as a form of $(N-1)$-dimensional vector w rather than $u = \begin{pmatrix} w \\ 0 \end{pmatrix}$, the form becomes

$$w^T Sw - 2w^T Rv + v^T \Xi v;$$

the minimizer of the latter expression in w is given by $w_* = S^{-1}Rv$, i.e., w_* is exactly the vector of steady-state potentials at the nodes (cf. (4.7.8)). Thus, the steady-state potentials $u_*(v)$ form a minimizer of $G(\cdot)$. The value of the form G at this minimizer is $\frac{1}{2}\left[\Xi(Pu_*(v) - v)\right]^T (Pu_*(v) - v)$, and the vector $\Xi(Pu_*(v) - v)$ is exactly the vector of steady-state currents, see (4.7.1). Thus, the optimal value of G is $I_*^T(v)(Pu_*(v) - v)$, i.e., is nothing but H.

Variational Principle says that an upper bound $H \le h$ on the steady-state dissipated heat power is

$$\mathcal{H}(S, v) \equiv \min_u (Pu - v)^T S(Pu - v) \le h.$$

The left hand side in this bound is a *concave* function of S (as a minimum of linear functions of S). Therefore an upper bound on the dissipated steady-state heat power, the outer voltages being fixed, defines a *non-convex* set of "feasible" matrices S. If S, as it normally is the case, depends affinely on "free design parameters", the non-convex set of feasible S's defines a non-convex feasible set in the space of design parameters, so that incorporating an upper bound on the steady-state dissipated heat power would yield a non-convex (and hence – hardly tractable computationally) optimization problem.

Note that if we were designing an oven rather than a chip – i.e., were interested to get *at least* a prescribed heat power dissipation instead of *at most* a given one – the restriction would be captured by the realm of convex (specifically, semidefinite) programming (cf. Simple Lemma).

[14] We assume that the potential of the "ground" – the node N – is 0; with this convention, the notion of "steady-state potentials" becomes well-defined.

Good news on heat dissipated in transition period. The heat dissipated during a transition from the steady state associated with outer voltages v_- (the "old" steady state) and the steady state (the "new" one) associated with outer voltages v_+ is, in general, a senseless notion. Indeed, the transition period, rigorously speaking, is infinite. If the new steady state is "active", i.e., not all of the corresponding steady-state currents are zero, then the heat dissipation power during the transition will approach a positive quantity (the steady-state dissipated heat power for the new steady state), and the entire power energy dissipated during the (infinite!) transition period will be $+\infty$. There is, however, a case where this difficulty does not occur and we indeed may speak about the heat energy dissipated during the transition – this is the case when the "new" steady-state currents are zero. In this case, the dissipated heat power in course of transition goes to 0 exponentially fast, the decay rate being (bounded by) the Boyd time constant, and it indeed makes sense to speak about the heat dissipated during the transition. Now, there is a particular (although quite important) class of "simple" RC-circuits satisfying the assumption "the currents at a static steady-state are zero" – circuits of the type shown on the picture:

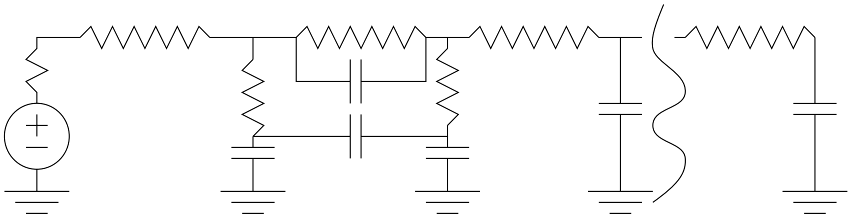


Figure 4.3. A "simple" RC-circuit

In such a circuit, there exists a single source of outer voltage, the resistors form a connected net which starts at one of the poles of the source and does not reach the other pole of the source ("the ground"). And each capacitor either links a node incident to a resistor and the ground ("capacitor of type I"), or links two nodes incident to resistors ("capacitor of type II"). Here the steady-state currents clearly are zero. Moreover, the steady-state potentials at all nodes incident to resistors are equal to the magnitude of the outer voltage; as about the voltage at a capacitor, it either equals to the magnitude of the outer voltage (for capacitors of type I), or is zero (for capacitors of type II).

For a "simple" circuit, the heat energy dissipated during transition can be found as follows. Assume that the outer voltage switches from its old value v_- of a magnitude μ_- to its new value v_+ of magnitude μ_+ [15]. Let us compute the heat dissipated during the transition period (assumed to start at $t = 0$). Denoting $u(t)$ the potentials, $I(t)$ the currents, $H(t)$ the power of dissipated heat at time instant $t \geq 0$, we get

$$H(t) = I^T(t)(Pu(t) - v_+) = (Pu(t) - v_+)^T \Xi (Pu(t) - v_+)$$

(we have used (4.7.1)); consequently, the heat $\mathbf{H}$ dissipated during the transition is

$$\mathbf{H} = \int_0^\infty (Pu(t) - v_+)^T \Xi (Pu(t) - v_+) dt.$$

[15] In spite of the fact that now we are speaking about a single-source circuit, it would be bad to identify the magnitude of the outer voltage and the voltage itself. According to our formalism, an outer voltage is a *vector* with the coordinates indiced by arcs of the circuit; a coordinate of this vector is the "physical magnitude" of the outer source "inside" an arc. Thus, μ_- is a number, and v_- is a vector with all but one zero coordinates; the only nonzero coordinate is equal to μ_- and corresponds to the arc containing the outer source.

Now, for a "simple" circuit we have $Pu_*(v_+) = v_+$, where, as before, $u_*(v_+)$ is the vector of steady-state potentials associated with the outer voltage v_+. Denoting by $\delta(t)$ the vector comprised of the first $N-1$ coordinates of $u(t) - u_*(v_+)$ (recall that all our potentials are normalized by the requirement that the potential of "the ground" – of the last (N-th) node – is identically equal zero) and recalling the definition of S, we can rewrite the expression for $\mathbf{H}$ as

$$\mathbf{H} = \int_0^\infty \delta^T(t) S \delta(t) dt.$$

It is clear from the origin of $\delta(\cdot)$ that this function solves the homogeneous version of (4.7.6) and that the initial condition for $\delta(\cdot)$ is $\delta(0) = (\mu_- - \mu_+)e$, e being $(N-1)$-dimensional vector of ones:

$$C\frac{d}{dt}\delta(t) = -S\delta(t), \delta(0) = (\mu_- - \mu_+)e.$$

Multiplying both sides in our differential equation by $\delta^T(t)$ and integrating in t, we get

$$\begin{array}{rcl} \mathbf{H} & = & \int_0^\infty \delta^T(t) S \delta(t) dt \\ & = & -\int_0^\infty \delta^T(t) C \delta'(t) dt \\ & = & -\frac{1}{2}\int_0^\infty \frac{d}{dt}\left[\delta^T(t) C \delta(t)\right] dt \\ & = & \frac{1}{2}\delta^T(0) C \delta(0) \\ & = & \frac{(\mu_- - \mu_+)^2}{2} e^T C e. \end{array}$$

Thus, in the case of a "simple" circuit the heat dissipated during the transition is

$$\mathbf{H} = \frac{(\mu_+ - \mu_-)^2}{2} e^T C e \quad [e = (1, ..., 1)^T \in \mathbf{R}^{N-1}]. \tag{4.7.12}$$

– it is a *linear* function of C.

4.7.2 Wire sizing

Modern sub-micron chips can be modeled as RC circuits; in these circuits, the resistors, physically, are the inter-connecting wires (and the transistors), and the capacitors model the capacitances between pairs of wires or a wire and the substrate. After the topology of a chip and the placement of its elements on a substrate are designed, engineers start to define the widths of the wires, and this is the stage where the outlined models could be used. In order to pose the wire sizing problem as an optimization program, one may think of a wire as being partitioned into rectangular segments of a prescribed length and treat the widths of these segments as design variables. A pair of two neighbouring wires (or a wire and the substrate) can be modeled by a RC-structure, as shown on Fig. 4.4:

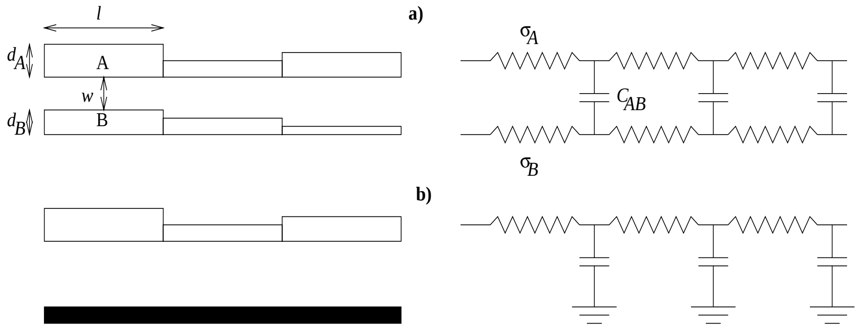


Figure 4.4. Wires (left) and the equivalent RC-structure (right)
a) – a pair of two neighbouring wires; b) – a wire and the substrate

A nice feature of this model is that both the conductances of the resulting resistors and the capacitances of the resulting capacitors turn out to be *linear* functions of our design parameters – the widths of the segments, or, which is the same, of the areas of the segments (the lengths of the segments are fixed!). E.g., for the RC-structure depicted on Fig. 4.4.a) one has

$$\begin{array}{rcl} c_{AB} & = & \kappa_{A,B}(s_A + s_B), \\ \sigma_A & = & \kappa_A s_A, \end{array}$$

s_A, s_B being the areas of the corresponding segments. The coefficients $\kappa_{A,B}, \kappa_A$ depend on a lot of parameters (e.g., on distances between the wires), but all these parameters at the phase of design we are speaking about are already set. Thus, in the wire sizing problem the matrices Ξ and Θ, and therefore the matrices S, C, R as well, are *affine functions* of the design variables – the areas of the segments:

$$C = \mathcal{C}(x), S = \mathcal{S}(x), R = \mathcal{R}(x)$$

(x stands for the design vector comprised of the areas of segments). As a result, we may pose numerous sizing-related problems as semidefinite programs, e.g.:

- We can minimize the total area occupied by the wires under the restriction that the speed (i.e., the time constant) of the resulting circuit should be at least something given; the arising semidefinite program is

$$\begin{array}{rrcl} & \sum_i x_i & \to & \min \\ \text{s.t.} & & & \\ & \mathcal{S}(x) - \kappa\mathcal{C}(x) & \succeq & 0 \\ & x & \geq & 0 \end{array}$$

  ($\kappa > 0$ is fixed);

- Whenever a circuit is "simple" (which indeed is the case for many chips), we can minimize the heat dissipated per a transition under the restriction that the speed is at least something given; the arising semidefinite program is

$$\begin{array}{rrcl} & e^T[\mathcal{C}(x)]e & \to & \min \\ \text{s.t.} & & & \\ & \mathcal{S}(x) - \kappa\mathcal{C}(x) & \succeq & 0 \\ & x & \geq & 0 \end{array}$$

- We can add to the above programs upper and lower bounds on the the areas of segments, as well as other linear constraints on the design variables, etc.

4.8 Applications, VI: Structural design

Structural design is an engineering area which has to do with mechanical constructions like *trusses* and *plates*. We already know what a truss is – a construction comprised of linked to each other thin elastic bars. A plate is a construction comprised of a material occupying a given domain, the mechanical properties of the material varying continuously from point to point. In engineering, design of plates is called *shape* design, this is why in the sequel we call objects of our interest *trusses* and *shapes* instead of trusses and plates.

It turns out that numerous problems of the type "given a type of material to be used, a *resource* (an upper bound on the amount of material to be used) and a set of *loading scenarios* – external loads of interest, find an optimal truss/shape – the one capable to withstand best of all the loads in question" can be casted as semidefinite programs, and semidefinite programming offers *the* natural way to model the problems and to process them analytically and numerically. The purpose of this section is to develop a unified semidefinite-programming-based approach to the outlined problems.

4.8.1 Building a model

Mechanical constructions we are about to consider (the so called *constructions with linear elasticity*) can be described by the following general story.

I. A construction C can be characterized by

I.1) A linear space $\mathbf{V} = \mathbf{R}^m$ of *virtual displacements* of C;

I.2) A positive semidefinite quadratic form

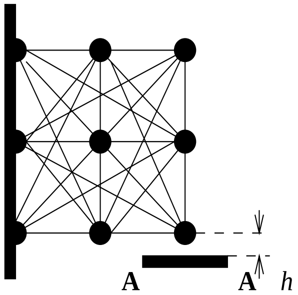
$$E_C(v) = \frac{1}{2} v^T A_C v$$

on the space of displacements; the value of this form at a displacement v is the potential energy stored by the construction as a result of the displacement. The (positive semidefinite symmetric) $m \times m$ matrix A_C of this form is called the *stiffness matrix* of C;

Example: truss. A truss (Sections 1.3.5, 3.4.3) fits I.1) – I.2).

I.3) A closed convex subset $\mathcal{V} \subset \mathbf{R}^m$ of *kinematically admissible displacements.*

Example (continued). In our previous truss-related considerations, there was no specific set of kinematically admissible displacements – we thought that in principle every virtual displacement $v \in \mathbf{R}^m$ may become an actual displacement, provided that an external load is chosen accordingly. However, sometimes the tentative displacements of the nodes are restricted by external *obstacles*, like the one you see on the picture:



What we see is a 9-node planar ground structure with 33 tentative bars and a rigid obstacle AA. This obstacle does not allow the South-Eastern node to move down more than by h and thus induces a linear inequality constraint on the vector of virtual displacements of the nodes.

II. An external load applied to the construction C can be represented by a vector $f \in \mathbf{R}^m$; the static equilibrium of C loaded by f is given by the following

Variational Principle: *A construction C is capable to carry an external load f if and only if the quadratic form*

$$E_C^f(v) = \frac{1}{2}v^T A_C v - f^T v \tag{4.8.1}$$

of displacements v attains its minimum on the set $\mathcal{V}$ of kinematically admissible displacements, and displacement yielding (static) equilibrium is a minimizer of $E_C^f(\cdot)$ on $\mathcal{V}$.

The minus minimum value of E_C^f on $\mathcal{V}$ is called the *compliance* of the construction C with respect to the load f:

$$\mathrm{Compl}_f(C) = \sup_{v\in\mathcal{V}}\left[f^T v - \frac{1}{2}v^T A_C v\right]$$

Example (continued). A reader not acquainted with the KKT optimality conditions may skip this paragraph.

We have seen in Section 3.4.3 that the Variational Principle does work for a truss. At that moment we, however, dealt with the particular "obstacle-free" case: $\mathcal{V} = \mathbf{V}$. What happens that there are obstacles? Assume that the obstacles are absolutely rigid and frictionless. When in course of truss deformation a moving node meets an obstacle, a contact force occurs and comes into play – it becomes a part of the external load. As a result, the equilibrium displacement is given by the equations

$$Av = f + \sum_l f_l, \tag{*}$$

where Av, as we remember, is the vector of reaction forces caused by the deformation of the truss and f_l's represent the contact forces coming from obstacles. "The nature" is free to choose these forces, with the only restriction that a contact force should be normal to the boundary of the corresponding obstacle (there is no friction!) and should "look" towards the truss. With these remarks in mind, one can easily recognize in (*) the usual KKT conditions for constrained minimum of E_C^f, the constraints being given by the obstacles. Thus, an equilibrium displacement is a KKT point of the problem of minimizing E_C^f over $\mathcal{V}$. Since the problem is convex, its KKT points are the same as the minimizers of E_C^f over $\mathcal{V}$.

The part of the story we have told so far relates to a particular system and does not address the question of which elements of the situation may be affected by the design: may we play with the space of virtual displacements? or with $\mathcal{V}$? or with the stiffness matrix? Different types of structural design problems deal with different answers to the outlined questions; we, however, will focus on the case when the only element we may play with is the stiffness matrix. Specifically, we assume that

III. The stiffness matrix A_C depends on mechanical characteristics $t_1, ..., t_n$ of "elements" $E_1, ..., E_n$ comprising the construction, and these characteristics t_i are *positive semidefinite symmetric $d \times d$ matrices*, d being given by the type of the construction. Specifically,

$$A_C = \sum_{i=1}^{n}\sum_{s=1}^{S} b_{is} t_i b_{is}^T, \tag{4.8.2}$$

where b_{is} are given $m \times d$ matrices.

At a first glance, III looks very strange: where positive semidefinite matrices t_i come from? Well, this is what Mechanics says on both types of constructions we actually are interested in – trusses and shapes.

Indeed, we know from Section 1.3.5 that the stiffness matrix of a truss is

$$A(t) = \sum_{i=1}^{n} t_i b_i b_i^T,$$

where $t_i \geq 0$ are bar volumes and b_i are certain vectors given by the geometry of the nodal set; and nonnegative reals t_i may be viewed as positive semidefinite 1×1 matrices...

Now, what about shapes? To see that III holds in this case as well, it requires an additional excursion to Mechanics we are about to start (and which can be omitted by a non-interested reader).

As we remember, a shape is comprised of material occupying a given 2D or 3D domain Ω, the mechanical properties of the material varying from point to point. Such a construction is *infinite-dimensional*: its virtual displacements are vector fields on Ω and altogether form certain linear space $\mathbf{V}$ of vector fields on Ω ($\mathbf{V}$ should not necessarily be comprised of all vector fields; e.g., some parts of the boundary of Ω may be fixed by boundary conditions, so that the displacement fields must vanish at these parts of the boundary).

The elasticity properties of the material at a point $P \in \Omega$ are represented by the *rigidity tensor* $E(P)$ which, mathematically, is a symmetric positive semidefinite $d \times d$ matrix, where $d = 3$ for planar and $d = 6$ for spatial shapes. Mechanics says that the density, at a point P, of the potential energy stored by a shape, the displacement of the shape being a vector field $v(\cdot)$, is

$$\frac{1}{2} s_P^T[v] E(P) s_P[v], \tag{4.8.3}$$

so that the potential energy stored in a deformated shape is

$$\frac{1}{2} \int_{\Omega} s_P^T[v] E(P) s_P[v] dP.$$

Here for a 2D shape

$$s_P[v] = \begin{pmatrix} \frac{\partial v_x(P)}{\partial x} \\ \frac{\partial v_y(P)}{\partial y} \\ \frac{1}{\sqrt{2}} \left[\frac{\partial v_x(P)}{\partial y} + \frac{\partial v_y(P)}{\partial x} \right] \end{pmatrix},$$

v_x and v_y being the $x-$ and the $y-$components of the 2D vector field $v(\cdot)$. Note that $s_P[v]$ can be obtained as follows: we first build the Jacobian of the vector field v at P – the matrix

$$J(P) = \begin{pmatrix} \frac{\partial v_x(P)}{\partial x} & \frac{\partial v_x(P)}{\partial y} \\ \frac{\partial v_y(P)}{\partial x} & \frac{\partial v_y(P)}{\partial y} \end{pmatrix},$$

and then symmeterize the matrix – build the symmetric 2×2 matrix

$$J^s(P) = \frac{1}{2}[J(P) + J^T(P)].$$

$s_P[v]$ is nothing but the vector of the coordinates of $J^s(P) \in \mathbf{S}^2$ in the natural orthonormal basis $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 2^{-1/2} \\ 2^{-1/2} & 0 \end{pmatrix}$ of the 3-dimensional space $\mathbf{S}^2$.

For a 3D shape, $s_P[v]$ is given by a completely similar construction: we build the 3×3 Jacobian of the 3D vector fields $v(\cdot)$ at the point P, symmeterize it and then pass from the 3×3 symmetric matrix we have got to the vector of coordinates of this matrix in the natural basis of the 6-dimensional space $\mathbf{S}^3$; we skip the corresponding explicit formulas.

An external load acting at a shape can be represented by a linear form $f[v]$ on the space of displacements; this form measures the work carried out by the load at a displacement. In typical cases, this functional looks like $\int_{\partial\Omega} f^T(P) v(P) dS(P)$, $f(P)$ being the field of external forces acting at the boundary. And Mechanics says that the equilibrium displacement field in the loaded shape minimizes the energy functional

$$\frac{1}{2}\int_{\Omega} s_P^T[v]E(P)s_P[v]dP - f[v]$$

over the set of kinematically admissible vector fields $v(\cdot)$. The minus minimum value of this functional is called the compliance of the shape with respect to the load in question.

As we see, the "true" model of a shape is infinite-dimensional. In order to get a "computationally tractable" model, the *Finite Element* approximation is used, namely,

1. The domain Ω is partitioned into finitely many non-overlapping cells $\Omega_1, ..., \Omega_n$, and the properties of the material are assumed to be constant within the cells:

$$E(P) = E_i \text{ for } P \in \Omega_i;$$

2. The infinite-dimensional space $\mathbf{V}$ of vector fields on Ω is "approximated" by its finite-dimensional subspace $\mathbf{V}^m$ spanned by m basic continuously differentiable displacement fields $w_1(\cdot), ..., w_m(\cdot)$; with this approximation, the set of kinematically admissible displacement fields shrinks to a set in $\mathbf{V}^m$.

With this approximation, the finite element model of a shape becomes as follows:

- A virtual displacement v becomes a vector from $\mathbf{R}^m$ (the actual displacement field corresponding to a vector $v = (v_1, ..., v_m)$ is, of course, $\sum_{i=1}^m v_i w_i(\cdot)$);
- The potential energy stored by the shape, v being the corresponding displacement, is

$$\frac{1}{2}\sum_{i=1}^n \int_{\Omega_i} v^T[s(P)E_i s^T(P)]v dP, \qquad s^T(P) = [s_P[w_1]; s_P[w_2]; ...; s_P[w_m]] \in \mathbf{M}^{dm};$$

- The linear functional $f[\cdot]$ representing a load becomes a usual linear form $f^T v$ on $\mathbf{R}^m$ (so that we can treat the vector f of the coefficients of this form as the load itself).
- The equilibrium displacement of the shape under a load f is the minimizer of the quadratic form

$$\frac{1}{2}v^T\left[\sum_{i=1}^n \int_{\Omega_i} s(P)E_i s^T(P)dP\right] v - f^T v$$

on a given set $\mathcal{V} \subset \mathbf{R}^m$ of kinematically admissible displacements, and the compliance is the minus minimum value of this form on $\mathcal{V}$.

It remains to note that, as is stated in Exercise 1.17, there exist a positive integer S and "cubature formulas"

$$\int_{\Omega_i} s_P^T[w_p]Es_P[w_q]dP = \sum_{s=1}^S \alpha_{is} s_{P_{is}}^T[w_p]Es_{P_{is}}[w_q] \quad \forall E \in \mathbf{S}^d \quad \forall p,q = 1, ..., m$$

with nonnegative weights α_{is}. Denoting by ω_i the measures of the cells Ω_i and setting

$$t_i = \omega_i E_i, \quad b_{is} = \alpha_{is}^{1/2}\omega_i^{-1/2}s(P_{is}),$$

we get

$$\int_{\Omega_i} s(P)E_i s^T(P)dP = \sum_{s=1}^S b_{is}t_i b_{is}.$$

Thus, we have represented a shape by a collection $t_1, ..., t_n$ of positive semidefinite $d \times d$ matrices, and the potential energy of a deformated shape now becomes

$$\frac{1}{2}v^T\left[\sum_{i=1}^n \int_{\Omega_i} s(P)E_i s^T(P)\right] v = \frac{1}{2}v^T\left[\sum_{i=1}^n\sum_{s=1}^S b_{is}t_i b_{is}^T\right] v,$$

v being the displacement. We see that a shape, after finite element discretization, fits III.

The concluding chapter of our story says how we measure the "material" used to build the construction:

IV. As far as the material "consumed" by a construction C is concerned, all "expenses" of this type are completely characterized by the vector $(\mathrm{Tr}(t_1), ..., \mathrm{Tr}(t_n))$ of the traces of the matrices $t_1, ..., t_n$ mentioned in III.

> For a truss, the indicated traces are exactly the same as t_i's themselves and are the volumes of the bars constituting the truss, so that IV is quite reasonable. For a shape, $\mathrm{Tr}(E(P))$ is a natural measure of the "material density" of the shape at a point $P \in \Omega$, so that IV again is quite reasonable.

Now we can formulate the general Structural Design problem we are interested in:

Problem 4.8.1 [Static Structural Design] *Given*

1. *A "ground structure", i.e.,*
   - *the space $\mathbf{R}^m$ of virtual displacements along with its closed convex subset $\mathcal{V}$ of kinematically admissible displacements,*
   - *a collection $\{b_{is}\}_{i=1,...,n,s=1,...,S}$ of $m \times d$ matrices,*
2. *A set $T = \{(t_1, ..., t_n) \mid t_i \in \mathbf{S}^d_+ \quad \forall i\}$ of "admissible designs",*

   and
3. *A set $\mathcal{F} \subset \mathbf{R}^m$ of "loading scenarios",*

find a "stiffest, with respect to $\mathcal{F}$, admissible construction", i.e., find a collection $t \in T$ which minimizes the worst, over $f \in \mathcal{F}$, compliance of the construction with respect to f:

$$\mathrm{Compl}_{\mathcal{F}}(t) \equiv \sup_{f\in\mathcal{F}} \sup_{v\in\mathcal{V}} \left[f^T v - \frac{1}{2} v^T \left[\sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \right] v \right] \to \min \mid t \in T.$$

4.8.2 The standard case

The Static Structural Design (SSD) problem 4.8.1 in its general form is a little bit "diffuse" – we did not say what are the geometries of the set $\mathcal{V}$ of kinematically admissible displacements, of the set T of admissible designs and of the set $\mathcal{F}$ of loading scenarios. For all applications known to us these geometries can be specialized as follows.

S.1. The set $\mathcal{V}$ of kinematically admissible displacements is a polyhedral set:

$$\mathcal{V} = \{v \in \mathbf{R}^m \mid Rv \le r\} \qquad [R \in \mathbf{M}^{qm}], \tag{4.8.4}$$

and the system of linear inequalities $Rv \le r$ satisfies the Slater condition: there exists $\bar{v}$ such that $R\bar{v} < r$.

S.2. The set T of admissible designs is given by simple linear constraints on the *traces* of the positive semidefinite rigidity matrices t_i – upper and lower bounds on $\mathrm{Tr}(t_i)$ and an upper bound on the "total material resource" $\sum_{i=1}^n \mathrm{Tr}(t_i)$:

$$\begin{array}{c} T = \{t = (t_1, ..., t_n) \mid t_i \in \mathbf{S}^d_+, \underline{\rho}_i \le \mathrm{Tr}(t_i) \le \overline{\rho}_i, \sum_{i=1}^n \mathrm{Tr}(t_i) \le w\} \\ [0 \le \underline{\rho}_i < \overline{\rho}_i < \infty, \sum_{i=1}^n \underline{\rho}_i < w] \end{array} \tag{4.8.5}$$

S.3. The set $\mathcal{F}$ of loading scenarios is either a finite set:

$$\mathcal{F} = \{f_1, ..., f_k\}, \tag{4.8.6}$$

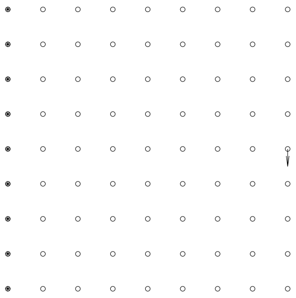
(*multi-load structural design*), or an ellipsoid:

$$\mathcal{F} = \{f = Qu \mid u^T u \leq 1\} \quad [Q \in \mathbf{M}^{mk}] \tag{4.8.7}$$

(*robust structural design*).

The interpretation of the multi-load setting is quite clear: the construction is supposed to work under different loading conditions, and we intend to control its stiffness for a number of typical loading scenarios. To motivate the robust setting, consider the following example.

Assume we are designing a planar truss – a cantilever; the 9 × 9 nodal structure and the only load of interest f^* are as shown on the picture:



9 × 9 ground structure and the load of interest

The optimal single-load design yields a nice truss as follows:

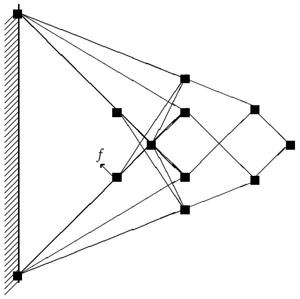


Figure 4.5. Optimal cantilever (single-load design); the compliance is 1.000

The compliance of the optimal truss with respect to the load of interest is 1.000.

Now, what happens if, instead of the load of interest f^*, the truss is affected by a "small occasional load" f shown on the picture, the magnitude ($\equiv$ the Euclidean length) of f being just 0.5% of the magnitude of f^*? The results are disastrous: the compliance is increased by factor 8.4 (!) In fact, our optimal cantilever is highly instable: it can collapse when a bird will try to build a nest in a "badly placed" node of the construction...

In order to ensure stability of our design, we should control the compliance not only with respect to a restricted set of "loads of interest", but also with respect to all small enough "occasional loads" somehow distributed along the nodes. The simplest way to do it is to add to the original finite set of loads of interest the ball comprised of all "occasional" loads of magnitude not exceeding some level. There are, however, two difficulties in this approach:

– from the viewpoint of mathematics, it is not that easy to deal with the set of loading scenarios of the form "the union of a ball and a finite set"; it would be easier to handle either a finite set, or an ellipsoid.

– from the engineering viewpoint, the difficulty is to decide where the occasional loads should be applied. If we allow them to be distributed along all 9 × 9 nodes of the original ground structure, the resulting design will incorporate all these nodes (otherwise its compliance with respect to some occasional loads will be infinite), which makes no sense. What we indeed are interested in, are occasional loads distributed along the nodes which will be used by the resulting construction, but how could we know these nodes in advance?

As about the first difficulty, a natural way to overcome it is to take, as $\mathcal{F}$, the "ellipsoidal envelope" of the original – finite – set of the loads of interest and a small ball, i.e., to choose as $\mathcal{F}$ the centered at the origin ellipsoid of the smallest volume containing the original loading scenarios and the ball.

In order to overcome, to some extent, the second difficulty, we could use the two-stage scheme: at the first stage, we take into consideration the loads of interest only and solve the corresponding single/multi-load problem. At the second stage, we resolve the problem, treating the set of nodes actually used by the structure we have obtained as our new nodal set, and taking as $\mathcal{F}$ the ellipsoidal envelope of the loads of interest and the ball comprised of all "small" occasional nodes distributed along the nodes of our new nodal set.

Let us look what this approach yields in our cantilever example. The cantilever depicted on Fig. 4.5 uses 12 nodes from the original 81-node grid; two of these 12 nodes are fixed. Taking the resulting 12 nodes as our new nodal set and allowing all pair connections of these nodes, we get a new – reduced – ground structure with 20 degrees of freedom. Now let us define $\mathcal{F}$ as the ellipsoidal envelope of f^* and the ball comprised of all loads of the Euclidean norm not exceeding 10% of the norm of f^*. The 20-dimensional ellipsoid $\mathcal{F}$ is very simple: one of its principal half-axes is f^*, and the remaining 19 half-axes are of the length $0.1 \parallel f^* \parallel_2$ each, the directions of these half-axes forming a basis in the orthogonal complement to f^* in the 20-dimensional space of virtual displacements of our 12 nodes. Minimizing the worst-case, with respect to the ellipsoid of loads $\mathcal{F}$, compliance under the original design constraints, we come to a new cantilever depicted on Fig. 4.6:

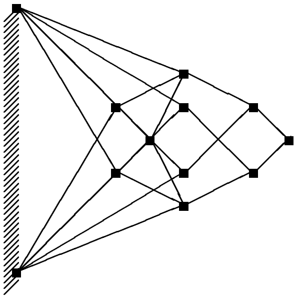


Figure 4.6. "Robust" cantilever

The maximum compliance, over the ellipsoid of loads $\mathcal{F}$, of the "robust" cantilever is 1.03, and its compliance with respect to the load of interest f^* is 1.0024 – i.e., it is only by 0.24% larger than the optimal compliance given by the single-load design. We see that when passing from the "nominal" single-load design to the robust one we loose basically nothing in optimality, while getting dramatic improvement in the stability (for the "nominal" design, the compliance with respect to a "badly chosen" occasional load of the magnitude $0.1 \parallel f^* \parallel_2$ may be as large as 32000!)

We conclude that ellipsoidal sets of loads indeed make sense.

We shall refer to the Static Structural Design problem with the data satisfying **S.1** – **S.3** as to the *Standard* SSD problem; speaking on this problem, we always assume that it satisfies the following assumption:

S.4. $\sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \succ 0$ whenever $t_i \succ 0$, $i = 1, ..., n$.
This assumption forbids "rigid body" motions of the ground structure: if all rigidities are positive definite, then the potential energy stored by the construction under *any* nontrivial displacement is strictly positive.

4.8.3 Semidefinite reformulation of the Standard SSD problem

In order to get a semidefinite reformulation of the Standard Static Structural Design problem, we start with building semidefinite representation for the compliance. Thus, our goal is to get an SDR for the set

$$\mathcal{C} = \left\{ (t, f, \tau) \in \left(\mathbf{S}^d_+\right)^n \times \mathbf{R}^m \times \mathbf{R} \mid \mathrm{Compl}_f(t) \equiv \sup_{v \in \mathbf{R}^m : Rv \le r} \left[f^T v - \frac{1}{2} v^T \left[\sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \right] v \right] \le \tau \right\}.$$

The required SDR is given by the following

Proposition 4.8.1 *Let $t = (t_1, ..., t_n) \in \mathbf{S}^d_+$ and $f \in \mathbf{R}^m$. Then the inequality*

$$\mathrm{Compl}_f(t) \le \tau$$

is satisfied if and only if there exists a nonnegative vector μ of the same dimension q as the one of columns of R such that the matrix

$$\mathcal{A}(t, f, \tau, \mu) = \begin{pmatrix} 2\tau - 2r^T\mu & -f^T + \mu^T R \\ -f + R^T\mu & \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \end{pmatrix}$$

is positive semidefinite. Thus, the epigraph of $\mathrm{Compl}_f(t)$ *(regarded as a function of $t \in \left(\mathbf{S}^d_+\right)^n$ and $f \in \mathbf{R}^m$) admits the SDR*

$$\begin{array}{rcl} \begin{pmatrix} 2\tau - 2r^T\mu & -f^T + \mu^T R \\ -f + R^T\mu & \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \end{pmatrix} & \succeq & 0; \\ t_i & \succeq & 0, \ i = 1, ..., n; \\ \mu & \ge & 0. \end{array} \tag{4.8.8}$$

Proof. First let us explain where the result comes from. By definition, $\mathrm{Compl}_f(t) \le \tau$ if

$$\sup_{v : Rv \le r} \left[f^T v - \frac{1}{2} v^T A(t) v \right] \le \tau \qquad \left[A(t) = \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \right]$$

The supremum in the left hand side is taken over v varying in a set given by linear constraints $Rv \le r$. If we add "penalized constraints" $\mu^T(r - Rv)$ to the objective, μ being a nonnegative weight vector, and then remove the constraints, passing to the supremum of the penalized objective over the entire space, i.e., to the quantity

$$\phi_f(t, \mu) \equiv \sup_{v \in \mathbf{R}^m} \left[f^T v - \frac{1}{2} v^T A(t) v + \mu^T (r - Rv) \right],$$

then we end up with something which is $\ge \mathrm{Compl}_f(t)$; consequently, <u>if</u> there exists $\mu \ge 0$ such that

$$f^T v - \frac{1}{2} v^T A(t) v + \mu^T (r - Rv) \le \tau \quad \forall v \in \mathbf{R}^m,$$

<u>then</u> we have $\tau \ge \mathrm{Compl}_f(t)$. On the other hand, the *Lagrange Duality* says that under the Slater condition from **S.1** the quantity $\phi_f(t, \mu)$ for properly chosen $\mu \ge 0$ is *exactly* the supremum of $f^T v - \frac{1}{2} v^T A(t) v$ over v satisfying $Rv \le r$; thus, if $\tau \ge \mathrm{Compl}_f(t)$, then $\tau \ge \phi_f(t, \mu)$ for some $\mu \ge 0$. Thus, believing in the Lagrange Duality, we come to the following observation:

(!) *The inequality* $\mathrm{Compl}_f(t) \le \tau$ *is equivalent to the existence of a* $\mu \ge 0$ *such that* $\phi_f(t,\mu) \le \tau$.

It remains to note that the inequality $\phi_f(t,\mu) \le \tau$ says that the unconstrained – over the entire space – minimum of certain convex quadratic form, namely, of the form

$$Q(v) = [\tau - r^T\mu] + \frac{1}{2}v^T A(t)v + (-f + R^T\mu)^T v$$

– is nonnegative; by Simple Lemma (see Section 4.3.1), the latter fact is equivalent to the positive semidefiniteness of the matrix $\mathcal{A}(t,f,\tau,\mu)$.

The outlined reasoning does not satisfy us: in our course, we do not deal with Lagrange Duality. All we have is Conic Duality (in fact, equivalent to the Lagrange one). In order to be "self-sufficient", let us derive (!) from Conic Duality.

Our first observation is as follows:

(*) *Let* $t_1,...,t_n \in \mathbf{S}^d_+$, $f \in \mathbf{R}^m$ *and* $\tau \in \mathbf{R}$ *be fixed. Consider the following system of inequalities with variables* $v \in \mathbf{R}^m$, $\sigma \in \mathbf{R}$:

$$\begin{array}{rcl} \frac{1}{2}v^T A(t)v + \sigma - f^T v & \le & 0 \\ & & [A(t) = \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T] \\ R^T v & \le & r \end{array} \qquad (S(t,f))$$

Then $(t,f,\tau) \in \mathcal{C}$ *if and only if* $\tau \ge \sigma$ *for all solutions to* $(S(t,f))$, *i.e., if and only if the linear inequality* $\tau \ge \sigma$ *is a consequence of the system* $(S(t,f))$.

Indeed, the σ-components of the solutions to $(S(t,f))$ are exactly those σ's which do not exceed the value of the quadratic form

$$E^f_t(v) = f^T v - \frac{1}{2}v^T A(t)v$$

at certain point of the set $\mathcal{V} = \{v \mid Rv \le r\}$. Consequently, to say that a given τ is $\ge$ all these σ's is exactly the same as to say that τ is $\ge$ the supremum of the form $E^f_t(v)$ over $v \in \mathcal{V}$, i.e., is exactly the same as to say that $\tau \ge \mathrm{Compl}_f(t)$.

Now, $(S(t,f))$ is nothing but a linear vector inequality. Indeed, the quadratic inequality in $(S(t,f))$ is a conic quadratic inequality:

$$\begin{array}{c} \frac{1}{2}v^T A(t)v + \sigma - f^T v \le 0 \\ \Updownarrow \\ v^T A(t)v + 2\sigma - 2f^T v \le 0 \\ \Updownarrow \\ \| B^T(t)v \|_2^2 - 2f^T v + 2\sigma \le 0, \\ B(t) = [b_{11}t_1^{1/2}, b_{12}t_1^{1/2}, ..., b_{1S}t_1^{1/2}; ...; b_{n1}t_n^{1/2}, ..., b_{nS}t_n^{1/2}] \quad [A(t) = B(t)B^T(t)] \\ \Updownarrow \\ \begin{pmatrix} B^T(t)v \\ \frac{1}{2} + \sigma - f^T v \\ \frac{1}{2} - \sigma + f^T v \end{pmatrix} \ge_{\mathbf{L}} 0 \end{array}$$

so that $(S(t,f))$ is the linear vector inequality in variables v, σ, τ:

$$Q\begin{pmatrix} v \\ \sigma \end{pmatrix} - q \equiv \begin{pmatrix} B^T(t)v \\ \frac{1}{2} + \sigma - f^T v \\ \frac{1}{2} - \sigma + f^T v \\ r - Rv \end{pmatrix} \ge_{\mathbf{K}} 0,$$

where $\mathbf{K}$ is the direct product of the Lorentz cone and the nonnegative orthant of appropriate dimensions.

Note that the resulting linear vector inequality is strictly feasible. Indeed, due to the Slater assumption in **S.1**, we may choose v to make strict the inequalities $r - Rv \ge 0$; after v is chosen, we may make σ to be negative enough to make strict also the "conic quadratic" part of our vector inequality.

Now recall that we know what are the necessary and sufficient conditions for a linear inequality to be a consequence of a strictly feasible linear vector inequality: they are given by Proposition 2.4.2. In the case in question these conditions are equivalent to the existence of a nonnegative vector μ of the same dimension q as the one of columns in R and a vector $\zeta \in \mathbf{L}$ such that

$$\begin{array}{c} [v^T;\sigma]Q^T\begin{pmatrix}\zeta\\ \mu\end{pmatrix} = -\sigma \quad \forall (v,\sigma)\in\mathbf{R}^m\times\mathbf{R};\\ [\zeta^T;\mu^T]q \geq -\tau. \end{array}$$

Recalling the origin of Q and q, we come to the following conclusion:

(**) *Let $t \in \left(\mathbf{S}^d_+\right)^n$, $f \in \mathbf{R}^m$ and $\tau \in \mathbf{R}$. Then* $\mathrm{Compl}_f(t) \leq \tau$ *if and only if there exist d-dimensional vectors ξ_{is}, reals α,β and a vector ν such that*

$$\begin{array}{llcl} (a) & (\beta-\alpha)f - R^T\mu + \sum_{i=1}^n\sum_{s=1}^s b_{is}t_i^{1/2}\xi_{is} & = & 0;\\ (b) & \alpha-\beta & = & -1;\\ (c) & \frac{1}{2}(\alpha+\beta)+r^T\mu & \leq & \tau;\\ (d) & \mu & \geq & 0;\\ (e) & \beta & \geq & \sqrt{\alpha^2+\sum_{i=1}^n\sum_{s=1}^S \xi_{is}^T\xi_{is}}. \end{array} \tag{4.8.9}$$

Now consider the conic quadratic inequality

$$\begin{pmatrix} B^T(t)v\\ \frac{1}{2}-r^T\mu+\sigma+[-f+R^T\mu]^Tv\\ \frac{1}{2}+r^T\mu-\sigma-[-f+R^T\mu]^Tv \end{pmatrix} \geq_{\mathbf{L}} 0 \tag{4.8.10}$$

v,σ being the variables and $\mu \geq 0$ being a vector of parameters, and let us ask ourselves when the inequality $\sigma \leq \tau$ is a consequence of this (clearly strictly feasible) vector inequality. According to Proposition 2.4.2 this is the case if and only if there exist vectors $\xi_{is} \in \mathbf{R}^d$ and reals α,β such that

$$\begin{array}{llcl} (a) & (\alpha-\beta)[-f+R^T\mu] + \sum_{i=1}^n\sum_{s=1}^S b_{is}t_i^{1/2}\xi_{is} & = & 0;\\ (b) & \alpha-\beta & = & -1;\\ (c) & \frac{1}{2}(\alpha+\beta)+(\alpha-\beta)r^T\mu & \leq & \tau;\\ (e) & \beta & \geq & \sqrt{\alpha^2+\sum_{i=1}^n\sum_{s=1}^S \xi_{is}^T\xi_{is}}. \end{array} \tag{4.8.11}$$

Comparing (4.8.9) and (4.8.11), we come to the conclusion as follows:

(***) *Let $t \in \left(\mathbf{S}^d_+\right)^n$, $f \in \mathbf{R}^m$ and $\tau \in \mathbf{R}$. Then* $\mathrm{Compl}_f(t) \leq \tau$ *if and only if there exists $\mu \geq 0$ such that the inequality $\sigma \leq \tau$ is a consequence of the conic quadratic inequality* (4.8.10).

It remains to note that the conic quadratic inequality (4.8.10) is equivalent to the scalar quadratic inequality

$$v^TA(t)v - 2r^T\mu + 2\sigma + 2[-f+R^T\mu]v \leq 0.$$

Consequently, (***) says that $\mathrm{Compl}_f(t) \leq \tau$ if and only if there exists $\mu \geq 0$ such that the following implication holds true:

$$\forall (v,\sigma): \quad \sigma \leq [f-R^T\mu]^Tv - \frac{1}{2}v^TA(t)v + r^T\mu) \Rightarrow \sigma \leq \tau.$$

But the latter implication clearly holds true if and only if

$$\tau \geq \max_{v\in\mathbf{R}^m}\left[[f-R^T\mu]^Tv - \frac{1}{2}v^TAv + r^T\mu\right]. \tag{4.8.12}$$

Thus, $\tau \geq \mathrm{Compl}_f(t)$ if and only if there exists $\mu \geq 0$ such that (4.8.12) takes place; but this is exactly the statement (!) we need. ■

SDR of the epigraph of the compliance immediately implies a semidefinite reformulation of the *multi-load* Standard SSD problem:

$$\begin{array}{llcl} & \tau & \to & \min\\ \text{s.t.} & & & \\ (a) & \begin{pmatrix} 2\tau-2r^T\mu_l & -f_l^T+\mu_l^TR\\ -f_l+R^T\mu_l & \sum_{i=1}^n\sum_{s=1}^S b_{is}t_ib_{is}^T \end{pmatrix} & \succeq & 0,\ l=1,...,k;\\ (b) & t_i & \succeq & 0,\ i=1,...,n;\\ (c) & \sum_{i=1}^n \mathrm{Tr}(t_i) & \leq & w;\\ (d) & \underline{\rho}_i \leq \mathrm{Tr}(t_i) & \leq & \overline{\rho}_i,\ i=1,...,n;\\ (e) & \mu_l & \geq & 0,\ l=1,...,k, \end{array} \tag{4.8.13}$$

the design variables being $t_i \in \mathbf{S}^d$, $i = 1, ..., n$, μ_l – vectors of the same dimension q as the one of the columns in R, $l = 1, ..., k$, and $\tau \in \mathbf{R}$, and $f_1, ..., f_k$ being the loading scenarios. Indeed, the LMI's (a) along with nonnegativity constraints (e) express the fact that the worst-case, over the loads $f_1, ..., f_k$, compliance of the construction yielded by the rigidities $t_1, ..., t_n$ does not exceed τ (see Proposition 4.8.1), while the remaining constraints (b), (c), (d) express the fact that $t = (t_1, ..., t_n)$ is an admissible design.

When passing to the *robust* Standard SSD problem – the one where $\mathcal{F}$ is an ellipsoid:

$$\mathcal{F} = \{f = Qu \mid u^T u \leq 1\}$$

rather than a finite set, we meet with a difficulty: our objective now is

$$\mathrm{Compl}_{\mathcal{F}}(t) = \sup_{f \in \mathcal{F}} \mathrm{Compl}_f(t),$$

i.e., it is the supremum of *infinitely many* SD-representable functions; and our calculus does not offer us tools to build an SDR for such an aggregate. This difficulty reflects the essence of the matter: if there are obstacles, the robust version of the SSD problem is extremely difficult (at least as difficult as an NP-complete combinatorial problem). In the "obstacle-free" case, however, it is easy to get an SDR for $\mathrm{Compl}_{\mathcal{F}}(t)$, provided that $\mathcal{F}$ is an ellipsoid:

Proposition 4.8.2 *Let the set of kinematically admissible displacements coincide with the space $\mathbf{R}^m$ of all virtual displacements: $\mathcal{V} = \mathbf{R}^m$, and let $\mathcal{F}$ be an ellipsoid:*

$$\mathcal{F} = \{f = Qu \mid u^T u \leq 1\} \quad [Q \in \mathbf{M}^{mk}].$$

Then the function $\mathrm{Compl}_{\mathcal{F}}(t)$, regarded as a function of $t = (t_1, ..., t_n) \in \left(\mathbf{S}^d_+\right)^n$, is SD-representable: for $t \in \left(\mathbf{S}^d_+\right)^n$ one has

$$\mathrm{Compl}_{\mathcal{F}}(t) \leq \tau \Leftrightarrow \begin{pmatrix} 2\tau I_k & Q^T \\ Q & \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \end{pmatrix} \succeq 0. \tag{4.8.14}$$

Consequently, in the case in question the Standard SSD problem can be posed as the following semidefinite program:

$$\begin{array}{rcl} & \tau & \rightarrow & \min \\ \textit{s.t.} & & & \\ & \begin{pmatrix} 2\tau I_k & Q^T \\ Q & \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \end{pmatrix} & \succeq & 0; \\ & t_i & \succeq & 0, \ i = 1, ..., n; \\ & \sum_{i=1}^n \mathrm{Tr}(t_i) & \leq & w; \\ & \underline{\rho}_i \leq \mathrm{Tr}(t_i) & \leq & \overline{\rho}_i, \ i = 1, ..., n. \end{array} \tag{4.8.15}$$

Proof. Let, as always, $A(t) = \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T$. We have

$$\begin{array}{rcll}
\mathrm{Compl}_{\mathcal{F}} & \le & \tau & \Leftrightarrow \\
(Qu)^T v - \frac{1}{2} v^T A(t) v & \le & \tau \quad \forall v \quad \forall (u : u^T u \le 1) & \Leftrightarrow \\
(Qu)^T v - \frac{1}{2} v^T A(t) v & \le & \tau \quad \forall v \quad \forall (u : u^T u = 1) & \Leftrightarrow \\
(Q \parallel w \parallel_2^{-1} w)^T v - \frac{1}{2} v^T A(t) v & \le & \tau \quad \forall v \forall (w \neq 0) & \Leftrightarrow \\
(Qw)^T \underbrace{(\parallel w \parallel_2 v)}_{-y} - \frac{1}{2} (\parallel w \parallel_2 v)^T A(t) (\parallel w \parallel_2 v) & \le & \tau w^T w \quad \forall v \quad \forall (w \neq 0) & \Leftrightarrow \\
2\tau w^T w + 2 w^T Q^T y + y^T A(t) y & \ge & 0 \quad \forall y \forall (w \neq 0) & \Leftrightarrow \\
2\tau w^T w + 2 w^T Q^T y + y^T A(t) y & \ge & 0 \quad \forall y \in \mathbf{R}^m, w \in \mathbf{R}^k & \Leftrightarrow \\
\begin{pmatrix} 2\tau I_k & Q^T \\ Q & A(t) \end{pmatrix} & \succeq & 0. &
\end{array}$$

$\blacksquare$

"Universal" semidefinite form of the Standard SSD problem. To allow for unified treatment of both multi-load Standard SSD problem (4.8.13) and the robust obstacle-free problem (4.8.15), it makes sense to note that both formulations are covered by the following generic semidefinite program:

$$\begin{array}{llcl}
 & \tau & \to & \min \\
\text{s.t.} & & & \\
(a) & \begin{pmatrix} 2\tau I_p + \mathcal{D}_l z + D_l & [\mathcal{E}_l z + E_l]^T \\ [\mathcal{E}_l z + E_l] & \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \end{pmatrix} & \succeq & 0, \ l = 1, ..., K; \\
(b) & t_i & \succeq & 0, \ i = 1, ..., n; \\
(c) & \sum_{i=1}^n \mathrm{Tr}(t_i) & \le & w; \\
(d) & \underline{\rho}_i \le \mathrm{Tr}(t_i) & \le & \overline{\rho}_i, \ i = 1, ..., n; \\
(e) & z & \ge & 0,
\end{array} \qquad (\mathrm{Pr})$$

where

- the design variables are $t_i \in \mathbf{S}^d$, $i = 1, ..., n$, $z \in \mathbf{R}^N$, $\tau \in \mathbf{R}$;
- the data are given by the $m \times d$ matrices b_{is}, affine mappings

  $$z \mapsto \mathcal{D}_l z + D_l : \mathbf{R}^N \to \mathbf{S}^d; \quad z \mapsto \mathcal{E}_l z + E_l : \mathbf{R}^N \to \mathbf{M}^{mp}, \ l = 1, ..., K.$$

  and the reals $\underline{\rho}_i, \overline{\rho}_i$, $i = 1, ..., n$, $w > 0$.

Indeed,

- The multi-load problem (4.8.13) corresponds to the case of $p = 1$, $K = k$ (the number of loading scenarios),

  $$z = (\mu^1, ..., \mu^k) \in \mathbf{R}^q \times ... \times \mathbf{R}^q, \mathcal{D}_l z + D_l = -2r^T \mu^l, \quad \mathcal{E}_l z + E_l = -f_l + R^T \mu^l$$

  (f_l is the l-th load of interest).

- The robust problem (4.8.15) corresponds to the case of $K = 1$, $p = k$ (the dimension of the loading ellipsoid). In fact, in the robust problem there should be no z at all; however, in order to avoid extra comments, we introduce one-dimensional "redundant" variable z and set

  $$\mathcal{E}_1 = 0, E_1 = Q; \mathcal{D}_1 z + D_1 = -2z I_k.$$

It is immediately seen that the resulting problem (Pr), namely, the problem

$$\tau \to \min \mid \begin{pmatrix} 2(\tau - z)I_k & Q^T \\ Q & \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \end{pmatrix} \succeq 0 \ \& \ (\mathrm{Pr}.(b-d)) \ \& \ z \geq 0$$

is equivalent to (4.8.15).

Note that when converting the problems of our actual interest (4.8.13) and (4.8.15) to the generic form (Pr), we ensure the following property of the resulting problem:

S.5. For every $l = 1, ..., K$, there exists $\alpha_l \in \mathbf{S}^p_{++}$ and $V_l \in \mathbf{M}^{mp}$ such that

$$\sum_{l=1}^K [\mathcal{D}_l^* \alpha_l + 2\mathcal{E}_l^* V_l] < 0.$$

Indeed, in the case when the original problem is the multi-load problem (4.8.13), we have $\mathbf{S}^p = \mathbf{R}$, $\mathbf{M}^{mp} = \mathbf{R}^m$ and

$$\sum_{l=1}^K [\mathcal{D}_l^* \alpha_l + 2\mathcal{E}_l^* V_l] = \begin{pmatrix} -2\alpha_1 r + 2RV_1 \\ -2\alpha_2 r + 2RV_2 \\ \cdots \\ -2\alpha_K r + 2RV_K \end{pmatrix};$$

the latter vector is negative when all α_l are equal to 1 and all V_l are equal to strictly feasible solution of the system $Rv \leq r$; such a solution exists by **S.1.**

In the case when the original problem is the obstacle-free robust problem (4.8.15), we have

$$\mathcal{D}_1^* \alpha_1 + 2\mathcal{E}_1^* V_1 = -2\,\mathrm{Tr}(\alpha_1),$$

and to make the result negative, it suffices to set $\alpha_1 = I_p$.

From now, speaking about (Pr), we assume that the data of the problem satisfy **S.3**, **S.4** and **S.5.**

Remark 4.8.1 *The problem* (Pr) *is strictly feasible.*

Indeed, let us choose somehow $z > 0$. By **S.3**, we can choose $t_i \succ 0$, $i = 1, ..., n$, to satisfy the strict versions of the inequalities (Pr.(b, c, d)). By **S.4** the matrix $\sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T$ is positive definite; but then, by Lemma on the Schur Complement, LMI's (Pr.(a)) are satisfied as strict for all large enough values of τ.

4.8.4 From primal to dual

From the viewpoint of numerical processing, a disadvantage of the problem (Pr) is its huge design dimension. Consider, e.g., the case of a multi-load design of an obstacle-free truss with M-node ground structure. In this case (Pr) is a semidefinite program of the design dimension $n+1$, $n = O(M^2)$ being the number of tentative bars. The program includes k (k is the number of loading scenarios) "big" LMI's (each of the row size $m+1$, m being the number of degrees of freedom of the set of nodes; $m \approx 2M$ for planar and $m \approx 3M$ for spatial constructions) and a number of scalar linear inequality constraints. For a 15×15 planar nodal grid with fixed most-left nodes we get $M = 225$, $n+1 = 25096$, $m = 420$. Even an LP program with 25,000 variables should not be treated as a "small" one; a semidefinite program of such a design dimension definitely is not accessible for the existing software. The situation, however, is not that bad, and the way to overcome the difficulty is offered by duality. Namely, it turns out that the problem dual to (Pr) (by itself it is even more difficult for straightforward numerical solution than (Pr)) can be simplified a lot: it admits analytical elimination of most of the variables. E.g., the dual to the outlined multi-load truss problem can be converted to a semidefinite program with nearly mk design variables; for the indicated sizes and 3-scenario setting, the design dimension of the latter problem is about 1300, which is within the range of applicability of the existing solvers.

We are about to build the problem dual to (Pr) and to process it analytically.

Step 0. Building the dual. Applying to (Pr) our formalism for passing from a semidefinite program to its dual, we come to the following semidefinite program:

$$
\begin{array}{lrcl}
\text{maximize} & -\phi & \equiv & -\sum_{l=1}^{K}\operatorname{Tr}(D_l\alpha_l + 2E_l^TV_l) \\
 & & & -\sum_{i=1}^{n}[\overline{\rho}_i\sigma_i^+ - \underline{\rho}_i\sigma_i^-] - w\gamma \\
\text{s.t.} & & & \\
 & \begin{pmatrix} \alpha_l & V_l^T \\ V_l & \beta_l \end{pmatrix} & \succeq & 0,\ l=1,...,K \\
 & & & [\alpha_l\in\mathbf{S}^p, \beta_l\in\mathbf{S}^m, V_l\in\mathbf{M}^{mp}], \\
 & \tau_i & \succeq & 0,\ i=1,...,n \\
 & & & [\tau_l\in\mathbf{S}^{D_l}], \\
 & \sigma_i^+,\sigma_i^- & \geq & 0,\ i=1,...,n \\
 & & & [\sigma_i^+,\sigma_i^-\in\mathbf{R}], \\
 & \gamma & \geq & 0 \\
 & & & [\gamma\in\mathbf{R}], \\
 & \eta & \geq & 0 \\
 & & & [\eta\in\mathbf{R}^N], \\
 & 2\sum_{l=1}^{K}\operatorname{Tr}(\alpha_l) & = & 1, \\
 & \sum_{l=1}^{K}[\mathcal{D}_l^*\alpha_l + 2\mathcal{E}_l^*V_l] + \eta & = & 0, \\
 & \sum_{l=1}^{K}\sum_{s=1}^{S} b_{is}^T\beta_l b_{is} + \tau_i & & \\
 & +[\sigma_i^- - \sigma_i^+ - \gamma]I_d & = & 0,\ \ i=1,...,n,
\end{array}
\qquad (\mathrm{D}_{\mathrm{ini}})
$$

the design variables being α_l,β_l,V_l, $l=1,...,K$.

Step 1. Eliminating η and $\{\tau_i\}_{i=1}^n$. ($\mathrm{D}_{\mathrm{ini}}$) clearly is equivalent to the problem

$$
\begin{array}{lrcl}
\text{minimize} & \phi & \equiv & \sum_{l=1}^{K}\operatorname{Tr}(D_l\alpha_l + 2E_l^TV_l) \\
 & & & +\sum_{i=1}^{n}[\overline{\rho}_i\sigma_i^+ - \underline{\rho}_i\sigma_i^-] + w\gamma \\
\text{s.t.} & & & \\
(a) & \begin{pmatrix} \alpha_l & V_l^T \\ V_l & \beta_l \end{pmatrix} & \succeq & 0,\ l=1,...,K; \\
(b) & \sigma_i^+,\sigma_i^- & \geq & 0,\ l=1,...,n; \\
(c) & \gamma & \geq & 0; \\
(d) & 2\sum_{l=1}^{K}\operatorname{Tr}(\alpha_l) & = & 1; \\
(e) & \sum_{l=1}^{K}[\mathcal{D}_l^*\alpha_l + 2\mathcal{E}_l^*V_l] & \leq & 0; \\
(f) & \sum_{l=1}^{K}\sum_{s=1}^{S} b_{is}^T\beta_l b_{is} & \preceq & [\gamma+\sigma_i^+ - \sigma_i^-]I_d, \\
 & & & i=1,...,n.
\end{array}
\qquad (\mathrm{D}')
$$

Note that passing from ($\mathrm{D}_{\mathrm{ini}}$) to ($\mathrm{D}'$), we have switched from maximization of $-\phi$ to minimization of ϕ, so that the optimal value in (D') is minus the one of ($\mathrm{D}_{\mathrm{ini}}$).

Step 2. Eliminating $\{\beta_l\}_{l=1}^K$. We start with observing that (D$'$) is strictly feasible. Indeed, let us set, say, $\sigma_i^\pm = 1$. By **S.5** there exist positive definite matrices α_l and rectangular matrices V_l of appropriate sizes satisfying the strict version of (D$'.(e)$); by normalization, we may enforce these matrices to satisfy (D$'.(d)$). Given indicated α_l, V_l and choosing "large enough" β_l, we enforce validity of the strict versions of (D$'.(a)$). Finally, choosing large enough $\gamma > 0$, we enforce strict versions of (D$'.(c)$) and (D$'.(f)$).

Note that the same arguments demonstrate that

Remark 4.8.2 *Problem* (D$_{\rm ini}$) *is strictly feasible.*

Since (D$'$) is strictly feasible, its optimal value is the same as in problem (D$''$) obtained from (D$'$) by adding to the set of constraints the constraints

$$(g) \quad \alpha_l \succ 0, \ l = 1, ..., K.$$

Now note that if a collection

$$(\alpha = \{\alpha_l\}_{l=1}^K, V = \{V_l\}_{l=1}^K, \beta = \{\beta_l\}_{l=1}^K, \sigma = \{\sigma_i^\pm\}_{i=1}^n, \gamma)$$

is a feasible solution to (D$''$), then the collection

$$(\alpha, V, \beta(\alpha, V) = \{\beta_l(\alpha, V) = V_l\alpha_l^{-1}V_l^T\}_{l=1}^K, \sigma, \gamma)$$

also is a feasible solution to (D$''$) with the same value of the objective; indeed, from LMI's (D$'.(a)$) by Lemma on the Schur Complement it follows that $\beta_l(\alpha, V) \preceq \beta_l$, so that replacing β_l with $\beta_l(\alpha, V)$ we preserve validity of the LMI's (D$'.(f)$) as well as (D$'.(a)$). Consequently, (D$''$) is equivalent to the problem

$$\begin{array}{lrcl}
\text{minimize} & \phi & \equiv & \sum_{l=1}^K \mathrm{Tr}(D_l\alpha_l + 2E_l^TV_l) \\
 & & & + \sum_{i=1}^n[\overline{\rho}_i\sigma_i^+ - \underline{\rho}_i\sigma_i^-] + w\gamma \\
\text{s.t.} & & & \\
(b) & \sigma_i^+, \sigma_i^- & \geq & 0, \ i = 1, ..., n; \\
(c) & \gamma & \geq & 0; \\
(d) & 2\sum_{l=1}^K \mathrm{Tr}(\alpha_l) & = & 1; \\
(e) & \sum_{l=1}^K[\mathcal{D}_l^*\alpha_l + 2\mathcal{E}_l^*V_l] & \leq & 0; \\
(f') & \sum_{l=1}^K\sum_{s=1}^S b_{is}^TV_l\alpha_l^{-1}V_l^Tb_{is} & \leq & [\gamma + \sigma_i^+ - \sigma_i^-]I_d, \ i = 1, ..., n, \\
(g) & \alpha_l & \succ & 0, \ l = 1, ..., K.
\end{array} \qquad (\mathrm{D}''')$$

Now note that the system of LMI's (D$'''.(g)$)&(D$'''.(f')$) is equivalent to the system of LMI's

$$\begin{array}{rcl}
\begin{pmatrix} A(\alpha) & B_i^T(V) \\ B_i(V) & (\gamma + \sigma_i^+ - \sigma_i^-)I_d \end{pmatrix} & \succ & 0, \ i = 1, ..., n, \\
A(\alpha) & \succ & 0,
\end{array} \qquad (4.8.16)$$

where

$$\begin{array}{rcl}
\alpha & = & \{\alpha_l \in \mathbf{S}^p\}_{l=1}^K, \\
V & = & \{V_l \in \mathbf{M}^{pm}\}_{l=1}^K, \\
A(\alpha) & = & \mathrm{Diag}(\overbrace{\alpha_1, ..., \alpha_1}^{S \text{ times}}, \overbrace{\alpha_2, ..., \alpha_2}^{S \text{ times}}, ..., \overbrace{\alpha_K, ..., \alpha_K}^{S \text{ times}}), \\
B_i(V) & = & [b_{i1}^TV_1, b_{i2}^TV_1, ..., b_{iS}^TV_1; b_{i1}^TV_2, b_{i2}^TV_2, ..., b_{iS}^TV_2; ...; b_{i1}^TV_K, b_{i2}^TV_K, ..., b_{iS}^TV_K].
\end{array} \qquad (4.8.17)$$

Indeed, the difference of the left and of the right hand sides of (D$'''$.(f')) is the Schur complement of the angular block of the left hand side matrix in (4.8.16), and it remains to apply the Lemma on the Schur Complement. Consequently, (D$'''$) is equivalent to the problem

$$
\begin{array}{lrcl}
\text{minimize} & \phi & \equiv & \sum_{l=1}^K \mathrm{Tr}(D_l\alpha_l + 2E_l^T V_l) \\
 & & & + \sum_{i=1}^n [\overline{\rho}_i \sigma_i^+ - \underline{\rho}_i \sigma_i^-] + w\gamma \\
\text{s.t.} & & & \\
 & \begin{pmatrix} A(\alpha) & B_i^T(V) \\ B_l(V) & (\gamma + \sigma_i^+ - \sigma_i^-) I_d \end{pmatrix} & \succeq & 0,\ i = 1, ..., n, \\
 & \sigma_i^+, \sigma_i^- & \geq & 0,\ i = 1, ..., n; \\
 & \gamma & \geq & 0; \\
 & 2\sum_{l=1}^K \mathrm{Tr}(\alpha_l) & = & 1; \\
 & \alpha_l & \succ & 0,\ l = 1, ..., K; \\
 & \sum_{l=1}^K [\mathcal{D}_l^* \alpha_l + 2\mathcal{E}_l^* V_l] & \leq & 0.
\end{array}
$$

Problem (D$'''$) is strictly feasible along with (D$'$), so that its optimal value remains unchanged when we remove from (D$'''$) the strict LMI's, thus coming to the final form of the problem dual to (Pr):

$$
\begin{array}{lrcl}
\text{minimize} & \phi & \equiv & \sum_{l=1}^K \mathrm{Tr}(D_l\alpha_l + 2E_l^T V_l) \\
 & & & + \sum_{i=1}^n [\overline{\rho}_i \sigma_i^+ - \underline{\rho}_i \sigma_i^-] + w\gamma \\
\text{s.t.} & & & \\
 & \begin{pmatrix} A(\alpha) & B_i^T(V) \\ B_i(V) & (\gamma + \sigma_i^+ - \sigma_i^-) I_d \end{pmatrix} & \succeq & 0,\ l = 1, ..., N, \\
 & \sigma_i^+, \sigma_i^- & \geq & 0,\ i = 1, ..., n; \\
 & \gamma & \geq & 0; \\
 & 2\sum_{l=1}^K \mathrm{Tr}(\alpha_l) & = & 1; \\
 & \sum_{l=1}^K [\mathcal{D}_l^* \alpha_l + 2\mathcal{E}_l^* V_l] & \leq & 0,
\end{array}
\tag{Dl}
$$

the design variables of the problem being

$$\alpha = \{\alpha_l \in \mathbf{S}^p\}_{l=1}^K, V = \{V_i \in \mathbf{M}^{mp}\}_{l=1}^K, \sigma = \{\sigma_i^\pm \in \mathbf{R}\}_{i=1}^n, \gamma \in \mathbf{R}.$$

As we have seen, both the primal problem (Pr) and its dual ($\mathrm{D_{ini}}$) are strictly feasible (Remarks 4.8.1, 4.8.2). Consequently, both (Pr) and ($\mathrm{D_{ini}}$) are solvable with equal optimal values (the Conic Duality Theorem) and with bounded level sets (see Exercise 2.12). By its origin, the optimal value in the problem (Dl) is minus the optimal value in ($\mathrm{D_{ini}}$), and of course (Dl) inherits from ($\mathrm{D_{ini}}$) the property to have bounded level sets and is therefore solvable. Thus, we get the following

Proposition 4.8.3 *Both problems* (Pr), (Dl) *are strictly feasible, solvable and possess bounded level sets. The optimal values in these problems are negations of each other.*

The case of "simple bounds". In the case when there are no actual bounds on $\mathrm{Tr}(t_i)$ (formally it means that $\underline{\rho}_i = 0, \overline{\rho}_i = w$ for all i), the dual problem (Dl) can be further simplified, namely, we can eliminate the σ-variables. Indeed, consider a feasible solution to (Dl). When replacing all $\sigma_i^\pm$ with zeros, simultaneously increasing γ by $\delta = \max[0, \max_i(\sigma_i^+ - \sigma_i^-)]$, we clearly preserve feasibility and add to the objective the quantity

$$\begin{array}{rcl}
w\delta - \sum_{i=1}^n \left[\overline{\rho}_i\sigma_i^+ - \underline{\rho}_i\sigma_i^-\right] & = & w\delta - \sum_{i=1}^n \overline{\rho}_i\sigma_i^+ \quad [\text{since } \underline{\rho}_i = 0] \\
& \le & w\delta - w\sum_{i=1}^n \sigma_i^+ \quad [\text{since } \overline{\rho}_i \ge w] \\
& \le & w\max_i \sigma_i^+ - w\sum_{i=1}^n \sigma_i^+ \quad [\text{since } \delta \le \max_i \sigma_i^+ \text{ due to } \sigma_i^\pm \ge 0] \\
& \le & 0 \quad [\text{since } \sigma_i^+ \ge 0],
\end{array}$$

i.e., we gain in the objective value. Thus, we loose nothing when setting in (Dl) $\sigma_i^\pm = 0$, thus coming to the problem

$$\begin{array}{lrcl}
\text{minimize} & \phi & = & \sum_{l=1}^K \mathrm{Tr}(D_l\alpha_l + 2E_l^T V_l) + w\gamma \\
\text{s.t.} & & & \\
& \begin{pmatrix} A(\alpha) & B_i^T(V) \\ B_i(V) & \gamma I_d \end{pmatrix} & \succeq & 0,\ i = 1, ..., n, \\
& \sum_{l=1}^K [\mathcal{D}_l^*\alpha_l + 2\mathcal{E}_l^* V_l] & \le & 0, \\
& 2\sum_{l=1}^K \mathrm{Tr}(\alpha_l) & = & 1, \\
& \gamma & \ge & 0,
\end{array} \qquad (\mathrm{Dl_{sb}})$$

the design variables being $\alpha = \{\alpha_l \in \mathbf{S}^p\}_{l=1}^K$, $V = \{V_l \in \mathbf{M}^{pm}\}_{l=1}^K$, $\gamma \in \mathbf{R}$.

To understand how fruitful was our effort, let us compare the dimensions of the problem $(\mathrm{Dl_{sb}})$ with those of the original problem (Pr) in the simplest case of a k-load "obstacle-free" truss design problem with simple bounds. As we have seen, in this case the design dimension of the primal problem (Pr) is $O(M^2)$, M being the cardinality of the nodal grid, and (Pr) involves k LMI's with row sizes $m = O(M)$ and $n+1$ scalar linear inequality constraints. The design dimension of $(\mathrm{Dl_{sb}})$ is just $k + mk + 1$ (in the case in question, $p = 1$); the dual problem includes n LMI's of the row size $(k+1)$ each (in the case in question, $S = 1$), two scalar linear inequalities and a single scalar linear equality constraint. We see that if the number of loading scenarios k is a small integer (which normally is the case), the design dimension of the dual problem is $O(M)$, i.e., it is by orders of magnitude less than the design dimension of the primal problem (Pr). As a kind of penalization, the dual problem involves a lot of non-scalar LMI's ($n = O(M^2)$ instead of k non-scalar LMI's in (Pr)), but all these LMI's are "small" – of row size $k + 1 = O(1)$ each, while the non-scalar LMI's in (Pr) are "large" – of row size $O(M)$ each. As a result, when solving (Pr) and $(\mathrm{Dl_{sb}})$ by the best known so far numerical techniques (the interior-point algorithms), the computational effort for (Pr) turns out to be $O(M^6)$, while for $(\mathrm{Dl_{sb}})$ it is only $O(k^3M^3)$, which, for large M and small k, does make a difference! Of course, there is an immediate concern about the dual problem: the actual design variables are not seen in it at all; how to recover a (nearly) optimal construction from a (nearly) optimal solution to the dual problem? In fact, however, there is no reason to be concerned: the required recovering routines do exist and are cheap computationally.

4.8.5 Back to primal

Problem (Dl) is not exactly the dual of (Pr) – it is obtained from this dual by eliminating part of the variables. What happens when we pass from (Dl) to its dual? It turns out that we end up with a nontrivial (and instructive) equivalent reformulation of (Pr), namely, with the problem

$$
\begin{array}{lrcl}
\text{minimize} & \tau & & \\
\text{s.t.} & & & \\
(a) & \left(\begin{array}{c|ccc|c|ccc}
2\tau I_p + \mathcal{D}_l z + D_l & [q^l_{11}]^T & \cdots & [q^l_{1S}]^T & \cdots & [q^l_{n1}]^T & \cdots & [q^l_{nS}]^T \\ \hline
q^l_{11} & t_1 & & & & & & \\
\cdots & & \ddots & & & & & \\
q^l_{1S} & & & t_1 & & & & \\ \hline
\cdots & & & & \ddots & & & \\ \hline
q^l_{n1} & & & & & t_n & & \\
\cdots & & & & & & \ddots & \\
q^l_{nS} & & & & & & & t_n
\end{array}\right) & \succeq & 0,\ l = 1,...,K; \\
(b) & t_i & \succeq & 0,\ i = 1,...,n; \\
(c) & \underline{\rho}_i \leq \mathrm{Tr}(t_i) & \leq & \overline{\rho}_i,\ i = 1,...,n; \\
(d) & \sum_{i=1}^n \mathrm{Tr}(t_i) & \leq & w; \\
(e) & \sum_{i=1}^n \sum_{s=1}^S b_{is} q^l_{is} & = & \mathcal{E}_l z + E_l,\ l = 1,...,k; \\
(vi) & z & \geq & 0,
\end{array}
\tag{Pr$^+$}
$$

the design variables in the problem being symmetric $d \times d$ matrices t_i, $i = 1,...,n$, $d \times p$ matrices q^l_{is}, $l = 1,...,K, i = 1,...,n, s = 1,...,S$, real τ and $z \in \mathbf{R}^N$. Problem (Pr$^+$) is not the straightforward dual of (Dl); it is obtained from this dual by eliminating part of the variables. Instead of boring derivation of (Pr$^+$) via duality, we prefer to give a direct proof of equivalence between (Pr) and (Pr$^+$):

Proposition 4.8.4 *A collection $(\{t_i\}_{i=1}^n, z, \tau)$ is a feasible solution to* (Pr) *if and only if it can be extended by properly chosen $\{q^l_{is} \mid l = 1,...,K, i = 1,...,n, s = 1,...,S\}$ to a feasible solution to* (Pr$^+$).

Proof. "if" part: let a collection

$$(\{t_i\}_{i=1}^n, z, \tau, \{q^l_{is} \mid l = 1,...,K, i = 1,...,n, s = 1,...,S\})$$

be a feasible solution to (Pr$^+$); all we should prove is the validity of the LMI's (Pr.(a)). Let us fix $l \leq K$; we should prove that for every pair (x, y) of vectors of appropriate dimensions we have

$$x^T[2\tau I_p + \mathcal{D}_l z + D_l]x + 2x^T[E_i + \mathcal{E}_l z]^T y + y^T[\sum_{i=1}^n \sum_{s=1}^S b^T_{is} t_i b^T_{is}]y \geq 0. \tag{4.8.18}$$

Indeed, in view of (Pr$^+$.(e)) the left hand side of (4.8.18) is equal to

$$
\begin{array}{rcl}
x^T[2\tau I_p + \mathcal{D}_l z + D_l]x & & \\
+2x^T[\sum_{i=1}^n \sum_{s=1}^S b_{is} q^l_{is}]^T y + y^T[\sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b^T_{is}]y & = & x^T[2\tau I_p + \mathcal{D}_l z + D_l]x \\
& & +2\sum_{i=1}^n \sum_{s=1}^S x^T[q^l_{is}]^T y_{is} \\
& & +\sum_{i=1}^n \sum_{s=1}^S y^T_{is} t_i y_{is}, \\
& & y_{is} = b^T_{is} y.
\end{array}
$$

The resulting expression is nothing but the value of the quadratic form with the matrix from the left hand side of the corresponding LMI (Pr$^+$.(a)) at the vector comprised of x and $\{y_{is}\}_{i,s}$, and therefore it is nonnegative, as claimed.

"only if" part: let

$$(\{t_i\}_{i=1}^n, z, \tau)$$

be a feasible solution to (Pr). Let us fix l, $1 \leq l \leq K$, and let us set

$$f_l = \mathcal{E}_l z + E_l.$$

For every $x \in \mathbf{R}^p$ the quadratic form of $y \in \mathbf{R}^M$:

$$x^T[2\tau I_p + \mathcal{D}_l z + d_l] + 2x^T f_l^T y + y^T A(t) y \qquad \left[A(t) = \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T\right]$$

is nonnegative, i.e., the equation

$$A(t)y = f_l x$$

is solvable for every x; of course, we can choose its solution to be linear in x:

$$y = Y_l x;$$

note that then

$$A(t)Y_l x = f_l x \quad \forall x,$$

i.e.

$$A(t)Y_l = f_l.$$

Let us now set

$$[q_{is}^l]^T = Y_l^T b_{is} t_i; \tag{4.8.19}$$

then

$$\sum_{i=1}^n \sum_{s=1}^S b_{is} q_{is}^l = \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T Y_l = A(t) Y_l = f_l.$$

Recalling the definition of f_l, we see that extending $(\{t_i\}, z, \tau)$ by $\{q_{is}^l\}$ we ensure the validity of $(\mathrm{Pr}^+.(e))$. It remains to verify that the indicated extensions ensures the validity of LMI's $(\mathrm{Pr}^+.(a))$ as well. What we should verify is that for every collection $\{y_{is}\}$ of vectors of appropriate dimension and for every $x \in \mathbf{R}^p$ we have

$$F(x, \{y_{is}\}) \equiv x^T[2\tau I_p + \mathcal{D}_l z + D_l]x + 2x^T \sum_{i=1}^n \sum_{s=1}^S [q_{is}^l]^T y_{is} + \sum_{i=1}^n \sum_{s=1}^S y_{is}^T t_i y_{is} \geq 0. \tag{4.8.20}$$

Given x, let us set

$$y_{is}^* = -b_{is}^T Y_l x,$$

and let us prove that the collection $\{y_{is}^*\}$ minimizes $F(x, \cdot)$, which is immediate: $F(x, \cdot)$ is convex quadratic form, and its partial derivative w.r.t. y_{is} at the point $\{y_{is}^*\}$ is equal to (see (4.8.19))

$$2q_{is}^l x + 2t_i y_{is}^* = 2[t_i b_{is}^T Y_l x - t_i b_{is}^T Y_l x] = 0$$

for all i, s. It remains to note that

$$\begin{array}{rcl}
F(x, \{y_{is}^*\}) & = & x^T[2\tau I_p + \mathcal{D}_l z + D_l]x - 2x^T \sum_{i=1}^n \sum_{s=1}^S [q_{is}^l]^T b_{is}^T Y_l x \\
 & & + \sum_{i=1}^n \sum_{s=1}^S x^T Y_l^T b_{is} t_i b_{is}^T Y_l x \\
 & = & x^T[2\tau I_p + \mathcal{D}_l z + D_l]x - 2x^T [\sum_{i=1}^n \sum_{s=1}^S b_{is} q_{is}^l]^T Y_l x \\
 & & + x^T Y_l^T A(t) Y_l x \\
 & = & x^T[2\tau I_p + \mathcal{D}_l z + D_l]x - 2x^T[\mathcal{E}_l z + E_l]^T Y_l x + x^T Y_l^T A(t) Y_l x \\
 & & \text{[due to already proved } (\mathrm{Pr}^+.(e))] \\
 & = & (x^T; -x^T Y_l^T) \begin{pmatrix} \tau I_p + \mathcal{D}_l z + D_l & [\mathcal{E}_l z + E_l]^T \\ \mathcal{E}_l z + E_l & A(t) \end{pmatrix} \begin{pmatrix} x \\ -Y_l x \end{pmatrix} \\
 & \geq & 0 \\
 & & \text{[since } (\{t_i\}, z, \tau) \text{ is feasible for (Pr)]}
\end{array}$$

Thus, the minimum of $F(x, \{y_{is}\})$ in $\{y_{is}\}$ is nonnegative, and therefore (4.8.20) indeed is valid. ■

4.8.6 Explicit forms of the Standard Truss and Shape problems

Let us list the explicit forms of problems (Pr), (Dl), (Pr$^+$) for the standard cases of the multi-load and robust Static Truss/Shape Design.

Multi-Load Static Truss Design: Here

$$\mathcal{V} = \{v \in \mathbf{R}^m \mid Rv \le r\} \quad [\dim(r) = q]; \qquad \mathcal{F} = \{f_1, ..., f_k\}.$$

The settings are:

- (Pr):

$$\begin{array}{rcl}
\tau & \to & \min \\
\begin{pmatrix} 2\tau - 2r^T\mu_l & -f_l^T + \mu_l^T R \\ -f_l + R^T\mu_l & \sum_{i=1}^n b_i b_i^T t_i \end{pmatrix} & \succeq & 0, \ l = 1, ..., k; \\
\underline{\rho}_i \le t_i & \le & \overline{\rho}_i, i = 1, ..., n; \\
\sum_{i=1}^n t_i & \le & w; \\
\mu_l & \ge & 0, \ l = 1, ..., k; \\
{[\tau, t_i, \mu_l \in \mathbf{R}^q]} & &
\end{array}$$

- (Dl):

$$\begin{array}{rcl}
-2\sum_{l=1}^k f_l^T v_l + \sum_{i=1}^n [\overline{\rho}_i \sigma_i^+ - \underline{\rho}_i \sigma_i^-] + w\gamma & \to & \min \\
\left(\begin{array}{ccc|c} \alpha_1 & & & b_i^T v_1 \\ & \ddots & & \cdots \\ & & \alpha_k & b_i^T v_k \\ \hline b_i^T v_1 & \cdots & b_i^T v_k & \gamma + \sigma_i^+ - \sigma_i^- \end{array}\right) & \succeq & 0, \ i = 1, ..., n; \\
\sigma_i^{\pm} & \ge & 0, \ i = 1, ..., n; \\
\gamma & \ge & 0; \\
Rv_l & \le & \alpha_l r, \ l = 1, ..., k; \\
2\sum_{l=1}^k \alpha_l & = & 1. \\
{[\alpha_l, \sigma_i^{\pm}, \gamma \in \mathbf{R}, v_l \in \mathbf{R}^m]} & &
\end{array}$$

- (Pr$^+$):

$$\begin{array}{rcl}
\tau & \to & \min \\
\left(\begin{array}{c|ccc} 2\tau - 2r^T\mu_l & q_1^l & \cdots & q_n^l \\ \hline q_1^l & t_1 & & \\ \cdots & & \ddots & \\ q_n^l & & & t_n \end{array}\right) & \succeq & 0, \ l = 1, ..., k; \\
\underline{\rho}_i \le t_i & \le & \overline{\rho}_i, \ i = 1, ..., n; \\
\sum_{i=1}^n t_i & \le & w; \\
\sum_{i=1}^n q_i^l b_i & = & f_l - R^T\mu_l, \ l = 1, ..., k; \\
\mu_l & \ge & 0, \ l = 1, ..., k; \\
{[\tau, t_i, q_i^l \in \mathbf{R}, \mu_l \in \mathbf{R}^q]} & &
\end{array}$$

Robust Obstacle-Free Static Truss Design. Here

$$\mathcal{V} = \mathbf{R}^m; \quad \mathcal{F} = \{f = Qu \mid u^T u \le 1\} \quad [Q \in \mathbf{M}^{mk}].$$

The settings are:

- (Pr):

$$\begin{array}{rcl}
\tau & \to & \min \\
\begin{pmatrix} 2\tau I_k & Q^T \\ Q & \sum_{i=1}^n b_i b_i^T t_i \end{pmatrix} & \succeq & 0; \\
\underline{\rho}_i \le t_i & \le & \overline{\rho}_i, i = 1, ..., n; \\
\sum_{i=1}^n t_i & \le & w. \\
{[\tau, t_i \in \mathbf{R}]} & &
\end{array}$$

- (Dl):

$$\begin{array}{rcl}
2\,\mathrm{Tr}(Q^T V) + \sum_{i=1}^n [\overline{\rho}_i \sigma_i^+ - \underline{\rho}_i \sigma_i^-] + w\gamma & \to & \min \\
\begin{pmatrix} \alpha & V^T b_i \\ b_i^T V & \gamma + \sigma_i^+ - \sigma_i^- \end{pmatrix} & \succeq & 0, \; i = 1, ..., n; \\
\sigma_i^\pm & \ge & 0, \; i = 1, ..., n; \\
\gamma & \ge & 0; \\
2\,\mathrm{Tr}(\alpha) & = & 1. \\
{[\alpha \in \mathbf{S}^k, \sigma_i^\pm, \gamma \in \mathbf{R}, V \in \mathbf{M}^{mk}]} & &
\end{array}$$

- (Pr$^+$):

$$\begin{array}{rcl}
\tau & \to & \min \\
\left(\begin{array}{c|ccc} 2\tau I_k & q_1^T & \cdots & q_n^T \\ \hline q_1 & t_1 & & \\ \cdots & & \ddots & \\ q_n & & & t_n \end{array}\right) & \succeq & 0; \\
\underline{\rho}_i \le t_i & \le & \overline{\rho}_i, \; i = 1, ..., n; \\
\sum_{i=1}^n t_i & \le & w; \\
\sum_{i=1}^N b_i q_i & = & Q; \\
{[t_i \in \mathbf{R}, q_i^T \in \mathbf{R}^k]} & &
\end{array}$$

Multi-Load Static Shape Design. Here

$$\begin{array}{rcl}
t_i & \in & \mathbf{S}^d, \; d = \begin{cases} 3, & \text{planar shape} \\ 6, & \text{spatial shape} \end{cases}, \\
\mathcal{F} & = & \{f_1, ..., f_k\}, \\
\mathcal{V} & = & \{v \in \mathbf{R}^m \mid Rv \le r\} \quad [\dim\,(r) = q].
\end{array}$$

The settings are

- (Pr):

$$\begin{array}{rcl}
\tau & \to & \min \\
\begin{pmatrix} 2\tau - 2r^T \mu_l & -f_l^T + \mu_l^T R \\ -f_l + R^T \mu_l & \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \end{pmatrix} & \succeq & 0, \; l = 1, ..., k; \\
t_i & \succeq & 0, \; i = 1, ..., n; \\
\underline{\rho}_i \le \mathrm{Tr}(t_i) & \le & \overline{\rho}_i, i = 1, ..., n; \\
\sum_{i=1}^n \mathrm{Tr}(t_i) & \le & w; \\
\mu_l & \ge & 0, \; l = 1, ..., k; \\
{[\tau \in \mathbf{R}, t_i \in \mathbf{S}^d, \mu_l \in \mathbf{R}^q]} & &
\end{array}$$

- (Dl):

$$\begin{array}{rcl}
-2\sum_{l=1}^k f_l^T v_l + \sum_{i=1}^n [\overline{\rho}_i \sigma_i^+ - \underline{\rho}_i \sigma_i^-] + w\gamma & \to & \min \\
\left(\begin{array}{ccc|c|ccc|c}
\alpha_1 & & & & & & & v_1^T b_{i1} \\
& \ddots & & & & & & \cdots \\
& & \alpha_1 & & & & & v_1^T b_{iS} \\ \hline
& & & \ddots & & & & \cdots \\ \hline
& & & & \alpha_k & & & v_k^T b_{i1} \\
& & & & & \ddots & & \cdots \\
& & & & & & \alpha_k & v_k^T b_{iS} \\
b_{i1}^T v_1 & \cdots & b_{iS}^T v_1 & \cdots & b_{i1}^T v_k & \cdots & b_{iS}^T v_k & (\gamma + \sigma_i^+ - \sigma_i^-) I_d
\end{array}\right) & \succeq & 0,\ i = 1, ..., n; \\
\sigma_i^\pm & \geq & 0,\ i = 1, ..., n; \\
\gamma & \geq & 0; \\
R v_l & \leq & \alpha_l r,\ l = 1, ..., k; \\
2\sum_{l=1}^k \alpha_l & = & 1. \\
{}[\alpha_l, \sigma_i^\pm, \gamma \in \mathbf{R}, v_l \in \mathbf{R}^m] & &
\end{array}$$

- (Pr$^+$):

$$\begin{array}{rcl}
\tau & \to & \min \\
\left(\begin{array}{c|ccc|c|ccc}
2\tau - 2r^T\mu_l & [q_{11}^l]^T & \cdots & [q_{1S}^l]^T & \cdots & [q_{n1}^l]^T & \cdots & [q_{nS}^l]^T \\ \hline
q_{11}^l & t_1 & & & & & & \\
\cdots & & \ddots & & & & & \\
q_{1S}^l & & & t_1 & & & & \\ \hline
\cdots & & & & \ddots & & & \\ \hline
q_{n1}^l & & & & & t_n & & \\
\cdots & & & & & & \ddots & \\
q_{nS}^l & & & & & & & t_n
\end{array}\right) & \succeq & 0,\ l = 1, ..., k; \\
\underline{\rho}_i \leq \mathrm{Tr}(t_i) & \leq & \overline{\rho}_i,\ i = 1, ..., n; \\
\sum_{i=1}^n \mathrm{Tr}(t_i) & \leq & w; \\
\sum_{i=1}^n \sum_{s=1}^S b_{is} q_{is}^l & = & f_l - R^T \mu_l,\ l = 1, ..., k; \\
\mu_l & \geq & 0,\ l = 1, ..., k; \\
{}[\tau \in \mathbf{R}, t_i \in \mathbf{S}^d, q_{is}^l \in \mathbf{R}^d, \mu_l \in \mathbf{R}^q] & &
\end{array}$$

Robust Obstacle-Free Static Shape Design. Here

$$\begin{array}{rcl}
t_i & \in & \mathbf{S}^d,\ d = \begin{cases} 3, & \text{planar shape} \\ 6, & \text{spatial shape} \end{cases}, \\
\mathcal{F} & = & \{f = Qu \mid u^T u \leq 1\} \qquad [Q \in \mathbf{M}^{mk}], \\
\mathcal{V} & = & \mathbf{R}^m.
\end{array}$$

The settings are

- (Pr):

$$\begin{array}{rcl}
2\tau & \to & \min \\
\begin{pmatrix} \tau I_k & Q^T \\ Q & \sum_{i=1}^n \sum_{s=1}^S b_{is} t_i b_{is}^T \end{pmatrix} & \succeq & 0; \\
t_i & \succeq & 0,\ i = 1, ..., n; \\
\underline{\rho}_i \leq \mathrm{Tr}(t_i) & \leq & \overline{\rho}_i, i = 1, ..., n; \\
\sum_{i=1}^n \mathrm{Tr}(t_i) & \leq & w. \\
{}[\tau \mathbf{R}, t_i \in \mathbf{S}^d & &
\end{array}$$

- (Dl):

$$
\begin{array}{rcl}
2\,\mathrm{Tr}(Q^TV)+\sum_{i=1}^n[\overline{\rho}_i\sigma_i^+-\underline{\rho}_i\sigma_i^-]+w\gamma & \to & \min \\
\left(\begin{array}{ccc|c}
\alpha & & & V^Tb_{i1} \\
 & \ddots & & \cdots \\
 & & \alpha & V^Tb_{iS} \\
\hline
b_{i1}^TV & \cdots & b_{iS}^TV & (\gamma+\sigma_i^+-\sigma_i^-)I_d
\end{array}\right) & \succeq & 0,\ i=1,...,n; \\
\sigma_i^\pm & \geq & 0,\ i=1,...,n; \\
\gamma & \geq & 0; \\
2\,\mathrm{Tr}(\alpha) & = & 1. \\
{}[\alpha\in\mathbf{S}^k,\sigma_i^\pm,\gamma\in\mathbf{R},V\in\mathbf{M}^{mk}] & &
\end{array}
$$

- (Pr$^+$):

$$
\begin{array}{rcl}
\tau & \to & \min \\
\left(\begin{array}{c|ccc|c|ccc}
2\tau I_k & [q_{11}]^T & \cdots & [q_{1S}]^T & \cdots & [q_{n1}]^T & \cdots & [q_{nS}]^T \\
\hline
q_{11} & t_1 & & & & & & \\
\cdots & & \ddots & & & & & \\
q_{1S} & & & t_1 & & & & \\
\hline
\cdots & & & & \ddots & & & \\
\hline
q_{n1} & & & & & t_n & & \\
\cdots & & & & & & \ddots & \\
q_{nS} & & & & & & & t_n
\end{array}\right) & \succeq & 0; \\
\underline{\rho}_i\leq\mathrm{Tr}(t_i) & \leq & \overline{\rho}_i,\ i=1,...,n; \\
\sum_{i=1}^n\mathrm{Tr}(t_i) & \leq & w; \\
\sum_{i=1}^n\sum_{s=1}^S b_{is}q_{is} & = & Q; \\
{}[\tau\in\mathbf{R},t_i\in\mathbf{S}^d,q_{is}\in\mathbf{M}^{dk}] & &
\end{array}
$$

4.9 Applications, VII: Extremal ellipsoids

In our course we already have met, on different occasions, with the notion of an ellipsoid – a set E in $\mathbf{R}^n$ which can be represented as the image of the unit Euclidean ball under an affine mapping:

$$E=\{x=Au+c\mid u^Tu\leq 1\}\quad[A\in\mathbf{M}^{nq}] \tag{Ell}$$

Ellipsoids are very convenient mathematical entities:

- it is easy to specify an ellipsoid – just to point out the corresponding matrix A and vector c;
- the family of ellipsoids is closed with respect to affine transformations: the image of an ellipsoid under an affine mapping again is an ellipsoid;
- there are many operations, like minimization of a linear form, computation of volume, etc., which are easy to carry out when the set in question is an ellipsoid, and is difficult to carry out for more general convex sets.

By the indicated reasons, ellipsoids play important role in different areas of applied mathematics; in particular, people use ellipsoids to approximate more complicated sets. Just as a simple motivating example, consider a discrete-time linear time invariant controlled system:

$$
\begin{array}{rcl}
x(t+1) & = & Ax(t)+Bu(t),\ t=0,1,... \\
x(0) & = & 0
\end{array}
$$

and assume that the control is norm-bounded:

$$\| u(t) \|_2 \leq 1 \quad \forall t.$$

The question is what is the set X_T of all states "reachable in a given time T", i.e., the set of all possible values of $x(T)$. We can easily write down the answer:

$$X_T = \{x = Bu_{T-1} + ABu_{T-2} + A^2Bu_{T-3} + ... + A^{T-1}Bu_0 \mid \| u_t \|_2 \leq 1, t = 0, ..., T-1\},$$

but this answer is not "explicit"; just to check whether a given vector x belongs to X_T requires to solve a nontrivial conic quadratic problem, the complexity of the problem being the larger the larger is T. In fact the geometry of X_T may be very complicated, so that there is no possibility to get a "tractable" explicit description of the set. This is why in many applications it makes sense to use "simple" – ellipsoidal - approximations of X_T; as we shall see, approximations of this type can be computed in a recurrent and computationally efficient fashion.

It turns out that the natural framework for different problems of the "best possible" approximation of convex sets by ellipsoids is given by semidefinite programming. In this section we intend to consider a number of basic problems of this type.

Preliminaries on ellipsoids. According to our definition, an ellipsoid in $\mathbf{R}^n$ is the image of the unit Euclidean ball in certain $\mathbf{R}^q$ under an affine mapping; e.g., for us a segment in $\mathbf{R}^{100}$ is an ellipsoid; indeed, it is the image of one-dimensional Euclidean ball under affine mapping. In contrast to this, in geometry an ellipsoid in $\mathbf{R}^n$ is usually defined as the image of the n-dimensional unit Euclidean ball under an invertible affine mapping, i.e., as the set of the form (Ell) with additional requirements that $q = n$, i.e., that the matrix A is square, and that it is nonsingular. In order to avoid confusion, let us call these "true" ellipsoids *full-dimensional*. Note that a full-dimensional ellipsoid E admits two nice representations:

- First, E can be represented in the form (Ell) with *positive definite symmetric* A:

$$E = \{x = Au + c \mid u^T u \leq 1\} \quad [A \in \mathbf{S}^n_{++}] \tag{4.9.1}$$

  Indeed, it is clear that if a matrix A represents, via (Ell), a given ellipsoid E, the matrix AU, U being an orthogonal $n \times n$ matrix, represents E as well. It is known from Linear Algebra that by multiplying a nonsingular square matrix from the right by a properly chosen orthogonal matrix, we get a positive definite symmetric matrix, so that we always can parameterize a full-dimensional ellipsoid by a positive definite symmetric A.

- Second, E can be given by a strictly convex quadratic inequality:

$$E = \{x \mid (x-c)^T D(x-c) \leq 1\} \quad [D \in \mathbf{S}^n_{++}]. \tag{4.9.2}$$

  Indeed, one may take $D = A^{-2}$, where A is the matrix from the representation (4.9.1).

In the sequel we deal a lot with *volumes* of full-dimensional ellipsoids. Since an invertible affine transformation $x \mapsto Ax + b : \mathbf{R}^n \to \mathbf{R}^n$ multiplies the volumes of n-dimensional domains by $|\,\mathrm{Det}\, A|$, the volume of a full-dimensional ellipsoid E given by (4.9.1) is $\kappa_n \mathrm{Det}\, A$, where κ_n is the volume of the n-dimensional unit Euclidean ball. In order to avoid meaningless constant factors, it makes sense to pass from the usual n-dimensional volume $\mathrm{mes}_n(G)$ of a domain G to its *normalized volume*

$$\mathrm{Vol}(G) = \kappa_n^{-1} \mathrm{mes}_n(G),$$

i.e., to choose, as the unit of volume, the volume of the unit ball rather than the one of the cube with unit edges. From now on, speaking about volumes of n-dimensional domains, we always mean their normalized volume (and omit the word "normalized"). With this convention, the volume of a full-dimensional ellipsoid E given by (4.9.1) is just

$$\mathrm{Vol}(E) = \mathrm{Det}\ A,$$

while for an ellipsoid given by (4.9.1) the volume is

$$\mathrm{Vol}(E) = [\mathrm{Det}\ D]^{-1/2}\,.$$

Outer and inner ellipsoidal approximations. It was already mentioned that our current goal is to realize how to solve basic problems of "the best" ellipsoidal approximation E of a given set S. There are two types of these problems:

- *Outer approximation*, where we are looking for the "smallest" ellipsoid E *containing* the set S;

- *Inner approximation*, where we are looking for the "largest" ellipsoid E *contained* in the set S.

In both these problems, a natural way to say when one ellipsoid is "smaller" than another one is to compare the volumes of the ellipsoids. The main advantage of this viewpoint is that it results in *affine-invariant* constructions: an invertible affine transformation multiplies volumes of all domains by the same constant and therefore preserves ratios of volumes of the domains.

Thus, what we are interested in are the *largest volume ellipsoid(s) contained in a given set* S and the *smallest volume ellipsoid(s) containing a given set* S. In fact these *extremal ellipsoids* are unique, provided that S is a *solid* – a closed and bounded convex set with a nonempty interior, and are not too bad approximations of the set:

Theorem 4.9.1 [Löwner – Fritz John] *Let $S \subset \mathbf{R}^n$ be a solid. Then*

(i) *There exists and is uniquely defined the largest volume full-dimensional ellipsoid $E_{\rm in}$ contained in S. The concentric to $E_{\rm in}$ n times larger (in linear sizes) ellipsoid contains S; if S is central-symmetric, then already $\sqrt{n}$ times larger than $E_{\rm in}$ concentric to $E_{\rm in}$ ellipsoid contains S.*

(ii) *There exists and is uniquely defined the smallest volume full-dimensional ellipsoid $E_{\rm out}$ containing S. The concentric to $E_{\rm out}$ n times smaller (in linear sizes) ellipsoid is contained in S; if S is central-symmetric, then already $\sqrt{n}$ times smaller than $E_{\rm out}$ concentric to $E_{\rm out}$ ellipsoid is contained in S.*

The proof is the subject of Exercise 4.32.

The existence of th extremal ellipsoids is, of course, a good news; but how to compute these ellipsoids? The possibility to compute efficiently (nearly) extremal ellipsoids heavily depends on the description of S. Let us start with two simple examples.

Inner ellipsoidal approximation of a polytope. Let S be a polyhedral set *given by a number of linear equalities:*

$$S = \{x \in \mathbf{R}^n \mid a_i^T x \leq b_i,\ i = 1, ..., m\}.$$

Proposition 4.9.1 *Assume that S is a full-dimensional polytope (i.e., is bounded and possesses a nonempty interior). Then the largest volume ellipsoid contained in S is*

$$E = \{x = Z_* u + z_* \mid u^T u \le 1\},$$

where Z_*, z_* are given by an optimal solution to the following semidefinite program:

$$\begin{array}{lrcl} & t & \to & \max \\ s.t. & & & \\ (a) & t & \le & (\mathrm{Det}\, Z)^{1/(2n+1)}, \\ (b) & Z & \succeq & 0, \\ (c) & \| Z a_i \|_2 & \le & b_i - a_i^T z, \ i = 1, ..., m, \end{array} \tag{In}$$

with the design variables $Z \in \mathbf{S}^n, z \in \mathbf{R}^n, t \in \mathbf{R}$.

Note that (In) *indeed is a semidefinite program: both* (In.(a)) *and* (In.(c)) *can be represented by LMI's, see items* 15d *and* 1-14 *in Section 4.2.*

Proof. Indeed, an ellipsoid (4.9.1) is contained in S if and only if

$$a_i^T (Au + c) \le b_i \quad \forall u : u^T u \le 1,$$

or, which is the same, if and only if

$$\| A a_i \|_2 + a_i^T c = \max_{u : u^T u \le 1} [a_i^T A u + a_i^T c] \le b_i.$$

Thus, (In.$(b-c)$) just express the fact that the ellipsoid $\{x = Zu + z \mid u^T u \le 1\}$ is contained in S, so that (In) is nothing but the problem of maximizing (a positive power of) the volume of an ellipsoid over ellipsoids contained in S. ■

We see that if S is a polytope *given by a set of linear inequalities*, then the problem of the best inner ellipsoidal approximation of S is an explicit semidefinite program and as such can be efficiently solved. In contrast to this, if S is a polytope *given as a convex hull of finite set:*

$$S = \mathrm{Conv}\{x_1, ..., x_m\},$$

then the problem of the best inner ellipsoidal approximation of S is "computationally intractable" – in this case, it is difficult just to check whether a given candidate ellipsoid is contained in S.

Outer ellipsoidal approximation of a finite set. Let S be a polyhedral set *given as a convex hull of a finite set of points:*

$$S = \mathrm{Conv}\{x_1, ..., x_m\}.$$

Proposition 4.9.2 *Assume that S is a full-dimensional polytope (i.e., possesses a nonempty interior). Then the smallest volume ellipsoid containing S is*

$$E = \{x \mid (x - c_*)^T D_* (x - c_*) \le 1\},$$

where c_*, D_* are given by an optimal solution (t_*, Z_*, z_*, s_*) to the semidefinite program

$$\begin{array}{lrcl} & t & \to & \max \\ s.t. & & & \\ (a) & t & \le & (\mathrm{Det}\ Z)^{1/(2n+1)}, \\ (b) & Z & \succeq & 0, \\ (c) & \begin{pmatrix} s & z^T \\ z & Z \end{pmatrix} & \succeq & 0, \\ (d) & x_i^T Z x_i - 2x_i^T z + s & \le & 1,\ i = 1, ..., m, \end{array} \qquad \text{(Out)}$$

with the design variables $Z \in \mathbf{S}^n, z \in \mathbf{R}^n, t, s \in \mathbf{R}$ via the relations

$$D_* = Z_*; c_* = Z_*^{-1} z_*.$$

Note that (Out) *indeed is a semidefinite program, cf. Proposition 4.9.1.*

Proof. Indeed, let us pass in the description (4.9.2) from the "parameters" D, c to the parameters $Z = D, z = Dc$, thus coming to the representation

$$E = \{x \mid, x^T Z x - 2x^T z + z^T Z^{-1} z \le 1\}. \qquad (!)$$

The ellipsoid of the latter type contains the points $x_1, ..., x_m$ if and only if

$$x_i^T Z x_i - 2x_i^T z + z^T Z^{-1} z \le 1,\ i = 1, ..., m,$$

or, which is the same, if and only if there exists $s \ge z^T Z^{-1} z$ such that

$$x_i^T Z x_i - 2x_i^T z + s \le 1,\ i = 1, ..., m.$$

Recalling Lemma on the Schur Complement, we see that the constraints (Out.$(b-d)$) say exactly that the ellipsoid (!) contains the points $x_1, ..., x_m$. Since the volume of such an ellipsoid is $(\mathrm{Det}\ Z)^{-1/2}$, (Out) is the problem of maximizing a negative power of the volume of an ellipsoid containing the finite set $\{x_1, ..., x_m\}$, i.e., the problem of finding the smallest volume ellipsoid containing this finite set. It remains to note that an ellipsoid is convex, so that it is exactly the same – to say that it contains a finite set $\{x_1, ..., x_m\}$ and to say that it contains the convex hull of this finite set. ∎

We see that if S is a polytope *given as a convex hull of a finite set*, then the problem of the best outer ellipsoidal approximation of S is an explicit semidefinite program and as such can be efficiently solved. In contrast to this, if S is a polytope *given by a list of inequality constraints*, then the problem of the best outer ellipsoidal approximation of S is "computationally intractable" – in this case, it is difficult just to check whether a given candidate ellipsoid contains S.

4.9.1 Ellipsoidal approximations of unions/intersections of ellipsoids

Speaking informally, Proposition 4.9.1 deals with inner ellipsoidal approximation of the intersection of "degenerate" ellipsoids, namely, half-spaces (a half-space is just a very large Euclidean ball!) Similarly, Proposition 4.9.2 deals with the outer ellipsoidal approximation of the union of degenerate ellipsoids, namely, points (a point is just a ball of zero radius!). We are about to demonstrate that when passing from "degenerate" ellipsoids to the "normal" ones, we still have a possibility to reduce the corresponding approximation problems to explicit semidefinite programs. The key observation here is as follows:

Proposition 4.9.3 *An ellipsoid*

$$E = \{x = Au + z \mid u^T u \leq 1\} \quad [A \in \mathbf{M}^{nq}]$$

is contained in the full-dimensional ellipsoid

$$W = \{x \mid (x-c)^T B^T B(x-c) \leq 1\} \quad [B \in \mathbf{M}^{nn}, \mathrm{Det}\, B \neq 0]$$

if and only if there exists λ *such that*

$$\begin{pmatrix} I_n & B(z-c) & BA \\ (z-c)^T B^T & 1-\lambda & \\ A^T B^T & & \lambda I_q \end{pmatrix} \succeq 0 \tag{4.9.3}$$

as well as if and only if there exists λ *such that*

$$\begin{pmatrix} B^{-1}(B^{-1})^T & z-c & A \\ (z-c)^T & 1-\lambda & \\ A^T & & \lambda I_q \end{pmatrix} \succeq 0 \tag{4.9.4}$$

Proof. We clearly have

$$\begin{array}{c} E \subset W \\ \Updownarrow \\ u^T u \leq 1 \Rightarrow (Au + z - c)^T B^T B(Au + z - c) \leq 1 \\ \Updownarrow \\ u^T u \leq t^2 \Rightarrow (Au + t(z-c))^T B^T B(Au + t(z-c)) \leq t^2 \\ \Updownarrow \mathcal{S}\text{-Lemma} \\ \exists \lambda \geq 0 : [t^2 - (Au + t(z-c))^T B^T B(Au + t(z-c))] - \lambda[t^2 - u^T u] \geq 0 \quad \forall (u,t) \\ \Updownarrow \\ \exists \lambda \geq 0 : \begin{pmatrix} 1 - \lambda - (z-c)^T B^T B(z-c) & -(z-c)^T B^T BA \\ A^T B^T B(z-c) & \lambda I_q - A^T B^T BA \end{pmatrix} \succeq 0 \\ \Updownarrow \\ \exists \lambda \geq 0 : \begin{pmatrix} 1-\lambda & \\ & \lambda I_q \end{pmatrix} - \begin{pmatrix} (z-c)^T B^T \\ A^T B^T \end{pmatrix} \begin{pmatrix} B(z-c) & BA \end{pmatrix} \succeq 0 \end{array}$$

Now note that in view of Lemma on the Schur Complement the matrix $\begin{pmatrix} 1-\lambda & \\ & \lambda I_q \end{pmatrix} - \begin{pmatrix} (z-c)^T B^T \\ A^T B^T \end{pmatrix} \begin{pmatrix} B(z-c) & BA \end{pmatrix}$ is positive semidefinite if and only if the matrix in (4.9.3) is so. Thus, $E \subset W$ if and only if there exists a nonnegative λ such that the matrix in (4.9.3), let it be called $P(\lambda)$, is positive semidefinite. Since the latter matrix can be positive semidefinite only when $\lambda \geq 0$, we have proved the first statement of the proposition. To prove the second statement, note that the matrix in (4.9.4), let it be called $Q(\lambda)$, is closely related to $P(\lambda)$:

$$Q(\lambda) = SP(\lambda)S^T, \quad S = \begin{pmatrix} B^{-1} & & \\ & 1 & \\ & & I_q \end{pmatrix} \succ 0,$$

so that $Q(\lambda)$ is positive semidefinite if and only if $P(\lambda)$ is so. ■

Here are the consequences of Proposition 4.9.3.

Inner ellipsoidal approximation of an intersection of full-dimensional ellipsoids. Let

$$W_i = \{x \mid (x - c_i)^T B_i^2 (x - c_i) \le 1\} \quad [B_i \in \mathbf{S}^n_{++}],$$

$i = 1, ..., m$, be given full-dimensional ellipsoids in $\mathbf{R}^n$; assume that the intersection W of these ellipsoids possesses a nonempty interior. Then the problem of the best inner ellipsoidal approximation of W is an explicit semidefinite program, namely, the program

$$\begin{array}{rcl} & t & \to & \max \\ \text{s.t.} & & & \\ & t & \le & (\mathrm{Det}\, Z)^{1/(2n+1)}, \\ \begin{pmatrix} I_n & B_i(z - c_i) & B_i Z \\ (z - c_i)^T B_i & 1 - \lambda_i & \\ Z B_i & & \lambda_i I_q \end{pmatrix} & & \succeq & 0, \; i = 1, ..., m, \\ & Z & \succeq & 0 \end{array} \qquad \text{(InEll)}$$

with the design variables $Z \in \mathbf{S}^n$, $z \in \mathbf{R}^n$, $\lambda_i, t \in \mathbf{R}$. The largest ellipsoid contained in $W = \cap_{i=1}^m W_i$ is given by an optimal solution $Z_*, z_*, t_*, \{\lambda_i^*\})$ of (InEll) via the relation

$$E = \{x = Z_* u + z_* \mid u^T u \le 1\}.$$

Indeed, by Proposition 4.9.3 the LMI's

$$\begin{pmatrix} I_n & B_i(z - c_i) & B_i Z \\ (z - c_i)^T B_i & 1 - \lambda_i & \\ Z B_i & & \lambda_i I_q \end{pmatrix} \succeq 0, \; i = 1, ..., m$$

express the fact that the ellipsoid $\{x = Zu + z \mid u^T u \le 1\}$ with $Z \succeq 0$ is contained in every one of the ellipsoids W_i, i.e., is contained in the intersection W of these ellipsoids. Consequently, (InEll) is exactly the problem of maximizing (a positive power of) the volume of an ellipsoid over the ellipsoids contained in W.

Outer ellipsoidal approximation of a union of ellipsoids. Let

$$W_i = \{x = A_i u + r_i \mid u^T u \le 1\} \quad [A_i \in \mathbf{M}^{nk_i}],$$

$i = 1, ..., m$, be given ellipsoids in $\mathbf{R}^n$; assume that the convex hull W of the union of these ellipsoids possesses a nonempty interior. Then the problem of the best outer ellipsoidal approximation of W is an explicit semidefinite program, namely, the program

$$\begin{array}{rcl} & t & \to & \max \\ \text{s.t.} & & & \\ & t & \le & (\mathrm{Det}\, Z)^{1/(2n+1)}, \\ \begin{pmatrix} I_n & Zc_i - z & ZA_i \\ (Zc_i - z)^T & 1 - \lambda_i & \\ A_i^T Z & & \lambda_i I_{k_i} \end{pmatrix} & & \succeq & 0, \; i = 1, ..., m, \\ & Z & \succeq & 0 \end{array} \qquad \text{(OutEll)}$$

with the design variables $Z \in \mathbf{S}^n$, $z \in \mathbf{R}^n$, $\lambda_i, t \in \mathbf{R}$. The smallest ellipsoid containing $W = \mathrm{Conv}(\cup_{i=1}^m W_i)$ is given by an optimal solution $Z_*, z_*, t_*, \{\lambda_i^*\})$ of (OutEll) via the relation

$$E = \{x \mid (x - c_*) Z_*^2 (x - c_*) \le 1\}, \quad c_* = Z_*^{-1} z_*.$$

Indeed, by Proposition 4.9.3 for $Z \succ 0$ the LMI's

$$\begin{pmatrix} I_n & Zc_i - z & ZA_i \\ (Zc_i - z)^T & 1 - \lambda_i & \\ A_i^T Z & & \lambda_i I_{k_i} \end{pmatrix} \succeq 0, \; i = 1, ..., m$$

express the fact that the ellipsoid $E = \{x \mid (x - Z^{-1}z)^T Z^2 (x - Z^{-1}z) \leq 1\}$ contains every one of the ellipsoids W_i, i.e., contains the convex hull W of the union of these ellipsoids. The volume of the ellipsoid E is $(\mathrm{Det}\; Z)^{-1}$; consequently, (OutEll) is exactly the problem of maximizing a *negative* power (i.e., of minimizing a positive power) of the volume of an ellipsoid over the ellipsoids containing W.

4.9.2 Approximating sums of ellipsoids

Let us come back to our motivating example, where we were interested to build ellipsoidal approximation of the set X_T of all states $x(T)$ where a given discrete time invariant linear system

$$\begin{array}{rcl} x(t+1) & = & Ax(t) + Bu(t), \; t = 0, ..., T-1 \\ x(0) & = & 0 \end{array}$$

can be driven in time T by a control $u(\cdot)$ satisfying the norm bound

$$\| u(t) \|_2 \leq 1, \; t = 0, ..., T-1.$$

How could we build such an approximation recursively? Let X_t be the set of all states where the system can be driven in time $t \leq T$, and assume that we have already built inner and outer ellipsoidal approximations E^t_{in} and E^t_{out} of the set X_t:

$$E^t_{\mathrm{in}} \subset X_t \subset E^t_{\mathrm{out}}.$$

Let also

$$E = \{x = Bu \mid u^T u \leq 1\}.$$

Then the set

$$F^{t+1}_{\mathrm{in}} = AE^t_{\mathrm{in}} + E \equiv \{x = Ay + z \mid y \in E^t_{\mathrm{in}}, z \in E\}$$

clearly is contained in X_{t+1}, so that a natural recurrent way to define an inner ellipsoidal approximation of X_{t+1} is to take as E^{t+1}_{in} the largest volume ellipsoid contained in F^{t+1}_{in}. Similarly, the set

$$F^{t+1}_{\mathrm{out}} = AE^t_{\mathrm{out}} + E \equiv \{x = Ay + z \mid y \in E^t_{\mathrm{out}}, z \in E\}$$

clearly covers X_{t+1}, and the natural recurrent way to define an outer ellipsoidal approximation of X_{t+1} is to take as E^{t+1}_{out} the smallest volume ellipsoid containing F^{t+1}_{out}.

Note that the sets F^{t+1}_{in} and F^{t+1}_{out} are of the same structure: each of them is the *arithmetic sum* $\{x = v + w \mid v \in V, w \in W\}$ of two ellipsoids V and W. Thus, we come to the problem as follows: *Given two ellipsoids W, V, find the best inner and outer ellipsoidal approximations of their arithmetic sum $W + V$.* In fact, it makes sense to consider a little bit more general problem:

Given m ellipsoids $W_1, ..., W_m$ in $\mathbf{R}^n$, find the best inner and outer ellipsoidal approximations of the arithmetic sum

$$W = \{x = w_1 + w_1 + ... + w_m \mid w_i \in W_i, \; i = 1, ..., m\}$$

of the ellipsoids $W_1, ..., W_m$.

In fact, we have posed two different problems: the one of inner approximation of W (let this problem be called (I)) and the other one, let it be called (O), of outer approximation. It turns out that both these problems are very difficult (at least when m is not once for ever fixed). There exist, however, "computationally tractable" *approximations* of both (I) and (O) we are about to consider.

In considerations to follow we assume, for the sake of simplicity, that the ellipsoids $W_1, ..., W_m$ are full-dimensional (which is not a severe restriction – a "flat" ellipsoid can be easily approximated by a "nearly flat" full-dimensional ellipsoid). Besides this, we may assume without loss of generality that all our ellipsoids W_i are centered at the origin. Indeed, we have $W_i = c_i + V_i$, where c_i is the center of W_i and $V_i = W_i - c_i$ is centered at the origin; consequently,

$$W_1 + ... + W_m = (c_1 + ... + c_m) + (V_1 + ... + V_m),$$

so that the problems (I) and (O) for the ellipsoids $W_1, ..., W_m$ can be straightforwardly reduced to similar problems for the centered at the origin ellipsoids $V_1, ..., V_m$.

Preliminaries: the polar of a convex set. To proceed, we need to recall an extremely useful geometric notion – the one of the polar of a convex set. Let X be a closed convex and containing origin set in $\mathbf{R}^n$. The set

$$X_* = \{y \in \mathbf{R}^n \mid y^T x \leq 1 \quad \forall x \in X\}$$

is called the *polar* of X. The basic properties of the polars are as follows (all sets in question belong to certain fixed $\mathbf{R}^n$):

1. Let X be a closed convex set containing the origin. Then so is its polar X_*. Moreover, twice taken polar is the original set: $(X_*)_* = X$.

2. The polarity transformation reverts inclusions: if X, Y are closed convex sets containing the origin, then $X \subset Y$ if and only if $Y_* \subset X_*$.

3. The polar of a centered at the origin full-dimensional ellipsoid $X = \{x \mid x^T Dx \leq 1\}$ is the ellipsoid $X_* = \{y \mid y^T D^{-1} y \leq 1\}$; in particular, for centered at the origin full-dimensional ellipsoid one has
$$\mathrm{Vol}(X)\,\mathrm{Vol}(X_*) = 1.$$

Given full-dimensional centered at the origin ellipsoids $W_i = \{x \mid x^T B_i^2 x \leq 1\}$, $i = 1, ..., m$, let us ask ourselves what is the *polar* W_* of their arithmetic sum $W = W_1 + ... + W_m$. The interest in the polar is natural: both W and its polar W_* clearly are central-symmetric, so that in problems of their inner/outer ellipsoidal approximation we loose nothing when restricting ourselves with the ellipsoids centered at the origin. But then, by the just presented properties of the polar the problem of best outer/inner ellipsoidal approximation for W is exactly the same as the problem of the best inner/outer ellipsoidal approximation for W_*; perhaps W_* is more convenient for processing than W?

In order to understand what is W_*, note that a vector $y \in \mathbf{R}^n$ belongs to W_* if and only if

$$\max\{y^T(w_1 + ... + w_m) \mid w_i \in W_i\} \leq 1,$$

i.e., if and only if

$$\sum_{i=1}^m \max\{y^T w \mid w \in W_i\} \leq 1.$$

On the other hand, it is immediately seen that

$$\max\{y^T w \mid w^T B_i^2 w \leq 1\} = \sqrt{y^T B_i^{-2} y}.$$

Thus, we come to the result as follows:

$$(W_1 + ... + W_m)_* = \{y \mid \sum_{i=1}^m \sqrt{y^T B_i^{-2} y} \leq 1\}. \tag{4.9.5}$$

Problem (O). As we remember, the problem of the best outer ellipsoidal approximation of $W = W_1 + ... + W_m$ is equivalent to the problem of the best inner ellipsoidal approximation of W_*: the larger centered at the origin ellipsoid E_* we are capable to inscribe into W_*, the smaller ellipsoid $E = \mathrm{Polar}(E_*)$ containing W we get. Let us try to find a "large" ellipsoid in W_* as follows: let $\nu = (\nu_1, ..., \nu_m)$ be a vector with positive entries and the sum of entries ≤ 1. We associate with such a vector the convex set

$$W_*^\nu = \{y \mid y^T B_i^{-2} y \leq \nu_i^2, \ i = 1, ..., m\}.$$

By construction, W_*^ν is intersection of the centered at the origin ellipsoids

$$E_i^\nu = \{y \mid y^T (\nu_i B_i)^{-2} y \leq 1\};$$

by (4.9.5), this intersection is contained in W_*. We already know how to find the largest volume ellipsoid E_*^ν contained in $W_*^\nu = \cap_{i=1}^m E_i^\nu$; this ellipsoid is centered at the origin and is contained in W_*. Thus, we get a systematic way to produce centered at the origin and contained in W_* ellipsoids – every "weight vector" ν of the outlined type produces such an ellipsoid, namely, E_*^ν. The best (i.e., the largest), over all weight vectors ν, ellipsoid we can get in this way not necessarily is the largest volume ellipsoid contained in W_*, but may be thought of as an "approximation" to the latter "difficult to compute" entity.

In order to implement our plan, note that a centered at the origin ellipsoid

$$E = \{y \mid y^T Z^{-2} y \leq 1\} \quad [Z \succ 0] \tag{4.9.6}$$

is contained in the ellipsoid $E_i^\nu = \{y \mid y^T (\nu_i B_i)^{-2} y \leq 1\}$ ($i \leq m$ is fixed) if and only if $Z^{-2} \succeq (\nu_i B_i)^{-2}$, i.e., if and only if $Z^2 \preceq \nu_i^2 B_i^2$, or, which is the same by the Lemma on the Schur Complement, if and only if

$$\begin{pmatrix} \nu_i B_i^2 & Z \\ Z & \nu_i I_n \end{pmatrix} \succeq 0.$$

We arrive at the following result:

Proposition 4.9.4 *Given m centered at the origin full-dimensional ellipsoids*

$$W_i = \{x \in \mathbf{R}^n \mid x^T B_i^2 x \leq 1\} \quad [B_i \succ 0],$$

$i = 1, ..., m$, let us associate with these ellipsoids the semidefinite program

$$\begin{array}{rcl} t & \to & \max \\ s.t. & & \\ t & \leq & (\mathrm{Det}\, Z)^{1/(2n+1)}; \\ \begin{pmatrix} \nu_i B_i^2 & Z \\ Z & \nu_i I_n \end{pmatrix} & \succeq & 0, \ i = 1, ..., m; \\ Z & \succeq & 0; \\ \sum_{i=1}^m \nu_i & \leq & 1 \end{array} \tag{$\tilde{\mathrm{O}}$}$$

with variables $t, \nu_i \in \mathbf{R}, Z \in \mathbf{S}^n$. Every feasible solution $(Z, \{\nu_i\})$ to this problem with positive value of the objective produces ellipsoid

$$E = \{x \mid x^T Z^2 x \leq 1\}$$

which contains the arithmetic sum $W_1 + ... + W_m$ of the ellipsoids $W_1, ..., W_m$. The largest volume ellipsoid E one can obtain in this way is given by optimal solution to $(\tilde{\mathrm{O}})$.

Problem (I). Let us represent the given centered at the origin ellipsoids W_i as the images of unit Euclidean balls under linear mappings:

$$W_i = \{x = A_i u \mid u^T u \leq 1\} \quad [A_i \succ 0],$$

$i = 1, ..., m$. A natural way to generate ellipsoids contained in the arithmetic sum $W = W_1 + ... + W_m$ of the given ellipsoids is to note that whenever X_i are $n \times n$ matrices of spectral norms not exceeding one, then the ellipsoid

$$E(X_1, ..., X_M) = \{x = (A_1 X_1 + A_2 X_2 + ... + A_m X_m) u \mid u^T u \leq 1\}$$

is contained in W (indeed, whenever $\| u \|_2 \leq 1$, we have also $\| X_i u \|_2 \leq 1$, so that $A_i X_i u \in W_i$ and, consequently, $(\sum_{i=1}^m A_i X_i) u \in W$). Thus, every collection of square matrices with spectral norms not exceeding 1 produces an ellipsoid contained in W, and we could use the largest volume ellipsoid of this form (i.e., the one corresponding to the largest $|\operatorname{Det}(A_1 X_1 + ... + A_m X_m)|$) as a surrogate of the largest volume ellipsoid contained in W. Recall that we know how to express a bound on the spectral norm of a matrix via LMI:

$$|X| \leq t \Leftrightarrow \begin{pmatrix} tI_n & -X^T \\ -X & tI_n \end{pmatrix} \succeq 0 \quad [X \in \mathbf{M}^{nn}]$$

(item 16 of Section 4.2). The difficulty, however, is that the matrix $\sum_{i=1}^m A_i X_i$ specifying the ellipsoid $E(X_1, ..., X_m)$, although being linear in the "design variables" X_i, is not necessarily symmetric positive semidefinite, and we do not know how to maximize the determinant over general-type square matrices. We may, however, use the following fact from Linear Algebra:

Lemma 4.9.1 *Let $Y = S + C$ be a square matrix represented as a sum of symmetric matrix S and skew-symmetric (i.e., $C^T = -C$) matrix C. Assume that S is positive definite. Then*

$$|\operatorname{Det}(Y)| \geq \operatorname{Det}(S).$$

Proof. We have $Y = S + C = S^{1/2}(I + \Sigma)S^{1/2}$, where $\Sigma = S^{-1/2} C S^{-1/2}$ is skew-symmetric along with C. We have $|\operatorname{Det}(Y)| = \operatorname{Det}(S) |\operatorname{Det}(I + \Sigma)|$; it remains to note that all eigenvalues the skew-symmetric matrix Σ are purely imaginary, so that the eigenvalues of $I + \Sigma$ are ≥ 1 in absolute value, whence $|\operatorname{Det}(I + \Sigma)| \geq 1$. ■

In view of Lemma, it makes sense to impose on $X_1, ..., X_m$, besides the requirement that their spectral norms are ≤ 1, also the requirement that the "symmetric part"

$$S(X_1, ..., X_m) = \frac{1}{2}\left[\sum_{i=1}^m A_i X_i + \sum_{i=1}^m X_i^T A_i\right]$$

of the matrix $\sum_i A_i X_i$ is positive semidefinite, and to maximize under these constraints the quantity $\operatorname{Det}(S(X_1, ..., X_m))$ – a lower bound on the volume of the ellipsoid $E(X_1, ..., X_m)$. With this approach, we come to the following result:

Proposition 4.9.5 *Let $W_i = \{x = A_i u \mid u^T u \leq 1\}$, $A_i \succ 0$, $i = 1,..,m$. Consider the semidefinite program*

$$
\begin{array}{lrcl}
 & t & \to & \max \\
s.t. & & & \\
(a) & t & \leq & \left(\mathrm{Det}\left(\frac{1}{2}\sum_{i=1}^m [X_i^T A_i + A_i X_i]\right)\right)^{1/(2n+1)} \\
(b) & \sum_{i=1}^m [X_i^T A_i + A_i X_i] & \succeq & 0 \\
(c) & \begin{pmatrix} I_n & -X_i^T \\ -X_i & I_n \end{pmatrix} & \succeq & 0, \ i = 1,...,m
\end{array}
$$

with design variables $X_1,...,X_m \in \mathbf{M}^{nn}, t \in \mathbf{R}$. Every feasible solution $(\{X_i\}, t)$ to this problem produces ellipsoid

$$E(X_1,...,X_m) = \{x = (\sum_{i=1}^m A_i X_i) u \mid u^T u \leq 1\}$$

contained in the arithmetic sum $W_1 + ... + W_m$ of the original ellipsoids, and the volume of this ellipsoid is at least t.

4.10 Assignments to Lecture 4

4.10.1 Around positive semidefiniteness, eigenvalues and $\succeq$-ordering

Criteria for positive semidefiniteness

Recall the criterion of positive definiteness of a symmetric matrix:

> [Sylvester] *A symmetric $m \times m$ matrix $A = [a_{ij}]_{i,j=1}^{m}$ is positive definite if and only if all its angular minors*
> $$\mathrm{Det}\left([a_{ij}]_{i,j=1}^{k}\right),$$
> *$k = 1, ..., n$, are positive.*

Exercise 4.1 [5] *Prove that a symmetric $m \times m$ matrix A is positive semidefinite if and only if all its principal minors (i.e., determinants of square sub-matrices symmetric w.r.t. the diagonal) are nonnegative.*

> Hint: look at the angular minors of the matrices $A + \epsilon I_n$ for small positive ϵ.

Demonstrate by example that nonnegativity of angular minors of a symmetric matrix is not sufficient for the positive semidefiniteness of the matrix.

Exercise 4.2 [5] [Diagonal-dominated matrices] *Let $A = [a_{ij}]_{i,j=1}^{m}$ be a symmetric matrix satisfying the relation*

$$a_{ii} \geq \sum_{j \neq i} |a_{ij}|, \; i = 1, ..., m.$$

Prove that A is positive semidefinite.

Diagonalization

Exercise 4.3 [3] *Prove the following standard facts of Linear Algebra:*

1) *If A is symmetric positive semidefinite $m \times m$ matrix and P is a $n \times m$ matrix, then the matrix PAP^T is positive semidefinite.*

2) *A symmetric $m \times m$ matrix A is positive semidefinite if and only if it can be represented as $A = Q\Lambda Q^T$, where Q is orthogonal, and Λ is diagonal with nonnegative diagonal entries. What are these entries? What are the columns of Q?*

3) *Let A, B be two symmetric matrices of the same size and let A be positive definite. Then there exist nonsingular square matrix Q and diagonal matrix Λ such that*

$$A = QQ^T, B = Q\Lambda Q^T.$$

Variational characterization of eigenvalues

The basic fact about eigenvalues of a symmetric matrix is the following

> **Variational Description of Eigenvalues** [Weil] *Let A be a symmetric $m \times m$ matrix and $\lambda(A) = (\lambda_1(A), ..., \lambda_m(A))$ be the vector of eigenvalues of A taken with their multiplicities and written down in the non-ascendent order:*
> $$\lambda_1(A) \geq \lambda_2(A) \geq ... \geq \lambda_m(A).$$

Then for every $i = 1, ..., m$ one has:

$$\lambda_i(A) = \min_{E\in\mathcal{E}_i} \max_{v\in E, v^T v=1} v^T A v,$$

where $\mathcal{E}_i$ is the family of all linear subspaces of $\mathbf{R}^m$ of dimension $m - i + 1$.

Exercise 4.4 ^{15} *Prove the Variational Description of Eigenvalues.*

Exercise 4.5 ^{15} *Derive from the Variational Description of Eigenvalues the following facts:*

(i) [Monotonicity of the vector of eigenvalues] *If $A \succeq B$, then $\lambda(A) \geq \lambda(B)$;*

(ii) [Interlacing of Eigenvalues] *Let $A \in \mathbf{S}^m$, and let E be a linear subspace of $\mathbf{R}^m$ of codimension $k < m$ (i.e., of dimension $m-k$). Let A_E be the restriction of the operator $x \mapsto Ax$ onto E, i.e., the operator $x \mapsto PAx : E \to E$, P being the orthoprojector onto E (in terms of matrices: let $e_1, ..., e_{n-k}$ be an orthonormal basis in E; you may think of A_E as of the $(n-k)\times(n-k)$ matrix with the entries $e_i^T A e_j$, $i, j = 1, ..., n-k$). Then for every $i \leq n-k$ one has*

$$\lambda_i(A) \geq \lambda_i(A_E) \geq \lambda_{i+k-1}(A).$$

Recall how a function of symmetric matrix is defined. Let A be a symmetric $m \times m$ matrix and

$$p(t) = \sum_{i=0}^{k} p_i t^i$$

be a real polynomial on the axis. It is clear what is $p(A)$: by definition,

$$p(A) = \sum_{i=0}^{k} p_i A^i \in \mathbf{S}^m.$$

This definition is compatible with the arithmetic: when you add/multiply polynomials, you add/multiply the "values" of these polynomials at a fixed symmetric matrix:

$$(p+q)(A) = p(A) + q(A); (p\cdot q)(A) = p(A)q(A).$$

A nice feature of this definition is that

> (*) *For $A \in \mathbf{S}^m$, the matrix $p(A)$ depends only on the restriction of p on the spectrum (set of eigenvalues) of A: if p and q are two polynomials such that $p(\lambda_i(A)) = q(\lambda_i(A))$ for $i = 1, ..., m$, then $p(A) = q(A)$.*
>
> Indeed, we can represent a symmetric matrix A as $A = U^T\Lambda U$, where U is orthogonal and Λ is diagonal; the diagonal entries of Λ are exactly the eigenvalues of A. Since $UU^T = I$, we have $A^i = U^T\Lambda^i U$; consequently,
>
> $$p(A) = U^T p(\Lambda) U,$$
>
> and since the matrix $p(\Lambda)$ depends on the restriction of p on the spectrum of A only, the result follows.
>
> As a byproduct of our reasoning, we get "explicit" representation of $p(A)$ in terms of the *spectral decomposition* $A = U^T\Lambda U$ (U is orthogonal, Λ is diagonal with diagonal $\lambda(A)$):
>
> (**) *The matrix $p(A)$ is just $U^T\,\mathrm{Diag}(p(\lambda_1(A)), ..., p(\lambda_n(A)))U$.*

(*) allows to speak about arbitrary functions of matrices, not necessarily polynomials:

> *Let A be symmetric matrix and f be a real-valued function defined at least at the spectrum of A. By definition, the matrix $f(A)$ is defined as $p(A)$, where p is a polynomial coinciding with f on the spectrum of A.* (The definition makes sense: by (*), $p(A)$ depends only on the restriction of p on the spectrum of A, i.e., every "polynomial continuation" $p(\cdot)$ of f from the spectrum of A on the entire axis results in the same $p(A)$).

"Calculus of functions of a symmetric matrix" we just have defined is fully compatible with the usual arithmetic:

$$(f+g)(A) = f(A)+g(A); (\mu f)(A) = \mu f(A); (f\cdot g)(A) = f(a)g(A); (f\circ g)(A) = f(g(A)),$$

provided that the functions in question are well-defined on the spectrum of the corresponding matrix. And of course the spectral decomposition of $f(A)$ is just $f(A) = U^T \operatorname{Diag}(f(\lambda_1(A)), ..., f(\lambda_m(A)))U$, $A = U^T \operatorname{Diag}(\lambda_1(A), ..., \lambda_m(A))U$ being the spectral decomposition of A.

Note that "Calculus of functions of symmetric matrices" becomes very unusual when we are trying to operate with functions of several (non-commuting) matrices. E.g., it is generally *not* true that $\exp\{A+B\} = \exp\{A\}\exp\{B\}$ (the right hand side matrix may be even non-symmetric!). It is also generally not true that if f is monotone and $A \succeq B$, then $f(A) \succeq f(B)$, etc.

Exercise 4.6 ^{5} *Demonstrate by example that the relation $0 \preceq A \preceq B$ not necessarily implies that $A^2 \preceq B^2$.*

By the way, the relation $0 \preceq A \preceq B$ does imply that $0 \preceq A^{1/2} \preceq B^{1/2}$.

Sometimes, however, we can get "weak" matrix versions of usual arithmetic relations. E.g.,

Exercise 4.7 ^{3} *Let f be a nondecreasing function on the axis, and let $A \succeq B$. Prove that $\lambda(f(A)) \le \lambda(f(B))$.*

The strongest (and surprising) "weak" matrix version of a usual inequality is as follows.

Let $f(t)$ be a *proper* convex function on the axis; by definition, it means that f is a function on the axis taking real values and the value $+\infty$ such that

– the set $\operatorname{Dom} f$ of the values of argument where f is finite is convex and nonempty;

– if a sequence $\{t_i \in \operatorname{Dom} f\}$ converges to a point t and the sequence $f(t_i)$ has a limit, then $t \in \operatorname{Dom} f$ and $f(t) \le \lim\limits_{i\to\infty} f(t_i)$ (it is called "lower semicontinuity").

E.g., the function $f(x) = \begin{cases} 0, & 0 \le t \le 1 \\ +\infty, & \text{otherwise} \end{cases}$ is proper. In contrast to this, the functions

$$g(x) = \begin{cases} 0, & 0 < t \le 1 \\ 1, & t = 0 \\ +\infty, & \text{for all remaining } t \end{cases}$$

and

$$h(x) = \begin{cases} 0, & 0 < t < 1 \\ +\infty, & \text{otherwise} \end{cases}$$

are not proper, although are convex: a proper function cannot "jump up" at an endpoint of its domain, as it is the case for g, and is forbidden to take value $+\infty$ at a point, if it takes values $\le a < \infty$ in every neighbourhood of the point, as it is the case for h.

For a proper function f, its *Legendre transformation* f_* is defined as

$$f_*(s) = \sup_t [ts - f(t)].$$

It turns out that the Legendre transformation of a proper function also is proper, and that twice taken Legendre transformation of a proper function is this very function.

The Legendre transformation (which, by the way, can be defined for convex functions on $\mathbf{R}^n$ as well) underlies a lot of standard inequalities. Indeed, by definition of f_* we have

$$f_*(s) + f(t) \geq st \quad \forall s,t; \tag{L}$$

Specifying somehow f and computing f_*, we can get from the general inequality (L) a lot of useful inequalities. E.g.,

- If $f(t) = \frac{1}{2}t^2$, then $f_*(t) = \frac{1}{2}t^2$, and (L) becomes the standard inequality

$$st \leq \frac{1}{2}t^2 + \frac{1}{2}t^2;$$

- If $1 < p < \infty$ and $f(t) = \begin{cases} \frac{t^p}{p}, & t \geq 0 \\ +\infty, & t < 0 \end{cases}$, then $f_*(s) = \begin{cases} \frac{s^q}{q}, & s \geq 0 \\ +\infty, & s < 0 \end{cases}$, q being given by $\frac{1}{p} + \frac{1}{q} = 1$, and (L) becomes the Young inequality

$$s,t \geq 0 \Leftarrow ts \leq \frac{t^p}{p} + \frac{s^q}{q},\ 1 < p,q < \infty, \frac{1}{p} + \frac{1}{q} = 1.$$

Now, what happens with (L) if s,t are symmetric matrices? Of course, both sides of (L) still make sense and are matrices, but we have no hope to say something reasonable about the relation between these matrices (e.g., the right hand side in (L) is not necessarily symmetric). However,

Exercise 4.8 ^{10} *Let f_* be a proper function with the domain* $\mathrm{Dom}\, f_* \subset \mathbf{R}_+$, *and let f be the Legendre transformation of f. Then for every pair symmetric matrices X, Y of the same size with the spectrum of X belonging to* $\mathrm{Dom}\, f$ *and the spectrum of Y belonging to* $\mathrm{Dom}\, f_*$ *one has*

$$\lambda(f(X)) \geq \lambda\left(Y^{1/2}XY^{1/2} - f_*(Y)\right)^{16)}$$

Birkhoff's Theorem

Surprisingly enough, one of the most useful facts about eigenvalues of symmetric matrices is the following, essentially combinatorial, statement (it does not mention the word "eigenvalue" at all).

> **Birkhoff's Theorem.** *Consider the set $\mathcal{S}_m$ of double-stochastic $n \times n$ matrices, i.e., square matrices $[p_{ij}]_{i,j=1}^m$ satisfying the relations*
>
> $$\begin{array}{rcl} p_{ij} & \geq & 0,\ i,j = 1,...,m; \\ \sum_{i=1}^m p_{ij} & = & 1,\ j = 1,...,m; \\ \sum_{j=1}^m p_{ij} & = & 1,\ i = 1,...,m. \end{array}$$

[16)] In the scalar case, our inequality reads $f(x) \geq y^{1/2}xy^{1/2} - f_*(y)$, which is an equivalent form of (L) for the case of $\mathrm{Dom}\, f_* \subset \mathbf{R}_+$.

A matrix P belongs to $\mathcal{S}_m$ if and only if it can be represented as a convex combination of $n \times n$ permutation matrices:

$$P \in \mathcal{S}_m \Leftrightarrow \exists \{\lambda_i \geq 0, \sum_i \lambda_i = 1) : P = \sum_i \lambda_i \Pi^i,$$

where all Π^i are permutation matrices (with exactly one nonzero, namely, equal to 1, element in every row and every column).

An immediate corollary of the Birkhoff Theorem is the following fact:

(*) *Let $f : \mathbf{R}^m \to \mathbf{R} \cup \{+\infty\}$ be a convex symmetric function (symmetry means that the value of the function remains unchanged when we permute the coordinates in an argument), let $x \in \mathrm{Dom}\, f$ and $P \in \mathcal{S}_m$. Then*

$$f(Px) \leq f(x).$$

The proof is immediate: by Birkhoff's Theorem, Px is a convex combination of a number of permutations x^i of x. Since f is convex, we have

$$f(Px) \leq \max_i f(x^i) = f(x),$$

the concluding equality being given by the symmetry of f.

The role of (*) in numerous questions related to eigenvalues is based upon the following simple

Observation. *Let A be a symmetric $m \times m$ matrix. Then the diagonal $\mathrm{Dg}(A)$ of the matrix A is the image of the vector $\lambda(A)$ of the eigenvalues of A under multiplication by a double stochastic matrix:*

$$\mathrm{Dg}(A) = P\lambda(A) \text{ for some } P \in \mathcal{S}_m$$

Indeed, consider the spectral decomposition of A:

$$A = U^T \mathrm{Diag}(\lambda_1(A), ..., \lambda_m(A))U$$

with orthogonal $U = [u_{ij}]$. Then

$$A_{ii} = \sum_{j=1}^m u_{ji}^2 \lambda_j(A) \equiv (P\lambda(A))_i,$$

where the matrix $P = [u_{ji}^2]_{i,j=1}^m$ is double stochastic.

Combining Observation and (*), we conclude that if f is a convex symmetric function on $\mathbf{R}^m$, then for every $m \times m$ symmetric matrix one has

$$f(\mathrm{Dg}(A)) \leq f(\lambda(A)).$$

Moreover, let $\mathcal{O}_m$ be the set of all orthogonal $m \times m$ matrices. For every $V \in \mathcal{O}$, the matrix $V^T AV$ has the same eigenvalues as A, so that for a convex symmetric f one has

$$f(\mathrm{Dg}(V^T AV)) \leq f(\lambda(V^T AV)) = f(\lambda(A)),$$

whence

$$f(\lambda(A)) \geq \max_{V\in\mathcal{O}_m} f(\mathrm{Dg}(V^TAV)).$$

In fact the inequality here is equality, since for properly chosen $V \in \mathcal{O}_m$ we have $\mathrm{Dg}(V^TAV) = \lambda(A)$. We have arrived at the following result:

> *(**) Let f be a symmetric convex function on $\mathbf{R}^m$. Then for every symmetric $m\times m$ matrix A one has*
>
> $$f(\lambda(A)) = \max_{V\in\mathcal{O}_m} f(\mathrm{Dg}(V^TAV)),$$
>
> *$\mathcal{O}_m$ being the set of all $m\times m$ orthogonal matrices.*
>
> *In particular, the function*
>
> $$F(A) = f(\lambda(A))$$
>
> *is convex in $A \in \mathbf{S}^m$ (as the maximum of a family of convex in A functions $F_V(A) = f(\mathrm{Dg}(V^TAV))$, $V \in \mathcal{O}_m$.)*

Exercise 4.9 ^{10} *Let $A = [a_{ij}]$ be a symmetric $m \times m$ matrix. Prove that*

(i) *Whenever $p \geq 1$, one has $\sum_{i=1}^m |a_{ii}|^p \leq \sum_{i=1}^m |\lambda_i(A)|^p$;*

(ii) *Whenever A is positive semidefinite, $\prod\limits_{i=1}^m a_{ii} \geq \mathrm{Det}\,(A)$;*

(iii) *Let for $x \in \mathbf{R}^m$ the function $S_k(x)$ be the sum of k largest entries of x (i.e., the sum of the first k entries in the vector obtained from x by writing down the coordinates of x in the non-ascending order). Prove that $S_k(x)$ is a convex symmetric function of x and derive from this observation that*

$$S_k(\mathrm{Dg}(A)) \leq S_k(\lambda(A)).$$

Hint: note that $S_k(x) = \max_{1\leq i_1<i_2<...<i_k\leq m} \sum_{l=1}^k x_{i_l}$.

(iv) [Trace inequality] **Whenever $A, B \in \mathbf{S}^m$, one has*

$$\lambda^T(A)\lambda(B) \geq \mathrm{Tr}(AB).$$

Exercise 4.10 ^{5} *Prove that if $A \in \mathbf{S}^m$ and $p, q \in [1,\infty]$ are such that $\frac{1}{p}+\frac{1}{q} = 1$, then*

$$\max_{B\in\mathbf{S}^m:\|\lambda(B)\|_q=1} \mathrm{Tr}(AB) = \| \lambda(A) \|_p .$$

In particular, $\| \lambda(\cdot) \|_p$ is a norm on $\mathbf{S}^m$, and the conjugate of this norm is $\| \lambda(\cdot) \|_q$, $\frac{1}{p}+\frac{1}{q} = 1$.

Exercise 4.11 ^{200} *Prove that the functions*

$$\begin{array}{rcl} F(X) & = & \sum_{i=1}^m |\lambda_i(X)|^{7/3} \\ G(X) & = & \sum_{i=1}^m \max(\lambda_i(X), 0)^{7/3} \end{array}$$

of $X \in \mathbf{S}^m$ are convex and find SDR's of these functions.

Cauchy's inequality for matrices

Exercise 4.12 [10] *The standard Cauchy's inequality says that*

$$|\sum_i x_i y_i| \leq \sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2} \tag{4.10.1}$$

for reals x_i, y_i, $i = 1, ..., n$; this inequality is exact in the sense that for every collection $x_1, ..., x_n$ there exists a collection $y_1, ..., y_n$ with $\sum\limits_i y_i^2 = 1$ which makes (4.10.1) an equality.

(i) *Prove that whenever $X_i, Y_i \in \mathbf{M}^{p,q}$, one has*

$$\sigma(\sum_i X_i^T Y_i) \leq \lambda\left(\left[\sum_i X_i^T X_i\right]^{1/2}\right) \parallel \lambda\left(\sum_i Y_i^T Y_i\right) \parallel_\infty^{1/2} \tag{$*$}$$

where $\sigma(A) = \lambda([AA^T]^{1/2})$ is the vector of singular values of a matrix A arranged in the non-ascending order.

Prove that for every collection $X_1, ..., X_n \in \mathbf{M}^{p,q}$ there exists a collection $Y_1, ..., Y_n \in \mathbf{M}^{p,q}$ with $\sum\limits_i Y_i^T Y_i = I_q$ which makes (*) equality.

(ii) *Prove the following "matrix version" of the Cauchy inequality: whenever $X_i, Y_i \in \mathbf{M}^{p,q}$, one has*

$$|\sum_i \mathrm{Tr}(X_i^T Y_i)| \leq \mathrm{Tr}\left(\left[\sum_i X_i^T X_i\right]^{1/2}\right) \parallel \lambda(\sum_i Y_i^T Y_i) \parallel_\infty^{1/2}, \tag{$**$}$$

and for every collection $X_1, ..., X_n \in \mathbf{M}^{p,q}$ there exists a collection $Y_1, ..., Y_n \in \mathbf{M}^{p,q}$ with $\sum\limits_i Y_i^T Y_i = I_q$ which makes (**) an equality.

Minors of positive semidefinite matrices

A well-known fact of Linear Algebra is that a symmetric $m \times m$ matrix (A_{ij}) is positive semidefinite if and only if it is a Gram matrix:

> there exists a system of m vectors f_i such that
>
> $$A_{ij} = f_i^T f_j$$
>
> for all i, j.

The goal of the subsequent exercises is to prove the following nice extension of the "if" part of this result:

> (!) *Let $F_1, ..., F_m$ be $p \times q$ rectangular matrices. Then the $m \times m$ matrix*
>
> $$A_{ij} = \mathrm{Det}\,(F_i^T F_j),\ \ i, j = 1, ..., m$$
>
> *is positive semidefinite.*

(!) is an immediate consequence of the following fact:

(!!) *Let B be $pm \times pm$ positive semidefinite symmetric matrix partitioned into m^2 blocks B^{ij} of sizes $p \times p$ each, as in the following example:*

$$B = \left(\begin{array}{cc|cc|cc} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ b_{12} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ \hline b_{13} & b_{23} & b_{33} & b_{34} & b_{35} & b_{36} \\ b_{14} & b_{24} & b_{34} & b_{44} & b_{45} & b_{46} \\ \hline b_{15} & b_{25} & b_{35} & b_{45} & b_{55} & b_{56} \\ b_{16} & b_{26} & b_{36} & b_{46} & b_{56} & b_{66} \end{array}\right)$$

6 × 6 symmetric matrix and its partitioning into 2 × 2 blocks

Then the $m \times m$ matrix $B^{(p)}$:

$$B^{(p),\det}_{ij} = \mathrm{Det}\,(B^{ij}), \ i,j = 1, ..., m$$

is positive semidefinite.

Moreover, if $0 \preceq B \preceq C$, then

$$0 \preceq B^{(p),\det} \preceq C^{(p),\det}.$$

Exercise 4.13 ^{3} *Prove the implication* (!!)⇒(!).

In order to prove (!!), it makes sense to use the following construction. Let us fix a positive integer k. For an integer $n \geq k$, let $\mathcal{I}_n$ denote the family of all k-element subsets $\bar{i} = \{i_1 < i_2 < ... < i_k\}$ of the index set $\{1, 2, ..., n\}$. Now, given an $n \times n$ matrix A (not necessarily symmetric), let us define $\binom{n}{k} \times \binom{n}{k}$ matrix $\overline{A}$ as follows: the rows and the columns of $\overline{A}$ are indiced by elements of $\mathcal{I}_n$, and $\overline{A}_{\bar{i}\bar{j}}$ is the $k \times k$ minor of A formed by elements of the rows from the set $\bar{i}$ and the columns from the set $\bar{j}$. A nice property of the mapping $A \mapsto \overline{A}$ is that it preserves multiplication and transposition:

$$\overline{A} \cdot \overline{B} = \overline{AB}; \quad \overline{A^T} = (\overline{A})^T \qquad (*)$$

Exercise 4.14 ^{5} 1) *Verify* (*).

2) *Derive* (!!) *from* (*).

Hint: use the result of Exercise 4.3.3)

Recall that if A is an $m \times m$ matrix, then its adjoint A^{adj} is the $m \times m$ matrix with entries A^{adj}_{ij} which are algebraic complements (in A) to cells i, j.

Exercise 4.15 ^{10} 1) *Prove that* $\left.\frac{d}{dt}\right|_{t=0} \mathrm{Det}\,(A + tB) = \mathrm{Tr}(B^T A^{\mathrm{adj}})$.

2) *Derive from* (!!) *and* 1) *that if B is symmetric $pm \times pm$ positive semidefinite matrix partitioned into m^2 blocks B^{ij} of sizes $p \times p$ each, then the $pm \times pm$ matrix*

$$B^{(p),\mathrm{adj}} = \begin{pmatrix} (B^{11})^{\mathrm{adj}} & ... & (B^{1m})^{\mathrm{adj}} \\ ... & ... & ... \\ (B^{m1})^{\mathrm{adj}} & ... & (B^{mm})^{\mathrm{adj}} \end{pmatrix}$$

is symmetric positive semidefinite.

3) *Derive* 2) *directly from* (!!) *and from the following observation:*

For a square matrix A, let A^+ be the matrix with A^+_{ij} being the determinant of the matrix obtained from A by eliminating the row i and the column j. Then, in the notation from 2),

$$B^{(p),\mathrm{adj}} = D^T B^{(p),+} D,$$
$$B^{(p),+} = \begin{pmatrix} (B^{11})^+ & \dots & (B^{1m})^+ \\ \dots & \dots & \dots \\ (B^{m1})^+ & \dots & (B^{mm})^+ \end{pmatrix}.$$

with properly chosen diagonal matrix D independent of B.

4) *Using the same line of argument as in* 3)*, prove that the mappings*

$$\begin{array}{rcl} B & \mapsto & B^{(p),+} \\ B & \mapsto & B^{(p),\mathrm{adj}} \end{array}$$

are $\succeq$-monotone mappings of $\mathbf{S}^{pm}_+$ into $\mathbf{S}^{pm}_+$:

$$0 \preceq B \preceq C \Rightarrow \left\{ \begin{array}{rcl} 0 \preceq B^{(p),+} & \preceq & C^{(p),+} \\ 0 \preceq B^{(p),\mathrm{adj}} & \preceq & C^{(p),\mathrm{adj}} \end{array} \right.$$

Note that already the $\succeq$-monotonicity of the mapping

$$A \mapsto A^{\mathrm{adj}} : \mathbf{S}^p_+ \to \mathbf{S}^p_+$$

(stated by 4) as applied with $m = 1$) is a rather surprising fact. Indeed, the mapping $A \mapsto A^{-1}$: int $\mathbf{S}^p_+ \to$ int $\mathbf{S}^p_+$ which is "very close" to the mapping $A \mapsto A^{\mathrm{adj}}$ (since $A^{-1} = [\det(A)]^{-1}(A^{\mathrm{adj}})^T$; for symmetric A, you may omit the transposition) is $\succeq$-antimonotone:

$$0 \prec A \preceq B \Rightarrow A^{-1} \succeq B^{-1}.$$

4.10.2 SD representations of epigraphs of convex polynomials

Mathematically speaking, the central question concerning "expressive abilities" of Semidefinite Programming is how wide is the family of convex sets which are SDr. By definition, an SDr set is the projection of the inverse image of $\mathbf{S}^m_+$ under affine mapping. In other words, every SDr set is a projection of a convex set given by a number of polynomial inequalities; indeed, the cone $\mathbf{S}^m$ is a convex set given by a number of polynomial inequalities saying that all principal minors of matrix are nonnegative. Consequently, the inverse image of $\mathbf{S}^m_+$ under an affine mapping also is a convex set given by a number of (non-strict) polynomial inequalities. And it is known that every projection of such a set also is given by a number of polynomial inequalities (both strict and non-strict). We conclude that

> *A SD-representable set always is a convex set given by finitely many polynomial inequalities (strict and non-strict).*

A natural (and seemingly very difficult) question is whether the inverse is true – whether a convex set given by a number of polynomial inequalities always is SDr. This question can be simplified in many ways – we may fix the dimension of the set, we may assume the polynomials participating in inequalities to be convex, we may fix the degrees of the polynomials, etc.; to the best of our knowledge, all these questions are open.

The goal of the subsequent exercises is to answer affirmatively on the simplest question from the outlined series:

Let $\pi(x)$ be a convex polynomial of one variable. Then the epigraph of the polynomial – the set

$$\{(t,x) \in \mathbf{R}^2 \mid t \geq \pi(x)\}$$

is SDr.

Let us fix a nonnegative integer k and consider the curve

$$p(x) = (1, x, x^2, ..., x^{2k})^T \in \mathbf{R}^{2k+1}.$$

Let Π_k be the closure of the convex hull of values of the curve. How can one describe Π_k?

A convenient way to answer this problem is to pass to *matrix representation* of all objects involved. Namely, let us associate with a vector $\xi = (\xi_0, \xi_1, ..., \xi_{2k}) \in \mathbf{R}^{2k+1}$ the $(k+1) \times (k+1)$ symmetric matrix

$$\mathcal{M}(\xi) = \begin{pmatrix} \xi_0 & \xi_1 & \xi_2 & \xi_3 & \cdots & \xi_k \\ \xi_1 & \xi_2 & \xi_3 & \xi_4 & \cdots & \xi_{k+1} \\ \xi_2 & \xi_3 & \xi_4 & \xi_5 & \cdots & \xi_{k+2} \\ \xi_3 & \xi_4 & \xi_5 & \xi_6 & \cdots & \xi_{k+3} \\ \cdots & \cdots & \cdots & \cdots & \ddots & \cdots \\ \xi_k & \xi_{k+1} & \xi_{k+2} & \xi_{k+3} & \cdots & \xi_{2k} \end{pmatrix},$$

so that

$$[\mathcal{M}(\xi)]_{ij} = \xi_{i+j}, \ i,j = 2, ..., k.$$

Transformation $\xi \mapsto \mathcal{M}(\xi) : \mathbf{R}^{2k+1} \to \mathbf{S}^{k+1}$ is a linear embedding; the image of Π_k under this embedding is the closure of the convex hull of values of the curve

$$P(x) = \mathcal{M}(p(x)).$$

It follows that the image $\widehat{\Pi}_k$ of Π_k under the mapping $\mathcal{M}$ possesses the following properties:

(i) $\widehat{\Pi}_k$ belongs to the image of $\mathcal{M}$, i.e., to the subspace H_k of $\mathbf{S}^{2k+1}$ comprised of *Hankel* matrices – matrices with entries depending on the sum of indices only:

$$H_k = \left\{ X \in \mathbf{S}^{2k+1} | i+j = i'+j' \Rightarrow X_{ij} = X_{i'j'} \right\};$$

(ii) $\widehat{\Pi}_k \subset \mathbf{S}^{k+1}_+$ (indeed, every matrix $\mathcal{M}(p(x))$ is positive semidefinite);

(iii) For every $X \in \widehat{\Pi}_k$ one has $X_{00} = 1$.

It turns out that properties (i) – (iii) characterize $\widehat{\Pi}_k$:

> (!) *A symmetric $(k+1) \times (k+1)$ matrix X belongs to $\widehat{\Pi}_k$ if and only if it possesses the properties* (i) – (iii)*: its entries depend on sum of indices only (i.e., $X \in H_k$), X is positive semidefinite and $X_{00} = 1$.*

(!) is a particular case of the classical results related to the so called "moment problem". The goal of the subsequent exercises is to give a simple alternative proof of this statement.

Note that the mapping $\mathcal{M}^* : \mathbf{S}^{k+1} \to \mathbf{R}^{2k+1}$ conjugate to the mapping $\mathcal{M}$ is as follows:

$$(\mathcal{M}^* X)_l = \sum_{i=0}^{l} X_{i,l-i}, \ l = 0, 1, ..., 2k,$$

and we know something about this mapping: namely, Example 18a (Lecture 4) says that

(*) *The image of the cone $\mathbf{S}_+^{k+1}$ under the mapping $\mathcal{M}^*$ is exactly the cone of coefficients of nonnegative on the entire axis polynomials of degree $\leq 2k$.*

Exercise 4.16 ^{10} *Derive* (!) *from* (*).

(!), among other useful things, implies the result we need:

> *Let $\pi(x) = \pi_0 + \pi_1 x + \pi_2 x^2 + ... + \pi_{2k} x^{2k}$ be a convex polynomial on the axis of degree $2k$. Then the epigraph of π is SDr:*
>
> $$\{(t,x) \in \mathbf{R}^2 \mid t \geq p(x)\} = \mathcal{X}[\pi], \tag{4.10.2}$$
>
> where
>
> $$\mathcal{X}[\pi] \quad = \quad \left\{(t,x) \middle| \exists x_2,...,x_{2k} : \begin{array}{l} \begin{pmatrix} 1 & x & x_2 & x_3 & \dots & x_k \\ x & x_2 & x_3 & x_4 & \dots & x_{k+1} \\ x_2 & x_3 & x_4 & x_5 & \dots & x_{k+2} \\ x_3 & x_4 & x_5 & x_6 & \dots & x_{k+3} \\ \dots & \dots & \dots & \dots & \ddots & \dots \\ x_k & x_{k+1} & x_{k+2} & x_{k+3} & \dots & x_{2k} \end{pmatrix} \succeq 0, \\ \pi_0 + \pi_1 x + \pi_2 x_2 + \pi_3 x_3 + ... + \pi_{2k} x_{2k} \leq t \end{array} \right\} \tag{!!}$$

Exercise 4.17 ^{5} *Prove* (4.10.2).

Note that the set $\mathcal{X}[\pi]$ makes sense for an arbitrary polynomial π, not necessary for a convex one. What is the projection of this set onto the (t,x)-plane? The answer is surprisingly nice: this is the convex hull of the epigraph of the polynomial π!

Exercise 4.18 ^{5} *Let $\pi(x) = \pi_0 + \pi_1 x + ... + \pi_{2k} x^{2k}$ with $\pi_{2k} > 0$, and let*

$$G[\pi] = \mathrm{Conv}\{(t,x) \in \mathbf{R}^2 \mid t \geq p(x)\}$$

be the convex hull of the epigraph of π (the set of all convex combinations of points from the epigraph of π).

1) *Prove that $G[\pi]$ is a closed convex set.*

2) *Prove that*

$$G[\pi] = \{(t,x) \in \mathbf{R}^2 \mid \exists x_1,...,x_{2k} : (t_x, x_2, ..., x_{2k}) \in \mathcal{X}[\pi]\}.$$

4.10.3 Around the Lovasz capacity number

Recall that the Lovasz capacity number $\theta(\Gamma)$ of an n-node graph Γ is the optimal value in the following semidefinite program:

$$\lambda \to \min \mid \lambda I_n - \mathcal{L}(x) \succeq 0 \tag{L}$$

where the symmetric $n \times n$ matrix $\mathcal{L}(x)$ is defined as follows:

- the dimension of x is equal to the number of arcs in Γ, and the coordinates of x are indiced by these arcs;

- the element of $\mathcal{L}(x)$ in an "empty" cell ij – such that the nodes i and j are not linked by an arc in Γ – is 1;

- the elements of $\mathcal{L}(x)$ in a pair of symmetric "non-empty" cells ij, ji – such that the nodes i and j are linked by an arc – are equal to the coordinate of x indiced by the corresponding arc.

As we remember, the importance of $\theta(\Gamma)$ comes from the fact that $\theta(\Gamma)$ is a computable upper bound on the stability number $\alpha(\Gamma)$ of the graph. We have seen also that the Shor semidefinite relaxation of the problem of finding the stability number of Γ leads to a "seemingly stronger" upper bound on $\alpha(\Gamma)$, namely, the optimal value $\sigma(\Gamma)$ in the semidefinite program

$$\lambda \to \min \mid \begin{pmatrix} \lambda & -\frac{1}{2}(e+\mu)^T \\ -\frac{1}{2}(e+\mu) & A(\mu,\nu) \end{pmatrix} \succeq 0 \tag{Sh}$$

where $e = (1,...,1)^T \in \mathbf{R}^n$ and $A(\mu,\nu)$ is the matrix as follows:

- the dimension of ν is equal to the number of arcs in Γ, and the coordinates of ν are indiced by these arcs;
- the diagonal entries of $A(\mu,\nu)$ are $\mu_1,...,\mu_n$;
- the off-diagonal entries of $A(\mu,\nu)$ corresponding to "empty cells" are zeros;
- the off-diagonal entries of $A(\mu,\nu)$ in a pair of symmetric "non-empty" cells ij, ji are equal to the coordinate of ν indiced by the corresponding arc.

We have seen that (L) can be obtained from (Sh) when the variables μ_i are set to 1, so that $\sigma(\Gamma) \le \theta(\Gamma)$. Thus,

$$\alpha(\Gamma) \le \sigma(\Gamma) \le \theta(\Gamma). \tag{4.10.3}$$

Exercise 4.19 [5]

1) Prove that if (ξ,μ,ν) *is a feasible solution to* (Sh)*, then there exists a symmetric* $n\times n$ *matrix* A *such that* $\xi I_n - A \succeq 0$ *and at the same time the diagonal entries of* A *and the off-diagonal entries in the "empty cells" are* ≥ 1*. Derive from this observation that the optimal value in* (Sh) *is not less than the optimal value* $\theta'(\Gamma)$ *in the following semidefinite program:*

$$\xi \to \min \mid \xi I_n - X \succeq 0, X_{ij} \ge 1 \textit{ whenever } i,j \textit{ are not adjacent in } \Gamma \qquad \text{(Sc)}.$$

2) Prove that $\theta'(\Gamma) \ge \alpha(\Gamma)$*.*

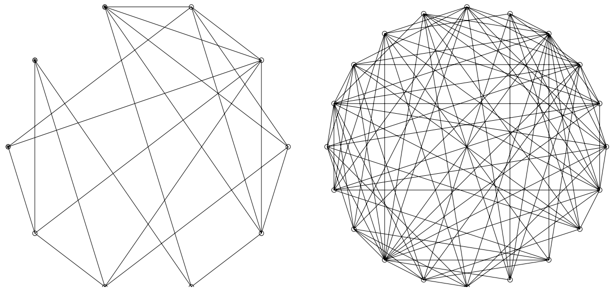
> Hint: Demonstrate that if all entries of a symmetric $k\times k$ matrix are ≥ 1, then the maximum eigenvalue of the matrix is at least k. Derive from this observation and the Interlacing Eigenvalues Theorem (Exercise 4.5.(ii)) that if a symmetric matrix contains a principal $k\times k$ submatrix with entries ≥ 1, then the maximum eigenvalue of the matrix is at least k.

The upper bound $\theta'(\Gamma)$ on the stability number of Γ is called the *Schrijver* capacity of graph Γ. Note that we have

$$\alpha(\Gamma) \le \theta'(\Gamma) \le \sigma(\Gamma) \le \theta(\Gamma).$$

A natural question is which inequalities in this chain may happen to be strict. To answer it, we have computed the quantities in question for about 2,000 random graphs with number of nodes from 8 to 20. In our experiments, the stability number was computed – by brute force – for graphs with ≤ 12 nodes; for no one of the latter graphs, the integral part of $\theta(\Gamma)$ was greater than $\alpha(\Gamma)$. Furthermore, $\theta(\Gamma)$ was not integer in 156 of our 2,000 experiments, and in 27 of these 156 cases the Schrijver capacity number $\theta'(\Gamma)$ was strictly less than $\theta(\Gamma)$. The quantities $\theta'(\cdot), \sigma(\cdot), \theta(\cdot)$ for 13 of these 27 cases are listed in the below table:

Graph #	# of nodes	α	θ'	σ	θ
1	20	?	4.373	4.378	4.378
2	20	?	5.062	5.068	5.068
3	20	?	4.383	4.389	4.389
4	20	?	4.216	4.224	4.224
5	13	?	4.105	4.114	4.114
6	20	?	5.302	5.312	5.312
7	20	?	6.105	6.115	6.115
8	20	?	5.265	5.280	5.280
9	9	3	3.064	3.094	3.094
10	12	4	4.197	4.236	4.236
11	8	3	3.236	3.302	3.302
12	12	4	4.236	4.338	4.338
13	10	3	3.236	3.338	3.338



Graphs # 13 (left) and # 8 (right); all nodes are on circumferences.

Exercise 4.20 ^{3} *Compute the stability numbers of the graphs # 8 and # 13.*

Exercise 4.21 ^{10} *Prove that* $\sigma(\Gamma) = \theta(\Gamma)$.

The *chromatic number* $\xi(\Gamma)$ of a graph Γ is the minimal number of colours such that one can color the nodes of the graph in such a way that no two adjacent (i.e., linked by an arc) nodes get the same colour[17]. The *complement* $\bar{\Gamma}$ of a graph Γ is the graph with the same set of nodes, and a pair of nodes in $\bar{Gamma}$ is linked by an arc if and only if these nodes are *not* linked by an arc in Γ.

Lovasz proved that for every graph

$$\theta(\Gamma) \le \xi(\bar{\Gamma}) \qquad (*)$$

so that

$$\alpha(\Gamma) \le \theta(\Gamma) \le \xi(\bar{\Gamma})$$

(Lovasz's Sandwich Theorem).

[17] E.g., when colouring a geographic map, it is convenient not to colour similarly a pair of countries with common border. It was observed that to meet this requirement for actual maps, 4 colours are sufficient. The "4-colour" Conjecture guesses that this is so for *any* geographic map. Mathematically, you can represent a map by a graph, the nodes being the countries, and two nodes being linked by an arc if and only if the corresponding countries have common border. A characteristic feature of such a graph is that it is *planar* – you may draw it on 2D plane in such a way that the arcs will not cross each other, meeting only at the nodes. Thus, mathematical form of the 4-colour Conjecture is that the chromatic number of any planar graph is at most 4; it indeed is true, but it took about 100 years to prove the conjecture!

Exercise 4.22 [10] *Prove* (*).

Hint: Let us colour the vertices of Γ in $k = \xi(\bar{\Gamma})$ colours in such a way that no two vertices of the same colour are adjacent in $\bar{\Gamma}$, i.e., every two nodes of the same colour are adjacent in Γ. Set $\lambda = k$, and let x be such that

$$[\mathcal{L}(x)]_{ij} = \begin{cases} -(k-1), & i \neq j, \ i,j \text{ are of the same colour} \\ 1, & \text{otherwise} \end{cases}$$

Prove that (λ, x) is a feasible solution to (L).

Exercise 4.23 [5] *Let* $A \in \mathbf{S}^m_+$. *Prove that*

$$\max\{x^T Ax \mid x_i = \pm 1, \ i = 1, ..., m\} = \max\{\frac{2}{\pi}\sum_{i=1}^{m} a_{ij}\arcsin(X_{ij}) \mid X \succeq 0, X_{ii} = 1, \ i = 1, ..., m\}.$$

Exercise 4.24 [10] *Let* $A \in \mathbf{S}^m$. *Prove that*

$$\max_{x:x_i=\pm 1, i=1,...,m} x^T Ax \geq \mathrm{Tr}(A).$$

Develop an efficient algorithm which, given A, *generates a point* x *with coordinates* ± 1 *such that* $x^T Ax \geq \mathrm{Tr}(A)$.

4.10.4 Around Lyapunov Stability Analysis

A natural mathematical model of the usual swing is the linear time invariant dynamic system

$$y''(t) = -\omega^2 y(t) - 2\mu y'(t) \tag{S}$$

with positive ω^2 and $0 \leq \mu < \omega$ (the term $\mu y'(t)$ represents friction). A general solution to this equation is

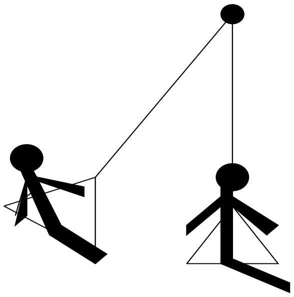
$$y(t) = a\cos(\omega' t + \phi_0)\exp\{-\mu t\}, \ \omega' = \sqrt{\omega^2 - \mu^2},$$

with free parameters a and ϕ_0, i.e., this is a decaying oscillation. Note that the equilibrium

$$y(t) \equiv 0$$

is stable – every solution to (S) exponentially fast converges to 0 along with its derivative.

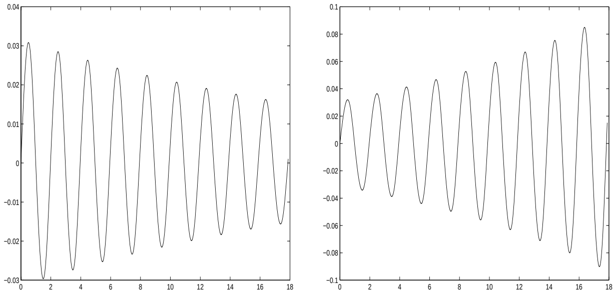
After stability is observed, an immediate question arises: how is it possible to swing on a swing? Everybody known from practice that it is possible. On the other hand, since the equilibrium is stable, it looks as if it was impossible to swing, without somebody's assistance, for a long time. The reason which makes swinging possible is highly nontrivial – *parametric resonance*. A swinging kid does not sit on the swing in a once for ever fixed position; what he does is shown on the picture:



As a result, the "effective length" of the swing l – the distance from the point where the ropes are fixed to the center of gravity of the system – becomes varying with time: $l = l(t)$. School mechanics says that $\omega^2 = g/l$, g being the gravity acceleration. Thus, the actual swing is a *time-varying* linear dynamic system:

$$y''(t) = -\omega^2(t)y(t) - 2\mu y'(t). \tag{S$'$}$$

And it turns out that for properly varied $\omega(t)$ the equilibrium $y(t) \equiv 0$ is *not* stable. A swinging kid is just varying $l(t)$ in a way which results in *instable* dynamic system (S$'$) and enjoys this instability.



$$y''(t) = -\frac{g}{l+h\sin(2\omega t)}y(t) - 2\mu y'(t), \quad y(0) = 0, y'(0) = 0.1$$

$$\left[l = 1\,[\mathrm{m}], g = 10\,[\tfrac{\mathrm{m}}{\mathrm{sec}^2}], \mu = 0.15[\tfrac{1}{\mathrm{sec}}], \omega = \sqrt{g/l}\right]$$

Graph of $y(t)$

left: $h = 0.125$: this kid is too small; he should grow up...
right: $h = 0.25$: this kid already can swing...

Exercise 4.25 [15] *Assume that you are given parameters l ("nominal length of the swing rope"), $h > 0$ and $\mu > 0$, and it is known that a swinging kid can vary the "effective length" of the rope within the bounds $l \pm h$, i.e., its movement is governed by the uncertain linear time-varying system*

$$y''(t) = -a(t)y(t) - 2\mu y'(t), \quad a(t) \in \left[\frac{g}{l+h}, \frac{g}{l-h}\right].$$

Try to identify the domain in the 3D-space of parameters l, μ, h where the system is stable, as well as the domain where its stability can be certified by a quadratic Lyapunov function. What is "the difference" between these two domains?

4.10.5 Around $\mathcal{S}$-Lemma

Recall that $\mathcal{S}$-Lemma states that if x^TAx and x^TBx are two homogeneous quadratic forms and the inequality

$$x^TAx \geq 0 \tag{I}$$

is strictly feasible (i.e., $x^TAx > 0$ for some x), then the inequality

$$x^TBx \geq 0 \tag{II}$$

is a consequence of the inequality (I):

$$\forall x : x^TAx \geq 0 \Rightarrow x^TBx \geq 0 \tag{$\Rightarrow$}$$

if and only if (II) is a "linear consequence" of (I):

$$B = \lambda A + \Delta, \ \lambda \geq 0, \Delta \succeq 0.$$

The form of this statement is already familiar to us: this is a kind of the Theorem on Alternative. The strongest extension of the Theorem on Alternative for systems of scalar *linear* inequalities to the case of *nonlinear* inequalities is the following statement:

> The Lagrange Duality Theorem: *Let $f_0, f_1, ..., f_m$ be convex functions on $\mathbf{R}^m$ such that the system of inequalities*
>
> $$f_i(x) \le 0,\ i = 1, ..., m \tag{S}$$
>
> *is strictly feasible (i.e., $f_i(\bar{x}) < 0$ for some $\bar{x}$ and all $i = 1, ..., m$). The inequality*
>
> $$f_0(x) \ge 0$$
>
> *is a consequence of the system (S) if and only if it can be obtained, in linear fashion, from (S) and a "trivially" true – valid on the entire $\mathbf{R}^n$ – inequality, i.e., if and only if there exist m nonnegative weights λ_i such that*
>
> $$f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \ge 0 \quad \forall x.$$

$\mathcal{S}$-Lemma deals with a very specific setup of this type (a single and very specific inequality in (S), a very specific f_0); the point, however, is that the Lemma is absolutely free of any convexity assumptions. The Lemma demonstrates an extremely useful "exceptional" phenomenon (see Exercise 4.29 below).

The goal of the subsequent exercises is to prove the $\mathcal{S}$-Lemma. The "if" part of the Lemma is evident, so that we will focus on the "only if" part of it. Thus, in the sequel it is assumed that we are given two quadratic forms $x^T Ax$ and $x^T Bx$ with symmetric matrices A, B such that $\bar{x}^T A\bar{x} > 0$ for some $\bar{x}$ and the implication $(\Rightarrow)$ is true. Our goal is to prove that

> (*) *Under outlined assumptions, there exists $\lambda \ge 0$ such that $B \succeq \lambda A$.*

The main tool we need is the following

Theorem 4.10.1 [General Helley Theorem] *Let $\{A_\alpha\}_{\alpha\in I}$ be a family of closed convex sets in $\mathbf{R}^n$ such that*

1. *Every $n+1$ sets from the family have a point in common;*
2. *There is a finite sub-family of the family such that the intersection of the sets from the sub-family is bounded.*

Then all sets from the family have a point in common.

Exercise 4.26 ^{3} *Prove the General Helley Theorem.*

Exercise 4.27 ^{5} *Show that (*) is a corollary of the following statement:*

> (**) *Let $x^T Ax$, $x^T Bx$ be two quadratic forms such that*
>
> $$x^T Ax \ge 0, x \ne 0 \Rightarrow x^T Bx > 0 \tag{$\Rightarrow'$}$$
>
> *Then there exists $\lambda \ge 0$ such that $B \succeq \lambda A$.*

Exercise 4.28 [10] *Given data A, B satisfying the premise of $(**)$, define the sets*

$$Q_x = \{\lambda \geq 0 : x^T Bx \geq \lambda x^T Ax\}.$$

1) Prove that every one of the sets Q_x is a closed nonempty convex set on the axis;
2) Prove that at least one of the sets Q_x is bounded;
*3) *Prove that every two sets $Q_{x'}, Q_{x''}$ have a point in common.*
4) Derive $(**)$ from 1) – 3), thus concluding the proof of the $\mathcal{S}$-Lemma.

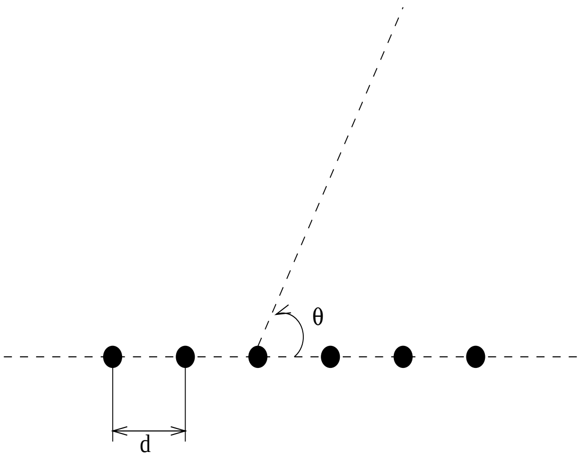
Exercise 4.29 [3] *Demonstrate by example that if $x^T Ax, x^T Bx, x^T Cx$ are three quadratic forms with symmetric matrices such that*

$$\begin{array}{c} \exists \bar{x} : \bar{x}^T A\bar{x} > 0, \bar{x}^T B\bar{x} > 0 \\ x^T Ax \geq 0, x^T Bx \geq 0 \Rightarrow x^T Cx \geq 0, \end{array}$$

then <u>not necessarily</u> there exist $\lambda, \mu \geq 0$ such that $C \succeq \lambda A + \mu B$.

4.10.6 Around Antenna Synthesis

Consider the Antenna Synthesis problem from Example 2, Section 4.6:

> Given an equidistant array of n harmonic oscillators placed along the X-axis in the XY-plane:
>
> 
>
>
> choose complex weights z_j, $j = 1, ..., n$, in order to get the modulus $|Z(\cdot)|$ of the corresponding diagram
>
> $$Z(\theta) = \sum_{l=0}^{n-1} z_l \exp\{-\mathrm{i}l\Omega(\theta)\}, \quad \Omega(\theta) = \frac{2\pi d\cos(\theta)}{\lambda}$$
>
> as close as possible to a given target.

Assume we are interested to get a diagram which is "concentrated" in the beam $-\phi_+ \leq \theta \leq \phi_+$. The natural design specifications in this case are as follows: we fix a ripple $\alpha > 1$, require from the diagram to satisfy the inequality

$$\frac{1}{\alpha} \leq |Z(\theta)| \leq \alpha, \; -\phi_+ \leq \theta \leq \phi_+$$

and minimize under this restriction the sidelobe attenuation level

$$\max_{\phi_- \leq |\theta| \leq \pi} |Z(\theta)|,$$

$\phi_- > \phi_+$ being a given sidelobe angle.

To solve the problem, one may use a simplified version of the approach presented in Section 4.6, namely, may pose the problem as

$$\begin{array}{llrcl}
 & & \epsilon & \to & \min \\
\text{s.t.} & & & & \\
(a) & 0 \leq R(\omega) \equiv r(0) + \sum_{l=1}^{n-1}(r(2l-1)\cos(l\omega) + r(2l)\sin(l\omega)) & & \leq & \epsilon,\ \omega \in \Gamma_1 \\
(b) & & 0 & \leq & R(\omega),\ \omega \in \Gamma_2 \\
(c) & & \alpha^{-2} \leq R(\omega) & \leq & \alpha^2,\ \omega \in \Gamma_3 \\
(d) & & 0 & \leq & R(\omega),\ \omega \in \Gamma_4,
\end{array}$$

where Γ_j, $j = 1, 2, 3, 4$, are fine finite grids in the segments

$$\begin{array}{rcl}
\Delta_1 & = & [-\pi, \omega_{\min}],\ \omega_{\min} = -\frac{2\pi d}{\lambda}, \\
\Delta_2 & = & [\omega_{\min}, \omega_-],\ \omega_- = \frac{2\pi d \cos(\phi_-)}{\lambda}, \\
\Delta_3 & = & [\omega_-, \omega_+],\ \omega_+ = \frac{2\pi d \cos(\phi_+)}{\lambda}, \\
\Delta_4 & = & [\omega_+, \pi].
\end{array} \qquad (*)$$

Note that the lower bounds in $(a), (b), (d)$ are aimed at ensuring that $R(\omega)$ is nonnegative on $[-\pi, \pi]$, which is a necessary and sufficient condition for $R(\Omega(\theta))$ to be of the form $|Z(\theta)|^2$ for some $Z(\theta)$. Of course, the indicated lower bounds ensure nonnegativity of R only on the grid $\Gamma = \cup_{j=1}^4 \Gamma_j$ in the segment $[-\pi, \pi]$, not on the segment itself, so that a solution to (*) sometimes should be modified to yield an actual diagram. Note that the settings we dealt with in Section 4.6 were free of this drawback: there we were ensuring nonnegativity of $R(\cdot)$ by restricting the coefficient vector r to belong to the SDr set of coefficients of nonnegative trigonometric polynomials. The "approximate" setting (*), however, has a definite advantage – this is just an LP program, not a semidefinite one. To utilize this advantage, we should know how to modify a solution to (*) in order to make the corresponding $R(\cdot)$ nonnegative on the entire segment.

Exercise 4.30 [5] *Assume that Γ is an M-point equidistant grid on $[-\pi, \pi]$:*

$$\Gamma = \{-\pi + \frac{2\pi j}{M} \mid j = 1, 2, ..., M\}$$

and that

$$M > \pi n(n-1).$$

Prove that if (ϵ, r) is a feasible solution to (*), then $r(0) \geq 0$ and the "regularized $R(\cdot)$" – the trigonometric polynomial

$$R(\omega) + \delta \equiv r(0) + \delta + \sum_{l=1}^{n-1}(r(2l-1)\cos(l\omega) + r(2l)\sin(l\omega)),$$

where the "regularization" δ is given by

$$\delta = \frac{\pi n(n-1)}{M - \pi n(n-1)},$$

is nonnegative on $[-\pi, \pi]$.

Consider now the particular data as follows:

- the number of ocillators in array is $n = 12$;

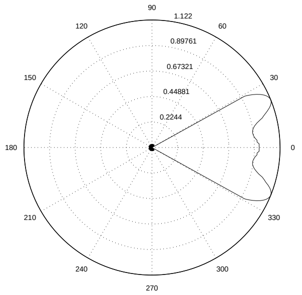
- the inter-element distance is $d = 0.25\lambda$;

- the (half) width of the beam $\phi_+ = 30^{\rm o}$;

- the sidelobe angle $\phi_- = 45^{\rm o}$;

- the ripple $\alpha = 1{\rm dB} = 10^{1/20} \approx 1.1220$.

When solving (*) with Γ being the equidistant 2048-point grid, the following pattern function $|Z^{\rm nom}(\theta)|$ is obtained:

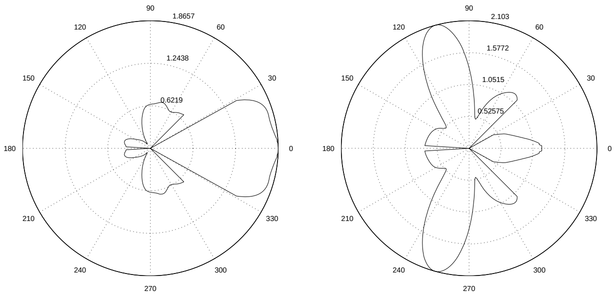


The dream...

The resulting sidelobe attenuation level $\sqrt{\epsilon}$ is $-30.25{\rm dB} = 10^{-30.25/20} \approx 0.031$. The result, however, is completely instable with respect to "implementation errors": when the weights $z_j^{\rm nom}$ of the resulting diagram $Z^{\rm nom}(\theta)$ are perturbed as

$$z_j^{\rm nom} \mapsto (1+\epsilon_j) z_j^{\rm nom},$$

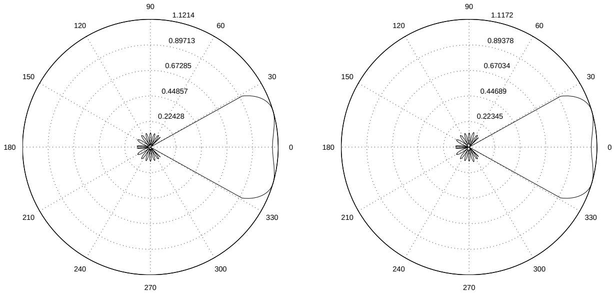
where ϵ_j are independent random (complex-valued) perturbations with zero mean and norm not exceeding $\epsilon\sqrt{2}$, the pattern function of an "actual" (random) diagram may look as follows:



The reality...

Left: perturbation level $\epsilon = 0.005\sqrt{2}$
sidelobe attenuation level $-3.13{\rm dB} \approx 0.694$; actual ripple $5.4{\rm dB} \approx 1.866$
Right: perturbation level $\epsilon = 0.01\sqrt{2}$
sidelobe attenuation level $6.45{\rm dB} \approx 2.103$

The pattern functions below are given by a "robust" design:



Robust design, "actual" pattern functions

Left: perturbation level $\epsilon = 0.005\sqrt{2}$
sidelobe attenuation level -17.77dB ≈ 0.1292
Right: perturbation level $\epsilon = 0.01\sqrt{2}$
sidelobe attenuation level -17.41dB ≈ 0.135

Exercise 4.31 ^{10} *Think how to build a "robust" setting of the Antenna Synthesis problem from Example 2, Section 4.6.*

4.10.7 Around ellispoidal approximations

Exercise 4.32 ^{15} *Prove the Löwner – Fritz John Theorem (Theorem 4.9.1).*

Exercise 4.33 ^{10} *Consider a discrete time controlled dynamic system*

$$\begin{array}{rcl} x(t+1) & = & Ax(t) + bu(t), \ t \geq 0 \\ x(0) & = & 0, \end{array}$$

where $x(t) \in \mathbf{R}^n$ is state vector and $u(t) \in [-1,1]$ is control. A centered at the origin ellipsoid

$$W = \{x \mid x^T Z x \leq 1\} \quad [Z \succ 0]$$

is called <u>*invariant*</u>*, if*

$$x \in W \Rightarrow Ax \pm b \in W.$$

Prove that

1) If W is an invariant ellipsoid and $x(t) \in W$ for some t, then $x(t') \in W$ for all $t' \geq t$.

2) Assume that the vectors $b, Ab, A^2b, ..., A^{n-1}b$ are linearly independent. Prove that an invairant ellipsoid exists if and only if A is stable (all eigenvalues of A are < 1 in absolute value).

3) Assuming that A is stable, prove that an ellipsoid $\{x \mid x^T Z x \leq 1\}$ $[Z \succ 0]$ is invariant if and only if there exists $\lambda \geq 0$ such that

$$\begin{pmatrix} 1 - b^T Z b - \lambda & -b^T Z A \\ -A^T Z b & \lambda Z - A^T Z A \end{pmatrix} \succeq 0.$$

How could one use this fact to approximate the smallest volume invariant ellipsoid numerically?

Lecture 5

Computational tractability of convex programs

To the moment, we did not look at all how to *solve* optimization problems we dealt with. This is the issue we address now.

5.1 Numerical solution of optimization programs – preliminaries

5.1.1 Mathematical Programming programs

All optimization programs we were speaking about are covered by the following universal form of a *Mathematical Programming program*:

$$p_0(x) \to \min \mid x \in X(p) \subset \mathbf{R}^{n(p)} \tag{p}$$

where

- $n(p)$ is the *design dimension* of problem (p);
- $X(p) \subset \mathbf{R}^n$ is the *feasible domain* of the problem;
- $p(x) : \mathbf{R}^n \to \mathbf{R}$ is the *objective* of (p).

The Mathematical Programming form (p) of an optimization program is most convenient for investigating "solvability issues" in Optimization, this is why at this point we switch to optimization programs in the form of MP. Note that the optimization programs we dealt with to the moment – – the conic programs

$$c^T x \to \min \mid Ax - b \in \mathbf{K},$$

associated with a cone (closed, convex, pointed and with a nonempty interior) $\mathbf{K} \in \mathbf{R}^m$ can be easily converted to the MP format: it suffices to set

$$X(p) = \{x \mid Ax - b \in \mathbf{K}\}, p_0(x) = c^T x. \tag{5.1.1}$$

Note that the MP programs obtained from the conic ones possess a very important characteristic feature: they are *convex*.

Definition 5.1.1 *A Mathematical Programming program (p) is called convex, if*

- *The domain $X(p)$ of the program is a convex set: whenever $x, x' \in X$, the segment $\{y = \lambda x + (1-\lambda)x' \mid 0 \le \lambda \le 1\}$ linking the points x, x' also is contained in $X(p)$;*
- *The objective $p(x)$ is convex on $\mathbf{R}^{n(p)}$:*

$$\forall x, x' \; \forall \lambda \in [0,1]: \qquad p(\lambda x + (1-\lambda)x') \le \lambda p(x) + (1-\lambda)p(x').$$

One can immediately verify that (5.1.1) indeed defines a convex optimization problem.

Our interest in *convex* optimization programs, as opposed to other Mathematical Programming programs, comes from the following fact:

> (!) *Convex optimization programs are "computationally tractable": there exist solution methods which "efficiently solve" every convex optimization program satisfying "very mild" computability restrictions.*
>
> (!!) *In contrast to this, no efficient universal solution methods for nonconvex Mathematical Programming programs are known, and there are strong reasons to expect that no methods of this type exist.*

Note that even the first "positive" statement (!) to the moment is just a claim, not a theorem – we did not define yet what is a "solution method" and what does "efficiency" mean. The goal of this Lecture is to define all these notions and to convert (!) to a theorem.

5.1.2 Solution methods and efficiency

Intuitively, a (numerical) solution method is a computer code; when solving a particular optimization program, a computer loaded with this code inputs the data of the program, executes the code on these data and outputs the result – a real array representing the solution, or the message "no solution exists". The efficiency of such a solution method at a particular program is natural to measure by the *running time* of the code as applied to the data of the program – by the number of elementary operations performed by the computer when executing the code; the less is the running time, the higher is the efficiency.

When formalizing these intuitive considerations, we should specify a number of elements involved, namely

- *Model of computations:* What our computer can do, in particular, what are its "elementary operations"?
- *Encoding of program instances:* What are programs we intend to solve and what are the "data of particular programs" the computer works with?
- *Quality of solution:* Solution of what kind we expect to get? An exactly optimal or an approximate one? Even for simple convex programs, it would be unrealistic to expect that the data can be converted to an *exactly optimal* solution in *finitely many* elementary operations, so that we should ask for no more than an approximate solution. But then we should decide how to measure the quality of an approximate solution and should somehow inform the computer on the quality we expect to get.

Let us specify these elements in the way most convenient for our "subject domain" – optimization programs like Linear, Conic Quadratic and Semidefinite ones.

Model of computations. What we are about to describe is called "Real Arithmetic Model of Computations"; in order to avoid boring formalities, we restrict ourselves with a kind of a "semi-formal" description. Namely, we assume that the computations are carried out by an idealized version of the usual computer which is capable to store countably many reals and can perform operations of the standard *exact* real arithmetic with these reals – the four arithmetic operations, taking elementary functions, like cos and exp, and comparisons. Thus, as far as the logical part of executing a code is concerned, we deal with the usual von Neuman computer, and the idealization is that we assume the data stored in memory to be actual reals (not their floating point approximations), and the operations with these reals not to be subject to any rounding.

Families of optimization programs. We want to speak about methods for solving optimization programs (p) "of a given structure", like Linear, Conic Quadratic or Semidefinite ones. All programs (p) "of a given structure" form certain family $\mathcal{P}$, and we assume that every particular program in this family – every *instance* (p) of $\mathcal{P}$ – is specified by its particular *data* $\mathrm{Data}(p)$ which is simply a finite-dimensional real vector; you may think about the entries of this data vector as about particular values, corresponding to an instance, of coefficients of "generic" (given by the description of $\mathcal{P}$) analytic expressions for $p_0(x)$ and $X(p)$. This approach is in full accordance with our intuition, as is seen from the following examples.

1. *Linear Programming $\mathcal{LP}$.* Here instances (p) are all possible LP programs
$$p_0^T x \to \min \mid x \in X(p) = \{x : Ax - b \geq 0\}$$
and the data vector specifying a particular LP program (p) can be obtained by writing down, successively, first the dimensions n (number of variables) and m (number of constraints), then – n entries of the objective p, then – mn entries of the constraint matrix A (say, row by row), and finally – m entries of b.

2. *Conic Quadratic Programming $\mathcal{CQP}$.* Here instances are all possible conic quadratic programs
$$p_0^T x \to \min \mid x \in X(p) = \{x \mid \| D_i x - d_i \|_2 \leq e_i^T x - c_i, \; i = 1, ..., k\},$$
where D_i are $m_i \times \mathrm{dim}\; x$ matrices. A natural way to encode the data of a particular instance by a finite-dimensional data vector is to write down successively the sizes $n = \mathrm{dim}\; x$ (design dimension), k (number of conic quadratic inequality constraints), then – k dimensions of the constraints $m_1, ..., m_k$, then – n entries of p, and finally – the entries of (D_i, d_i, e_i, c_i), $i = 1, ..., k$.

3. *Semidefinite programming $\mathcal{SDP}$.* Here instances are all possible semidefinite programs
$$p_0^T x \to \min \mid x \in X(p) = \{x : \sum_{i=1}^n x_i A_i - B \succeq 0\}$$
with $m \times m$ symmetric matrices $A_1, ..., A_n, B$. In order to encode the data of an instance by a finite-dimensional vector, we write down successively n, m, then – n entries of p, and finally, row by row, the entries of the matrices $A_1, ..., A_n, B$.

When speaking about families of optimization programs and data vectors of instances, we always assume that the first entry in $\mathrm{Data}(p)$ is the design dimension $n(p)$ of the instance. The dimension of the vector $\mathrm{Data}(p)$ will be called the *size* of the instance:
$$\mathrm{Size}(p) = \mathrm{dim}\; \mathrm{Data}(p).$$

Accuracy of approximate solutions. Consider a generic problem $\mathcal{P}$ [1] and assume that we have somehow fixed an "infeasibility measure" of a vector $x \in \mathbf{R}^{n(p)}$ as a solution to an instance $(p) \in \mathcal{P}$, let this measure be denoted by $\mathrm{Infeas}_{\mathcal{P}}(x,p)$. In our general considerations, all we require from this measure is that

- $\mathrm{Infeas}_{\mathcal{P}}(x,p) \geq 0$, and $\mathrm{Infeas}_{\mathcal{P}}(x,p) = 0$ when x is feasible for (p) (i.e., when $x \in X(p)$);
- $\mathrm{Infeas}_{\mathcal{P}}(x,p)$ is a real-valued convex function of $x \in \mathbf{R}^{n(p)}$.

Examples:

1. *$\mathcal{LP}$ (continued):* A natural way to measure infeasibility of an $x \in \mathbf{R}^n$ as a candidate solution to an LP instance

$$(p): \qquad p_0^T x \to \min \mid Ax - b \geq 0$$

is to set

$$\mathrm{Infeas}_{\mathcal{LP}}(x,p) = \min\{t \geq 0 \mid Ax + te - b \geq 0\} \tag{5.1.2}$$

where e is the vector of ones of appropriate dimension. It is immediately seen that

$$\mathrm{Infeas}_{\mathcal{LP}}(x,p) = \max\left[0, \max_{i=1,...,m}[b_i - (Ax)_i]\right], \tag{5.1.3}$$

m being the dimension of the right hand side vector b. Thus, our infeasibility measure is just the maximum of violations of the linear constraints of the program at x.

2. *$\mathcal{CQP}$ (continued):* A natural way to measure infeasibility of an $x \in \mathbf{R}^n$ as a candidate solution to a CQP instance

$$(p): \qquad c^T x \to \min \;\| D_i x - d_i \|_2 \leq e_i^T x - c_i, \; i = 1, ..., k$$

is to set

$$\begin{array}{rcl}\mathrm{Infeas}_{\mathcal{CQP}}(x,p) & = & \min\left\{t \geq 0 : \| D_i x - d_i \|_2 \leq e_i^T x - c_i + t, \; i = 1, ..., k\right\} \\ & = & \max\left[0, \max_{i=1,...,k}[\| D_i x - d_i \|_2 - e_i^T x + c_i]\right]\end{array} \tag{5.1.4}$$

Note that geometrically this definition is very similar to the one we have used in the case of LP: in order to measure the violation of a vector inequality

$$Ax - b \geq_{\mathbf{K}} 0 \tag{V}$$

at a given point, we take a "central" interior point e of $\mathbf{K}$ and look what is the smallest nonnegative coefficient t such that after we add te to the left hand side of (V), we get a valid vector inequality. In the case of LP we have $\mathbf{K} = \mathbf{R}^m_+$, and the natural "central point" of the cone is the vector of ones, and we come to (5.1.2). In the case of $\mathbf{K} = \mathbf{L}^{m_1+1} \times ... \times \mathbf{L}^{m_k+1}$ the natural choice of e is $e = (e^1, ..., e^k)$, where $e^i \in \mathbf{R}^{m_i+1}$ has the first m_i coordinates equal to 0 and the last coordinate equal to 1 (i.e., e^i is the direction of the axis of symmetry of the corresponding Lorentz cone), and we come to (5.1.4).

[1] We use the words "generic program" as a synonym of "family of optimization programs".

3. *$\mathcal{SDP}$ (continued):* A natural way to measure infeasibility of an $x \in \mathbf{R}^n$ as a candidate solution to an instance

$$(p): \qquad c^T x \to \min \mid \mathcal{A}x - B \equiv \sum_{i=1}^{n} x_i A_i - B \succeq 0$$

is to set

$$\mathrm{Infeas}_{\mathcal{SDP}}(x,p) = \min\left\{t \geq 0 : \mathcal{A}x - B + tI \succeq 0\right\}, \tag{5.1.5}$$

I being the unit matrix of appropriate size (we again have used the above construction, taking, as the "central interior point" of a semidefinite cone $\mathbf{S}^m_+$, the unit matrix of the corresponding size).

4. *General Convex Programming problems.* Assume that the instances of a generic problem in question are of the form

$$(p): \quad p_0(x) \to \min \mid x \in X(p) = \{x \in \mathbf{R}^{n(p)} : p_i(x) \leq 0,\ i = 1, ..., m(p)\},$$

where $p_i(x) : \mathbf{R}^{n(p)} \to \mathbf{R}$, $i = 0, ..., m(p)$, are convex functions. Here a natural infeasibility measure is the maximum constraint violation

$$\mathrm{Infeas}(x,p) = \min\left\{t \geq 0 : p_i(x) \leq t,\ i = 1, ..., m(p)\right\} = \max\left[0, \max_{i=1,...,m(p)} p_i(x)\right], \tag{5.1.6}$$

cf. (5.1.2), (5.1.4).

Given an infeasibility measure, we can easily define the notion of an ϵ-solution to an instance $(p) \in \mathcal{P}$, namely, as follows. Let $\mathrm{Opt}(p) \in \{-\infty\} \cup \mathbf{R} \cup \{+\infty\}$ be the optimal value of the instance (i.e., the infimum of the values of the objective on the feasible set, if the instance is feasible, and $+\infty$ otherwise). A point $x \in \mathbf{R}^{n(p)}$ is called *an ϵ-solution* to (p), if

$$\mathrm{Infeas}_{\mathcal{P}}(x,p) \leq \epsilon \text{ and } p(x) - \mathrm{Opt}(p) \leq \epsilon,$$

i.e., if x is both "ϵ-feasible" and "ϵ-optimal" for the instance.

Solution methods. Now we can easily say what is a solution method $\mathcal{M}$ for a given family $\mathcal{P}$ of optimization programs. By definition, this is a code for our idealized Real Arithmetic computer; when solving an instance $(p) \in \mathcal{P}$, the computer first inputs the data vector $\mathrm{Data}(p)$ of the instance and a real $\epsilon > 0$ – the accuracy to which the instance should be solved, and then executes the code $\mathcal{M}$ on this input. We assume that the execution, on every input $(\mathrm{Data}(p), \epsilon > 0)$ with $(p) \in \mathcal{P}$, takes finitely many elementary operations of the computer, let this number be denoted by $\mathrm{Compl}_{\mathcal{M}}(p,\epsilon)$, and the resulting output
– either is an $n(p)$-dimensional vector $\mathrm{Res}_{\mathcal{M}}(p,\epsilon)$ which is an ϵ-solution to (p),
– or is a correct message "(p) is infeasible",
– or is a correct message "(p) is below unbounded".

Now let us define the central notion of the *complexity* of a method $\mathcal{M}$. We have agreed to measure this efficiency on the basis of the *running time* $\mathrm{Compl}_{\mathcal{M}}(p,\epsilon)$ – the number of elementary operations performed by the method when solving instance (p) within accuracy ϵ. This characteristic, however, depends on a particular choice of (p) and ϵ. And now comes the crucial step in our formalization – we are ready to say what does it mean that $\mathcal{M}$ is 'efficient"

– *polynomial time* – on $\mathcal{P}$. By definition, it means that there exists a *polynomial* $\pi(s,\tau)$ such that

$$\mathrm{Compl}_{\mathcal{M}}(p,\epsilon) \leq \pi\left(\mathrm{Size}(p), \ln\left(\frac{\mathrm{Size}(p) + \parallel \mathrm{Data}(p) \parallel_1 + \epsilon^2}{\epsilon}\right)\right) \quad \forall (p) \in \mathcal{P}\ \forall \epsilon > 0; \qquad (5.1.7)$$

here $\parallel u \parallel_1 = \sum_{i=1}^{\dim u} |u_i|$. This definition is by no means a "self-evident" way to formalize the common sense notion of an "efficient" computation method; however, there are strong reasons (taking their roots in combinatorial optimization) to use exactly this way of formalization. Let us present a transparent "common sense" interpretation of our definition. The quantity

$$\mathrm{Digits}(p,\epsilon) = \ln\left(\frac{\mathrm{Size}(p) + \parallel \mathrm{Data}(p) \parallel_1 + \epsilon^2}{\epsilon}\right) \qquad (5.1.8)$$

may be thought of as a "number of accuracy digits" in an ϵ-solution; at least this is the case when (p) is fixed and $\epsilon \to +0$, so that the numerator in the fraction under the logarithm becomes unimportant. With this interpretation, polynomiality of $\mathcal{M}$ means a very simple thing: when we increase the size of an instance and the required number of accuracy digits by absolute constant factors (say, by 2), the running time increases by no more than another absolute constant factor. Roughly speaking, when $\mathcal{M}$ is polynomial time, a constant factor times progress in the performance of our Real Arithmetic computer results in a constant factor progress in sizes of instances and numbers of accuracy digits to which these instances can be solved in a fixed time. In contrast to this, for a non-polynomial-time method, say, one with the complexity

$$\mathrm{Compl}_{\mathcal{M}}(p,\epsilon) \geq O\left(\exp\{\mathrm{Size}(p)\} f(\epsilon)\right)$$

no conclusions of this type can be done: if with the computer we have had we were able to solve, in an appropriate time, instances of, say, size 100 and now we get a 10 times faster computer, all we can hope for is the progress in sizes of "available" instances from 100 to 102. Similarly, if we are using a method with the complexity

$$\mathrm{Compl}_{\mathcal{M}}(p,\epsilon) = O\left(f(\mathrm{Size}(p))\frac{1}{\epsilon}\right),$$

and were able to achieve accuracy, say, 3 digits and now are getting a 10 times faster computer, the resulting progress in accuracy is just one digit more.

Note that for typical polynomial time methods the upper bound (5.1.7) is just linear in the number of accuracy digits:

$$\mathrm{Compl}_{\mathcal{M}}(p,\epsilon) \leq \pi\left(\mathrm{Size}(p)\right) \mathrm{Digits}(p,\epsilon) \quad \forall (p) \in \mathcal{P}\ \forall \epsilon > 0.$$

Such a complexity bound admits even more transparent interpretation: the computational effort in solving problems from $\mathcal{P}$ is proportional to the number of accuracy digits we want to get, the proportionality coefficient – the "price" of an accuracy digit – being polynomial in the size of an instance.

The final point in our formalization is the notion of a *polynomially solvable* family $\mathcal{P}$ of optimization programs: $\mathcal{P}$ is called polynomially solvable, if it admits a polynomial time solution method. Polynomial solvability of a generic optimization problem $\mathcal{P}$ is a *theoretical* synonym of "computational tractability" of $\mathcal{P}$. As far as *computational practice* is concerned, polynomiality of $\mathcal{P}$ is neither necessary, nor *sufficient* condition for "practical tractability" of the instances of $\mathcal{P}$ (simply because there cannot exist conditions of this type – whatever are the complexity bounds, the sizes of problems we can solve in practice are limited). Not every polynomial time method can be used in practice (think about a polynomial time method for solving LP programs with complexity proportional to $\mathrm{Size}^8(p)$). On the other hand, a theoretically non-polynomial method not necessary is bad in practice (the most famous exception of this type is the Simplex method for LP) – the complexity is a worst-case-oriented notion, and why should we bother that much about the worst-case in practice? The "experimental fact", however, is that for those generic optimization problems (first of all, for LP) which were for a long time routinely solved by theoretically bad, although practically efficient, methods, "theoretically good" *and* practically efficient methods eventually were discovered, so that the theoretical complexity studies finally do have strong impact on computational practice.

As it was already mentioned, the main goal of this Lecture is to demonstrate that convex optimization problems are "computationally tractable", and at this point we nearly understand what should be proved. The proof itself, however, comes from a "quite unexpected" side – from considerations which have no straightforward connection with the above chain of complexity-related definitions.

5.2 "Black box"-represented convex programs

Consider a convex program

$$f(x) \to \min \mid x \in X \subset \mathbf{R}^n, \tag{C}$$

where $f : \mathbf{R}^n \to \mathbf{R}$ is a convex function and X is a *closed and bounded* convex set *with a nonempty interior*. Moreover, assume that we know in advance that X is neither "too large", nor "too flat", namely

A. We are given in advance an $R \in (0, \infty)$ such that X is contained in the centered at the origin ball $\{x \mid \| x \|_2 \leq R\}$, and an $r > 0$ such that X contains a Euclidean ball $\{x \mid \| x - \bar{x} \|_2 \leq r\}$ (Note that only the radius r of the "small" ball is known, not the center of the ball!). [2)]

To proceed, we need to recall two important constructions of Convex Analysis.

The Separation Theorem states that if X is a nonempty convex set in $\mathbf{R}^n$ and $x \in \mathbf{R}^n \backslash X$, then x can be separated from X by a hyperplane: there exists $e \neq 0$ such that

$$e^T x \geq \sup_{y \in X} e^T y. \tag{5.2.1}$$

A *Separation Oracle* $\mathrm{Sep}(X)$ for X is a "routine" (a "black box") which, given on input a point $x \in \mathbf{R}^n$, checks whether this point belongs to X; if it is the case, the routine reports that $x \in X$, and if $x \notin X$, $\mathrm{Sep}(X)$ reports that it is the case and returns a nonzero vector e which separates x and X in the sense of (5.2.1).

Subgradient. Let $f : \mathbf{R}^n \to \mathbf{R}$ is a convex function. A vector $e \in \mathbf{R}^n$ is called a *subgradient* of f at a point $x \in \mathbf{R}^n$, if

$$f(y) \geq f(x) + e^T(y - x) \quad \forall y \in \mathbf{R}^n; \tag{5.2.2}$$

[2)] We could weaken to some extent our "a priori knowledge"; however, in our further applications the "strong" assumption **A** will be automatically satisfied.

in other words, a subgradient f at x is the "slope" of an affine function which is $\leq f$ everywhere and is equal to f at x. The set of all subgradients of f at a point x is denoted by $\partial f(x)$ and is called the *subdifferential* of f at x. A fundamental result of Convex Analysis is that *if* $f : \mathbf{R}^n \to \mathbf{R}$ *is convex, then* $\partial f(x) \neq \emptyset$ *for all* x [3]. You can immediately verify that if f is differentiable at x, then $\partial f(x)$ is a singleton comprised of the usual gradient of f at x.

A *first order oracle* $\mathcal{O}(f)$ for f is a routine (a "black box") which, given on input a point $x \in \mathbf{R}^n$, returns on output the value $f(x)$ and a subgradient $f'(x) \in \partial f(x)$ of f at x.

Assume that we are interested to solve convex program (C) and have an access to a separation oracle $\mathrm{Sep}(X)$ for the feasible domain of the problem and to a first order oracle $\mathcal{O}(f)$ for the objective. How could we act? There is an extremely transparent way to solve the problem – the *Ellipsoid method* – which is nothing but a multi-dimensional extension of the usual bisection.

Ellipsoid method: the idea. Our central observation is as follows: assume that we already have found an n-dimensional ellipsoid

$$E = \{x = c + Bu \mid u^T u \leq 1\} \qquad [B \in \mathbf{M}^{n,n}, \det(B) \neq 0]$$

which covers the optimal set X_* of (C) (note that (C) is solvable – since its feasible set is assumed to be compact, and the objective – to be convex on the entire $\mathbf{R}^n$ and therefore continuous[4]). How could we cover the same optimal set with a smaller ellipsoid?

The answer is immediate.

1) Let us call the separation oracle $\mathrm{Sep}(X)$, the center c of the current ellipsoid being the input. There are two possible cases:

a) $\mathrm{Sep}(X)$ will say that $c \notin X$ and will return a separator e:

$$e \neq 0, e^T c \geq \sup_{y \in X} e^T y. \tag{5.2.3}$$

Note that in this case we can replace our current localizer E of the optimal set X_* by a smaller one – namely, by the "half-ellipsoid"

$$\widehat{E} = \{x \in E \mid e^T x \leq e^T c\}.$$

Indeed, by assumption X_* is contained in E; when passing from E to $\widehat{E}$, we cut off all points x of E where $e^T x > e^T c$, and by (5.2.3) all these points are outside of X and therefore outside of $X_* \subset X$. Thus, $X_* \subset \widehat{E}$.

b) $\mathrm{Sep}(X)$ says that $c \in X$. In this case let us call the first order oracle $\mathcal{O}(f)$; the oracle will return us the value $f(c)$ and a subgradient $e \partial f(c)$ of f at c. Now again two cases are possible:

b.1) $e = 0$. In this case we are done – c is a minimizer of f on X. Indeed, $c \in X$, and (5.2.2) now reads

$$f(y) \geq f(c) + 0^T (y - c) = f(c) \quad \forall y \in \mathbf{R}^n.$$

[3] In fact, if f is only partially defined convex function, then $\partial f(x)$ is nonempty at every point from the relative interior of the domain of f, and you can easily prove that statement by applying the Separation Theorem to the point $(x, f(x))$ and the convex set $\{(x,t) \mid t > f(x)\}$ in $\mathbf{R}^{n+1}$; we, however, have no necessity to consider the case of a "partially defined" f.

[4] A simple fact (try to prove it) is that a function which is convex in a neighbourhood of a point x is continuous in this neighbourhood

Thus, c is a global – on the entire $\mathbf{R}^n$ – minimizer of f, and we are in the situation $c \in X$, so that c minimizes f on X as well.

b.2) $e \neq 0$. In this case (5.2.2) reads

$$e^T(x-c) > 0 \Rightarrow f(x) > f(c),$$

so that replacing the ellipsoid E with the half-ellipsoid

$$\widehat{E} = \{x \in E \mid e^T x \le e^T c\}$$

we ensure the inclusion $X_* \subset \widehat{E}$. Indeed, $X_* \subset E$ by assumption, and when passing from E to $\widehat{E}$, we cut off all points of E where $e^T x > e^T c$ and, consequently, where $f(x) > f(c)$; since $c \in X$, no one of these points may belong to the set X_* of minimizers of f on X.

c) We have seen that as a result of operations described in a), b) we either terminate with an exact minimizer of f on X, or obtain a “half-ellipsoid”

$$\widehat{E} = \{x \in E \mid e^T x \le e^T c\} \qquad [e \neq 0]$$

which contains X_*. It remains to use the following simple geometric fact:

> (*) *Let $E = \{x = c + Bu \mid u^T u \le 1\}$ $(\det B \neq 0)$ be an n-dimensional ellipsoid and $\widehat{E} = \{x \in E \mid e^T x \le e^T c\}$ $(e \neq 0)$ be a “half” of E. If $n > 1$, then the set $\widehat{E}$ is covered by the ellipsoid*
>
> $$\begin{array}{c} E^+ = \{x = c^+ + B^+ u \mid u^T u \le 1\}, \\ c^+ = c - \frac{1}{n+1} Bp, \\ B^+ = B\left(\frac{n}{\sqrt{n^2-1}}(I_n - pp^T) + \frac{n}{n+1} pp^T\right) = \frac{n}{\sqrt{n^2-1}} B + \left(\frac{n}{n+1} - \frac{n}{\sqrt{n^2-1}}\right)(Bp)p^T, \\ p = \frac{B^T e}{\sqrt{e^T B B^T e}} \end{array} \tag{5.2.4}$$
>
> *and if $n = 1$, then the set $\widehat{E}$ is covered by the ellipsoid (which now is just a segment)*
>
> $$\begin{array}{c} E^+ = \{x \mid c^+ B^+ u \mid |u| \le 1\}, \\ c^+ = c - \frac{1}{2}\frac{Be}{|Be|}, \\ B_+ = \frac{1}{2} B. \end{array}$$
>
> *In all cases, the n-dimensional volume of the ellipsoid E^+ is less than the one of E:*
>
> $$\mathrm{Vol}(E^+) = \left(\frac{n}{\sqrt{n^2-1}}\right)^{n-1} \frac{n}{n+1} \mathrm{Vol}(E) \le \exp\{-1/(2n)\}\,\mathrm{Vol}(E) \tag{5.2.5}$$
>
> *(in the case of $n = 1$, $\left(\frac{n}{\sqrt{n^2-1}}\right)^{n-1} = 1$).*

(*) says that there exists – and can be explicitly written down – an ellipsoid $E^+ \supset \widehat{E}$ with the volume constant times less than the one of E. Since E^+ covers $\widehat{E}$, and the latter set, as we have seen, covers X_*, E^+ covers X_*. Now we can iterate the above construction, thus obtaining a sequence of ellipsoids $E, E^+, (E^+)^+, \ldots$ with volumes going to 0 at a linear rate (depending on the dimension n only) which “collapses” to the set X_* of optimal solutions of our problem – exactly as in the usual bisection!

Note that (*) is just an exercise in elementary calculus. Indeed, the ellipsoid E is given as an image of the unit Euclidean ball $W = \{u \mid u^T \leq 1\}$ under the one-to-one affine mapping $u \mapsto c + Bu$; the half-ellipsoid $\widehat{E}$ is then the image, under the same mapping, of the half-ball

$$\widehat{W} = \{u \in W \mid p^T u \leq 0\}$$

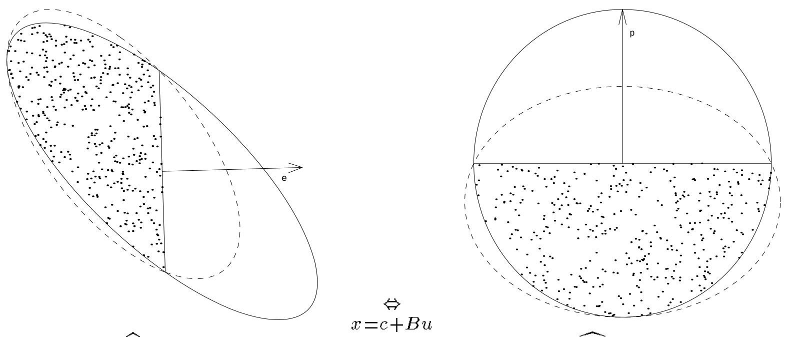
p being the unit vector from (5.2.4); indeed, if $x = c + Bu$, then $e^T x \leq e^T c$ if and only if $e^T Bu \leq 0$, or, which is the same, if and only if $p^T u \leq 0$. Now, instead of covering $\widehat{E}$ by a small in volume ellipsoid E^+, we may cover with a small ellipsoid W^+ the half-ball $\widehat{W}$ and then take E^+ to be the image of W^+ under our affine mapping:

$$E^+ = \{x = c + Bu \mid u \in W^+\}. \tag{5.2.6}$$

Indeed, if W^+ contains $\widehat{W}$, then the image of W^+ under our affine mapping $u \mapsto c + Bu$ contains the image of $\widehat{W}$, i.e., contains $\widehat{E}$. And since the ratio of volumes of two bodies remain invariant under affine mapping (passing from a body to its image under an affine mapping $u \mapsto c + Bu$, we just multiply the volume by $|\det B|$), we have

$$\frac{\mathrm{Vol}(E^+)}{\mathrm{Vol}(E)} = \frac{\mathrm{Vol}(W^+)}{\mathrm{vol}(W)}.$$

Thus, the problem of finding a "small" ellipsoid E^+ containing the half-ellipsoid $\widehat{E}$ can be reduced to the one of finding a "small" ellipsoid W^+ containing the half-ball $\widehat{W}$, as shown on the picture:



E, $\widehat{E}$ and E^+ W, $\widehat{W}$ and W^+

Now, the problem of finding a small ellipsoid containing $\widehat{W}$ is very simple: our "geometric data" are invariant with respect to rotations around the p-axis, so that we may look for W^+ possessing the same rotational symmetry. Such an ellipsoid W^+ is given by just 3 parameters: its center should belong to our symmetry axis, i.e., should be $-hp$ for certain h, one of the half-axes of the ellipsoid, let its length be μ, should be directed along p, and the remaining $n-1$ half-axes should be of the same length λ and be orthogonal to p. For our 3 parameters h, μ, λ we have 2 equations expressing the fact that the boundary of W^+ should pass through the "South pole" $-p$ of W and trough the "equator" $\{u \mid u^T u = 1, p^T u = 0\}$; indeed, W^+ should contain $\widehat{W}$ and thus – both the pole and the equator, and since we are looking for W^+ with the smallest possible volume, both the pole and the equator should be on the boundary of W^+. Using our 2 equations to express μ and λ via h, we end up with a single "free" parameter h, and the volume of W^+ (i.e., $\mathrm{const}\mu\lambda^{n-1}$) becomes an explicit function of h; minimizing this function in h, we find the "optimal" ellipsoid W^+, check that it indeed contains $\widehat{W}$ (i.e., that our geometric intuition was correct) and then convert W^+ into E^+ according to (5.2.6), thus coming to the explicit formulas (5.2.4) – (5.2.5); implementation of the outlined scheme takes from 10 to 30 minutes, depending on how many miscalculations will be done.

It should be mentioned that although the indicated scheme is quite straightforward and elementary, the fact that it works is not evident in advance: it might happen that the smallest volume ellipsoid containing a half-ball is the original ball itself! This would be the death of our idea – instead of a sequence of ellipsoids collapsing to the solution set X_*, we would get a "stationary" sequence $E, E, E...$; this pitiful possibility in fact does not take place, and this is a great favour the nature does to Convex Optimization.

Ellipsoid method: the construction. There is a small problem with implementing our idea of "trapping" the optimal set X_* of (C) by a "collapsing" sequence of ellipsoids. The only thing we can ensure is that all our ellipsoids contain X_* and that their *volumes* rapidly – at a linear rate – converge to 0. However, the linear sizes of the ellipsoids should not necessarily go to 0 – it may happen that the ellipsoids are shrinking in some directions and are not shrinking (or even become larger) in other directions (look what happens if we apply the construction to minimizing a function of 2 variables which in fact depends only on the first coordinate). Thus, to the moment it is unclear how to build a sequence of points converging to X_*. This difficulty, however, can be easily resolved: as we shall see, we can comprise this sequence of the best feasible solutions generated so far. Another issue which to the moment remains open is when to terminate the method; as we shall see in a while, this issue also can be settled properly.

The precise description of the Ellipsoid method as applied to (C) is as follows (in this description, we assume that $n \geq 2$, which of course does not restrict generality):

The Ellipsoid Method.

Initialization. Recall that when formulating (C) it was assumed that the feasible set X of the problem is contained in the ball $E_0 = \{x \mid \| x \|_2 \leq R\}$ of a given radius R and contains an (unknown) Euclidean ball of a known radius $r > 0$. The ball E_0 will be our initial ellipsoid; thus, we set

$$c_0 = 0, \ B_0 = RI, \ E_0 = \{x = c_0 + B_0 u \mid u^T u \leq 1\};$$

note that $E_0 \supset X$.

We also set

$$\rho_0 = R, \ L_0 = 0.$$

The quantities ρ_t will be the "radii" of the ellipsoids to be built, i.e., the radii of the Euclidean balls of the same volume as the one of the ellipsoids in question. The quantities L_t will be our guesses for the variation

$$\mathrm{Var}_R(f) = \max_{x \in E_0} f(x) - \min_{x \in E_0} f(x)$$

of the objective on the initial ellipsoid E_0; we shall use these guesses in the termination test.

Finally, we input the accuracy $\epsilon > 0$ to which we want to solve the problem.

Step t, $t = 1, 2, ...$ At the beginning of step t, we have the previous ellipsoid

$$E_{t-1} = \{x = c_{t-1} + B_{t-1} u \mid u^T u \leq 1\} \qquad [c_{t-1} \in \mathbf{R}^n, B_{t-1} \in \mathbf{M}^{n,n}, \det B_{t-1} \neq 0]$$

(i.e., have c_{t-1}, B_{t-1}) along with the quantities $L_{t-1} \geq 0$ and

$$\rho_{t-1} = |\det B_{t-1}|^{1/n}.$$

At step t, we act as follows (cf. the preliminary description of the method):

1) We call the separation oracle $\mathrm{Sep}(X)$, c_{t-1} being the input. It is possible that the oracle reports that $x_{t-1} \notin X$ and provides us with a separator

$$e \neq 0 : \quad e^T c_{t-1} \geq \sup_{y \in X} e^T y.$$

In this case we call step t *non-productive*, set

$$e_t = e, \;\; L_t = L_{t-1}$$

and go to rule 3) below. Otherwise – i.e., when $c_{t-1} \in X$ – we call step t *productive* and go to rule 2).

2) We call the first order oracle $\mathcal{O}(f)$, c_{t-1} being the input, and get the value $f(c_{t-1})$ and a subgradient $e \in \partial f(c_{t-1})$ of f at the point c_{t-1}. It is possible that $e = 0$; in this case we terminate and claim that c_{t-1} is an optimal solution to (C). In the case of $e \neq 0$ we set

$$e_t = e,$$

compute the quantity

$$\ell_t = \max_{y \in E_0} [e_t^T y - e_t^T c_{t-1}] = R \parallel e_t \parallel_2 - e_t^T c_{t-1},$$

update L by setting

$$L_t = \max\{L_{t-1}, \ell_t\}$$

and go to rule 3).

3) We set

$$\widehat{E}_t = \{x \in E_{t-1} \mid e_t^T x \leq e_t^T c_{t-1}\}$$

(cf. the definition of $\widehat{E}$ in our preliminary description of the method) and define the new ellipsoid

$$E_t = \{x = c_t + B_t u \mid u^T u \leq 1\}$$

by setting (see (5.2.4))

$$\begin{array}{rcl} p_t & = & \frac{B_{t-1}^T e_t}{\sqrt{e_t^T B_{t-1} B_{t-1}^T e_t}} \\ c_t & = & c_{t-1} - \frac{1}{n+1} B_{t-1} p_t, \\ B_t & = & \frac{n}{\sqrt{n^2-1}} B_{t-1} + \left(\frac{n}{n+1} - \frac{n}{\sqrt{n^2-1}}\right) (B_{t-1} p_t) p_t^T. \end{array} \tag{5.2.7}$$

We also set

$$\rho_t = |\det B_t|^{1/n} = \left(\frac{n}{\sqrt{n^2-1}}\right)^{(n-1)/n} \left(\frac{n}{n+1}\right)^{1/n} \rho_{t-1}$$

(see (5.2.5)) and go to rule 4).

4) [Termination test]. We check whether the inequality

$$\frac{\rho_t}{r} < \frac{\epsilon}{L_t + \epsilon} \tag{5.2.8}$$

is satisfied. If it is the case, we terminate and output, as the result of the solution process, the best (with the smallest value of f) of the "search points" $c_{\tau-1}$ associated with *productive* steps $\tau \leq t$ (we shall see that these productive steps indeed exist, so that the result of the solution process is well-defined). If (5.2.8) is not satisfied, we go to step $t+1$.

Just to get some feeling how the method works, here is a 2D illustration. The problem is

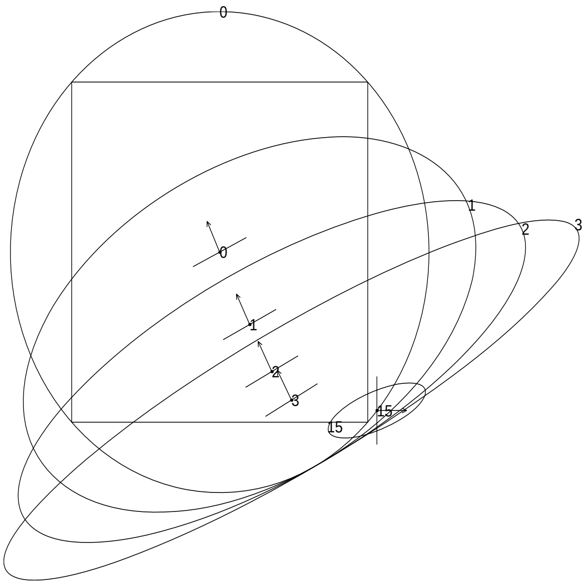
$$\begin{array}{rcl} f(x) & = & \frac{1}{2}(1.443508244x_1 + 0.623233851x_2 - 7.957418455)^2 \\ & & +5(-0.350974738x_1 + 0.799048618x_2 + 2.877831823)^4 \to \min \\ -1 \leq x_1, x_2 & \leq & 1, \end{array}$$

the optimal solution is $x_1^* = 1, x_2^* = -1$, and the exact optimal value is 70.030152768...

The best (with the smallest value of the objective) feasible solutions to the problem found in course of first t steps of the method, $t = 1, 2, ..., 256$, are shown in the following table:

t	best value	t	best value
1	374.61091739	16	76.838253451
2	216.53084103	...	...
3	146.74723394	32	70.901344815
4	112.42945457	...	...
5	93.84206347	64	70.031633483
6	82.90928589	...	...
7	82.90928589	128	70.030154192
8	82.90928589	...	...
...	...	256	70.030152768

The initial phase of the process looks as follows:



Ellipses E_{t-1} and search points c_{t-1}, $t = 1, 2, 3, 4, 16$
Arrows: gradients of the objective $f(x)$
Unmarked segments: tangents to the level lines of $f(x)$

Ellipsoid method: complexity analysis. We are about to establish our key result:

Theorem 5.2.1 *Let the Ellipsoid method be applied to convex program* (C) *of dimension $n \geq 2$ such that the feasible set X of the problem contains a Euclidean ball of a given radius $r > 0$ and* *is contained in the ball $E_0 = \{\| x \|_2 \leq R\}$ of a given radius R. For every input accuracy $\epsilon > 0$, the Ellipsoid method terminates after no more than*

$$N_{(\mathrm{C})}(\epsilon) = \mathrm{Ceil}\left(2n^2 \left[\ln\left(\frac{R}{r}\right) + \ln\left(\frac{\epsilon + \mathrm{Var}_R(f)}{\epsilon}\right)\right]\right) + 1 \tag{5.2.9}$$

steps, where

$$\mathrm{Var}_R(f) = \max_{E_0} f - \min_{E_0} f,$$

Vol *is the n-dimensional volume and* Ceil(a) *is the smallest integer $\geq a$. Moreover, the result $\widehat{x}$ generated by the method is a feasible ϵ-solution to* (C)*:*

$$\widehat{x} \in X \text{ and } f(x) - \min_X f \leq \epsilon. \tag{5.2.10}$$

Proof. We should prove the following pair of statements:

(i) The method terminates in course of the first $N_{(\mathrm{C})}(\epsilon)$ steps

(ii) The result $\widehat{x}$ is a feasible ϵ-solution to the problem.

1^0. Comparing the preliminary and the final description of the Ellipsoid method and taking into account the initialization rule, we see that if the method does not terminate before step t or terminates at this step according to rule 4), then

$$\begin{array}{llcl} (a) & E_0 & \supset & X; \\ (b) & E_\tau & \supset & \widehat{E}_\tau = \left\{x \in E_{\tau-1} \mid e_\tau^T x \leq e_\tau^T c_{\tau-1}\right\}, \ \tau = 1, ..., t; \\ (c) & \mathrm{Vol}(E_\tau) & = & \rho_\tau^n \,\mathrm{Vol}(E_0) \\ & & = & \left(\frac{n}{\sqrt{n^2-1}}\right)^{n-1} \frac{n}{n+1} \mathrm{Vol}(E_{\tau-1}) \\ & & \leq & \exp\{-1/(2n)\} \,\mathrm{vol}(E_{\tau-1}), \ \tau = 1, ..., t. \end{array} \tag{5.2.11}$$

Note that from (c) it follows that

$$\rho_\tau \leq \exp\{-\tau/(2n^2)\} R, \ \tau = 1, ..., t. \tag{5.2.12}$$

2^0. We claim that

> *If the Ellipsoids method terminates at certain step t, then the result $\widehat{x}$ is well-defined and is a feasible ϵ-solution to (C).*

Indeed, there are only two possible reasons for termination. First, it may happen that $c_{t-1} \in X$ and $f'(c_{t-1}) = 0$ (see rule 2)). From our preliminary considerations we know that in this case c_{t-1} is an optimal solution to (C), which is even more that we have claimed. Second, it may happen that at step t relation (5.2.8) is satisfied. Let us prove that the claim of 2^0 takes place in this case as well.

2^0.a) Let us set

$$\nu = \frac{\epsilon}{\epsilon + L_t} \in (0, 1].$$

By (5.2.8), we have $\rho_t / r < \nu$, so that there exists ν' such that

$$\frac{\rho_t}{r} < \nu' < \nu \quad [\leq 1]. \tag{5.2.13}$$

Let x_* be an optimal solution to (C), and X^+ be the "ν'-shrinkage" of X to x_*:

$$X^+ = x_* + \nu'(X - x_*) = \{x = (1 - \nu')x_* + \nu' z \mid z \in X\}. \tag{5.2.14}$$

We have

$$\mathrm{Vol}(X^+) = (\nu')^n \,\mathrm{Vol}(X) \geq \left(\frac{r\nu'}{R}\right)^n \mathrm{Vol}(E_0) \tag{5.2.15}$$

(the concluding inequality is given by the fact that X contains a Euclidean ball of the radius r), while

$$\mathrm{Vol}(E_t) = \left(\frac{\rho_t}{R}\right)^n \mathrm{Vol}(E_0) \tag{5.2.16}$$

by definition of ρ_t. Comparing (5.2.15), (5.2.16) and taking into account that $\rho_r < r\nu'$ by (5.2.13), we conclude that $\mathrm{Vol}(E_t) < \mathrm{Vol}(X^+)$ and, consequently, X^+ cannot be contained in E_t. Thus, there exists a point y which belongs to X^+, i.e.,

$$y = (1-\nu')x_* + \nu' z \qquad [z \in X], \tag{5.2.17}$$

and does *not* belong to E_t.

2^0.b) Since y does not belong to E_t and at the same time belongs to $X \subset E_0$ along with x_* and z (X is convex!), we see that there exists a $\tau \leq t$ such that $y \in E_{\tau-1}$ and $y \notin E_\tau$. By (5.2.11.b), every point x from the complement of E_τ in $E_{\tau-1}$ satisfies the relation $e_\tau^T x > e_\tau^T c_{\tau-1}$. Thus, we have

$$e_\tau^T y > e_\tau^T c_{\tau-1} \tag{5.2.18}$$

2^0.c) Observe that the step τ for sure is productive. Indeed, otherwise, by construction of the method, e_t would separate X from $c_{\tau-1}$, and (5.2.18) would be impossible – we know that $y \in X$! By the way, we have proved that if the method terminates at a step t, then at least one of the steps $1, ..., t$ is productive, so that the result is well-defined.

Since the step τ is productive, e_τ is a subgradient of f at x_τ (description of the method!), so that

$$f(u) \geq f(c_{\tau-1}) + e_\tau^T(u - c_{\tau-1}) \quad \forall u,$$

and in particular for $u = x_*$. On the other hand, $z \in X \subset E_0$, so that by the definition of ℓ_τ and L_τ we have

$$e_\tau^T(z - c_{\tau-1}) \leq \ell_\tau \leq L_\tau.$$

Thus,

$$\begin{array}{l} f(x_*) \geq f(c_{\tau-1}) + e_\tau^T(x_* - c_{\tau-1}) \\ L_\tau \geq e_\tau^T(z - c_{\tau-1}) \end{array}$$

Multiplying the first inequality by $(1-\nu')$, the second – by ν' and adding the results, we get

$$\begin{array}{rcl} (1-\nu')f(x_*) + \nu' L_\tau & \geq & (1-\nu')f(c_{\tau-1}) + e_\tau^T([(1-\nu')x_* + \nu' z] - c_{\tau-1}) \\ & = & (1-\nu')f(c_{\tau-1}) + e_\tau^T(y - c_{\tau-1}) \\ & & \text{[see (5.2.17)]} \\ & \geq & (1-\nu')f(c_{\tau-1}) \\ & & \text{[see (5.2.18)]} \end{array}$$

and we come to

$$\begin{array}{rcl} f(c_{\tau-1}) & \leq & f(x_*) + \frac{\nu' L_\tau}{1-\nu'} \\ & \leq & f(x_*) + \frac{\nu' L_t}{1-\nu'} \\ & & [\text{since } L_\tau \leq L_t \text{ in view of } \tau \leq t] \\ & \leq & f(x_*) + \epsilon \\ & & [\text{by definition of } \nu \text{ and since } \nu' < \nu] \\ & = & \mathrm{Opt(C)} + \epsilon. \end{array}$$

We see that *there exists a productive (i.e., with feasible $c_{\tau-1}$) step $\tau \leq t$ such that the corresponding search point $c_{\tau-1}$ is ϵ-optimal.* Since we are in the situation where the result $\widehat{x}$ is the best (with the smallest value of f) of the feasible search point generated in course of the first t steps, $\widehat{x}$ also is feasible and ϵ-optimal, as claimed in 2^0.

3^0 It remains to verify that the method does terminate in course of the first $N = N_{(\mathrm{C})}(\epsilon)$ steps. Assume, on the contrary, that it is not the case, and let us lead this assumption to a contradiction.

Observe, first, that for every productive step t we have

$$c_{t-1} \in X \text{ and } e_t = f'(c_{t-1}),$$

whence, by the definition of a subgradient and the variation $\mathrm{Var}_R(f)$,

$$u \in E_0 \Rightarrow \mathrm{Var}_R(f) \geq f(u) - f(c_{t-1}) \geq e_t^T(u - c_{t-1}),$$

whence

$$\ell_t \equiv \max_{u \in E_0} e_t^T(u - c_{t-1}) \leq \mathrm{Var}_R(f).$$

Looking at the description of the method, we conclude that

$$L_t \leq \mathrm{Var}_R(f) \qquad \forall t. \tag{5.2.19}$$

Since we have assumed that the method does not terminate in course of the first N steps, we have

$$\frac{\rho_N}{r} \geq \frac{\epsilon}{\epsilon + L_N}. \tag{5.2.20}$$

The right hand side in this inequality is $\geq \epsilon/(\epsilon + \mathrm{Var}_R(f))$ by (5.2.19), while the left hand side is $\leq \exp\{-N/(2n^2)\}R$ by (5.2.12). We get

$$\exp\{-N/(2n^2)\}R/r \geq \frac{\epsilon}{\epsilon + \mathrm{Var}_R(f)} \Rightarrow N \leq 2n^2 \left[\ln\left(\frac{R}{r}\right) + \ln\left(\frac{\epsilon + \mathrm{Var}_R(f)}{\epsilon}\right)\right],$$

which is the desired contradiction with the definition of $N = N_{(\mathrm{C})}(\epsilon)$. ■

5.3 Polynomial solvability of Convex Programming

Equipped with the Ellipsoid method, we are ready to formulate the "mild assumptions" under which a family $\mathcal{P}$ of convex optimization programs is polynomially solvable.Our assumptions are those of *polynomial computability*, *polynomial growth* and *polynomial boundedness of feasible sets.* When formulating these assumptions, we shall associate with $\mathcal{P}$ a number of positive "characteristic constants"; their particular values are of no importance for us, the only thing which counts is that these constants exist. In order to simplify notation, we denote all these constants by the same symbol χ, so that this symbol, even in different places of the same equation, may have different values (cf. the usual conventions on how one interprets symbols like $o(1)$).

Polynomial computability. Let $\mathcal{P}$ be a family of convex optimization programs, and let $\mathrm{Infeas}_{\mathcal{P}}(x,p)$ be the corresponding measure of infeasibility of candidate solutions. We say that our family is *polynomially computable*, if there exist two codes $\mathcal{C}_{\mathrm{obj}}$, $\mathcal{C}_{\mathrm{cons}}$ for the Real Arithmetic computer such that

1. For every instance $(p) \in \mathcal{P}$, the computer, given on input the data vector of the instance (p) along with a point $x \in \mathbf{R}^{n(p)}$ and executing the code $\mathcal{C}_{\mathrm{obj}}$, outputs the value $p_0(x)$ and a subgradient $e(x) \in \partial p_0(x)$ of the objective p_0 of the instance at the point x, and the running time (i.e., total number of operations) of this computation $T_{\mathrm{obj}}(x,p)$ is bounded from above by a polynomial of the size of the instance:

$$\forall\left((p) \in \mathcal{P}, x \in \mathbf{R}^{n(p)}\right): \quad T_{\mathrm{obj}}(x,p) \leq \chi \mathrm{Size}^{\chi}(p) \quad [\mathrm{Size}(p) = \dim \mathrm{Data}(p)]. \tag{5.3.1}$$

2. For every instance $(p) \in \mathcal{P}$, the computer, given on input the data vector of the instance (p), a point $x \in \mathbf{R}^{n(p)}$ and an $\epsilon > 0$ and executing the code $\mathcal{C}_{\mathrm{cons}}$, reports on output whether $\mathrm{Infeas}_{\mathcal{P}}(x,p) \leq \epsilon$, and if it is not the case, outputs a linear form e which separates the point x from all those points y where $\mathrm{Infeas}_{\mathcal{P}}(y,p) \leq \epsilon$:

$$\forall\left(y, \mathrm{Infeas}_{\mathcal{P}}(y,p) \leq \epsilon\right): \qquad e^T x > e^T y, \tag{5.3.2}$$

the running time $T_{\mathrm{cons}}(x,\epsilon,p)$ of the computation being bounded by a polynomial of the size of the instance and of the "number of accuracy digits":

$$\forall\left((p) \in \mathcal{P}, x \in \mathbf{R}^{n(p)}, \epsilon > 0\right): \quad T_{\mathrm{cons}}(x,\epsilon,p) \leq \chi\left(\mathrm{Size}(p) + \mathrm{Digits}(p,\epsilon)\right)^{\chi}. \tag{5.3.3}$$

Note that the vector e in (5.3.2) is not supposed to be nonzero; when it is 0, (5.3.2) simply says that there are no points y with $\mathrm{Infeas}_{\mathcal{P}}(y,p) \leq \epsilon$.

Polynomial growth. We say that a family $\mathcal{P}$ of convex programs equipped with an infeasibility measure $\mathrm{Infeas}_{\mathcal{P}}(x,p)$ is a family with *polynomial growth*, if the objectives and the infeasibility measures, as functions of x, grow polynomially with $\| x \|_1$, the degree of the polynomial being a power of $\mathrm{Size}(p)$:

$$\begin{array}{l}\forall\left((p) \in \mathcal{P}, x \in \mathbf{R}^{n(p)}\right): \\ |p_0(x)| + \mathrm{Infeas}_{\mathcal{P}}(x,p) \leq \left(\chi\left[\mathrm{Size}(p) + \| x \|_1 + \| \mathrm{Data}(p) \|_1\right]\right)^{(\chi \mathrm{Size}^{\chi}(p))}.\end{array} \tag{5.3.4}$$

Examples. Let us verify that the families of Linear, Conic Quadratic and Semidefinite programs equipped with the corresponding infeasibility measures (see Section 5.1.2) are polynomially computable with polynomial growth.

1. <u>$\mathcal{LP}$ *(continued):*</u> Polynomial computability means that given the data (n,m,p_0,A,b) of an LP instance

   $$(p): \qquad p_0^T x \to \min \mid Ax - b \geq 0 \quad [A: m \times n]$$

   and given an $x \in \mathbf{R}^n$, $\epsilon > 0$, we are able to compute efficiently, in the aforementioned sense,

   (a) the value of the objective at x $p_0(x) = p_0^T x$,

   (b) the subgradient of the objective $p_0(\cdot)$ at x (which is just p_0 !);

(c) whether

$$\mathrm{Infeas}_{\mathcal{LP}}(x,p) = \max\left[0, \max_{i=1,...,m}[b_i - (Ax)_i]\right]$$

is $\leq \epsilon$, and if it is not the case, to compute a vector e such that

$$e^Tx > e^Ty \quad \forall\,(y : \mathrm{Infeas}_{\mathcal{LP}}(y) \leq \epsilon\}\,. \tag{5.3.5}$$

The straightforward implementation of (a) and (b) requires χn operations, and, of course,

$$n \leq \mathrm{Size}(p) = \dim\ \mathrm{Data}(P) = 1 + n + nm + m.$$

Thus, we have no problems with $\mathcal{C}_{\mathrm{obj}}$.

To build $\mathcal{C}_{\mathrm{cons}}$, let us compute straightforwardly the value of $\mathrm{Infeas}_{\mathcal{LP}}(x,p)$ according to the explicit formula for this particular infeasibility measure; this computation requires just linear in $\mathrm{Size}(p)$ number of arithmetic operations. If the resulting value of the infeasibility measure is $> \epsilon$, so that we should compute a separator e, we also have not that much to do: we are in the situation when the residual $b_{\hat{i}} - (Ax)_{\hat{i}}$ of one of our constraints is $> \epsilon$, and we can take, as e, the minus $\hat{i}$-th row of the constraint matrix; indeed, with this choice

$$e^Tx = -(Ax)_{\hat{i}} > \epsilon - b_{\hat{i}},$$

while for every candidate solution y with $\mathrm{Infeas}_{\mathcal{LP}}(y,p) \leq \epsilon$ we have

$$e^Ty = -(Ay)_{\hat{j}} \leq \epsilon - b_{\hat{i}}.$$

Thus, both $\mathcal{C}_{\mathrm{obj}}$ and $\mathcal{C}_{\mathrm{cons}}$ can be chosen to have running time just $\chi\,\mathrm{Size}(p)$.

The fact that $\mathcal{LP}$ is of polynomial growth is evident.

2. *$\mathcal{CQP}$ (continued):* Here the instances are

$$(p): \qquad p_0^Tx \to \min \mid \parallel D_ix - d_i \parallel_2 \leq e_i^Tx - c_i,\ i = 1,...,k,$$

and the infeasibility measure is

$$\begin{array}{rcl} \mathrm{Infeas}_{\mathcal{CQP}}(x,p) & = & \max\left[0, \max\limits_{i=1,...,k}[\parallel D_ix - d_i \parallel_2 - e_i^Tx + c_i]\right] \\ & = & \max\left[0, \max\limits_{i=1,...,k} p_i(x)\right], \\ p_i(x) & = & \parallel D_ix - d_i \parallel_2 - e_i^Tx + c_i,\ i = 1,...,k. \end{array} \tag{5.3.6}$$

To verify that the family is polynomially computable, let us denote by m_i the number of rows in D_i and by n the dimension of x. Observe that

$$\mathrm{Size}(p) \geq n + \sum_{i=1}^{k}(m_i + 1)(n + 1)$$

(the right hand side is the total number of entries in p_0 and all collections (D_i, d_i, e_i, c_i), $i = 1,...,k$). Now we can build $\mathcal{C}_{\mathrm{obj}}$ and $\mathcal{C}_{\mathrm{cons}}$ as follows. Given x and ϵ, we can straightforwardly compute the value $p_0(x) = p_0^Tx$ of the objective at x, the subgradient p_0 of the objective, and the value of the infeasibility measure $\mathrm{Infeas}_{\mathcal{CQP}}(x)$ at x in $\chi\,\mathrm{Size}(p)$

operations. After $\mathrm{Infeas}_{\mathcal{CQP}}(x,p)$ is computed, we check whether this quantity is $>\epsilon$. If it is the case, we should also build a separator e. To this end let us look at the largest (at x) of the "constraints" $p_i(x)$ (see (5.3.6)), let $\widehat{\imath}$ be its index. By (5.3.6), the relation $\mathrm{Infeas}_{\mathcal{CQP}}(x,p)>\epsilon$ means exactly that $p_{\widehat{\imath}}(x)>\epsilon$, while for every y with $\mathrm{Infeas}_{\mathcal{CQP}}(y,p)\leq\epsilon$ we have $p_{\widehat{\imath}}(y)\leq\epsilon$. It follows that we can choose, as e, any subgradient of $p_{\widehat{\imath}}(\cdot)$ at the point x, since then

$$\mathrm{Infeas}_{\mathcal{CQP}}(y,p)\leq\epsilon\Rightarrow p_{\widehat{\imath}}(y)<p_{\widehat{\imath}}(x)\Rightarrow e^Ty<e^Tx,$$

the concluding $\Rightarrow$ being given by the definition of a subgradient:

$$e\in\partial f(y)\Rightarrow f(y)\geq f(x)+e^T(y-x)\ \forall y\Leftrightarrow e^T(x-y)\geq f(x)-f(y)\ \forall y.$$

On the other hand, a subgradient of every $p_i(\cdot)$ is easy to compute. Indeed, we have $p_i(x)=\parallel D_ix-d_i\parallel_2-e_i^Tx+c_i$. If x is such that $D_ix-d_i\neq 0$, then p_i is differentiable at x, so that its subgradient at x is the usual gradient

$$\nabla p_i(x)=-e_i+\frac{1}{\parallel D_ix-d_i\parallel_2}D_i^T(D_ix-d_i),$$

and it can be computed in $\chi m_in\leq\chi\,\mathrm{Size}(p)$ operations. And if $D_ix-d_i=0$, then, as it is immediately seen, just $-e_i$ is a subgradient of $p_i(x)$ at x.

Thus, $\mathcal{CQP}$ is "easily" polynomially computable – $\mathcal{C}_{\mathrm{obj}}$ and $\mathcal{C}_{\mathrm{cons}}$ can be built to have running times $\chi\,\mathrm{Size}(p)$.

The fact that $\mathcal{CQP}$ is a family with polynomial growth is evident.

3. *$\mathcal{SDP}$ (continued):* Here the instances are semidefinite programs

$$p_0^Tx\to\min\mid x\in X(p)=\{x\mid \mathcal{A}x-B\equiv\sum_{j=1}^n x_jA_j-B\succeq 0\},$$

and the infeasibility measure is

$$\mathrm{Infeas}_{\mathcal{SDP}}(x,p)=\min\left\{t\geq 0:\mathcal{A}x-B+tI\succeq 0\right\}.$$

To verify that the family if polynomially computable, observe first that if m is the row size of the matrices A_j, B, then

$$\mathrm{Size}(p)=\dim\,\mathrm{Data}(p)\geq n+(n+1)m^2 \tag{5.3.7}$$

(the right hand side is the total number of entries in $p_0,A_1,...,A_n,B$). Same as in the previous cases, given $\mathrm{Data}(p)$ and x, we have no problems with computing the value and the subgradient (which is just p_0) of our linear objective $p_0(x)=p_0^Tx$ in $\chi n\leq\chi\,\mathrm{Size}(p)$ operations, so that there is no problem with $\mathcal{C}_{\mathrm{obj}}$.

As about $\mathcal{C}_{\mathrm{cons}}$, let us start with the observation that there exists Linear Algebra algorithm $\mathcal{S}$ which, given on input a symmetric $m\times m$ matrix A, checks in $O(m^3)$ operations whether A is positive semidefinite, and if it is not the case, generates a vector ξ such that $\xi^TA\xi<0$.

As a simple example of such an algorithm $\mathcal{S}$, we may use the Lagrange scheme (explained in every Linear Algebra textbook) of representing a quadratic form $\eta^T A\eta$ as a weighted sum of squares of (linearly independent) linear forms:

$$\eta^T A\eta = \sum_{j=1}^{m} \lambda_j (q_j^T \eta)^2,$$

with m linearly independent vectors $q_1, ..., q_m$. This scheme is a simple algorithm (with running time $O(m^3)$) which converts A into the collection of weights λ_j and vectors q_j, $j = 1, ..., m$. In order to check whether a given symmetric $m \times m$ matrix A is positive semidefinite, we may run this Lagrange algorithm on A. If all resulting λ_j are nonnegative, A is positive semidefinite. And if one of λ_j, say, λ_1, turns out to be negative, we can find a vector ξ such that $q_1^T\xi = 1$, $q_j^T\xi = 0$, $j = 2, ..., m$ to get a "certificate" of the fact that A is not positive semidefinite:

$$\xi^T A\xi = \lambda_1 (q_i^T \xi)^2 = \lambda_1 < 0.$$

Note that to find ξ is the same as to solve the linear system $q_j^T\xi = \begin{cases} 1, & j=1 \\ 0, & j = 2, ..., m \end{cases}$ with a nonsingular matrix, i.e., this computation requires just $O(m^3)$ operations.

Equipped with $\mathcal{S}$, let us implement $\mathcal{C}_{\rm cons}$ as follows. Given x and $\epsilon > 0$, we compute the matrix

$$A = \sum_{j=1}^{n} x_j A_j - B + \epsilon I.$$

Note that by the definition of our infeasibility measure, $\mathrm{Infeas}_{\mathcal{SDP}}(x, p) \leq \epsilon$ if and only if A is positive semidefinite. In order to check whether this indeed is the case, we apply to A the algorithm $\mathcal{S}$. If $\mathcal{S}$ reports that $A \succeq 0$, we conclude that $\mathrm{Infeas}_{\mathcal{SDP}}(x, p) \leq \epsilon$ and stop. If A is not positive semidefinite, $\mathcal{S}$ returns a corresponding certificate – a vector ξ such that $\xi^T A\xi < 0$. Let us set

$$e = (-\xi^T A_1 \xi, ..., -\xi^T A_n \xi)^T;$$

we claim that e can be used as the separator $\mathcal{C}_{\rm cons}$ should return in the case of $\mathrm{Infeas}_{\mathcal{SDP}}(x, p) > \epsilon$. Indeed, we have

$$0 > \xi^T A\xi = \xi^T \left[\sum_j x_j A_j - B + \epsilon I\right] \xi,$$

i.e.,

$$e^T x > \xi^T[-B + \epsilon I]\xi.$$

On the other hand, for every y with $\mathrm{Infeas}_{\mathcal{SDP}}(y, p) \leq \epsilon$ the matrix $\sum_j y_j A_j - B + \epsilon I$ is positive semidefinite, so that

$$0 \leq \xi^T \left[\sum_j y_j A_j - B + \epsilon I\right] \xi,$$

whence

$$e^T y \leq \xi^T[-B + \epsilon I]\xi.$$

Thus,

$$\mathrm{Infeas}_{\mathcal{SDP}}(y,p) \le \epsilon \Rightarrow e^T y \le \xi^T[-B_\epsilon I]\xi < e^T x,$$

and e indeed is the required separator.

It remains to note that the running time of the routine $\mathcal{C}_{\rm cons}$ we have built is $\chi n m^2$ operations to compute A, χm^3 operations more to run $\mathcal{S}$ and $\chi n m^2$ operations to convert ξ into e. Thus, the running time of $\mathcal{C}_{\rm cons}$ as applied to $\mathrm{Data}(p), x, \epsilon$ does not exceed $\chi(n+m)m^2 \le \chi \mathrm{Size}^{3/2}(p)$ (see (5.3.7)).

We have seen that $\mathcal{SDP}$ is polynomially computable. The fact that the family is of polynomial growth is evident.

4. *General Convex Programming problems (continued):* Consider a family $\mathcal{P}$ of convex optimization programs with instances of the form

$$(p) \qquad p_0(x) \to \min \mid x \in X(p) = \{x \mid p_i(x) \le 0, \ i = 1,..,m(p)\}$$

$(p_i(\cdot) : \mathbf{R}^{n(p)} \to \mathbf{R}$ are convex, $i = 0, ..., m(p))$ equipped with the infeasibility measure (5.1.6):

$$\begin{array}{rcl}\mathrm{Infeas}_{\mathcal{P}}(x,p) & = & \min\{t \ge 0 : p_j(x) - t \le 0, \ j = 1, ..., m(p)\} \\ & = & \max\left[0, \max_{j=1,...,m(p)} p_j(x)\right]\end{array}$$

Assume that

I. The vector-function $p(x) = (p_0(x), ..., p_{m(p)}(x))^T$, $(p) \in \mathcal{P}$, is polynomially computable: there exists a code $\mathcal{C}$ which, given on input the data vector $\mathrm{Data}(p)$ of an instance and a point $x \in \mathbf{R}^{n(p)}$, returns the values $p_i(x)$ and subgradients $p_i'(x)$ $i = 0, ..., m(p)$ of all components of the function at x, the running time $T(p)$ of the computation being bounded by a polynomial of the size of the instance:

$$\forall (p) \in \mathcal{P} : \quad T(p) \le \chi \mathrm{Size}^{\chi}(p)$$

II. The vector-function $p(x)$ is of polynomial growth:

$$\begin{array}{l}\forall \left((p) \in \mathcal{P}, x \in \mathbf{R}^{n(p)}\right) : \\ \| p(x) \|_1 \le (\chi [\mathrm{Size}(p) + \| x \|_1 + \| \mathrm{Data}(p) \|_1])^{(\chi \mathrm{Size}^{\chi}(p))}.\end{array}$$

We claim that under these assumptions $\mathcal{P}$ is polynomially computable and is of polynomial growth. The second of these statements is evident. To verify the first of them, note that $\mathcal{C}_{\rm obj}$ is readily given by $\mathcal{C}$. The code $\mathcal{C}_{\rm cons}$ can be implemented in the same way as in the case of Linear and Conic Quadratic programs, namely, as follows. Given $\mathrm{Data}(p)$, $x \in \mathbf{R}^{n(p)}$ and an $\epsilon > 0$, we first run $\mathcal{C}$ on $\mathrm{Data}(p), x$ to get $p(x)$ and $p'(x) = \{p_i'(x)\}_{i=0}^{m(p)}$. Note that this computation, as a byproduct, gives us $m(p)$ (since $m(p)+1$ is the number of entries in the vector $p(x)$ we get); besides this, we may be sure that

$$\max[n(p), m(p)] \le \chi \mathrm{Size}^{\chi}(p). \tag{5.3.8}$$

Indeed, the running time of executing $\mathcal{C}$ – which is $\chi \mathrm{Size}^{\chi}(p)$ – cannot be less that the time required to read $n(p)$ entries of x and to write $m(p)+1$ entries of $p(x)$.

After $p(x)$ is obtained, we compute the quantity $g(x) = \max_{i=1,...,m(p)} p_i(x)$ and check whether this quantity is $\le \epsilon$. If it is the case, we report that $\mathrm{Infeas}_{\mathcal{P}}(x,p) = \max[0, g(x)]$

is $\leq \epsilon$ and stop. In the case of $g(x) > \epsilon$, we find the largest – with the value $g(x)$ – of the reals $p_i(x)$, $i = 1, ..., m(p)$, let the index of this real be $\widehat{i}$, and report, as the required separator e, the vector $p'_{\widehat{i}}(x)$. The fact that it indeed is a separator is immediate: if $\mathrm{Infeas}_{\mathcal{P}}(y,p) \leq \epsilon$, then $p_{\widehat{i}}(y) \leq \epsilon < p_{\widehat{i}}(x)$, whence, by the definition of a subgradient,

$$e^T(y-x) \leq p_{\widehat{i}}(y) - p_{\widehat{i}}(x) < 0.$$

It remains to note that all our additional to running $\mathcal{C}$ manipulations require $O(m(p))$ operations, and the latter quantity is $\leq \chi \mathrm{Size}^{\chi}(p)$ in view of (5.3.8).

The last assumption we need is as follows:

Polynomial boundedness of feasible sets. We say that a family of convex optimization problems $\mathcal{P}$ has polynomially bounded feasible sets, if the feasible set $X(p)$ of every instance $(p) \in \mathcal{P}$ is bounded and is contained in the centered at the origin Euclidean ball of "not too large" radius:

$$\begin{array}{l}\forall (p) \in \mathcal{P}: \\ X(p) \subset \left\{ x \in \mathbf{R}^{n(p)} : \| x \|_2 \leq \left(\chi \left[\mathrm{Size}(p) + \| \mathrm{Data}(p) \|_1\right]\right)^{\chi \mathrm{Size}^{\chi}(p)} \right\}.\end{array} \qquad (5.3.9)$$

Note that this assumption is <u>not</u> satisfied for typical families of convex optimization programs. E.g., given the data of an LP instance, we cannot bound the size of the feasible set by a function of the norm of the data vector; to understand that it indeed is the case, look at the subfamily of $\mathcal{LP}$ comprised of (one-dimensional!) instances

$$x \to \min \mid \delta x \geq 1$$

with $\delta > 0$.

Note, however, that we can impose the property of polynomial boundedness of feasible sets by brute force: just assuming that the description of $X(p)$ includes an explicit box constraint

$$|x_j| \leq R(p)/n^{1/2}(p), \ j = 1, ..., n(p)$$

$R(p)$ being an element of the data vector $\mathrm{Data}(p)$. Thus, given a family $\mathcal{P}$ of convex programs, we may pass from it to the family $\mathcal{P}^+$ as follows: instances (p^+) of $\mathcal{P}^+$ are pairs $((p), R)$, (p) being an instance of $\mathcal{P}$ and R being a positive real; if (p) is the optimization program

$$(p): \quad p_0(x) \to \min \mid x \in X(p) \subset \mathbf{R}^{n(p)},$$

then $(p^+) = ((p), r)$ is the optimization program

$$(p^+): \quad p_0(x) \to \min \mid x \in X(p^+) = \{x \in X(p) \mid |x_j| \leq R n^{-1/2}(p), \ j = 1, ..., n(p)\},$$

and the data vector of (p^+) is the data vector of p extended by the value of R:

$$\mathrm{Data}(p^+) = (\mathrm{Data}^T(p), R)^T.$$

Note that the resulting family $\mathcal{P}^+$ has polynomially bounded feasible sets: by construction, for every $(p^+) = ((p), R) \in \mathcal{P}^+$ we have

$$X(p^+) \subset \{x \in \mathbf{R}^{n(p)} \mid \| x \|_2 \leq R \leq \| \mathrm{Data}(p^+) \|_1\}.$$

As applied to the families like those of Linear, Conic Quadratic and Semidefinite programming, the outlined "brute force" ensuring the property we are speaking about "shrinks" the family: adding box constraints to a Linear/Conic Quadratic/Semidefinite program, we again get a program of the same structure. It follows that if we insist – and to get polynomial solvability results, we do insist – on the property of polynomial boundedness of the feasible sets, we cannot deal with the entire families $\mathcal{LP}$, $\mathcal{CQP}$, $\mathcal{SDP}$, etc., only with their subfamilies $\mathcal{LP}^+$, $\mathcal{CQP}^+$, $\mathcal{SDP}^+$,... thus restricting our subject. Note that from the viewpoint of practical computations, whatever it means, there is no restriction at all. Indeed, when solving a real-world optimization problem, we newer loose much when adding to the original formulation of the problem box constraints like $|x_j| \leq 10^{400}$, if not $|x_j| \leq 10^{12}$ – in any case, in actual computations there is no possibility to get a solution of that huge magnitude, not speaking on the fact that such a solution hardly could make practical sense.

5.3.1 Polynomial solvability of Convex Programming

We are about to establish our central result (which is the exact form of the claim (!)).

Theorem 5.3.1 *Let $\mathcal{P}$ be a family of convex optimization programs equipped with infeasibility measure* $\mathrm{Infeas}_{\mathcal{P}}(\cdot,\cdot)$. *Assume that the family is polynomially computable, with polynomial growth and with polynomially bounded feasible sets. Then $\mathcal{P}$ is polynomially solvable.*

Proof is readily given by the Ellipsoid method. Our plan is as follows: assume we are given a positive ϵ and the data vector of an instance $(p) \in \mathcal{P}$:

$$(p): \quad p_0(x) \to \min \mid x \in X(p) \subset \mathbf{R}^{n(p)}$$

and want to compute an ϵ-solution to the instance or to conclude that the instance is infeasible[5]. Since our instances are with polynomially bounded feasible sets, we can extract from $\mathrm{Data}(p)$ an a priori upper bound $R(p)$ on Euclidean norms of feasible solutions to the instance, thus converting (p) into an equivalent program

$$\overline{(p)}: \quad p_0(x) \to \min \mid x \in \bar{X}(p) = \{x \in X(p) \mid \| x \|_2 \leq R(p)\}.$$

Now, in order to find an ϵ-solution to the latter problem, it suffices to find a feasible ϵ-solution to the "augmented" problem

$$(p_\epsilon): \quad p_0(x) \to \min \mid x \in X = \{x \mid \mathrm{Infeas}_{\mathcal{P}}(x,p) \leq \epsilon, \| x \|_2 \leq R(p)\}. \tag{5.3.10}$$

A feasible ϵ-solution to the latter problem can be found by the Ellipsoid method, provided that we can equip the problem with a separation oracle for X, a first order oracle for $p_0(x)$ and to point out $r = r(p,\epsilon) > 0$ such that X contains a Euclidean ball of radius r. As we shall see in a while, our a priori assumptions on the family allow us to build all these entities in such a way that the resulting Ellipsoid-method-based solution method will be a polynomial time one.

Let us implement our plan.

[5] Since all our instances are with bounded feasible sets, we should not bother about possibility for (p) to be below unbounded.

Specifying $R(p)$**.** Since the problems of the family are with polynomially bounded feasible sets, $X(p)$ is contained in the centered at the origin Euclidean ball E_0 of the radius

$$R(p) = (\chi\,[\mathrm{Size}(p) + \parallel \mathrm{Data}(p) \parallel_1])^{\chi \mathrm{Size}^\chi(p)}, \tag{5.3.11}$$

where $\chi > 0$ is certain "characteristic constant" of $\mathcal{P}$ and therefore is known to us a priori. Given Data(p), we compute $R(p)$ according to (5.3.11), which requires a polynomial in Size(p) number of real arithmetic operations.

Specifying $r(p,\epsilon)$**.** We now need to find an $r(p,\epsilon) > 0$ in such a way that the feasible set X of the augmented problem (5.3.10) contains a ball of the radius $r(p,\epsilon)$. Interpreting this target literally, we immediately conclude that it is unachievable, since X can be empty (this is the case when (p) is "heavily infeasible" – it does not admit even ϵ-feasible solutions). We, however, can define a necessary $r(p,\epsilon)$ for the case when (p) is feasible, namely, as follows. To save notation, let us set

$$g(x) = \mathrm{Infeas}_{\mathcal{P}}(x,p).$$

From the polynomial growth property we know that both $p_0(x)$ and $g(x)$ are "not very large" in $E_0 = \{x \mid \parallel x \parallel_2 \leq R(p)\}$, namely,

$$\begin{array}{lrcl}
(a) & \mathrm{Var}_{R(p)}(p_0) & \leq & V(p), \\
(b) & g(x) & \leq & V(p) \quad \forall (x, \parallel x \parallel_2 \leq R(p)), \\
(c) & V(p) & = & (\chi\,[\mathrm{Size}(p) + \max[\parallel x \parallel_1 \mid \parallel x \parallel_2 \leq R(p)] + \parallel \mathrm{Data}(p) \parallel_1])^{(\chi \mathrm{Size}^\chi(p))} \\
 & & = & \left(\chi\left[\mathrm{Size}(p) + n^{1/2}(p)R(p) + \parallel \mathrm{Data}(p) \parallel_1\right]\right)^{(\chi \mathrm{Size}^\chi(p))},
\end{array} \tag{5.3.12}$$

χ being a "characteristic constant" of $\mathcal{P}$ and therefore being known. We compute $V(p)$ according to (5.3.12) (which again takes a polynomial in Size(p) number of operations) and set

$$r(p,\epsilon) = \frac{\epsilon}{V(p)+\epsilon} R(p). \tag{5.3.13}$$

We claim that

> (*) *If* (p) *is feasible, then the feasible set* X *of problem* (5.3.10) *contains a Euclidean ball of the radius* $r(p,\epsilon)$.
>
> Indeed, by definition of an infeasibility measure, $g(x)$ is a convex nonnegative real-valued function on $\mathbf{R}^{n(p)}$; if (p) is feasible, then g attains value 0 at certain point $\bar{x} \in E_0 \equiv \{x \mid \parallel x \parallel_2 \leq R(p)\}$. Consider the "shrinkage" of E_0 to $\bar{x}$ with coefficient $\nu = r(p,\epsilon)/R(p)$ (note that $\nu \in (0,1)$ by (5.3.13)):
>
> $$Y = (1-\nu)\bar{x} + \nu E_0 = \{x = (1-\nu)\bar{x} + \nu z \mid \parallel z \parallel_2 \leq R(p)\}.$$
>
> On one hand, Y is a Euclidean ball of radius $\nu R(p) = r(p,\epsilon)$. On the other hand, for every $x = (1-\nu)\bar{x} + \nu z \in Y$ ($\parallel z \parallel_2 \leq R(p)$) we have
>
> $$g(x) \leq (1-\nu)g(\bar{x}) + \nu g(z) \leq \nu g(z) \leq \nu V(p) \leq \epsilon$$
>
> (recall that g is convex and satisfies (5.3.12.b)).

Mimicking the oracles. The separation oracle $\mathrm{Sep}(X)$ for the feasible set X can be built as follows. Given x, it first checks whether $\| x \|_2 \leq R(p)$. If it is not the case, then already $e = x$ separates x from E_0 (and therefore form $X \subset E_0$), and $\mathrm{Sep}(x)$ reports that $x \not\in X$ and is separated from X by $e = x$. If $\| x \|_2 \leq R(p)$, the oracle calls $\mathcal{C}_{\rm cons}$, forwarding to it $\mathrm{Data}(p), x, \epsilon$. If the result returned to $\mathrm{Sep}(X)$ by $\mathcal{C}_{\rm cons}$ is the claim that $g(x) = \mathrm{Infeas}_{\mathcal{P}}(x,p)$ is $\leq \epsilon$ (i.e., if $x \in X$), $\mathrm{Sep}(X)$ reports that $x \in X$ and stops. If the result returned to $\mathrm{Sep}(X)$ by $\mathcal{C}_{\rm cons}$ is the claim that $g(x) > \epsilon$ along with a vector e such that

$$g(y) \leq \epsilon \Rightarrow e^T y < e^T x,$$

$\mathrm{Sep}(X)$ reports that $x \not\in X$ and outputs, as the required separator, either e (if $e \neq 0$), or an arbitrary vector $e' \neq 0$ (if $e = 0$). It is clear that $\mathrm{Sep}(X)$ works correctly (in particular, the case of $e = 0$ can arise only when X is empty, and in this case every nonzero vector separates x from X). Note that the running time $T_{\rm Sep}$ of $\mathrm{Sep}(X)$ (per a single call to the oracle) does not exceed $O(n(p))$ plus the running time of $\mathcal{C}_{\rm cons}$, i.e., it does not exceed $T_{\rm cons}(x,\epsilon,p) + O(n(p))$. Since $\mathcal{P}$ is polynomially computable, we have

$$n(p) \leq \chi \mathrm{Size}^{\chi}(p) \tag{5.3.14}$$

(indeed, $n(p) \leq T_{\rm obj}(x,p)$, since $\mathcal{C}_{\rm obj}$ should at least read $n(p)$ entries of an input value of x). Combining (5.3.14) and (5.3.3), we conclude that

$$T_{\rm Sep} \leq \chi \left(\mathrm{Size}(p) + \mathrm{Digits}(p,\epsilon)\right)^{\chi}. \tag{5.3.15}$$

The first order oracle $\mathcal{O}(p_0)$ is readily given by $\mathcal{C}_{\rm obj}$, and its running time $T_{\mathcal{O}}$ (per a single call to the oracle) can be bounded as

$$T_{\mathcal{O}} \leq \chi \mathrm{Size}^{\chi}(p), \tag{5.3.16}$$

see (5.3.1).

Running the Ellipsoid method. After we have built $R(p)$, $r(p,\epsilon)$, $\mathrm{Sep}(X)$ and $\mathcal{O}(p_0)$, we may apply to problem (5.3.10) the Ellipsoid method as defined above. The only precaution we should make is what will happen if X *does not* contain a ball of the radius $r(p,\epsilon)$; this may happen only in the case when (p) is infeasible (see (*)), but how could we know whether (p) is or is not feasible? To resolve the difficulty, let us act as follows. If (p) is feasible, then, by Theorem 5.2.1, the Ellipsoid method would terminate after no more than

$$\mathrm{Ceil}\left(2n^2(p)\left[\ln\left(\frac{R(p)}{r(p,\epsilon)}\right) + \ln\left(\frac{\epsilon + \mathrm{Var}_{R(p)}(p_0)}{\epsilon}\right)\right]\right) + 1$$

steps and will produce a feasible ϵ-solution to (5.3.10), i.e., an ϵ-solution to (p). The indicated number of steps can be bounded from above by the quantity

$$N(p,\epsilon) = \mathrm{Ceil}\left(2n^2(p)\left[\ln\left(\frac{R(p)}{r(p,\epsilon)}\right) + \ln\left(\frac{\epsilon + V(p)}{\epsilon}\right)\right]\right) + 1, \tag{5.3.17}$$

since $\mathrm{Var}_{R(p)}(p_0) \leq V(p)$ by (5.3.12.a). Let us "by force" terminate the Ellipsoid method if it intends to perform more than $N = N(p,\epsilon)$ steps. When using this "emergency stop", we define the result generated by the method as the best (with the smallest value of $p_0(\cdot)$) of the search points c_{t-1} associated with productive steps $t \leq N$, if there were productive steps; if no productive steps in course of our run are encountered, the result of the solution process is the conclusion that (p) is infeasible.

Correctness. We claim that the outlined implementation of the Ellipsoid method is correct – i.e., when the result of it is an approximate solution $\widehat{x}$, this is an ϵ-solution to (p), and is the result is the conclusion "(p) is infeasible", this conclusion is true. Indeed, if (p) is feasible, then the arguments used in the previous paragraph demonstrate that $\widehat{x}$ is well-defined and is an ϵ-solution of (p). If (p) is infeasible, then the result, by construction, is either the correct conclusion that (p) is infeasible, or a point $\widehat{x}$ such that $\mathrm{Infeas}_{\mathcal{P}}(\widehat{x}, p) \leq \epsilon$; such a point, in the case of *infeasible* (p), is an ϵ-solution of (p), since in the case in question $\mathrm{Opt}(p) = +\infty$ and therefore $p_0(x) \leq \mathrm{Opt}(p) + \epsilon$ for every x.

Polynomiality. It remains to verify that the running time of the solution method we have presented is as it should be for a polynomial time method. Observe, first, that all "preliminary" computations we need – those to specify $R(p)$, $V(p)$, $r(p,\epsilon)$, $N(p,\epsilon)$ – require no more than $\chi\mathrm{Size}^{\chi}(p)$ operations (we have already seen that this is the case for $R(p)$, $V(p)$ and $r(p,\epsilon)$; given these quantities, it takes just χ operations to compute $N(p,\epsilon)$). Thus, all we need is to get a polynomial time bound for the running time of the Ellipsoid method. This is immediate: the method performs no more than $N(p,\epsilon)$ steps, and the "arithmetic cost" of a step does not exceed the quantity

$$T = T_{\mathrm{Sep}} + T_{\mathcal{O}} + \chi n^2(p),$$

the concluding term in the right hand side representing the arithmetic cost of updating (5.2.7), computing ℓ_t and all other "out-of-oracles" operations required by the method. Thus, the overall running time $T(p,\epsilon)$ of our solution method can be bounded as

$$\begin{array}{rcl}
T(p,\epsilon) & \leq & \chi\mathrm{Size}^{\chi}(p) + N(p,\epsilon)\left[T_{\mathrm{Sep}} + T_{\mathcal{O}} + \chi n^2(p)\right] \\
& \leq & \chi\mathrm{Size}^{\chi}(p) + N(p,\epsilon)[\chi\,(\mathrm{Size}(p) + \mathrm{Digits}(p,\epsilon))^{\chi} + \chi\mathrm{Size}^{\chi}(p)] \\
& & \text{[we have used (5.3.15), (5.3.16) and (5.3.14)]} \\
& \leq & \chi\mathrm{Size}^{\chi}(p)\mathrm{Digits}^{\chi}(p,\epsilon) \\
& & \text{[see (5.1.8), (5.3.17), (5.3.13), (5.3.12)]}
\end{array}$$

so that the method indeed is a polynomial time one. ■

5.4 Difficult problems and NP-completeness

The most basic motivation for our "convexity-oriented" approach to optimization is that, as was announced in Preface and as we know now, Convex Optimization programs are "computationally tractable". In several places we also claimed that such and such problems are "hard" of "computationally intractable". What do these words actually mean? Without answering this question, a lot of our activity would become seemingly senseless: e.g., why should we bother about semidefinite relaxations of combinatorial problems like MAXCUT? What is wrong with these problems as they are? If we claim that something – e.g., Convex Programming – is "good", we should understand what does "bad" mean: "good", at least on the Earth and in science, is a relative notion...

To understand what does "computational intractability" mean, let us outline briefly the basic results of CCT – *Combinatorial Complexity Theory.*

5.4.1 CCT – quick introduction

A generic combinatorial problem, like a generic optimization problem, is a family $\mathcal{P}$ of *problem instances*, each instance $(p) \in \mathcal{P}$ being specified by a finite-dimensional *data vector*

Data(p). However, the data vectors now are assumed to be *Boolean vectors* – with entries taking values $0, 1$ only, so that the data vectors are, actually, finite binary words.

The model of computations in CCT also is more restrictive (and, in a sense, more realistic) that the Real Arithmetic model we dealt with; now our computer is capable to store only integers (i.e., finite binary words), and its operations are *bitwise*: we are allowed to multiply, add and compare integers, but now the "cost" of a single operation of this type depends on the bit length of the operands: to add and to compare two ℓ-bit integers, it takes $O(\ell)$ "bitwise" elementary operations, and to multiply a pair of ℓ-bit integers it costs $o(\ell^2)$ elementary operations [6]

In CCT, a solution to an instance (p) of a generic problem $\mathcal{P}$ is a finite binary word y such that the pair $(\text{Data}(p), y)$ satisfies certain "verifiable condition" $\mathcal{A}(\cdot,\cdot)$. Namely, it is assumed that there exists a code $\mathcal{M}$ for the above "Integer Arithmetic computer" such that executing the code on every input pair x, y of finite binary words, the computer after finitely many elementary operations terminates and outputs either "yes", if $\mathcal{A}(x, y)$ is satisfied, or "no", if $\mathcal{A}(x, y)$ is not satisfied.

For example, the *Shortest Path* problem:

> *Given a graph with arcs assigned nonnegative integer lengths, two nodes a, b in the graph and a positive integer d, find a path from a to b of total length not exceeding d, or detect that no such path exists*

or the problem *Stones* from Lecture 4:

> *Given n positive integers $a_1, ..., a_n$, find a vector $x = (x_1, ..., x_n)^T$ with coordinates ± 1 such that $\sum_i x_i a_i = 0$, or detect that no such vector exists*

are generic combinatorial problems. Indeed, the data of instances of both problems, same as candidate solutions to the instances, can be naturally encoded by finite sequences of integers. In turn, finite sequences of integers can be easily encoded by finite binary words – you just encode binary digit `0` of an integer as `00`, binary digit `1` – as `11` and use `01` and `10` to represent the commas separating integers in the sequence from each other and to represent the sign `-`, respectively:

$$5, -3, 7 \Rightarrow \underbrace{110011}_{101_2=5}\underbrace{01}_{,}\underbrace{10}_{-}\underbrace{1111}_{11_2=3}\underbrace{01}_{,}\underbrace{111111}_{111_2=7}$$

And of course for every one of these two problems you can easily point out a code for the "Integer Arithmetic computer" which, given on input two binary words $x = \text{Data}(p)$, y encoding the data vector of an instance (p) of the problem and a candidate solution, respectively, verifies in finitely many "bit" operations whether y represents or does not represent a solution to (p).

A solution algorithm for a generic problem $\mathcal{P}$ is a code $\mathcal{S}$ for the Integer Arithmetic computer which, given on input the data vector Data(p) of an instance $(p) \in \mathcal{P}$, after finitely many operations either returns a solution to the instance, or a (correct!) claim that no solution exists. The running time $T_{\mathcal{S}}(p)$ of the solution algorithm on instance (p) is exactly the number of elementary (i.e., bit) operations performed in course of executing $\mathcal{S}$ on Data(p).

[6] In fact two ℓ-bit integers can be multiplied in $O(\ell \ln \ell)$ bitwise operations, but for us it makes no difference; the only fact we need is that the "bitwise cost" of operations with integers is at least the bit size and at most a fixed polynomial of the bit size of the operands.

A *solvability test* for a generic problem $\mathcal{P}$ is defined similarly to a solution algorithm, but now all we want of the code is to say (correctly!) whether the input instance is or is not solvable, just "yes" or "no", without constructing a solution in the case of the "yes" answer.

The *complexity* of a solution algorithm/solvability test $\mathcal{S}$ is defined as

$$\mathrm{Compl}_{\mathcal{S}}(\ell) = \max\{T_{\mathcal{S}}(p) \mid (p) \in \mathcal{P}, \mathrm{length}(\mathrm{Data}(p)) \leq \ell\},$$

where $\mathrm{length}(x)$ is the bit length (i.e., number of bits) of a finite binary word x. The algorithm/test is called *polynomial time*, if its complexity is bounded from above by a polynomial of ℓ.

Finally, a generic problem $\mathcal{P}$ is called to be *polynomially solvable*, if it admits a polynomial time solution algorithm. If $\mathcal{P}$ admits a polynomial time solvability test, we say that $\mathcal{P}$ is *polynomially verifiable.*

Classes P and NP. A generic problem $\mathcal{P}$ is said to belong to the class P, if it is polynomially solvable.

A generic problem $\mathcal{P}$ is said to belong to the class NP, if the corresponding condition $\mathcal{A}$ possesses the following two properties:

> I. $\mathcal{A}$ is polynomially computable, i.e., the running time $T(x,y)$ (measured, of course, in elementary "bit" operations) of the associated code $\mathcal{M}$ is bounded from above by a polynomial of the *binary length* $\mathrm{length}(x)+\mathrm{length}(y)$ of the input:
>
> $$T(x,y) \leq \chi(\mathrm{length}(x)+\mathrm{length}(y))^{\chi} \quad \forall (x,y).$$
>
> Thus, the first property of an NP problem states that *given data* $\mathrm{Data}(p)$ *of a problem instance* p *<u>and</u> a candidate solution* y*, it is easy to check whether* y *is an actual solution of* (p) – to verify this fact, it suffices to compute $A(\mathrm{Data}(p),y)$, and this computation requires polynomial in $\mathrm{length}(\mathrm{Data}(p))+\mathrm{length}(y)$ time.
>
> The second property of an NP problem makes its instances even more easier:
>
> II. A solution to an instance (p) of a problem cannot be "too long" as compared to the data of the instance: there exists χ such that
>
> $$\mathrm{length}(y) > \pi(\mathrm{length}(x)) \equiv \chi \mathrm{length}^{\chi}(x) \Rightarrow \mathcal{A}(x,y) = \text{"}\mathtt{no}\text{"}.$$

Note that there is no problem to build a "brute force" solution algorithm for an NP problem: given $\mathrm{Data}(p)$, you just look successively at finite binary words "0","1","00","01","10","11",... and compute $A(\mathrm{Data}(p),y)$, y being the current candidate solution, to check whether y is a solution to (p). If it is the case, you stop, otherwise pass to the next candidate solution. After all candidates of length $\leq \pi(\mathrm{length}(x))$ are checked and no solution is found, you conclude that (p) has no solution and terminate. Of course, this brute force algorithm is not polynomial – its complexity exceeds $2^{\pi(\ell)}$.

As it is immediately seen, both Shortest Path and Stones problems belong to NP. There is, however, a dramatic difference between these two problems: the first one is polynomially solvable, while for the second no polynomial time solution algorithms are known. Moreover, the second problem is "as difficult as a problem from NP can be" – it is NP-complete.

Definition 5.4.1 (i) *Let $\mathcal{P}$, $\mathcal{Q}$ be two problems from* NP. *Problem $\mathcal{Q}$ is called to be polynomially reducible to $\mathcal{P}$, if there exists a polynomial time algorithm $\mathcal{M}$ (i.e., a code for the Integer Arithmetic computer with the running time bounded by a polynomial of the length of the input) with the following property. Given on input the data vector* $\mathrm{Data}(q)$ *of an instance* $(q) \in \mathcal{Q}$, $\mathcal{M}$ *converts this data vector to the data vector* $\mathrm{Data}(p[q])$ *of an instance of $\mathcal{P}$ such that $(p[q])$ is solvable if and only if (q) is solvable.*

(ii) *A generic problem $\mathcal{P}$ from* NP *is called* NP*-complete, if every other problem $\mathcal{Q}$ from* NP *is polynomially reducible to $\mathcal{P}$.*

One of the most basic results of Theoretical Computer Science is that NP-*complete problems do exist* (the Stones problem is an example).

The importance of the notion of an NP-complete problem comes from the following fact:

> (!!!) *If a particular* NP*-complete problem is polynomially verifiable (i.e., admits a polynomial time solvability test), then every problem from NP is polynomially solvable:* P = NP.

(!!!) is an immediate consequence of the following two observations:

(A) *If there exists an* NP*-complete problem $\mathcal{P}$ which is polynomially verifiable, then every problem from* NP *is polynomially verifiable.*

Indeed, under the premise of our statement, we can build a polynomial time solvability test for a problem $\mathcal{Q}$ from NP as follows: given an instance $(q) \in \mathcal{Q}$, we first apply the corresponding polynomial time "reduction algorithm" $\mathcal{M}$ to convert $\mathrm{Data}(q)$ into $\mathrm{Data}(p[q])$, see item (i) of Definition 5.4.1. Let $\theta(\cdot)$ be a polynomial such that the running time of algorithm $\mathcal{M}$ on input of length $l = 1, 2, \ldots$ does not exceed $\theta(l)$. The quantity $\mathrm{length}(\mathrm{Data}(p[q]))$ is bounded by the running time of $\mathcal{M}$ on the input $\mathrm{Data}(q)$ – it takes one bit operation just to write down a single output bit. Since $\mathcal{M}$ is polynomial, we conclude that $\mathrm{length}(\mathrm{Data}(p[q])) \le \theta(\ell)$, where $\ell = \mathrm{length}(\mathrm{Data}(q))$. After $\mathrm{Data}(p[q])$ is built, we run the polynomial time solvability test associated with $\mathcal{P}$ (we have assumed that it exists!) to check whether $(p[q])$ is solvable, thus detecting solvability/unsolvability of (q). The running time of this latter computation is bounded by $\pi(\mathrm{length}(\mathrm{Data}(p[q]))) \le \pi(\theta(\ell))$, π being a polynomial. Thus, the overall running time does not exceed $\theta(\ell) + \pi(\theta(\ell))$, which is a polynomial in $\ell = \mathrm{length}(\mathrm{Data}(q))$.

(B) *If every problem from* NP *is polynomially verifiable, every problem from* NP *is polynomially solvable as well.*

The idea of the proof (we skip the technical details) is as follows: assume we are interested to solve an instance $(p) \in \mathcal{P}$ ($\mathcal{P}$ is in NP) and have a polynomial time solvability test for the problem. We first check in polynomial time whether (p) is solvable. If the answer is "no", this is all we need; if the answer is "yes", we should find a solution itself. Applying our solvability test, we can decide in polynomial time whether (p) has a solution with the first bit 0. Whatever the answer will be, we will get the first bit y_1 of (some) solution: is the answer is "yes", this bit is 0, otherwise it is 1. Since $\mathcal{P}$ is in NP, we can check in polynomial time whether the single-bit word $y = y_1$ is a solution. If the answer is "no", we proceed in the same way: run our solvability test to check whether (p) has a solution with the first bit y_1 and the second bit 0, thus getting the first *two* bits of a solution, check whether the resulting two-bit word is a solution, and so on. Since the length of all possible solutions to (p) is bounded from above by a polynomial of the length of Data(p) ($\mathcal{P}$ is in NP!), we shall build a solution in a polynomial in $\ell = \text{length}(\text{Data}(p))$ number of calls to the solvability test and the "verification test" (the latter verifies whether a given y solves (p)), running time per call being bounded by a polynomial in ℓ, and the overall running time of building a solution to (p) turns out to be polynomial in ℓ.

As we have already mentioned, NP-complete problems do exist. Moreover, a throughout investigation of combinatorial problems carried out during last three decades (i.e., after the notion of an NP-complete problem and existence of these problems were discovered) demonstrates that

> *Basically all interesting combinatorial problems are in* NP*, and nearly all of them which were thought of to be "difficult" – with no known polynomial time solution algorithms – are* NP*-complete. The list of* NP*-complete problems includes Integer and Boolean Linear Programming, the Traveling Salesman problem,* MAXCUT *and hundreds of other combinatorial problems, including that "simply-looking" ones as the Stones problem.*

There are, essentially, just two important generic combinatorial problems from NP which were not known to be polynomially solvable in 1970 and still are not in the list of NP-complete problems. The first of them is the graph isomorphism problem – given two graphs, decide whether they are isomorphic – whether there exist one-to-one correspondences between the set of nodes and the set of arcs of the graphs which preserve the node-arc incidence; the "complexity status" of this problem is still unknown. The second one is Linear Programming over rationals (i.e., Linear Programming with rational data). An LP program with rational data, if solvable, admits a rational solution, so that this family of problems can be treated as a generic combinatorial problem. In 1978 L. Khachiyan proved that LP over rationals is polynomially solvable.

Note that it still is unknown whether P=NP – whether NP-complete problems are or are not polynomially solvable; this question (which is qualified as "the most important open problem in Theoretical Computer Science") remains open for about 30 years.

Although we do not know whether P=NP – in a sense, whether there indeed exist difficult combinatorial problems – at the "practical level" the answer seems to be clear. Indeed, there are a lot of NP-complete problems; some of them, like Integer and Boolean Linear Programming programs and their numerous special cases – are of huge practical importance and are therefore subject, over decades, of intensive studies of thousands excellent researchers both in Academy and in Industry. However, no polynomial time algorithm for any one of these problems was found. With the discovery of the theory of NP-completeness it became clear that in a sense all research in the area of solution methods for combinatorial programs deals with a *single* problem (since polynomial time solvability of a particular NP-complete problem a researcher is focusing on would automatically imply polynomial time solvability of the problems attacked by all other researchers). Given the huge total effort invested in this research, we should conclude that it is "highly unprobable" that NP-complete problems are polynomially solvable. Thus, at the "practical level" the fact that certain problem is NP-complete is sufficient to qualify the problem as "computationally intractable", at least at our present level of knowledge.

5.4.2 From the Real Arithmetic Complexity theory to the CCT and back

The two complexity theories we have outlined – the one based on the Real Arithmetic computer and interested in finding ϵ-solutions for problems with real data, and the other one based on the Integer Arithmetic computer and interested in finding exact solutions to problems with binary data – are similar, but in no sense identical. To stress the difference, consider a simple computational problem – just one of solving a square system of linear equations

$$Ax = b \qquad (A, b)$$

with $n = n(A, b)$ unknowns and a nonsingular matrix A. The Real Arithmetic Complexity Theory will qualify the problem as follows:

> (R) *The family $\mathcal{L}$ comprised of problem instances (A, b) with real entries in A, b and nonsingular A is polynomially solvable: there exists an algorithm (e.g., the Gauss elimination method) which, as applied to any instance, finds the exact solution of the instance in no more than $O(n^3(A, b))$ operations of exact real arithmetic, which is a polynomial of the dimension* $\mathrm{Size}(A, b) = n^2(A, b) + n(A, b)$ *of the data vector of the instance.*

The CCT first will reject to speak about systems of linear equations with *real* data, since in this case neither the data, nor candidate solutions can be encoded by finite binary words. However, CCT is ready to speak about systems with *rational* data, and will qualify these systems as follows:

> (C) *The family $\mathcal{L}$ comprised of problem instances (A, b) with rational entries in A, b and nonsingular A is polynomially solvable: there exists a solution algorithm (e.g., the Gauss elimination method) which, as applied to any instance, finds an exact solution of the instance in no more than a polynomial $\pi(\ell)$ of bit operations, ℓ being the length of the instance data – the total number of bits in binary representations of the numerators and denominators of all entries of the data.*

We see that these two statements are in a "general position" – they say different things about different entities, and no one of them is a consequence of the other one; a priori it might happen that one of these statements is true and another is false, or both are false, or both are true (which indeed is the case). The essence of the difference comes not from the fact that (R) speaks about all systems, while (C) – only about the systems with rational data – this is a minor point. The essence of the difference is that in (C) an "elementary operation" is a *bit* operation, while in (R) it is an operation with reals; thus, (C), as compared to (R), deals with much more restricted set of elementary operations and therefore – with much more "strict" (and more realistic) notion of "computational effort". As a kind of compensation, (C) uses a less strict notion of a polynomial time method than (R): for (C), a method is polynomial if its running time (measured in "bit" operations) is bounded by a polynomial of the *total binary length of the data*, while for (R) this running time (measured in the number of Real Arithmetic operations) should be bounded by a polynomial of the *number of data entries* – of a quantity which is definitely less than the binary length of the data. E.g., when (C) says that systems of linear equations are polynomially solvable, it says nothing definite about the complexity of solving a system of two linear equations with two variables: the "bitwise" complexity of this simplest computational problem can be as large as you wish, provided that the coefficients are rationals with large numerators and denominators. In contrast to this, when (R) says that systems of linear equations are polynomially solvable, it says, in particular, that a system of two linear equations with two unknowns can be solved in $O(1)$ operations of real arithmetic.

Although the two outlined complexity theories, as we just have seen, are "in general position", every one of them can utilize (and does utilize) some results of its "counterpart". Sometimes a "Real Arithmetic polynomial time method", as restricted to a family of problems with rational data, can be converted to a CCT-polynomial time algorithm for a combinatorial problem, thus yielding CCT-polynomial solvability of this problem. "Borrowing efficient algorithms" in the opposite direction – from combinatorial problems to those with real data – does not make much sense; what the Real Arithmetic Complexity Theory does borrow from CCT, are techniques to recognize "computationally intractable" computational problems. In our course we are "at the side of optimization programs with real data", so that our primary interest is what can be borrowed from the CCT, not given to it. It makes sense, however, to start with an example of what can be *given* to the CCT.

CCT-polynomial solvability of Linear Programming over rationals

Let us start with some historical remarks. Linear Programming in all its major methodological and computational aspects was discovered by G. Dantzig in late 40's; in particular, he proposed the Simplex method for LP (1948) which for about 40 years was the only one – and extremely efficient – practical tool for solving LP programs; it still is one of the most efficient computational techniques known for LP and the most frequently used one. The theoretical justification of the method is that in the Real Arithmetic model of computations it finds an exact solution to any LP program (or detects correctly that no solution exists) in finitely many arithmetic operations, while its practical justification is that this number of operations typically is quite moderate. However, it was discovered that the worst-case complexity of the Simplex method is very bad: Klee and Minty (1964) have built a sub-family of LP programs $\{(p_n)\}_{n=1}^{\infty}$ with the following property: (p_n) has rational data, and the "size" of the instance (p_n) is polynomial in n, whether we measure the size as the number of data entries or as their total bit length; and the number of arithmetic operations performed by the Simplex method as applied to (p_n) is more than 2^n. Thus, the Simplex method is not polynomial time neither in the sense of Real Arithmetic Complexity theory, nor in the sense of the CCT. For about 15 years, the question whether LP is or is not polynomially solvable, was one of the major challenges in the area of computational complexity. The "CCT-version" of this question was answered affirmatively by L. Khachiyan in 1978, and the tool he used was borrowed from "convex optimization with real data" – it was the Ellipsoid method (1976). The sketch of Khachiyan's construction is as follows.

Linear Feasibility Problem. Let us start with the *LP feasibility problem*:

(FeasLP) *Given a system of linear inequalities*

$$Ax \leq b \tag{S}$$

with rational coefficients, check whether the system has a solution.

The polynomial time (in the sense of CCT!) algorithm for (FeasLP) proposed by Khachiyan is as follows.

We may assume without loss of generality that the columns of A are linearly independent (if it is not the case, we do not affect feasibility by eliminating columns which are linear combinations of the remaining ones, and this Linear Algebra operation takes time polynomial in the total bit length L of the data and results in a system of the same structure and with total bit length of the data being polynomial in L). Besides this, it is convenient to assume the data integer (we may multiply all coefficients involved in a particular inequality by their common denominator; this equivalent transformation results in a problem with the total bit length of the data polynomial in the one of the original problem). Thus, we may assume that the data in (S) are integer and the columns in A are linearly independent. Let L be the total bit length of the data, m be the number of inequalities, and n be the number of unknowns in (S).

The first step is to get an a priori upper bound on the norm of a solution to (S), assuming that such a solution exists. This can be done as follows.

It is a well-known fact of Linear Programming/Convex Analysis that if A has linearly independent columns and (S) has a solution, then (S) has an *extreme point* solution x as follows: n of the m inequalities of (S) at x are equalities, and the rows of A corresponding to these inequalities are linearly independent. In other words, x is the unique solution of the square system of linear equations

$$\widehat{A}x = \widehat{b}$$

where $\widehat{A}$ is a square nonsingular $n \times n$ submatrix of A and $\widehat{b}$ is the corresponding subvector of b. From the Cramer rules it follows that every coordinate in x is the ratio $\frac{\Delta'}{\Delta}$ of two determinants, with $\Delta = \det \widehat{A}$ and Δ' being the determinant of an $n \times n$ submatrix of the matrix $[\widehat{A}; \widehat{b}]$. Δ is a nonzero integer (all entries in A, b are integer!), and Δ' is not too large – the absolute value of a determinant does not exceed the product of Euclidean lengths of its rows (Hadamard's inequality expressing an evident geometric fact: the volume of a parallelotope does not exceed the product of its edges). Since the sum of binary lengths of all entries in A, b does not exceed L, the above product cannot be very large; a simple computation demonstrates that it does not exceed $2^{O(1)L}$ (all $O(1)$'s are absolute constants). We conclude that the absolute values of all entries in x do not exceed $2^{O(1)L}$, so that $\| x \|_2 \leq 2^{O(1)L}\sqrt{n} \leq 2^{O(1)L}$ (we have used the evident fact that both n and m do not exceed L – it takes at least one bit to represent every one of the mn entries of A).

The second step is to look at the minimum value g_* of the residual

$$g(x) = \max\left[0, \max_{i=1,...,m}[(Ax)_i - b_i]\right];$$

note that this minimum value is 0 if (S) is solvable and is > 0 otherwise. What we are interested in is to understand whether g_* can be positive, but "very small". The answer is "no": if $g_* > 0$, then $g_* > 2^{-O(1)L}$.

Indeed, g_* is the optimal value in the feasible LP program

$$t \to \min \mid t \geq 0, (Ax)_i - b_i \leq t, \ i = 1, ..., m. \qquad \mathrm{S}'$$

The binary length L' of the data in this problem is of oder of L, and the problem for sure is solvable. From the theory of Linear Programming it is well-known that if an LP has a solution, then it has an extreme point, in the aforementioned sense, solution. The coordinates of the latter solution, in particular, its t-coordinate (i.e., the optimal value in (S$'$), i.e., g_*) again are ratios of determinants, now coming from the matrix $[A; b; e]$, e being the vector of ones. Thus, g_* is the ratio of two integers and these integers, same as above, do not exceed $2^{O(1)L'} = 2^{O(1)L}$. It follows that if $g_* > 0$, then $g_* \geq 2^{-O(1)L}$ – the numerator in the ratio representing g_*, being nonzero integer, should be at least one, and the denominator cannot be larger than $2^{O(1)L}$.

The third step. We already know that if (S) is feasible, the minimum of g in the ball $E_0 = \{x \mid \| x \|_2 \leq 2^{O(1)L}\}$ is zero (since then this ball contains a solution to (S)). We know also that if (S) is infeasible, then the minimum of g in E_0 is at least $2\epsilon = 2^{-O(1)L}$, since in the case in question already the minimum of g on the entire $\mathbf{R}^n$ admits the indicated lower bound. It follows that in order to check feasibility of (S) it suffices to find an ϵ-solution x_ϵ to the optimization problem

$$g(x) \to \min \mid x \in E_0; \tag{C}$$

if the value of g at x_ϵ will be $\leq \epsilon$, we will be sure that the true minimum value of g is less than ϵ, which, in view of the origin of ϵ, is possible only if the true optimal value in (C) is 0 and (S) is solvable. And if $g(x_\epsilon) > \epsilon$, the optimal value in (C) is > 0 (since x_ϵ is an ϵ-solution to (C)), and (S) is infeasible.

Now, g clearly is a convex function with easily (in $O(1)mn$ arithmetic operations) computable value and subgradient. It follows that an ϵ-solution to (C) can be found by the Ellipsoid method. Let us evaluate the complexity of this process. In the notation from Theorem 5.2.1 our case is the one of $X = E_0$ (i.e., $r = R = 2^{O(1)L}$) and, as is easily seen, $\mathrm{Var}_R(g) \leq R2^L$. Theorem 5.2.1 therefore says that the number of steps in the Ellipsoid method is bounded from above by $O(1)n^2 \ln\left(\frac{\epsilon + \mathrm{Var}_R(g)}{\epsilon}\right) \leq O(1)n^2 L$ (note that both ϵ^{-1} and $\mathrm{Var}_R(g)$ are of order of $2^{O(1)L}$). The number of arithmetic operations per step is $O(1)(n^2 + mn)$, where the n^2-term represents the "operation cost" of the method itself and the mn-term represents the computational expenses of computing $g(x), g'(x)$ at a given x and mimicking the separation oracle for the Euclidean ball E_0 (when proving Theorem 5.3.1, we have built such an oracle and have evaluated its running time – it is just $O(1)n$). Thus, the overall number of arithmetic operations required to find an ϵ-solution to (C) is $O(1)(n^2 + mn)n^2 L$, which is a polynomial in L (recall that $mn \leq L$).

We are nearly done; the only remaining problem is that the Ellipsoid method is a Real Arithmetic procedure, so that the polynomial in L complexity bound of checking feasibility in of (S) counts the number of operations of real arithmetic, and what we need is an Integer Arithmetic computer routine and a bound on the number of bit operations. Well, a quite straightforward (although boring) analysis demonstrates that we can obtain the same accuracy guarantees when implementing the Ellipsoid method on an "inexact arithmetic computer", where every elementary operation $+, -, /, \times, \sqrt{\cdot}$ is applied to $O(1)nL$-digit operands and rounds the exact result to the same $O(1)nL$ digits. Now every arithmetic operation "costs" a polynomial in L number of "bit" operations, and the overall "bit" complexity of the computation is polynomial in L.

From checking feasibility to finding a solution. It remains to explain how a CCT-polynomial time algorithm for checking feasibility of systems of linear inequalities with rational data can be converted into a CCT-polynomial time algorithm for solving LP programs with rational data. Observe, first, that to solve an LP problem is the same as to solve certain system of linear inequalities (S) (write down the constraints of the primal problem along with those of the dual and the linear equation saying that the duality gap is 0; of course, a linear equation can be written down as a pair of opposite linear inequalities). We already know how to check in polynomial time the feasibility of (S), and all we need is to understand how to find a feasible solution to (S) given that the system is feasible. The simplest way to do it is as follows. Let us take the first inequality $a^T x \leq b$ in (S), replace it with equality $a^T x = b$ and check whether the modified system we obtain is feasible. If it is not the case, we know that the hyperplane $a^T x = b$ does not intersect the solution set of (S); since this set is nonempty and convex, we conclude that every solution to the system (S$_1$) obtained from (S) by eliminating the first inequality is a solution to (S) as well. And if the modified system is feasible, let (S$_1$) be this modified system. Note that in both cases (S$_1$) is solvable, and every solution to (S$_1$) is a solution to (S) as well. Now let us look at (S$_1$); this system can have both inequalities and equalities. Let us take the first inequality of the system, if it exists, make it equality and check whether the modified system is feasible. If it is the case, this modified system will be our (S$_2$), and if it is not the case, (S$_2$) will be obtained from (S$_1$) by eliminating the first inequality in (S$_1$). Note that in both cases (S$_2$) is solvable, and every solution to it is a solution to (S$_1$) and therefore – to (S). Note also that the number of inequalities in (S$_2$) is by 2 less than that one in (S). Proceeding in the same way, we look in turn at all inequalities of the original system, check feasibility of certain "intermediate" system of equations and inequalities and, as a result, either make the inequality we are looking at an equality, or eliminate it at all, thus getting a new intermediate system. By construction, this system is solvable, and all its solutions are solutions to (S) as well. After m steps of this process (m is the number of inequalities in (S)) we terminate with a solvable system (S$_m$) of *equations*, and every solution to this system is a solution to (S). Note that all our intermediate systems are of the same total data length L as (S), so that the overall CCT-complexity of the outlined process is polynomial in L. It remains to note that we can use the standard Linear Algebra routines to find in polynomial in L time a solution to the solvable system of equations (S$_m$), thus getting – in polynomial time – a solution to (S).

Pay attention to the intrinsic mechanics of the outlined construction: its nucleus is a simple Real Arithmetic polynomial time routine, and this nucleus is "spoiled" by replacing Real Arithmetic operations with their inexact counterparts and is equipped with a completely "exterior" termination rules based on the fact that we are dealing with "percolated" – rational – data. Note that the more natural (although perhaps less "scientific") version of the question "whether LP is polynomially solvable" – namely, the question whether an LP program with rational/real data can be solved *exactly* in the number of Real Arithmetic operations bounded by a polynomial in the *size* $\mathrm{Size}(p) = \dim \mathrm{Data}(p)$ of instance – still remains open.

Difficult convex optimization problems

As it was already mentioned, what Real Arithmetic Complexity Theory can borrow from the Combinatorial Complexity Theory are techniques for detecting "computationally intractable" problems. Consider the situation as follows: we are given a family $\mathcal{P}$ of convex optimization programs [7] and want to understand whether the family is polynomially solvable. Well, Theorem 5.3.1 gives us sufficient conditions for polynomial solvability of $\mathcal{P}$, but what to do if one of these conditions is not satisfied? To be more concrete, let us look at the following family of convex optimization programs:

$$t \to \min \mid x \in X = \left\{ x \in \mathbf{S}^n \mid A \succeq x \succeq B, \max_{u \in C_n} u^T x u \le t \right\}, \tag{5.4.1}$$

where $C_n = \{u \in \mathbf{R}^n \mid |u_i| \le 1, i = 1, ..., n\}$ is the n-dimensional unit cube and A, B are symmetric matrices. Note that this problem is of essential interest for Robust Conic Quadratic Optimization we have mentioned in Lecture 4. We can say, in a natural way, what are data

[7] We could speak about other computational problems with "real data", in particular, nonconvex optimization ones, but recall that our subject is convex optimization.

vectors of instances and what is the associated infeasibility measure:

$$\mathrm{Infeas}(x,p) = \min\left\{\tau \geq 0 : x \preceq A + \tau I, x \succeq B - \tau I, \max_{u\in C_n} u^T x u \leq t + \tau\right\},$$

thus coming to a family of polynomial growth and with polynomially bounded feasible sets. The difficulty is with polynomial computability: we do not see an "easy" way to implement $\mathcal{C}_{\mathrm{cons}}$. Indeed, a direct way – just to compute $\mathrm{Infeas}(x,p)$ according to the definition of this function – fails, since no algorithms for computing the maximum of $g_x(u) = u^T x u$ over the unite cube with complexity less than 2^n operations are known, while $\mathrm{Size}(p)$ – the dimension of the data vector – is only of order n^2.

Now what? Perhaps we just do not see a "proper" way to implement $\mathcal{C}_{\mathrm{cons}}$ and should think more on this subject? For how long? Fortunately (or unfortunately, it depends on viewpoint), we can easily understand that our problem hardly is polynomially solvable. To explain the reason, let us forget for a moment about our particular family of convex programs and ask

> (?) *How could we convince ourselves that a given generic program $\mathcal{P}$ is "computationally intractable"?*

One of the ways to answer (?) is as follows. Assume that the objectives of the instances of $\mathcal{P}$ are polynomially computable and that we can point out a generic combinatorial problem $\mathcal{Q}$ *known to be* NP*-complete* which can be *reduced* to $\mathcal{P}$ in the following sense:

> *There exists a CCT-polynomial time algorithm $\mathcal{M}$ which, given on input the data vector* $\mathrm{Data}(q)$ *of an instance* $(q) \in \mathcal{Q}$*, converts it into a triple* $\mathrm{Data}(p[q])$, $\epsilon(q)$, $\mu(q)$ *comprised of the data vector of an instance* $(p[q]) \in \mathcal{P}$*, positive rational* $\epsilon(q)$ *and rational* $\mu(q)$ *such that* $(p[q])$ *is solvable and*
>
> *— if* (q) *is unsolvable, then the value of the objective of* $(p[q])$ *at every* $\epsilon(q)$*-solution to this problem is* $\leq \mu(q) - \epsilon(q)$*;*
>
> *— if* (q) *is solvable, then the value of the objective of* $(p[q])$ *at every* $\epsilon(q)$*-solution to this problem is* $\geq \mu(q) + \epsilon(q)$*.*

We claim that in the case in question we have all reasons to qualify $\mathcal{P}$ as a "computationally intractable" problem. Assume, on contrary, that $\mathcal{P}$ admits a polynomial time solution method $\mathcal{S}$, and let us look what happens if we apply this algorithm to solve $(p[q])$ within accuracy $\epsilon(q)$. Since $(p[q])$ is solvable, the method must produce an $\epsilon(q)$-solution $\widehat{x}$ to $(p[q])$. With additional "polynomial time effort" we may compute the value of the objective of $(p[q])$ at $\widehat{x}$ (recall that the objectives of instances from $\mathcal{P}$ are assumed to be polynomially computable). Now we can compare the resulting value of the objective with $\mu(q)$; by definition of reducibility, if this value is $\leq \mu(q)$, q is unsolvable, otherwise q is solvable. Thus, we get a correct "Real Arithmetic" solvability test for $\mathcal{Q}$. What is the (Real Arithmetic) running time of this test? By definition of a Real Arithmetic polynomial time algorithm, it is bounded by a polynomial of $s(q) = \mathrm{Size}(p[q])$ and

$$d(q) = \mathrm{Digits}((p[q]), \epsilon(q)) = \ln\left(\frac{\mathrm{Size}(p[q]) + \parallel \mathrm{Data}(p[q]) \parallel_1 + \epsilon^2(q)}{\epsilon(q)}\right).$$

Now note that if $\ell = \mathrm{length}(\mathrm{Data}(q))$, then the total number of bits in $\mathrm{Data}(p[q])$ and in $\epsilon(q)$ is bounded by a polynomial of ℓ (since the transformation $\mathrm{Data}(q) \mapsto (\mathrm{Data}(p[q]), \epsilon(q), \mu(q))$ takes CCT-polynomial time). It follows that both $s(q)$ and $d(q)$ are bounded by polynomials in ℓ, so that our "Real Arithmetic" solvability test for $\mathcal{Q}$ takes polynomial in $\mathrm{length}(\mathrm{Data}(q))$ number of arithmetic operations.

Recall that $\mathcal{Q}$ was assumed to be an NP-complete generic problem, so that it would be "highly unprobable" to find a CCT-polynomial time solvability test for this problem, while we have managed to build such a test, with the only (but important!) difference that our test is a Real Arithmetic one – it uses "incomparable more powerful" elementary operations. Well, a "reasonable" Real Arithmetic algorithm – any one which we can use in actual computations – *must* be tolerant to "small rounding errors" (cf. what was said about the Ellipsoid algorithm in the context of Linear Programming). Specifically, such an algorithm, as applied to a pair $((p),\epsilon)$ should be capable to "say" to the computer: "I need to work with reals with such and such number of binary digits before and after the dot, and I need all elementary operations with these reals to be precise within the same number of accuracy digits", and should preserve its performance and accuracy guarantees, provided that the computer meets the indicated requirement. Moreover, for a "reasonable" Real Arithmetic the "such and such number of digits before and after the dot" *must* be polynomial in $\mathrm{Size}(p)$ and $\mathrm{Digits}(p,\epsilon)$ [8]; and with these assumptions, our Real Arithmetic solvability test can be easily converted into a CCT-polynomial time solvability test for $\mathcal{Q}$, which – once again – hardly could exist. Thus, a Real Arithmetic polynomial time algorithm for $\mathcal{P}$ hardly could exist as well.

Since we do not know whether in fact NP-complete problems are "computationally intractable", the outlined reasoning does not *prove* that if you can reduce a NP-complete combinatorial problem to a generic program $\mathcal{P}$ with real data, the latter program is not polynomially solvable in the sense of Real Arithmetic Complexity theory; note, however, that (?) asks about "convince", not "prove".

As an illustration, consider the generic convex program $\mathcal{P}_0$ with instances (5.4.1) and let us demonstrate that the NP-complete problem Stones can be reduced to it, so that $\mathcal{P}_0$ is "computationally intractable". Indeed, let $(n, a=(a_1,...,a_n)^T)$ be the data of an instance (q) of Stones; recall that the instance is solvable, if there exist $u_i=\pm 1$ such that $\sum_i a_i u_i = 0$, and is unsolvable otherwise. Given (n,a), let us define the data of the instance $(p[q])\in\mathcal{P}_0$ as

$$A = B = \parallel a \parallel_2^2 I_n - aa^T,$$

and set

$$\epsilon(q) = \frac{1}{2(n+2)}, \mu(q) = n \parallel a \parallel_2^2 - \frac{1}{2}.$$

Let us demonstrate that this indeed is a reduction. Observe, first, that the conversion $\mathrm{Data}(q)\mapsto(\mathrm{Data}(p[q]),\epsilon(q),\mu(q))$ clearly is CCT-polynomial time. Now, since $A=B$, the feasible set of $(p[q])$ is

$$\{x = A = B, t \geq \max_{u\in C_n} u^T x u\},$$

and the optimal value in $(p[q])$ is

$$\mathrm{Opt}(p[q]) = \max_{u\in C_n} g(u), \;\; g(u) = u^T A u.$$

Since $A\succeq 0$ (check it!), the quadratic form g is convex, and therefore its maximum over C_n is the same as its maximum over the set of vertices of C_n (why?). If u

[8] In fact, this property normally is included into the very definition of a Real Arithmetic polynomial time algorithm; we prefer to skip these boring technicalities and to work with a simplified definition.

is a vertex of C_n (i.e., a vector with coordinates ±1), then the value of g at u is $n \parallel a \parallel_2^2 - (a^T u)^2$. Thus,

$$\mathrm{Opt}(p[q]) = \max\left\{ n \parallel a \parallel_2^2 - (a^T u)^2 \mid u_i = \pm 1,\ i = 1, ..., n \right\}.$$

If (q) is unsolvable – i.e., $a^T u \neq 0$ for all u with coordinates ±1 – then $(a^T u)^2 \geq 1$ for the indicated u (since for these u $a^T u$ is an integer), and we see that $\mathrm{Opt}(p[q]) \leq n \parallel a \parallel_2^2 - 1 = \mu(q) - 1/2$. On the other hand, if (q) is solvable, then the optimal value of the objective in $(p[q])$ is equal to $n \parallel a \parallel_2^2 = \mu(q) + 1/2$. Thus, the exact optimal value of $(p[q])$ is "quite sensitive" to solvability/unsolvability of (q). This is nearly what we need, but not exactly: we should prove that already the value of the objective of $(p[q])$ at any $\epsilon(q)$-solution to the problem "is sensitive" to the solvability status of (q). Let $(\widehat{x}, \widehat{t})$ be an $\epsilon(q)$-solution to $(p[q])$. In the case of unsolvable (q) we should have

$$\widehat{t} \leq \mathrm{Opt}(p[q]) + \epsilon(q) \leq \mu(q) - 1/2 + \epsilon(q) \leq \mu(q) - \epsilon(q). \tag{5.4.2}$$

Now assume that (q) is solvable. By definition of the infeasibility measure, we have

$$\begin{array}{rcl} \widehat{x} & \succeq & A - \epsilon(q) I_n, \\ \widehat{t} & \geq & \max_{u \in C_n} u^T \widehat{x} u - \epsilon(q) \end{array}$$

From (a), (b) it follows that

$$\begin{array}{rcl} \widehat{t} & \geq & \max_{u \in C_n} u^T \widehat{x} u - \epsilon(q) \\ & \geq & \max_{u \in C_n} \left[u^T A u - \epsilon(q) u^T u \right] - \epsilon(q) \\ & \geq & \max_{u \in C_n} u^T A u - n\epsilon(q) - \epsilon(q) \\ & = & \mathrm{Opt}(p[q]) - (n+1)\epsilon(q) \\ & = & \mu(q) + 1/2 - (n+1)\epsilon(q) \\ & \geq & \mu(q) + \epsilon(q). \end{array}$$

Combining the resulting inequality with (5.4.2), we see that the outlined construction indeed is a reduction of Stones to $\mathcal{P}_0$.

Generic convex program (5.4.1) illustrates the most typical source of "intractable" convex programs – *semi-infiniteness*. If we write (5.4.1) "explicitly", we get the problem

$$t \to \min \mid A \succeq x \succeq B, u^T x u \leq t \quad \forall u \in C_n,$$

and we see that the problem in fact has infinitely many simple convex constraints parameterized by a point $u \in C_n$. Computational tractability of a problem of this type depends on the geometry of the parameter set: if we replace the cube C_n by a simplex or an Euclidean ball, we get a polynomially computable (and polynomially solvable) generic program.

Bibliography

[1] Ben-Tal, A., and Nemirovski, A. (1996), ”Robust Solutions of Uncertain Linear Programs via Convex Programming” – to appear in *OR Letters*

[2] Lobo, M.S., Vanderbeghe, L., Boyd, S., and Lebret, H. (1997), ”Second-Order Cone Programming” – submitted to *Linear Algebra and Applications*, special issue on linear algebra in control, signals and image processing

[3] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory* – volume 15 of *Studies in Applied Mathematics*, SIAM, Philadelphia, 1994.

[4] Vanderberghe, L., Boyd, S., and El Gamal, A. (1996), ”Optimizing dominant time constant in RC circuits” – submitted to *IEEE Transactions on Computer-Aided Design.*

[5] Nemirovski, A. ”Polynomial time methods in Convex Programming” – in: J. Renegar, M. Shub and S. Smale, Eds., *The Mathematics of Numerical Analysis*, 1995 AMS-SIAM Summer Seminar on Mathematics in Applied Mathematics, July 17 – August 11, 1995, Park City, Utah. – Lectures in Applied Mathematics, v. 32 (1996): AMS, Providence, 543-589.

[6] Nemirovski, A. ”Interior point polynomial time methods in Convex Programming”, Lecture Notes (1996) – Faculty of Industrial Engineering and Management, Technion – Israel Institute of Technology

[7] Nesterov, Yu. ”Structure of non-negative polynomials and optimization problems”, Preprint DP 9749 (1997), C.O.R.E., Louvan-la-Neuve, Belgium.

[8] Nesterov, Yu. ”Semidefinite relaxation and non-convex quadratic optimization” – *Optimization Methods and Software* v. 12 (1997), 1-20.

[9] Wu S.-P., Boyd, S., and Vandenberghe, L. (1997), ”FIR Filter Design via Spectral Factorization and Convex Optimization” – to appear in Chapter 1 of *Applied Computational Control, Signal and Communications*, Biswa Datta, Ed., Birkhauser, 1997