Interior Point Methods

mini course

Systems and Control Group
Department of Applied Mathematics
The University of Twente

Spring 1997

Gjerrit Meinsma

Contents

1

Introduction

The general optimization problem can be stated as follows: Given a *feasible set* D and an *objective function* $f : D \to \mathbb{R}$, find

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in D \end{array} \tag{1.1}$$

If D is finite dimensional, then the optimization problem is *finite dimensional*. In this course, we will only consider finite dimensional optimization problems with $D \subset \mathbb{R}^n$. We classify these finite dimensional optimization problems as follows:

Unconstrained Optimization Problems In this case, $D = \mathbb{R}^n$.

Constrained Optimization Problems In this case, D is a non-trivial subset in $\mathbb{R}^n$ which is often described by a number of inequalities. Usually the feasible set D will be described by as series of inequalities

$$D = \{\, x : g_i(x) \leq 0, \ \ \forall i = 1, 2, \cdots, m \,\} \tag{1.2}$$

where $g_i : \mathbb{R}^n \to \mathbb{R}$ are given functions. Sometimes, matrix inequalities are used instead of scalar inequalities. For example, D may be a set in which a symmetric matrix function $G(x)$ is positive definite, i.e.,

$$D = \{\, x : G(x) > 0 \,\}$$

However, we note that every matrix inequality can be expressed as a number of scalar inequalities.

Decision Problems Associated with every optimization problem (1.1), there is a decision problem which is stated as follows: Given a feasible set $D \subset \mathbb{R}^n$, $f : D \to \mathbb{R}$ and any scalar α, determine if

$$f(x) < \alpha, \ \ \forall\, x \in D \tag{1.3}$$

The answer to the decision problem is binary: *yes* or *no*. The decision problem is easier than the optimization problem, however, an "efficient" algorithm for the decision problem will automatically lead to an "efficient" algorithm for the optimization problem.

For decision problems the theory of computational complexity can be used, such as the theory of NP-completeness[1]. Many classes of optimization problems have since been shown to be NP-hard.

Feasibility Problems In the above, we have implicitly assumed that the feasible set is nonempty. Although this assumption is often valid, determining if the feasible set is empty or not is often nontrivial when the set is described by inequalities (1.2). The task of the feasibility problem is to answer this question, and if affirmative, to find a *feasible member* $x \in D$. In fact, the decision problem mentioned above can be reformulated as a feasibility problem: determine if the following augmented feasible set is empty:

$$\hat{D} = \{x : f(x) \geq \alpha, x \in D\} \tag{1.4}$$

Obviously, $\hat{D}$ is empty if and only if the answer to the decision problem is *yes*. So in this sense, feasibility problems are very general.

The constrained optimization problems can be further categorized as follows:

Linear Programs (LPs) In this case, $f(x)$ is linear and D is described by a set of linear inequalities, i.e, given $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times n}$, solve

$$\begin{array}{ll} \text{minimize} & c^{\mathrm{T}}x \\ \text{subject to} & Ax \geq b \end{array} \tag{1.5}$$

The inequality above is taken to be elementwise.

LPs have been intensively studied for decades. Efficient algorithms for LPs exist that can handle upto several thousand variables.

Quadratic Programs (QPs) This case is the same as LP except that $f(x)$ is a quadratic function, i.e.,

$$f(x) = x^{\mathrm{T}}Qx + 2c^{\mathrm{T}}x \tag{1.6}$$

for some vector $c \in \mathbb{R}^n$ and symmetric matrix $Q \in \mathbb{R}^{n\times n}$. It turns out that the QP problem is much harder than the LP problem. In fact, the QP is one of the most challenging problems facing the researchers in the optimization theory and the class of QPs as a whole is NP-hard.

[1]Actually the theory of NP-completeness is designed for rational numbers, not for real numbers, although there are some tricks to make this possible. More recently a development of NP-completeness theory over the real numbers has been undertaken (Blum, Shub & Smale 1989) but I don't know its status.

Semidefinite Programs (SDPs) In this case, $f(x)$ is linear and D is described by a *linear matrix inequality*. That is, given a vector $c \in \mathbb{R}^n$ and symmetric matrices $F_i \in \mathbb{R}^{n\times n}$, $i = 0, 1, \ldots, n$, solve

$$\begin{array}{ll} \text{minimize} & c^{\mathrm{T}}x \\ \text{subject to} & F_0 + \sum_{i=1}^n x_i F_i \geq 0 \end{array} \tag{1.7}$$

It turns out that the SDP include the LP as a special case. The research in the last decade has offered efficient algorithms for solving SDPs.

In the systems and control community an SDP is often called an *eigenvalue problem (EVP)*; the feasibility problem whether or not $\{x : F_0 + \sum_{i=1}^n x_i F_i \geq 0\}$ is empty is called a *linear matrix inequality problem (LMI problem)*.

Convex Programs In this case f is a convex function and D is a convex set. Every LP and SDP is a convex program. A QP (1.6) is convex if Q is nonnegative definite.

1.1 About this course

The rest of this course consists of three chapters. These are:

Interior Point Methods for Convex Programs Interior point methods have been around for several decades. The emphasis in this course will be the more recent developments in interior point methods (IPMs) for LPs and SDPs, but for historical reasons we begin in the next chapter with the old IPMs for convex programs. The two methods discussed are the logarithmic barrier method and the method of centers. These methods are conceptually easy and are guaranteed to converge, but due to their generality they are not as sophisticated as the ones for LPs and SDPs.

Linear Programming This chapter examines two solutions of the well-known and perhaps the most important optimization problem: linear programs (LPs). The solution presented first is the classical Simplex algorithm. The other solution presented is a recent interior point algorithm with a guaranteed polynomial complexity. By combining the two algorithms, very efficient algorithms can be developed to solve large-size LPs.

Semidefinite Programming Not long after Karmakar's paper on IPMs in 1984, it was recognized that IPMs for LPs can be used much the same way for a matrix version of LPs as well. This matrix version is known as semi-definite programs (SDPs). SDPs arise quite often in engineering disciplines including statistics, systems and control, signal processing, etc. Techniques useful in generating SDP problems will be discussed. Some SDP examples are given. An interior point method will be introduced.

2

Interior Point Methods for Convex Programs

Covered in this in this chapter are

1. The logarithmic barrier method,
2. The center method.

These are methods to find global minimizers of a large class of finite dimensional convex programs. They are conceptually easy and they have been around for several decades. The logarithmic barrier function was introduced by Frisch (1955*a*) (see also (Frisch 1955*b*)). The method of centers was introduced by Huard (1967). Both methods fell out of favor in the beginning of the seventies, but recently they have been quite successful (in a somewhat altered form) for linear programs and semidefinite programs, which is the subject of the next two chapters.

The optimization problem that we consider in this chapter is

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & r_i(x) \geq 0, \quad (i = 1, \ldots, m) \end{array} . \tag{2.1}$$

The minimum will be denoted by f^*, the minimizer by x^*, and we assume that

$$f \text{ is convex, and the } r_i \text{ are concave.}$$

Under these assumptions, (2.1) is a *convex program* and, hence, any local minimum is a global minimum. The functions f and r_i are further assumed twice continuously differentiable. Most of the results to come actually hold under weaker assumptions, but we intend to use Newton's method.

The convex programs (2.1) form a very large class of problems, and includes for example all linear programs and semidefinite programs.

2.1 The logarithmic barrier method

Given a convex program (2.1), the *logarithmic barrier function* is defined as

$$\phi(x) \triangleq -\sum_{i=1}^{m} \log r_i(x).$$

The logarithmic barrier function is well defined in the interior $\{x : r_i(x) > 0, (i = 1, \ldots, m)\}$ of the feasible set, but because of the singularity of the logarithm at zero, the logarithmic barrier function grows to $+\infty$ as x approaches a boundary point of the feasible set.

Lemma 2.1.1. *The logarithmic barrier function is convex on the feasible set.*

Proof. It is sufficient to show that each $-\log r_i(x)$ is convex. Since r_i is assumed concave we have that

$$r_i(\lambda x + (1-\lambda)y) \geq \lambda r_i(x) + (1-\lambda) r_i(y)$$

for every $\lambda \in [0, 1]$. The function $-\log$ is decreasing and strictly convex, therefore, for every $\lambda \in [0, 1]$,

$$\begin{aligned} -\log(r_i(\lambda x + (1-\lambda)y)) &\leq -\log(\lambda r_i(x) + (1-\lambda) r_i(y)) \\ &\leq \lambda(-\log r_i(x)) + (1-\lambda)(-\log r_i(y)). \end{aligned}$$

This shows that $-\log r_i(x)$ is convex. ■

The *logarithmic barrier method* for the constrained convex program (2.1) is to solve instead for various values of $\alpha > 0$ the unconstraint convex problem

$$\text{minimize} \quad f(x) + \frac{1}{\alpha}\phi(x). \tag{2.2}$$

For a given $\alpha > 0$ the objective function here is convex and continuously differentiable in the interior of the feasible set. Hence any local minimum is global, and a minimum is readily computed using some iterative search. The term $\frac{1}{\alpha}\phi(x)$ acts as a barrier or a "repellent" of the boundary and this will ensure that the minimum is attained in the interior. This is why it is called an *interior point method*.

For large $\alpha > 0$ the extra term $\frac{1}{\alpha}\phi(x)$ in (2.2) hardly affects the minimization problem and only if x is very close to boundary will the extra term be of significance due to the singularity of the logarithm. We may expect then that for large $\alpha > 0$ a minimizer of the unconstrained (2.2) is close to a minimizer of (2.1). We have the following result.

Lemma 2.1.2. *Suppose the feasible set is bounded. Then for every $\alpha > 0$ a minimizer, x^*_α, of (2.2) exists, and*

$$\lim_{\alpha \to \infty} f(x^*_\alpha) = f^*.$$

In fact, for any $\alpha > 0$ there holds

$$0 \leq f(x_\alpha^*) - f^* \leq \frac{m}{\alpha}. \tag{2.3}$$

Here m is the number of constraints in (2.1).

Proof. This is an interesting proof. It uses the fact that the logarithm barrier method contains Langrangian dual information about a lower bound of f^* of the convex program (2.1). Let, as before, x_α^* denote a minimizer of (2.2). Because of differentiability we must have that the gradient of the objective function at x_α^* is zero, that is,

$$\nabla\big(f(x) + \frac{1}{\alpha}\phi(x)\big)\Big|_{x=x_\alpha^*} = \nabla f(x_\alpha^*) - \sum_i \frac{1}{r_i(x_\alpha^*)\alpha}\nabla r_i(x_\alpha^*) = 0. \tag{2.4}$$

The above gradient at x_α^* is also the gradient at x_α^* of another function

$$f(x) - \sum_i \lambda_i r_i(x), \quad \text{where } \lambda_i \triangleq \frac{1}{r_i(x_\alpha^*)\alpha}. \tag{2.5}$$

Note that the $\lambda_i = 1/(r_i(x_\alpha^*)\alpha)$ are positive, so the above function is convex, and, hence, x_α^* also minimizes (2.5). We then obtain a lower bound for f^*:

$$\begin{aligned}
f^* &\geq \min_{x \text{ feasible}} f(x) - \sum_i \lambda_i r_i(x) \qquad (\text{because } \lambda_i r_i(x) \geq 0)\\
&= f(x_\alpha^*) - \sum_i \lambda_i r_i(x_\alpha^*)\\
&= f(x_\alpha^*) - \sum_i \frac{1}{\alpha}\\
&= f(x_\alpha^*) - \frac{m}{\alpha}.
\end{aligned}$$

Since obviously $f^* \leq f(x_\alpha^*)$, we have (2.3). Convergence of $f(x_\alpha^*)$ to f^* is now obvious.

■

This is a strong result and is useful as a stopping criterion. The assumption that the feasible set is bounded is not that stringent (adding the constraint $\|x\|_2 \leq 10^{10000}$ makes it finite). The algorithm that combines the above results (and some additional insight which we left out) is called the *Sequential Unconstrained Minimization Technique (SUMT)*.

Sequential Unconstrained Minimization Technique (SUMT)

Input: strictly feasible $x^{(0)}$, tolerance $\epsilon > 0$,
initial barrier value $\alpha^{(0)}$, multiplicative factor $\beta > 1$.
Output: a feasible x such that $|f^* - f(x)| < \epsilon$.
Assumptions: The feasible set is bounded.

begin
$x := x^{(0)}$, $\alpha := \alpha^{(0)}$.
repeat
$v := -(\nabla^2 f(x) + \frac{1}{\alpha}\nabla^2\phi(x))^{-1}\,(\nabla f(x) + \frac{1}{\alpha}\nabla\phi(x))$; (Newton direction)
$\delta^* := \operatorname{argmin}_\delta f(x+\delta v) + \frac{1}{\alpha}\phi(x+\delta v)$; (a line search)
$x := x + \delta^* v$;
until $\|v\|$ very small
return if $m/\alpha < \epsilon$
$\alpha := \alpha\beta$.
end.

For large values of β the value of α changes so much that many costly Newton steps/line searches may be needed to find the next optimal $x^*_\alpha = x$. On the other hand, if β is only marginally more than 1, then α increases very slowly. There is trade off here.

The SUMT uses Newton's method and so we need to assume further that the Hessian is positive definite. This is hardly an assumption because for convex functions the Hessian is necessarily nonnegative definite, and only in the degenerate cases will it not be positive definite.

2.2 The method of centers

The *method of centers* is another method for the same convex program problem (2.1), where, as before, f and $-r_i$ are assumed twice continuously differentiable and convex.

Definition 2.2.1. The *analytic center* of a set of inequalities $r_i(x) > 0$, $(i = 1, \ldots, m)$ is defined as

$$x_{\mathrm{ac}} \triangleq \operatorname*{argmax}_{x \text{ feasible}} \prod_{i=1}^{m} r_i(x).$$

□

Strictly speaking the analytic center is *not* a property of the feasible set $\{x : r_i(x) \geq 0\}$, because a feasible set can be described by many different sets of inequalities, each with its own analytic center. What we do have is that the analytic center x_{ac} lies in the interior of the feasible set, and it is some sort of "center" of the feasible set.

The idea of the method of centers for convex program (2.1) is, again, to replace the constraint problem by a sequence of unconstrained convex optimization problems. Define $x^*_{\mathrm{c}}(t)$ as a function of $t \in \mathbb{R}$ as

$$\begin{aligned} x^*_{\mathrm{c}}(t) &\triangleq \text{analytic center of inequalities } f < t,\ r_i > 0,\ (i = 1, \ldots, m) \\ &= \operatorname*{argmax}_x\, (t - f(x)) \prod_{i=1}^{m} r_i(x) \\ &= \operatorname*{argmin}_x\, -\log(t - f(x)) - \sum_{i=1}^{m} \log(r_i(x)). \end{aligned} \tag{2.6}$$

The $x_c^*(t)$ is only well defined if the inequalities $f(x) < t$, $r_i(x) > 0$ are satisfied for at least

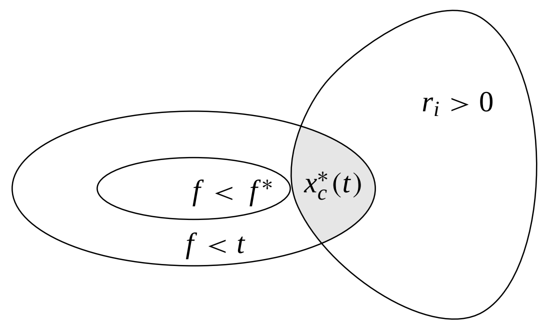

Figure 2.1: The method of centers.

one x. If we can find one such x, then finding $x_c^*(t)$ is easy because (2.6) is convex. As an example consider Fig. 2.1. In this figure two level sets are drawn for $f(x) = t$ and $f(x) = f^*$ (the two ellipsoid shaped curves) and also the feasible set $\{x : r_i(x) \geq 0\}$ is depicted. The shaded area, then, is the region where $f(x) < t$ and $r_i(x) > 0$ are feasible, and so $x_c^*(t)$ is somewhere in the center of this set. The optimal x^* is that point where the level set $\{x : f(x) = f^*\}$ touches the set $\{x : r_i(x) \geq 0\}$. Based on this picture it seems reasonable to believe that as t approaches f^*, the analytic center $x_c^*(t)$ will have to approach the minimizer x^* of the convex program (2.1).

Lemma 2.2.2. *If $t > f^*$ and the restricted feasible set $\{\, x : f(x) < t,\ r_i(x) \geq 0 \,\}$ is bounded, then $x_c^*(t)$ exists and*

$$0 \leq f(x_c^*(t)) - f^* \leq m(t - f(x_c^*(t))). \tag{2.7}$$

In particular, $\lim_{t \to f^*} f(x_c^*(t)) = f^*$. *Here m is the number of constraints (2.1).*

Proof. By differentiability we have that $\nabla(-\log(t - f(x)) - \sum \log(r_i(x))) = \frac{1}{t - f(x)} \nabla f(x) + \nabla \phi(x) = 0$ at $x = x_c^*(t)$. This implies that (2.4) is zero for $\alpha = 1/(t - f(x_c^*(t))$. For this α inequality (2.3) becomes (2.7). ∎

Method of Centers

Input: Strictly feasible $x^{(0)}$, tolerance $\epsilon > 0$,
initial upper bound $t > f(x^{(0)})$, and $0 < \theta < 1$.
Output: A feasible x such that $|f^* - f(x)| < \epsilon$.
Assumptions: The feasible set intersected with $\{x : f(x) < t\}$ is bounded.
begin
$x := x^{(0)}$

repeat
  $v := -[\nabla^2(-\log(t - f(x)) + \log\phi(x))]^{-1}$
    $\times \nabla(-\log(t - f(x)) + \phi(x))$ (Newton direction)
  $\delta^* := \operatorname{argmin}_\delta - \log(t - f(x + \delta v)) + \phi(x + \delta v)$ (line search)
  $x := x + \delta^* v$
until $\|v\|$ very small
return if $m(t - f(x)) < \epsilon$
$t := (1 - \theta) f(x) + \theta t$
end

Note that $f(x) < t$ so in the algorithm the update $t := (1 - \theta) f(x) + \theta t$ decreases t. As in the SUMT, we use here Newton's method and so we should demand that the Hessian is nonsingular.

Lemma 2.2.3. *The iterates t in the above algorithm converge to f^*.*

Proof. By Lemma 2.2.2 we have that $f^* \geq f(x) - m(t - f(x)) = (m + 1) f(x) - mt$. Therefore $f(x) \leq \frac{f^*+mt}{m+1}$. Let t^+ denote the update $t^+ := (1 - \theta) f(x) + \theta t$. Then

$$t^+ - f^* = (1 - \theta) f(x) + \theta t - f^* \leq (1 - \theta)\frac{f^* + mt}{m + 1} + \theta t - f^* = \frac{m + \theta}{m + 1}(t - f^*).$$

I.e., the error $t - f^*$ decreases each iteration by at least a factor $(m + \theta)/(m + 1) < 1$, hence it converges to zero. ■

Remark: A trivial variation of the method of centers and which appears to be quite effective in practice is to use m copies of the constraint $f(x) - t < 0$ instead of only one, that is, to use

$$\begin{aligned} x_c^*(t) &\triangleq \operatorname*{argmax}_x (t - f(x))^m \prod_{i=1}^m r_i(x) \\ &= \operatorname*{argmin}_x -m \log(t - f(x)) - \sum_{i=1}^m \log(r_i(x)). \end{aligned}$$

2.3 Interpretation: Central path

The set $\{\operatorname{argmin} f(x) + \frac{1}{\alpha}\phi(x) : a \in [0, \infty)\}$ is commonly referred to as the *central path*. Figure 2.2 depicts a typical central path. The central path starts at the analytic center x_{ac} and ends at an optimal solution x^* of the convex program. By definition the logarithmic barrier method generates points on the central path. Also the method of centers generates points on the central path (see the proof of Lemma 2.2.2). Although the notion of central path as such does not add anything new to two methods, it is a convenient way to think about these methods. We might use it in later chapters..

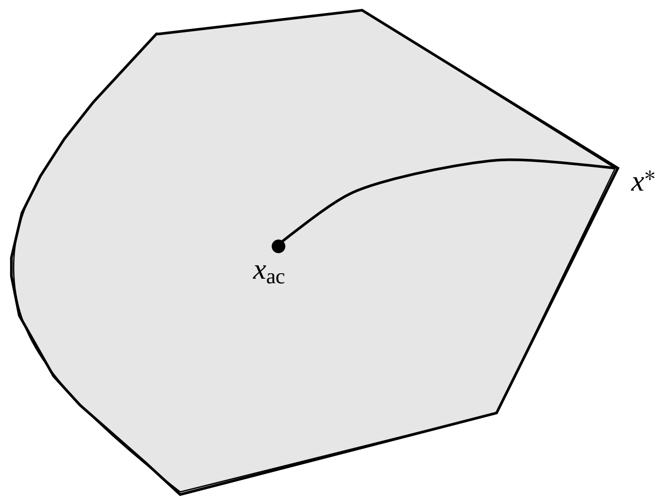

Figure 2.2: A central path.

2.4 Notes and References

The material of this chapter is mainly from den Hertog (1993) and Boyd & Vandenberghe (1995).

One problem we did not address is numerical stability of the computations. The main problem with the general logarithm barrier method is that the Hessian $(\nabla^2 f(x) + \frac{1}{\alpha}\nabla^2\phi(x))$ that is needed (and needs to be inverted) in the computation may become ill-conditioned as $\alpha \to \infty$. A draw back of the method of centers appears to be the fact that the initial upper bound t must always be chosen larger than the value of $f(x^{(0)})$, which may be very pessimistic (den Hertog 1993).

For more specialized problems like linear programs and semidefinite programs, singularities have in the past decade been removed (see next chapters), and polynomial running time has been proved.

3

Linear Programs

Covered in this chapter are

1. What is a linear program (LP)
2. The Simplex method for LPs
3. The concept of duality and the duality gap for LPs
4. An interior point method for LPs that terminates in polynomial time.

3.1 introduction

Suppose you want to determine the cheapest diet that satisfies certain minimal and maximal requirements for vitamins and calories and the like. This is an example of a linear program. Suppose you start a company with several branches around the world. You might be faced then with the problem to find the most cost effective way to get products from certain branches to other branches for assembly. This is a linear program.

The diet and transportation problem are just two of the many, many real life problems that can be cast as a linear program. We will encounter several in this chapter.

The theory of linear programs is rich and knows many highs. Quite impressive is the practical experience with the *Simplex method* as a method to solve linear programs. The Simplex method was invented by Dantzig (1951) right after the second world war (and was in fact motivated by military applications) and it allowed to solve by hand linear programs of reasonable size. With the advent of computers the Simplex method could be tested on larger problems and it has been very successful.

With the computer came also the interest in *complexity* of algorithms and in 1972 Klee & Minty showed that for specific problems the Simplex method may need an exponential number of iterations[1]. At about the same time it was proved that the general LP is in $\mathbf{NP} \cap \text{co}-\mathbf{NP}$, which suggests that there is a polynomial time algorithm for LPs. Finally in 1979 Khachian

[1]More precise: $2^n - 1$ iterations were needed to solve an LP with n variables and $2n$ constraints.

formulated an ellipsoid method that was guaranteed to work in polynomial time. This development was, however, *not* the end of the Simplex method because although Khachian's method runs in polynomial time, in practice it is much slower than the Simplex method.

Things changed in 1984 when in a famous paper Karmarkar presented a polynomial time algorithm that, as he claimed, for large problems could outdo the Simplex method by as much as a factor 100. Karmakar's paper inspired many people to look again at the interior point methods and this has resulted in a large number of different polynomial time interior point methods for LPs[2]. We present a Simplex method and we will look at Ye's interior point algorithm (Ye 1991) which at present achieves the best asymptotic running time.

3.2 LPs

The general *linear program* (*LP*) is to minimize a linear functional subject to the constraint that a set of linear equalities and inequalities is satisfied. More concrete, a problem is an LP if it can be written in the form

$$\begin{cases} \text{minimize} & \sum_{j=1}^{n} c_j x_j & \\ \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j = b_i & \forall i \in \{1, \dots, m\}, \\ & x_j \geq 0, & \forall j \in \{1, \dots, n\}, \end{cases}$$

where the $n, m \in \mathbb{N}$ and the $\{a_{ij}, b_i, c_j\}$ are given, and the minimization is with respect to the n variables $x_j \in \mathbb{R}$. It is usually more convenient to write the LP in matrix form. For example, if we define $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^{m\times 1}$, and $c, x \in \mathbb{R}^{n\times 1}$ as

$$A \triangleq \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \dots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, \quad b \triangleq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \quad c \triangleq \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}, \quad x \triangleq \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

then the above LP can be written as

$$\min_{x\in\mathbb{R}^n} c^{\mathrm{T}} x \quad \text{subject to} \quad Ax = b,\ x \geq 0.$$

Here "$x \geq 0$" is defined to mean that all entries of x are nonnegative.

LPs can be written in several other equivalent forms, but there are two popular standard forms. They are the one presented above and one which looks easier but which has some undesirable properties as we shall soon see.

Definition 3.2.1 (standard and canonical form). An LP is in *standard* form if it is of the form $\min_x\{c^{\mathrm{T}}x \,:\, Ax = b,\ x \geq 0\}$. It is in *canonical* form if it is of the form $\min_x\{c^{\mathrm{T}}x \,:\, Ax \geq b\}$. □

[2]There are more than 2000 papers on IPMs (den Hertog 1993).

The canonical form can be transformed into the standard form and vice versa, using the following transformations. 1) An unrestricted variable x_j can be replaced by a pair of nonnegative variables x_j^+ and x_j^- by letting $x_j = x_j^+ - x_j^-$. 2) Equalities $Ax = b$ can be replaced by a pair of inequalities $Ax \geq b, -Ax \geq -b$. 3) Inequalities $Ax \geq b$ can be replaced by a simpler looking $Ax - s = b, s \geq 0$, where now x *and* s are the variables.

Exercise 3.2.2. Show that the canonical form can be transformed into the standard form and vice versa. □

3.3 Examples of LPs

Example 3.3.1 (The diet problem). Suppose we want to form a diet that is most economical but satisfies certain daily intake requirements such as a minimal amount of vitamins, maximal amount of calories and so on.

More precisely, suppose we have n foods to choose from and that the jth food product costs c_j dollars per unit. Let x_j denote the number of units of food product j in our diet. Then

$$c^{\mathrm{T}} x$$

is the cost of the diet. Suppose we have m_1 “good” nutrients and that we want each day to have at least b_i units of nutrient number $i \in \{1, \ldots, m_1\}$. Suppose we have m_2 “bad” nutrients (let’s consider calories one of the bad nutrients), numbered $i = \{m_1 + 1, \ldots, m_1 + m_2\}$, whose intake each day we want to be below b_i. Let $a_{i,j}$ denote the amount of nutrient i in a unit of food product j. Then the cheapest diet—that is the cheapest combination of food products—that achieves all that (if any) is the solution of the LP

$$\min_x c^{\mathrm{T}} x, \quad \text{subject to} \quad \begin{bmatrix} I_{m_1} & 0 \\ 0 & -I_{m_2} \end{bmatrix} Ax \geq \begin{bmatrix} I_{m_1} & 0 \\ 0 & -I_{m_2} \end{bmatrix} b, \quad x \geq 0.$$

This is not yet in the standard form, but that is easily done. □

Example 3.3.2 (A transportation problem (Goldfarb & Todd 1989)). A company needs to ship a product from m locations to n destinations. Suppose location $i \in \{1, \ldots, m\}$ has a_i units of the product and that destination $j \in \{1, \ldots, n\}$ requires b_j units. Suppose that the total amount available equals the total amount required. Let c_{ij} denote the cost to ship one unit from location i to destination j.

What is the cheapest way to get the job done? It is the solution of an LP: Let x_{ij} denote the number of units that will be shipped from location i to destination j, and solve

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{subject to} & \sum_{j=1}^n x_{ij} = a_i \ \ \forall i \in \{1, \ldots, m\}, \\ & \sum_{i=1}^m x_{ij} = b_j \ \ \forall j \in \{1, \ldots, n\}, \\ & x_{ij} \geq 0 \ \ \forall i \in \{1, \ldots, m\}, \ \forall j \in \{1, \ldots, n\}. \end{array} \tag{3.1}$$

□

The transportation problem illustrates why modest sized problems may easily result in very large LPs. If, say, the number of locations and destinations in the above example are both 100, then the LP (3.1) involves 10000 variables. Goldfarb & Todd (1989) mention that LPs with as many as 63 million variables have been formulated and solved.

Example 3.3.3 (Chebyshev and ℓ_1 approximation). If a linear equation $Ax = b$ can not be solved exactly one often looks for a least squares solution, $\operatorname{argmin}_x \|Ax - b\|_2$. Another possibility is to do Chebyshev approximation, which is

$$\underset{x}{\text{minimize}} \; \|Ax - b\|_\infty,$$

where $\|c\|_\infty \triangleq \max_i |c_i|$. This can be cast as an LP in the variables $x \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$:

$$\begin{array}{ll} \text{minimize}_{x,\gamma} & \gamma \\ \text{subject to} & Ax - b \leq \underline{1}\gamma, \quad Ax - b \geq -\underline{1}\gamma. \end{array}$$

(The $\underline{1}$ is a useful short-hand for $\underline{1} = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^{\mathrm{T}}$.) Another possibility would be to minimize over x

$$\|Ax - b\|_1.$$

Here $\|c\|_1 \triangleq \sum_i |c_i|$. This is known as ℓ_1 approximation, and this also is an LP:

$$\begin{array}{ll} \text{minimize}_{x,y\in\mathbb{R}^n} & \sum_j^n y_j \\ \text{subject to} & Ax - b \leq y, \quad Ax - b \geq -y. \end{array}$$

□

Least squares solutions are by far the most popular, but in some instances ℓ_1 or Chebyshev approximation is better. As an example, suppose we are given a large collection of pairs of real numbers (x_i, y_i) with

$$y_i \approx kx_i + l,$$

for all i except for a few failures we would like to discard. To estimate k and l the ℓ_1 approximation is the best because its solution is the least sensitive to failures. It is in fact completely insensitive to the failures as long as there only a few of them. Chebyshev approximation is also very useful.

Example 3.3.4 (How computers calculate functions). Ever wondered how a computer calculates $\sin(x)$? Computers know how to perform addition and multiplication, but for transcendental functions like $\sin(x)$ and $\log(x)$ no finite method exists that uses only $\times$ and $+$. An obvious solution to this problem is to evaluate a sufficient number of terms of the Taylor series

$$\sin(x) = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \cdots$$

until the error is less than the machine precision. Suppose $x \in [-\pi/2, \pi/2]$ (this can always be achieved by subtracting multiples of π) then to obtain $\sin(x)$ within machine precision 10^{-15}, requires a polynomial of degree 17. The problem with the Taylor series is that it is unnecessarily accurate near $x = 0$ at the loss of accuracy for larger x. In most computer languages (or at least on my old Commodore-64 and some libraries of C-compilers I've seen) the value of $\sin(x)$ is instead computed via a polynomial approximation of lower degree $\sum_{k=1}^{n} \alpha_k x^{2k+1}$ where the stored numbers α_k are the solution of a Chebyshev approximation problem

$$\min_{\alpha_k \in \mathbb{R}} \max_{x \in [-pi/2, \pi/2]} |\sin(x) - \sum_{k=1}^{n} \alpha_k x^{2k+1}|.$$

The Chebyshev approximation is by definition the one that has the smallest guaranteed accuracy, which is exactly what we want of computer languages. To find α_k one might solve for a given n and a large number N of grid points $t_k \in [-\pi/2, \pi/2]$ the finite dimensional Chebyshev approximation problem

$$\min_{\alpha \in \mathbb{R}^n} \| \begin{bmatrix} \sin(t_1) \\ \vdots \\ \sin(t_N) \end{bmatrix} - \sum_{k=1}^{n} \alpha_k \begin{bmatrix} t_1^{2k+1} \\ \vdots \\ t_N^{2k+1} \end{bmatrix} \|_\infty.$$

This is an LP. (Remark: $\sin(x)$ can be computed even faster if we split the interval $[-\pi/2, \pi/2]$ in a few hundred tiny intervals and store for each of the subinterval a linear or cubic Chebyshev approximation.) $\square$

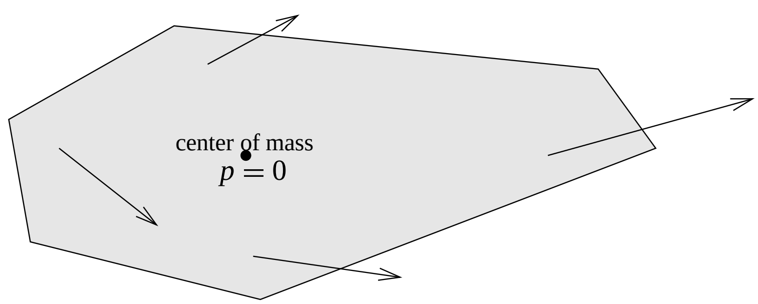

Figure 3.1: A thruster problem.

Example 3.3.5 (A thruster problem (Boyd & Vandenberghe 1995)). Figure 3.1 depicts a top view of a rigid body with several thrusters. There are n forces with magnitude u_i acting at $p_i = (p_{ix}, p_{iy})$ relative to the center of mass $p = 0$. The directions of the forces are denoted θ_i such that $\theta_i = 0$ corresponds to a horizontal force pointed to the right. We assume that the directions are fixed. Total forces and torque etcetera can be expressed as

$$\begin{array}{rrcl} \text{Total horizontal force:} & F_x & \triangleq & \sum_{i=1}^n u_i \cos\theta_i \\ \text{Total vertical force:} & F_y & \triangleq & \sum_{i=1}^n u_i \sin\theta_i \\ \text{Torque:} & T & \triangleq & \sum_{i=1}^n p_{iy} u_i \cos\theta_i - p_{ix} u_i \sin\theta_i \\ \text{Total fuel usage:} & & & u_1 + \ldots + u_n. \end{array}$$

There are several thruster problems that are LPs. For example, the problem to find forces $0 \le u_i \le 1$ that yield given desired forces and torque and minimize fuel usage is the LP

$$\begin{array}{ll} \text{minimize} & \underline{1}^{\mathrm{T}} u \\ \text{subject to} & Fu = f^{\mathrm{des}}, \ 0 \le u \le \underline{1}, \end{array}$$

where $\underline{1}$ is a short-hand for $\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^{\mathrm{T}}$ and

$$F = \begin{bmatrix} \cos\theta_1 & \cdots & \cos\theta_n \\ \sin\theta_1 & \cdots & \sin\theta_n \\ p_{1y}\cos\theta_1 - p_{1x}\sin\theta_1 & \cdots & p_{ny}\cos\theta_1 - p_{nx}\sin\theta_1 \end{bmatrix}, \quad f^{\mathrm{des}} = \begin{bmatrix} F_x^{\mathrm{des}} \\ F_y^{\mathrm{des}} \\ T^{\mathrm{des}} \end{bmatrix}.$$

This problem might not be feasible, but the following LP certainly is

$$\begin{array}{ll} \text{minimize} & \underline{1}^{\mathrm{T}} u + \gamma \| Fu - f^{\mathrm{des}} \|_\infty \\ \text{subject to} & 0 \le u \le \underline{1}. \end{array}$$

Here $\gamma > 0$ is some fixed number that specifies the trade off between fuel usage and deviation from desired forces/torque. Another possibility would be to solve for some fixed $\gamma > 0$ the LP

$$\begin{array}{ll} \text{minimize} & \underline{1}^{\mathrm{T}} u \\ \text{subject to} & 0 \le u \le \underline{1}, \ \ \| Fu - f^{\mathrm{des}} \|_\infty \le \gamma. \end{array}$$

□

3.4 Geometry of LPs

A lot of insight can be gained by looking at the LPs geometrically. Indeed the basic idea of Simplex method stems from a simple geometrical argument. First an example and a bit of terminology.

Definition 3.4.1 (Feasible). An LP $\min_x \{c^{\mathrm{T}} x : Ax = b, \ x \ge 0\}$ is *feasible* if the *feasible set* $\{x : Ax = b, \ x \ge 0\}$ is nonempty. Elements of the feasible set are called *feasible points* or, simply, *feasible*.

A feasible point x^* is an *optimal solution* of the LP if $c^{\mathrm{T}} x^* = \min_x \{c^{\mathrm{T}} x : Ax = b, \ x \ge 0\}$. The function $c^{\mathrm{T}} x$ which is to be minimized is often called the *objective function*. □

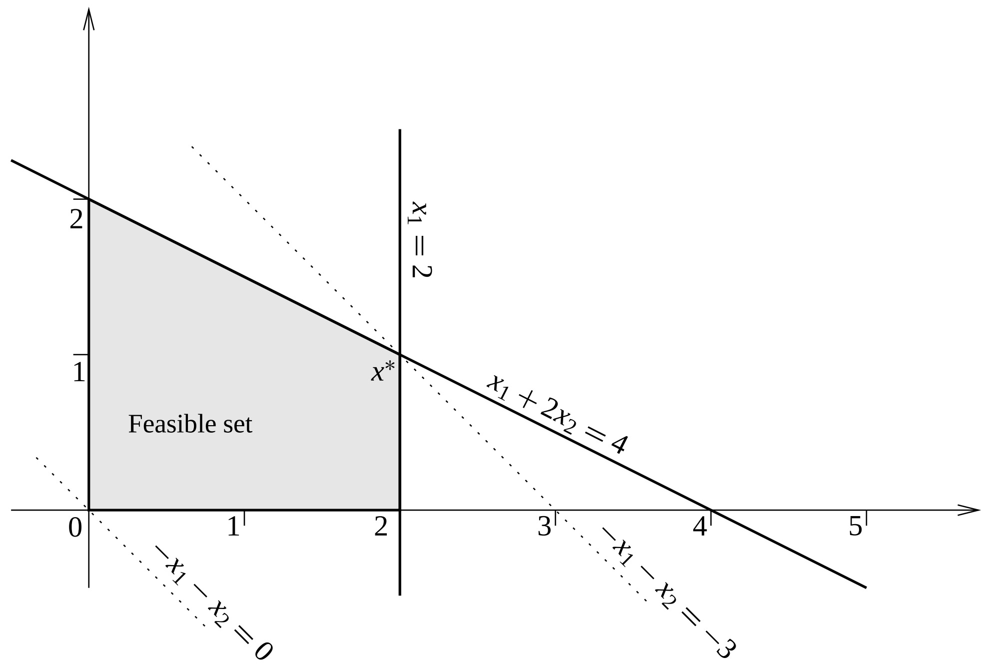

Figure 3.2: An LP with two variables.

Example 3.4.2. Suppose we wish the minimize $-x_1 - x_2$ subject to the constraints that

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_1 + 2x_2 \leq 4, \quad x_1 \leq 2.$$

The feasible set is depicted in Figure 3.2 as the shaded area. To get a feel for where the optimal solution x^* lies we draw for several values of $d \in \mathbb{R}$ the set of points where the objective function equals d. These sets are $\{(x_1, x_2) : -x_1 - x_2 = d\}$ and they form straight lines (the dotted lines in Figure 3.2). By varying d it should be clear that within the feasible set the objective function is the smallest (namely -3) at the point $x^* \triangleq (x_1, x_2) = (2, 1)$ as shown in the figure. In this example the optimal solution is apparently unique and lies at a corner of the feasible set. $\square$

The example suggests several things:

1. The set of feasible x (if bounded) is a convex polygon,
2. The optimal solution x^* is usually unique and occurs at a corner of the polygon, and in the unlikely case that it is not unique there is always a corner point which is optimal.

These two observations may be formalized algebraically. We need two notions which generalize "corners" and "convex polygons" to the n-dimensional case.

Definition 3.4.3 (Convex polyhedron and vertices in $\mathbb{R}^n$). An affine set in $\mathbb{R}^n$ of dimension $n-1$ is called a *hyper-plane* of $\mathbb{R}^n$. In other words, a set Ω is a hyper plane if-and-only if for some $d \in \mathbb{R}$ and $a \in \mathbb{R}^n$ we have $\Omega = \{x \in \mathbb{R}^n : a^{\mathrm{T}}x = d\}$.

The set of points on one side of a hyper-plane is called a *half-space*. *Closed half-spaces* include the hyper-plane and *open half-spaces* do not include the hyper-plane. In other words, Ω is a closed half-space of $\mathbb{R}^n$ if-and-only-if for some $d \in \mathbb{R}$ and $a \in \mathbb{R}^n$ we have $\Omega = \{x \in \mathbb{R}^n : a^T x \geq d\}$.

A subset of $\mathbb{R}^n$ is a *convex polyhedron* if it is the intersection of finitely many closed half-spaces of $\mathbb{R}^n$.

An element x of a convex polyhedron is a *vertex* or *extreme point* of the polyhedron if it can not be written as the average of two other elements from the polyhedron. □

Exercise 3.4.4. Prove that a convex polyhedron in $\mathbb{R}^n$ is indeed a convex set. □

Lemma 3.4.5. *For any LP the feasible set is a convex polyhedron. If the LP is in standard form,* $\min_x\{c^T x : Ax = b,\ x \geq 0\}$*, and if the minimum is finite then the LP has optimal solutions x and at least one of these optimal solutions is a vertex of the polyhedron.* □

The significance of this result is that for the minimization problem we need only worry about the vertices of which there finitely many. In principle, then, we have a means to find the minimum of the objective function: Generate all vertices and take the minimum of the objective function evaluated at each of these vertices. Generating the vertices is in way straightforward algebra, but the problem is that the number of vertices may grow exponentially with the number of constraints and, so this will most likely be a very inefficient method.

Proof of Lemma 3.4.5. Consider the general LP in standard form $\min_x\{c^T x : Ax = b,\ x \geq 0\}$. The feasible set $\{x : Ax = b,\ x \geq 0\}$ can also be written as the intersection of the half-spaces $\{x : x_j \geq 0\}$ and $\{x : a_k^T x \geq b_k\}$ and $\{x : -a_k^T x \geq -b_k\}$, and so the feasible set is (almost by definition) a convex polyhedron.

The main part is to show that for any feasible x there is a vertex x^v such that $c^T x^v \leq c^T x$.

If x is a vertex the claim holds obviously for $x^v = x$. If x is not a vertex it can, by definition, be expressed as the average of certain $x - y$ and $x + y$ from the polyhedron, with $y \neq 0$. Since the polyhedron is convex and since it contains both $x - y$ and $x + y$ we must have that $x + \lambda y$ is an element of the polyhedron for every $\lambda \in [-1, 1]$. This implies that whenever an entry x_k is zero also y_k is zero (otherwise $x \pm y$ can not both be nonnegative), moreover, we have that

$$Ay = \frac{1}{2}(A(x + y) - A(x - y)) = \frac{1}{2}(b - b) = 0.$$

The three cases below show that for some $\lambda^* \in \mathbb{R}$ the vector $x + \lambda^* y$ has less or equal cost and one more zero entry:

- Suppose that $c^T y < 0$. Then the objective function $c^T(x + \lambda y)$ at $x + \lambda y$ decreases as λ increases. Increase $\lambda > 0$ from zero until $x + \lambda y$ is about to become infeasible. This must occur at a certain *finite* $\lambda = \lambda^*$ because the minimum of the objective function is assumed finite. Since $A(x + \lambda y) = b$ for *every* $\lambda \in \mathbb{R}$, the only way that $x + \lambda y$ can become infeasible if we move beyond λ^* is when a certain entry $x_k - \lambda y_k$ changes sign at $\lambda = \lambda^*$. Remember that $x_j = 0$ implies that $y_j = 0$ so the changing in sign of

$x_k - \lambda y_k$ occurs in entry where $x_k > 0$. So have that $x - \lambda^* y$ has at least one more zero entry than x, which is what we had to show.

- Suppose that $c^{\mathrm{T}} y = 0$. Then the objective function $c^{\mathrm{T}}(x + \lambda y)$ at $x + \lambda y$ does not change at all if we vary λ. Define $\lambda^* = \min\{|x_k|/|y_k|\}$ where the minimum is taken over all indices k for which y_k is nonzero (and hence, x_k nonzero as well). It is easy to see that then both $x + \lambda^* y$ and $x - \lambda^* y$ are feasible and that one of them has at least one extra zero entry ($x_k \pm \lambda^* y_k$) than x.

- The case that $c^{\mathrm{T}} y > 0$ goes the same as the case that $c^{\mathrm{T}} y < 0$ (simply replace y with $-y$).

The three cases combined show that if x is not a vertex then we can always find a feasible $x + \lambda^* y$ which has at least one more zero entry than x and whose cost is not more than that of x. Repeat in this fashion with $x + \lambda^* y$ in place of x. Since a vector can have only as many zero entries as it has entries, the iteration must eventually reach a vertex x^v, and since we went "down hill" we end up with $c^{\mathrm{T}} x^v \leq c^{\mathrm{T}} x$.

It is now immediate that $\min_x\{c^{\mathrm{T}} x \,:\, Ax = b,\ x \geq 0\}$ is attained at one or more of the vertices. ■

Corollary 3.4.6. *Every polyhedron of the form* $\{x \in \mathbb{R}^n \,:\, Ax = b,\ x \geq 0\}$ *has a vertex.* □

Exercise 3.4.7. Give an example of an LP in canonical form $\{x \in \mathbb{R}^2 \,:\, Ax \geq b\}$ that has a well defined minimum, but whose minimum is not attained at a vertex of the feasible set $\{x \in \mathbb{R}^2 \,:\, Ax \geq b\}$. (This shows the interest in the standard form even though the canonical form is notationally more compact.) □

3.5 The Simplex method

The results in the previous section guarantee that the minimum of an LP in standard form will always be attained at a vertex (and possibly some other points, but who cares). This observation has led to the basic idea of the Simplex method, which is:

> Select any vertex of the feasible set. If the objective function is smaller at one of the adjacent vertices select that vertex, and repeat moving from vertex to adjacent vertex this way until no improvement in the objective function is possible.

Since at each step the objective function is made smaller it is not possible that a vertex is selected more than once and because there are only a finite number of vertices, the method will have to terminate. The Simplex method is now essentially nothing but a rule to decide at each step which of the adjacent vertices has to be selected as the new active vertex.

To formalize the idea we have to characterize "vertex" algebraically.

3.5.1 What are the adjacent vertices?

That is, what is an adjacent vertex in terms of matrix A and vector b?

Example 3.5.1. Consider again the LP of Example 3.4.2. In standard form the constraints of that example read

$$\begin{cases} x_1 + 2x_2 + s_1 &= 4, \\ x_1 + s_2 &= 2, \quad x_1 \geq 0, \quad x_2 \geq 0, \quad s_1 \geq 0, \quad s_2 \geq 0, \end{cases} \tag{3.2}$$

where now we have four variables x_1, x_2, s_1, s_2. An edge of the feasible set corresponds to one variable being zero. (See Figure 3.2). Corners, then, are characterized as having *two* variables equal to zero. For example, for the optimal solution $x^* = (x_1, x_2) = (2, 1)$ we have that (3.2) holds for $s_1 = 0$ and $s_2 = 0$. It is now easy to see that in this example two corners are adjacent if and only if they have one zero-valued variable in common. □

Picturing adjacent vertices in $\mathbb{R}^3$ is not that complicated either and what these examples suggest is that a vertex y is adjacent to a vertex x if the zero entries of y equal those of x except for one entry. We could have formalized this to higher dimensions, but that would only have complicated things. (In some degenerate cases for example adjacent vertices may have two or more different nonzero entries.) Instead we will work directly with vertices with most nonzero entries in common and leave the geometrical interpretation as a motivation only.

3.5.2 How do we know that x is a vertex?

That is to say, how do we know in terms of the data A and b that a vector x is a vertex of $\{x : Ax = b,\ x \geq 0\}$. That is the question we have to answer. From Example 3.5.1 we understand that vertices have a lot to do with the number of zero and nonzero entries of x.

Lemma 3.5.2. *An $x \in \{x \in \mathbb{R}^n : Ax = b,\ x \geq 0\}$ is a vertex of that polyhedron if and only if the columns of A that correspond to the nonzero entries of x are linearly independent.*

Proof. Let x be any element of the feasible set. For simplicity and without loss of generality we may assume that the first, say, q entries of x are the nonzero entries and that the remaining $n - q$ entries are the zero entries. Partition x accordingly as $x = \left[\begin{smallmatrix} \bar{x} \\ 0 \end{smallmatrix}\right]$, and also partition A accordingly as $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$. Then $Ax = A_1\bar{x} = b$, and $\bar{x} > 0$. We have to prove that x is a vertex if-and-only-if A_1 has full column rank.

Suppose A_1 does not have full column rank. There then exists a nonzero vector $\bar{w}$ such that $A_1\bar{w} = 0$. For small enough $\epsilon > 0$ we have that

$$A\begin{bmatrix} \bar{x} \pm \epsilon\bar{w} \\ 0 \end{bmatrix} = A_1(\bar{x} \pm \bar{w}) = b, \quad \text{and} \quad \begin{bmatrix} \bar{x} \pm \epsilon\bar{w} \end{bmatrix} > 0.$$

Therefore the vectors $y \triangleq \left[\begin{smallmatrix} \bar{x}+\epsilon\bar{w} \\ 0 \end{smallmatrix}\right]$ and $z \triangleq \left[\begin{smallmatrix} \bar{x}-\epsilon\bar{w} \\ 0 \end{smallmatrix}\right]$ are feasible points for some ϵ small enough. This shows that $x = (y + z)/2$ is the average of two different feasible points, and, hence, is not a vertex.

Suppose x is not a vertex. That is, $x = (y+z)/2$ for two feasible points $y \neq x$, $z \neq x$. Since $y \geq 0$ and $z \geq 0$ and $(y+z)/2 = x = \left[\begin{smallmatrix}\bar{x}\\0\end{smallmatrix}\right]$, we must have that y and z are also of the form $y = \left[\begin{smallmatrix}\bar{y}\\0\end{smallmatrix}\right]$ and $z = \left[\begin{smallmatrix}\bar{z}\\0\end{smallmatrix}\right]$. Finally, then, A_1 can not have full column rank because $\bar{x}-\bar{y}$ is nonzero and $A_1(\bar{x}-\bar{y}) = A(x-y) = Ax - Ay = b - b = 0$. ■

It's time for the Simplex method.

3.5.3 The Simplex method

As said earlier, the idea of the Simplex method is to move repeatedly from vertex to vertex in a direction that decreases the objective function. Now that we know how to characterize vertices we can state the algorithm in full detail. After that we go through the steps of the algorithm one at a time.

The Simplex Algorithm

Input: $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and an initial vertex x^v of $\{x : Ax = b,\ x \geq 0\}$.
Output: An optimal solution $x^v \in \mathbb{R}^n$ of $\operatorname{argmin}_x\{c^{\mathrm{T}}x : Ax = b,\ x \geq 0\}$.
Assumptions: A has full row rank.
Step 1: Partition x^v, c and A as

$$x^v = \begin{bmatrix} x^v_B \\ 0 \end{bmatrix}, \quad c = \begin{bmatrix} c_B \\ c_N \end{bmatrix}, \quad A = \begin{bmatrix} A_B & A_N \end{bmatrix}, \tag{3.3}$$

where $x^v_B, c_B \in \mathbb{R}^m$, $c_N \in \mathbb{R}^{n-m}$ and $A_B \in \mathbb{R}^{m\times m}$ with A_B nonsingular.
Step 2: Define $\bar{c}^{\mathrm{T}} \triangleq c_N^{\mathrm{T}} - c_B^{\mathrm{T}} A_B^{-1} A_N$. **Exit if** $\bar{c} \geq 0$ (then x^v is optimal).
Step 3: Select any index k such that $\bar{c}_k < 0$.
Step 4: Define $w := A_B^{-1} A_N e_k$ and determine
$\theta^* \triangleq \min_{1\leq j\leq m}\{\frac{x^v_j}{w_j} : w_j > 0\}$.
(e_k denotes the kth column of I_{n-m})
Step 5: Update

$$x^v := \begin{bmatrix} A_B^{-1}(b - \theta^* A_N e_k) \\ \theta^* e_k \end{bmatrix}.$$

(Then x^v is again a vertex)
Step 6: Let $p \in \{1, \ldots, m\}$ be an index such that $\theta^* = x^v_p / w_p$.
Step 7: Swap the pth and $m+k$th entries of x^v;
Swap the pth and $m+k$th column of A. (Then A and x^v are already of the form (3.3) and A_B is nonsingular)
Step 9: Goto Step 1.

An explanation of the steps follows below.

It is assumed in the algorithm that A has full row rank A. This is a natural assumption for the following reason. If A does not have full row rank then there is a nonsingular matrix U such that $UA = \left[\begin{smallmatrix} A_1 \\ 0 \end{smallmatrix}\right]$ with A_1 having full row rank. Partition Ub similarly as $\left[\begin{smallmatrix} b_1 \\ b_2 \end{smallmatrix}\right]$ with b_1 having as many rows as A_1. Then

$$Ax = b \;\Leftrightarrow\; UAx = Ub \;\Leftrightarrow\; \left[\begin{smallmatrix} A_1 \\ 0 \end{smallmatrix}\right] x = \left[\begin{smallmatrix} b_1 \\ b_2 \end{smallmatrix}\right].$$

If b_2 is nonzero then obviously the above equation has no solution x and, hence, the LP is infeasible. If on the other hand $b_2 = 0$ we may replace $Ax = b$ with the equivalent $A_1 x = b_1$. By construction A_1 has full row rank.

To find an initial vertex x^v as is required for the algorithm is a problem in its own right, and this is commonly called *phase 1*. To solve the phase 1 problem we will use.... the Simplex method! Honestly, this is no circular reasoning. Namely consider the augmented LP in variables x and y,

$$\min_{x \in \mathbb{R}^n,\, y \in \mathbb{R}^m} \sum_{i=1}^{m} y_i \quad \text{subject to} \quad Ax + y = b,\; x \geq 0,\; y \geq 0.$$

An initial vertex here is simply $(x = 0, y = b)$, so the Simplex method can be applied[3] to find an optimal solution (x^*, y^*). Obviously our original LP $\min_x \{c^{\mathrm{T}} x \,:\, Ax = b,\ x \geq 0\}$ is feasible if and only if $y^* = 0$ and, in that case, $x^v \triangleq x^*$ is the initial vertex for the original problem we are looking for. A neat trick.

By Lemma 3.5.2, the columns of A that correspond to the nonzero entries of x^v are linearly independent. Since A has full row rank m, the number of nonzero entries of x^v is at most m. This shows that, after possible permutation, we can assume that x^v and A are of the form (3.3) with A_B nonsingular and $x_B \in \mathbb{R}^m$. (The subscripts B and N stand for *basic* and *nonbasic*.) If we partition c and x likewise, as in (3.3), then the LP becomes

$$\min_{x_B,\, x_N} c_B^{\mathrm{T}} x_B + c_N^{\mathrm{T}} x_N\,, \quad \text{subject to} \quad A_B x_B + A_N x_N = b, \quad x_B \geq 0, \quad x_N \geq 0.$$

The good thing of this partition is that A_B is nonsingular and so the term x_B can be seen as a function of x_N,

$$x_B(x_N) = A_B^{-1}(b - A_N x_N).$$

We then obtain the objective function in terms of the free parameter x_N:

$$\begin{aligned} c^{\mathrm{T}} x &= c_B^{\mathrm{T}} x_B(x_N) + c_N^{\mathrm{T}} x_N \\ &= c_B^{\mathrm{T}} A_B^{-1}(b - A_N x_N) + c_N^{\mathrm{T}} x_N \\ &= c_B^{\mathrm{T}} A_B^{-1} b + (c_N^{\mathrm{T}} - c_B^{\mathrm{T}} A_B^{-1} A_N) x_N\,. \end{aligned}$$

[3] We need $b \geq 0$. That can be achieved by changing some signs if necessary.

For $x = x^v$ we have that $x_N = 0$ so at that vertex the objective function equals $c_B A_B^{-1} b$. If all entries of $\bar{c}^{\mathrm{T}} \triangleq (c_N^{\mathrm{T}} - c_B^{\mathrm{T}} A_B^{-1} A_N)$ are nonnegative then x^v is an optimal solution because for any feasible $x_N \geq 0$ the product $\bar{c}^{\mathrm{T}} x_N$ is then nonnegative and hence taking x_N nonzero can only increase the objective function. We have proved the validity of **Step 2**.

Next assume that some entry of $\bar{c}$, say entry $\bar{c}_k$, is less than zero. Parameterize x as a function of $\theta \in \mathbb{R}$ as

$$x(\theta) \triangleq \begin{bmatrix} x_B(x_N) \\ x_N \end{bmatrix}, \quad \text{where} \quad x_N \triangleq \theta e_k.$$

The e_k denotes the kth unit vector (i.e., the k column of I_{n-m}). Note that $x(0) = x^v$. The objective function at $x(\theta)$ is

$$c^{\mathrm{T}} x(\theta) = c_B^{\mathrm{T}} x_B(x_N) + c_N^{\mathrm{T}} x_N(\theta) = c_B^{\mathrm{T}} A_B^{-1} b + \bar{c}^{\mathrm{T}}(\theta e_k) = c_B^{\mathrm{T}} A_B^{-1} b + \theta \bar{c}_k.$$

Since $\bar{c}_k < 0$ we see that increasing θ decreases the objective function. It is readily verified that the θ^* in **Step 4** is the largest possible θ we can choose without $x(\theta)$ becoming infeasible.

That validity of the remainder of the steps of the Simplex algorithm is left as an exercise (Exercise 3.5.3).

Pivot rules

In some cases x_B may have a zero entry x_j, and in that case it is possible that we can not increase θ at all ($\theta^* = 0$). Nevertheless we may still proceed and interchange the pth and $m + k$th column and hope that in the next iteration $\theta^* > 0$.

Generally there is a freedom in choice of index k and p and over the years several different "pivot rules" that specify which k and p should be taken have been proposed. Among these are pivot rules that may get stuck in an infinite loop. (Apparently some of these are actually used in practice!) There are also pivot rules that are proven to avoid infinite loops.

All the well known pivot rules require in worst case an exponential number of iterations. It is still an open question whether or not there is a pivot rule that will run in polynomial time.

Exercise 3.5.3.

1. Write a Matlab macro for the Simplex method and use whatever pivot rule you want. Include the phase 1 problem.

2. The most time consuming steps are the linear equations that are to be solved at several stages. Try your Matlab macro on a few examples and print the difference of the matrix A_B^{-1} with the matrix A_B^{-1} after it has been updated in Step 5. This will indicate that

   $$\text{the new } A_B^{-1} = \text{the old } A_B^{-1} + W$$

   where W is a matrix of rank one. Find an expression for W in terms of the vector w as defined in Step 4. (Hint: Use that $w = A_B^{-1} A_N e_k$ and that the new A_B can be expressed as $A_{B,\mathrm{old}} + (A_{*(m+k)} - A_{*p}) e_p^{\mathrm{T}}$ where A_{*p} denotes the the pth column of A and e_p the pth column of the identiity I_p. Use the matrix inversion lemma.)

3. Show that in the Simplex algorithm A_B is nonsingular at all times and that x^v is a vertex at all times.

4. The findings of Item 2 can be used to speed up the Matlab macro. (Do not do this.) Would the speeded Matlab macro be as numerically reliable as the one of Item 1?

□

3.6 Lower bounds and duality

Duality is a very important concept in mathematics and in the theory of LPs in particular. Typically what happens is that "difficult" results are easy in the dual form and the other way around. Also, with duality results problems can often be transformed into something that looks completely different but is in fact equivalent. The aesthetics of duality theory is undisputed.

There are several ways to introduce duality for LPs. We choose the one based on the following classic duality theorem.

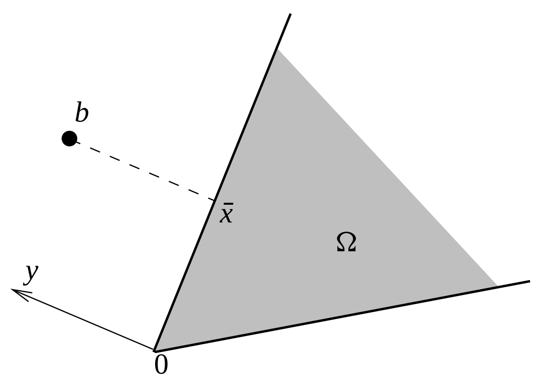

Figure 3.3: Separating hyper-plane.

Theorem 3.6.1 (Separating hyper-planes). *Let Ω be a closed convex cone of $\mathbb{R}^n$. Then $b \notin \Omega$ if and only if for some vector $y \in \mathbb{R}^n$ there holds*

$$y^{\mathrm{T}} b > 0, \quad \textit{and} \quad y^{\mathrm{T}} x \leq 0 \ \forall x \in \Omega.$$

Proof. Figure 3.3 sort of explains the idea. Since Ω is closed, the minimum $\|b-x\|_2$ over all $x \in \Omega$ is attained at some $\bar{x} \in \Omega$. Since Ω is a cone (see appendix) we have that $\lambda\bar{x} \in \Omega$ for all $\lambda > 0$. By definition then $\|\lambda\bar{x}-b\|_2^2$ is the smallest for $\lambda = 1$ and, hence, $\partial\|\lambda\bar{x}-b\|_2^2/\partial\lambda = 2(\lambda\bar{x}-b)^{\mathrm{T}}\bar{x} = 0$ for $\lambda = 1$. That is, $(\bar{x}-b)^{\mathrm{T}}\bar{x} = 0$. Define $y = b - \bar{x}$. We have

$$y^{\mathrm{T}} b = (b-\bar{x})^{\mathrm{T}}(b-\bar{x}+\bar{x}) = \|b-\bar{x}\|_2^2 > 0.$$

Let x be an arbitrary element of Ω. By convexity $\bar{x} + \mu(x - \bar{x})$ is an element of Ω for every $\mu \in [0, 1]$. Consider

$$\begin{aligned} \|b - (\bar{x} + \mu(x - \bar{x}))\|_2^2 &= \|y\|_2^2 - 2\mu y^{\mathrm{T}}(x - \bar{x}) + \mu^2 \|x - \bar{x}\|_2^2 \\ &= \|b - \bar{x}\|_2^2 - 2\mu y^{\mathrm{T}} x + o(\mu). \end{aligned}$$

Since $\bar{x}$ minimizes $\|b - x\|_2$ we must have that $y^{\mathrm{T}} x \leq 0$. ■

In other words, $b \cap \Omega = \emptyset$ iff there is hyper-plane $\{x : y^{\mathrm{T}} x = 0\}$ that separates b from the convex set Ω. This is easy to imagine in $\mathbb{R}^n$ (see Figure 3.3). As a consequence we obtain Farkas' lemma [4] about feasibility of LPs:

Lemma 3.6.2 (Farkas' Lemma). *Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ be given. There is an $x \in \mathbb{R}^n$ such that $Ax = b,\ x \geq 0$ if-and-only-if there is no $y \in \mathbb{R}^m$ such that $y^{\mathrm{T}} A \leq 0$ and $y^{\mathrm{T}} b > 0$.*

Proof. No x exists such that $Ax = b,\ x \geq 0$ if-and-only-if b does not intersect the closed cone $\{Ax : x \geq 0\}$. By the above separating hyper-plane theorem this is the case iff for some y we have that $y^{\mathrm{T}} b > 0$ and $\{y^{\mathrm{T}} Ax : x \geq 0\} \leq 0$. Obviously $y^{\mathrm{T}} Ax \leq 0$ for all $x \geq 0$ iff $y^{\mathrm{T}} A \leq 0$. ■

Farkas' Lemma exhibits a property typical to dual results, namely that "something does not exist" iff "something else does exist".

3.6.1 duality for LPs

In its original form Farkas' duality result says nothing about the objective function. To get a duality result that includes the objective function we might consider the problem whether or not for a given α the set

$$\{x : Ax = b,\ x \geq 0,\ c^{\mathrm{T}} x \leq \alpha\}$$

is feasible. This is a feasibility problem, and so Farkas' Lemma can be applied. This is the result:

Theorem 3.6.3. *Consider the "primal" LP*

$$z_P \triangleq \min_{x \in \mathbb{R}^n} \{ c^{\mathrm{T}} x : Ax = b,\ x \geq 0 \}$$

and "dual" LP

$$z_D \triangleq \max_{y \in \mathbb{R}^m} \{ y^{\mathrm{T}} b : y^{\mathrm{T}} A \leq c^{\mathrm{T}} \}.$$

If the primal LP is feasible with $z_P > -\infty$ or if the dual LP is feasible with $z_D < \infty$ then they are equivalent, i.e., then $z_P = z_D$.

[4] Farkas' lemma is very important, and is used for example to prove the Karush-Kuhn-Tucker conditions, on first order conditions for constraint optimization problems, see (Flecther 1987).

Proof. Take an arbitrary $\alpha \in \mathbb{R}$.

$$
\begin{aligned}
z_P > \alpha \quad &\Leftrightarrow \quad \nexists x \in \mathbb{R}^n \quad \text{s.t.} \quad c^T x \le \alpha,\ Ax = b,\ x \ge 0 \\
&\Leftrightarrow \quad \nexists x \in \mathbb{R}^n,\ x_{n+1} \in \mathbb{R} \quad \text{s.t.} \quad \begin{bmatrix} A & 0 \\ c^T & 1 \end{bmatrix} \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} = \begin{bmatrix} b \\ \alpha \end{bmatrix}, \quad \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} \ge 0 \\
&\Leftrightarrow \quad \exists y \in \mathbb{R}^m,\ y_{m+1} \in \mathbb{R} \quad \text{s.t.} \\
&\qquad\qquad \begin{bmatrix} y^T & y_{m+1} \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} > 0, \qquad \begin{bmatrix} y^T & y_{m+1} \end{bmatrix} \begin{bmatrix} A & 0 \\ c^T & 1 \end{bmatrix} \le 0 \\
&\Leftrightarrow \quad \exists y \in \mathbb{R}^m, \quad y_{m+1} \in \mathbb{R} \quad \text{s.t.} \\
&\qquad\qquad y^T b + y_{m+1}\alpha > 0, \qquad y^T A + y_{m+1} c^T \le 0, \quad y_{m+1} \le 0
\end{aligned}
\tag{3.4}
$$

If $y_{m+1} = 0$ the conditions of (3.4) is exactly that the primal is infeasible. If y_{m+1} is nonzero we may scale it to -1. In that case the conditions (3.4) are the same as that $z_D > \alpha$. So we have shown that $z_P > \alpha$ if and only if $z_D > \alpha$. Since this holds for any α, the primal and dual must be equivalent. ∎

3.6.2 The duality gap

Consider again the pair of primal/dual LPs, which we now denote as P and D:

$$
\begin{aligned}
P \quad &: \quad z_{\mathrm{P}} \triangleq \min_{x \in \mathbb{R}^n} \{ c^T x \,:\, Ax = b,\ x \ge 0 \}, \\
D \quad &: \quad z_{\mathrm{D}} \triangleq \max_{y \in \mathbb{R}^m} \{ y^T b \,:\, y^T A \le c^T \}.
\end{aligned}
$$

For any x feasible in P and y feasible in D there holds

$$
y^T b \le z_D = z_P \le c^T x.
$$

Therefore for any such x and y the optimal solution z_P must lie in the interval $[y^T b, c^T x]$. The difference $c^T x - y^T b$ is called the *duality gap*. The duality gap is nonnegative, and only if x is optimal in P and y is optimal in D is the duality gap zero. The duality gap therefore serves as an indicator how close feasible x and y are to being optimal. This is very useful. There is for example a version of the Simplex method that uses the duality gap, and the interior point algorithm discussed later uses it as well.

For later use we will rewrite the duality gap a bit. Consider the same primal/dual LPs, but written a bit differently:

$$
\begin{aligned}
P \quad &: \quad \min_{x \in \mathbb{R}^n} \{ c^T x \,:\, Ax = b,\ x \ge 0 \} \\
D \quad &: \quad \max_{y \in \mathbb{R}^m,\, s \in \mathbb{R}^n} \{ y^T b \,:\, y^T A + s^T = c^T,\ s^T \ge 0 \}
\end{aligned} .
$$

The difference is that we added a variable s in the description of the dual. It is readily verified that the duality gap $c^T x - b^T y$ is now simply

$$
s^T x.
$$

Remark: Dual optimization problems also exist for certain infinite dimensional optimization problems and "often" the duality gap between the optimal solutions is zero as well. If you have to compute a solution of an infinite dimensional optimization problem it makes sense to optimize over some "rich enough" finite dimensional subset, and hope for the best. If the dual is available you can do the same with the dual problem. The benefit is now that the duality gap between the two approximate optimal solutions can be computed and it will give a good indication (and hard bounds) on the accuracy of the approximate optimal solutions.

3.7 An interior point method

In this section we review an interior point algorithm which is one of the hot topics of the last couple of years. These IPMs are like the IPMs we discussed in the previous chapter, but their application to LPs allows for much more sophisticated results, theoretical results as well as practical results (speed of computation). There are hundreds of variations of IPMs. We will outline Ye's interior point method based on a primal-dual "barrier" function known as a primal-dual potential function (Ye 1991). A similar IPM is described in Freund (1991).

All IPMs for LPs share the following idea:

a) Construct a "barrier" function $G(x)$ that is well defined for strict feasible x and is $-\infty$ only at the optimal $x = x^*$.

b) Generate a sequence $\{x^{(k)}\}$ such that $\lim_{k\to\infty} G(x^{(k)}) = -\infty$.

c) Stop if $G(x^{(k)})$ is negative enough.

In many of the potential reduction methods the $x^{(k)}$ are themselves the solution of a minimization problem. Karmakar's method (1984) that started it all is essentially like that. Ye's method (1991) was the first in which the iterates $x^{(k)}$ are calculated in one go, while retaining the same order of number of iterates $x^{(k)}$.

As the name suggests, the iterates in IPMs are in the strict interior of the feasible set. This is at first sight uncomforting considering that the optimal x^* lies on the boundary of the feasible set. It turns out that this is hardly a problem. For LPs in which the data is rational there is a way to obtain exact optimal solutions (See notes and references). In the general case with real-valued data the iterates $x^{(k)}$ converge so fast, that for every tolerance $\epsilon > 0$ the running time to obtain ϵ-optimal solutions is polynomial and which depends only very mildly on ϵ. More on this later.

3.7.1 Ye's primal-dual IPM

The IPM of Ye (1991) that we explain in this section is a primal-dual method, that is, it solves the primal and dual LP at the same time. It should be no surprise that the duality gap $s^{\mathrm{T}}x$ will be used as a stopping criterion. The primal-dual LP is duplicated below

$$\begin{array}{lcl} P & : & \min_x\{\, c^{\mathrm{T}}x \,:\, Ax = b,\ x \geq 0\,\} \\ D & : & \max_{y,\,s}\{\, y^{\mathrm{T}}b \,:\, y^{\mathrm{T}}A + s^{\mathrm{T}} = c^{\mathrm{T}},\ s \geq 0\,\} \end{array} . \tag{3.5}$$

3.7.2 The barrier function

Todd & Ye (1990) introduced a barrier function called the *primal-dual potential function*. It aims to measure the distance of a feasible pair (x, s) to the boundary as well as give an indication about the duality gap.

Definition 3.7.1. The primal-dual potential function $G_q(x, s)$ associated with the primal-dual LP (3.5) is defined as

$$G_q(x,s) \triangleq q\log(s^{\mathrm{T}}x) - \sum_{j=1}^{n}\log(s_jx_j) \tag{3.6}$$

$$= (q-n)\log(s^{\mathrm{T}}x) + \log\left(\prod_{j=1}^{n}\frac{s^{\mathrm{T}}x}{s_jx_j}\right). \tag{3.7}$$

q may be any positive number. □

The first term in (3.6), $q\log(s^{\mathrm{T}}x)$, measures the duality gap: The more negative it is the smaller the duality gap. The second term in (3.6), $-\sum_{j=1}^{n}\log(x_js_j)$, measures the distance to the boundary of the feasible set: The more positive it is, the closer we are to the boundary. The potential function thus exhibits a trade of between the size of the duality gap and distance to the boundary, and the parameter q is the tuning parameter.

We want that $G_q \to -\infty$ if the duality gap $s^{\mathrm{T}}x \to 0$. Since $\frac{s^{\mathrm{T}}x}{s_jx_j} \geq 1$ we see from (3.7) that we then need to choose $q > n$. After playing with the formula for a while it can be seen that $q = n + \sqrt{n}$ is a good choice. We have the following result.

Lemma 3.7.2. *Take $q = n + \sqrt{n}$. Let (x, s) be feasible primal-dual vectors. If $G_q(x, s) \leq \sqrt{n}\log\epsilon$ then*

$$s^{\mathrm{T}}x \leq \epsilon.$$

Proof. Using (3.7) we see that $\sqrt{n}\log(s^{\mathrm{T}}x) = G_q(x,s) - \log\prod\frac{s^{\mathrm{T}}x}{s_jx_j} \leq \sqrt{n}\log\epsilon - \log\prod\frac{s^{\mathrm{T}}x}{s_jx_j}$. Because $\frac{s^{\mathrm{T}}x}{s_jx_j} \geq 1$, we have that $\sqrt{n}\log(s^{\mathrm{T}}x) \leq \sqrt{n}\log\epsilon$. ■

So if want to know the optimal $c^{\mathrm{T}}x^*$ with an error of at most 10^{-15} then we only need $G_q(x,s) \leq -\sqrt{n}(15\log 10) = -34.54\sqrt{n}$.

3.7.3 The algorithm

The idea is to generates iterates $\{x^{(k)}, s^{(k)}\}$ that reduce the potential function $G_q(x, s)$ each iteration by a constant positive amount. In view of Lemma 3.7.2 this then means that after $O(\sqrt{n})$ iterations the duality gap is ensured to be less than some given ϵ. Ye's algorithm shows that this can indeed be achieved. The algorithm is remarkably simple, although the derivation is a bit tricky, and some arguments seem to come from nowhere[5]. For $x \in \mathbb{R}^n$ the matrix $T_x \in \mathbb{R}^{n\times n}$ is defined as

$$T_x \triangleq \begin{bmatrix} x_1 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & x_n \end{bmatrix}.$$

Ye's primal-dual interior point algorithm

---

input: $m, n \in \mathbb{N}$, $\epsilon > 0$, and $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and initial
$x^{(0)},\ y^{(0)},\ s^{(0)}$ such that
$Ax^{(0)} = b,\ x^{(0)} > 0,\ y^{(0)\mathrm{T}}A + s^{(0)\mathrm{T}} = c^{\mathrm{T}},\ s^{(0)} > 0.$
output: A feasible pair $(x^{(k)}, s^{(k)})$ of (3.5) such that $s^{(k)\mathrm{T}}x^{(k)} < \epsilon$.
assumptions: A has full row rank.
begin
$q := n + \sqrt{n}$, $k := 0$
repeat
1) Do a scaling with respect to $x^{(k)}$:
$x' = T^{-1}_{x^{(k)}}x^{(k)} = \underline{1}, \quad s' = T_{x^{(k)}}s^{(k)}, \quad A' = AT_{x^{(k)}}$
2) $g := q/(s'^{\mathrm{T}}\underline{1})s' - \underline{1}$
3) $d := (I - A'^{\mathrm{T}}(A'A'^{\mathrm{T}})^{-1}A')g$
4) If $\|d\|_2 \geq 0.4$ then do a primal step: $x' := \underline{1} - \frac{1}{4\|d\|_2}d$,
else do a dual step: $s' := \frac{s'^{\mathrm{T}}\underline{1}}{q}(d + \underline{1})$
5) Scale back to original domain: $k := k + 1$,
$s^{(k)} := T^{-1}_{x^{(k-1)}}s',\ \ x^{(k)} := T_{x^{(k-1)}}x'$
until $s^{\mathrm{T}}x < \epsilon$
end

An explanation follows. As with the Simplex method, the assumption that A has full row rank is quite natural and can always be achieved. To find initial strictly feasible $x^{(0)}, s^{(0)} > 0$ is not always easy, but suppose we have such a pair. In Step 1 a transformation of the variables is performed. Interestingly the $x^{(k)}$ is transformed to

$$x' \triangleq T^{-1}_{x^{(k)}}x^{(k)} = \underline{1} \triangleq \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

[5]A theoretically more appealing IPM is discussed in the next chapter.

It is easy to verify that in the transformed variables the primal and dual LPs become

$$\begin{array}{lll} P & : & \min_{x'}\{ c'^{\mathrm{T}} x' \,:\, A'x' = b,\ x' \geq 0 \} \qquad (\text{with } c' := T^{-1}_{x^{(k)}}) \\ D & : & \max_{y',\, s'}\{ y'^{\mathrm{T}} b \,:\, y'^{\mathrm{T}} A' + s'^{\mathrm{T}} = c'^{\mathrm{T}},\ s' \geq 0 \} \end{array}. \tag{3.8}$$

It is important to note that the duality gap $s^{\mathrm{T}} x$ and the potential function $G_q(x, s)$ are invariant under such transformations[6].

One way to decrease $G_q(x', s')$ is to keep s' fixed and move x' in the direction $-g$, where g is defined as the gradient of G_q with respect to x evaluated at (x', s'):

$$\begin{aligned} g &\triangleq \nabla_x G_q(x, s)\big|_{(x,s)=(\underline{1},s')} \\ &= \frac{q}{s^{\mathrm{T}} x} s - \begin{bmatrix} \frac{1}{x_1} \\ \vdots \\ \frac{1}{x_n} \end{bmatrix}\Bigg|_{(x,s)=(\underline{1},s')} \\ &= \frac{q}{s'^{\mathrm{T}} x} s' - \underline{1}. \end{aligned}$$

This is Step 2 of the algorithm. The $-g$ is a direction of descent for G_q because

$$G_q(x' - \epsilon g, s') = G_q(x', s') - \epsilon g^{\mathrm{T}} \nabla_x G_q(x', s') + \mathrm{o}(\epsilon) = G_q(x', s') - \epsilon \|g\|_2^2 + \mathrm{o}(\epsilon).$$

We can not move x' in the direction of $-g$, however, because $A'(x' - \epsilon g) = b - \epsilon A' g$ is generally not equal to b, i.e., $x' - \epsilon g$ is not feasible. O.k., so why not move x' in the direction $-d$ where d is the projection of g onto the space $\{x : A'x = 0\}$. It is trivial to check that, in the usual inner product, d is given by the formula in Step 3. Note that $-d$ is still a direction of descent because

$$G_q(x' - \epsilon d, s') = G_q(x', s') - \epsilon d^{\mathrm{T}} \nabla_x G_q(x', s') + \mathrm{o}(\epsilon) = G_q(x', s') - \epsilon \|d\|_2^2 + \mathrm{o}(\epsilon).$$

Now consider Step 4 for the case that $\|d\|_2 \geq 0.4$. In that case x' is updated to $x' = \underline{1} - \frac{1}{4\|d\|_2} d$. By construction we have that $Ax' = b$, and from the expression $x' = \underline{1} - \frac{1}{4\|d\|_2} d$ it is further immediate that $x' > 0$. Therefore the updated x' is strictly feasible. The technical part is to show that

Lemma 3.7.3. *If $\|d\|_2 \geq 0.4$ and with x' updated to $x' = \underline{1} - \frac{1}{4\|d\|_2} d$ there holds*

$$G_q(x', s') \leq G_q(x^{(k)}, s^{(k)}) - \frac{7}{120}$$

[6] This componentwise transformation is known as *affine scaling*. The affine scaling used here is due to (Vanderbei, Meketon & Freedman 1986) and (Barnes 1986). Note that after scaling we have $x = \underline{1}$. A scaling can be interpreted as a change in the inner product (hence, affecting projections) or as a way to move x away from the boundary of the feasible set. Note however that although x is moved away from the boundary, the iterates s have to approach zero if the duality gap is to go to zero.

Proof. This is an exact copy of (Goemans 1994). As G_q is invariant under affine scaling we have that $G_q(x^{(k)}, s^{(k)}) = G_q(\underline{1}, s')$.

$$
\begin{aligned}
G_q(x', s') - G_q(\underline{1}, s') &= G_q(\underline{1}^{\mathrm{T}} s' - \frac{1}{4\|d\|_2} d, s') - G_q(\underline{1}, s') \\
&= q \log\bigl(\underline{1}^{\mathrm{T}} - \frac{d^{\mathrm{T}} s}{4\|d\|_2}\bigr) - \sum_{j=1}^{n} \log\bigl(1 - \frac{d_j}{4\|d\|_2}\bigr) - \sum_{j=1}^{n} \log s'_j \\
&\quad - q \log\bigl(\underline{1}^{\mathrm{T}} s'\bigr) + \sum_{j=1}^{n} \log 1 + \sum_{j=1}^{n} \log s'_j \\
&= q \log\bigl(1 - \frac{d^{\mathrm{T}} s'}{4\|d\|_2 \underline{1}^{\mathrm{T}} s}\bigr) - \sum_{j=1}^{n} \log\bigl(1 - \frac{d_j}{4\|d\|_2}\bigr).
\end{aligned}
$$

Using the relation

$$
-z - \frac{z^2}{2(1-a)} \leq \log(1-z) \leq -z \tag{3.9}
$$

which holds for $|z| \leq a < 1$ we get

$$
\begin{aligned}
G_q(x', s') - G_q(\underline{1}, s') &\leq -\frac{q d^{\mathrm{T}} s'}{4\|d\|_2 \underline{1}^{\mathrm{T}} s'} + \sum_{j=1}^{n} \frac{d_j}{4\|d\|_2} + \sum_{j=1}^{n} \frac{d_j^2}{16\|d\|_2^2 \frac{3}{4}} \text{ (for } a = \frac{1}{4}) \\
&= -\frac{q d^{\mathrm{T}} s'}{4\|d\|_2 \underline{1}^{\mathrm{T}} s'} + \frac{\underline{1}^{\mathrm{T}} d}{4\|d\|_2} + \frac{1}{24} \\
&= \frac{1}{4\|d\|_2} (e - \frac{q}{\underline{1}^{\mathrm{T}} s'})^{\mathrm{T}} d + \frac{1}{24} \\
&= \frac{1}{4\|d\|_2} (-g^{\mathrm{T}}) d + \frac{1}{24} \\
&= -\frac{\|d\|_2^2}{4\|d\|_2} + \frac{1}{24} \\
&= -\frac{\|d\|_2}{4} + \frac{1}{24} \\
&\leq -\frac{1}{10} + \frac{1}{24} = \frac{7}{120} = 0.0583333..
\end{aligned}
$$

We used here that $g^{\mathrm{T}} d = \|d\|_2^2$. ∎

Now consider the case that $\|d\|_2 < 0.4$. In this case Step 4 leaves x' unchanged and updates s' to $\underline{1}^{\mathrm{T}} s'/q(g-d)$. To see why the updated s' is strictly feasible we argue as follows. Before updating we know that s' is strictly feasible, therefore there is a y' such that

$$
y'^{\mathrm{T}} A' + s'^{\mathrm{T}} = c'^{\mathrm{T}}.
$$

Now $g - d = A'^{\mathrm{T}}(A'A'^{\mathrm{T}})^{-1}A'g$, that is, $g - d$ lies in the range of A'^{T}. Therefore for every μ there is a $\tilde{y}$ such that

$$\tilde{y}^{\mathrm{T}}A' + (s' + \mu(g-d))^{\mathrm{T}} = c'^{\mathrm{T}}.$$

For strict feasibility of $s' + \mu(g-d)$ it is therefore sufficient to show that all entries are positive. Take $\mu := (\underline{1}^{\mathrm{T}}s')/q > 0$. Then

$$\begin{aligned} s' + \mu(g-d) &= s' - \frac{\underline{1}^{\mathrm{T}}s'}{q}(g-d) \\ &= s' - \frac{\underline{1}^{\mathrm{T}}s'}{q}((\frac{q}{s'^{\mathrm{T}}x}s' - \underline{1}) - d) \\ &= \frac{\underline{1}^{\mathrm{T}}s'}{q}(d + \underline{1}). \end{aligned}$$

In Step 4 this update is performed only if $\|d\|_2 < 0.4$ and in such cases it is direct that $s' > 0$.

Lemma 3.7.4. *If $\|d\|_2 < 0.4$ and with s' updated to $s' := \frac{s'^{\mathrm{T}}\underline{1}}{q}(d + \underline{1})$ there holds*

$$G_q(x', s') \leq G_q(x^{(k)}, s^{(k)}) - \frac{1}{6}$$

Proof. This is an exact copy of (Goemans 1994). Let s' be as before updating and let $\tilde{s}$ be the update $\tilde{s} := \frac{s'^{\mathrm{T}}\underline{1}}{q}(d + \underline{1})$. Let t denote the duality gap before updating, $t := \underline{1}^{\mathrm{T}}s'$.

$$\begin{aligned} \sum_{j=1}^{n} \log \tilde{s}_j - n\log\Big(\frac{\underline{1}^{\mathrm{T}}\tilde{s}}{n}\Big) &= \sum_{j=1}^{n} \log\Big(\frac{t}{q}(1 + d_j)\Big) \;\; - n\log\Big(\frac{t}{q}(1 + \frac{\underline{1}^{\mathrm{T}}d}{n})\Big) \\ &= \sum_{j=1}^{n} \log(1 + d_j) \;\; - n\log(1 + \frac{\underline{1}^{\mathrm{T}}d}{n}) \\ &\geq \sum_{j=1}^{n} (d_j - \frac{d_j^2}{2\frac{3}{5}}) - n\frac{\underline{1}^{\mathrm{T}}d}{n} \quad \text{(using (3.9) with } a = \frac{2}{5}) \\ &\geq -\frac{\|d\|_2^2}{6/5} \geq -\frac{2}{15}. \end{aligned} \tag{3.10}$$

Using (3.10) and the inequality

$$\sum_{j=1}^{n} \log s_j \leq n \log \frac{\underline{1}^{\mathrm{T}}s}{n},$$

which follows from concavity of the logarithm function, we have

$$\begin{aligned} G_q(\underline{1}, \tilde{s}) - G_q(\underline{1}, s') &= q \log \frac{\underline{1}^{\mathrm{T}} \tilde{s}}{\underline{1}^{\mathrm{T}} s'} - \sum_{j=1}^{n} \log \tilde{s}_j + \sum_{j=1}^{n} \log s'_j \\ &\leq q \log \frac{\underline{1}^{\mathrm{T}} \tilde{s}}{\underline{1}^{\mathrm{T}} s'} + \frac{2}{15} - n \log\big(\frac{\underline{1}^{\mathrm{T}} \tilde{s}}{n}\big) + n \log\big(\frac{\underline{1}^{\mathrm{T}} s'}{n}\big) \\ &= \frac{2}{15} + \sqrt{n} \log \frac{\underline{1}^{\mathrm{T}} \tilde{s}}{\underline{1}^{\mathrm{T}} s'}. \end{aligned}$$

On the other hand $\underline{1}^{\mathrm{T}} \tilde{s} = \frac{t}{q}(n + \underline{1}^{\mathrm{T}} d)$ and recall that $t := \underline{1}^{\mathrm{T}} s$,

$$\frac{\underline{1}^{\mathrm{T}} \tilde{s}}{\underline{1}^{\mathrm{T}} s'} = \frac{1}{q}(n + \underline{1}^{\mathrm{T}} d) \leq \frac{1}{n + \sqrt{n}}(n + 0.4\sqrt{n}),$$

since, by Cauchy-Schwartz, $|\underline{1}^{\mathrm{T}} d| \leq \|e\|_2 \|d\|_2 = \sqrt{n}\|d\|_2$. Combining the above inequalities yields

$$\begin{aligned} G_q(\underline{1}, \tilde{s}) - G_q(\underline{1}, s') &\leq \frac{2}{15} + \sqrt{n} \log\big(1 - \frac{0.6\sqrt{n}}{n + \sqrt{n}}\big) \\ &\leq \frac{2}{15} - \frac{0.6n}{n + \sqrt{n}} \\ &\leq \frac{2}{15} - \frac{3}{10} = -\frac{1}{6}. \end{aligned}$$

■

Summary: Each iteration the potential function G_q is decreased by a positive amount of $\Delta := \min(7/120, 1/6) = 0.05833$ or more. Therefore by Lemma 3.7.2 the algorithm requires a maximum of

$$\frac{G_q(x^{(0)}, s^{(0)}) - \sqrt{n} \log \epsilon}{\Delta}$$

iterations before the duality gap $s^{\mathrm{T}} x$ is guaranteed to be less than ϵ. Ye further showed that an initial pair $(x^{(0)}, s^{(0)})$ can always be constructed cheaply such that $G_q(x^{(0)}, s^{(0)}) = O(\sqrt{n})$. As a result after $O(\sqrt{n})$ iterations the algorithm exits. In each iteration only trivial operations are needed and it easy to see that the most expensive step is the projection $d = (I - A'^{\mathrm{T}}(A'A'^{\mathrm{T}})^{-1}A')g$. To calculate the projection d we need $O(n^3)$ operations[7]. The overall algorithm therefore requires $O(n^{3.5})$ operations.

The value of $\Delta = 0.05833$ is small, but remember that this is a lower bound only and in practice G_q may decrease much more[8]. More important is the factor $n^{3.5}$. It can not be

---

[7]We might for example use Cholesky factorization to find the z such that $(A'A'^{\mathrm{T}})z = A'g$. That requires $O(n^3)$ steps (Golub & Loan 1983). Then $d := g - A'^{\mathrm{T}} z$.

[8]Actually nothing prohibits us from including a line search in $p > 0$ to minimize $G_q(x' - pd, s')$ or $G_q(x', s' + p(g - d))$. Approximate line searches are cheap and dramatically speed up the algorithm.

expected that IPMs beat the Simplex method for modest sized problems. IPMs are primarily useful for large problems with several thousands of variables (which is not uncommon in practice). For such large problems the order of complexity is the single most important number completely overshadowing the value of Δ. Ye's algorithm was the first to obtain an order of complexity of $O(n^{3.5})$. Later Anstreicher & Bosch (1992) showed that it can be brought down further to $O(n^3)$ by using some sort of approximate solutions to linear equations. Freund (1991) who developed almost the same algorithm as Ye, showed that $q = n + \sqrt{n}$ is the optimal choice of q.

Exercise 3.7.5. Write a Matlab macro for Ye's algorithm. Is there a reason to transform back to original domain (Step 5 of the algorithm)? Explain. □

3.8 Notes and References

The text of this chapter is mainly based on the lecture notes by Goemans (1994) and a paper by Goldfarb & Todd (1989).

Many pivot rules for the Simplex method may result in infinite loops, but there are pivot rules that can not get stuck in infinite loops. One of these is the "minimal index rule" (Bland 1977).

For all known pivot rules the number of pivots needed in worst case is exponential. It is as yet not known whether or not there is a pivot rule for which the Simplex method runs in polynomial time.

Many papers on interior point methods assume that the data are integer numbers (or rational numbers) and that all computations are done exact (so, the amount of bits to represent a number is not fixed a priori). It is shown that with such exact computation interior point algorithms, such as Ye's, are polynomial in time *and polynomial in the amount of bits required*. It is not known at present whether or not there are algorithms for LPs whose running time is polynomial and does not depend on the size of the (integer valued) entries of A and b.

After Karmakar's paper of 1984 the next breakthrough came with the book by Nesterov & Nemirovsky (1993). They showed that polynomial time interior point algorithms for LPs can in principle be generalized to almost any convex program. Their result relies on the existence of a barrier function with certain properties. They show that such a barrier function *exists*, but unfortunately *computing* it may still be very, very difficult.

Ye's (1991) was the first that required only $O(\sqrt{n})$ outer iterations without the need to follow the "central path" closely (den Hertog 1993, p.157). The primal-dual potential reduction methods work well in practice (den Hertog 1993, p.160).

4

Semidefinite Programs

Covered are:

1. Linear matrix inequalities (LMIs).
2. Properties of semidefinite programs (SDPs).
3. Review of an interior point polynomial time algorithm for SDPs.

Roughly speaking an SDP is the same as an LP except that the constraints are *matrix* inequalities instead of set of scalar inequalities. As such SDP is an extension of LP. It retains the convexity properties of LP but allows for much more general problems to be considered. One important such class is a class of convex *quadratic* programming problems with quadratic constraints. SDP have wide application. In control theory they are very popular these days, and for many NP-hard problems SDPs can be used to obtain meaningful lower or upper bounds.

Despite the similarities, SDPs and LPs have led different lives. The reason being that SDPs lack the kind of geometrical and combinatorial interpretation that plays such a prominent role in LPs.

The recent interior point algorithms for LPs make less explicit use of geometrical properties than the classic Simplex method, and one of the fortunes of this development has been the recognition that some form of interior point algorithm can also be used for semidefinite programs. As opposed to LPs, for SDPs the recent interior point methods are the first to efficiently solve the problem.

There will be a sense of déjà vu when we describe duality, duality gap and barrier functions, but they will not be quite the same.

4.1 Linear matrix inequalities (LMIs)

A *semidefinite program* (or SDP) is an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & c^{\mathrm{T}}x \\ \text{subject to} & F_0 + \sum_{i=1}^{n} x_i F_i \geq 0, \end{array} \tag{4.1}$$

where $x = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^n$ is the vector over which to optimize, $c \in \mathbb{R}^n$ is a given vector and F_0 and the $\{F_i\}$ are a set of symmetric matrices of equal dimension. The feasible set here is described by a matrix inequality

$$F(x) \triangleq F_0 + \sum_{i=1}^{n} x_i F_i \;\geq\; 0. \tag{4.2}$$

A matrix inequality is defined as follows.

Definition 4.1.1 (Nonnegative definite and positive definite). A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is *nonnegative definite*, denoted $A \geq 0$, if $z^{\mathrm{T}} A z \geq 0$ for all $z \in \mathbb{R}^n$.

A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is *positive definite*, denoted $A > 0$, if $z^{\mathrm{T}} A z > 0$ for all nonzero $z \in \mathbb{R}^n$. □

The appendix lists some more properties of nonnegative definite matrices. One particularly important result is that a symmetric matrix is nonnegative definite if and only if all its eigenvalues are nonnegative.

The inequality (4.2) is known as a *linear matrix inequality (LMI)* in the variable x. This is somewhat confusing considering that $F(x)$ is in fact *affine* in x. We also speak of an LMI if $\geq$ is replaced with $>$. In the rest of this section we examine the LMI in detail.

Lemma 4.1.2. *The feasible set* $\{x \,:\, F(x) \geq 0\}$ *is convex.*

Proof. Suppose $F(x) \geq 0$ and $F(y) \geq 0$. By linearity $F(\lambda x + (1-\lambda)y) = \lambda F(x) + (1-\lambda)F(y)$. Hence for any vector z and any $\lambda \in [0, 1]$ we have that

$$z^{\mathrm{T}}\big(F(\lambda x + (1-\lambda)y)\big)z = \lambda\, z^{\mathrm{T}} F(x) z + (1-\lambda)\, z^{\mathrm{T}} F(y) z \geq 0.$$

■

The feasibility problem to check whether or not the LMI $F(x) \geq 0$ is achieved by some x is sometimes called the *LMI problem*.

Other representations of LMIs Often the variables come in the form of entries of a matrix, say $X \in \mathbb{R}^{n\times n}$. For example the Lyapunov inequality

$$XA + A^{\mathrm{T}} X + I < 0 \tag{4.3}$$

is an LMI in (the entries of) X. We could of course write out the above inequality in terms of the entries x_i of X to obtain the form (4.2), but usually it is better to leave it in the condensed matrix format as it is.

There are many other equivalent ways to represent an LMI. One insultingly simple variation is the LMI $F(x) < 0$ i.e., with $F(x)$ negative definite instead of positive definite. In many cases negative definiteness is more natural.

Stacking LMIs Everyone knows that the intersection of two convex sets is again convex, but is the intersection of two convex sets described by LMIs again a convex set described by an LMI? The answer is yes, namely the intersection of two convex sets

$$\{x : F(x) \geq 0\} \cap \{x : G(x) \geq 0\}$$

can also be expressed by a "single" LMI

$$\{x : \begin{bmatrix} F(x) & 0 \\ 0 & G(x) \end{bmatrix} \geq 0\}.$$

Intersecting convex sets is thus matter of diagonally stacking the LMIs. As an example, the Lyapunov inequality (4.3) with X restricted to positive definite is again an LMI:

$$\begin{bmatrix} XA + A^{\mathrm{T}}X + I & 0 \\ 0 & -X \end{bmatrix} < 0.$$

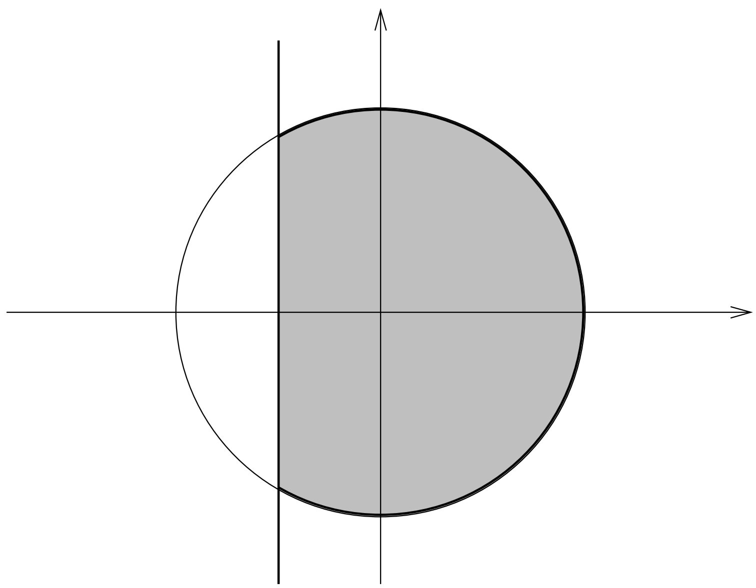

Figure 4.1: A typical feasible set defined by an LMI.

4.1.1 Quadratic feasible sets and Schur complements

Let us begin with an example.

Example 4.1.3. Consider the linear matrix inequality

$$F(x) \triangleq \begin{bmatrix} 1 & 0 & x_1 & 0 \\ 0 & 1 & x_2 & 0 \\ x_1 & x_2 & 1 & 0 \\ 0 & 0 & 0 & x_1 + \frac{1}{2} \end{bmatrix} \geq 0.$$

It is easy to verify that the eigenvalues of $F(x)$ are $x_1 + (1/2)$ and 1 and $1 \pm \sqrt{x_1^2 + x_2^2}$. Since $F(x) \geq 0$ iff all its eigenvalues are nonnegative (see appendix) we see that the feasible set $\{x : F(x) \geq 0\}$ is the intersection of the unit disc $\{x : \sqrt{x_1^2 + x_2^2} \leq 1\}$ with the half-space $\{\left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right] : x_1 \geq -1/2\}$. See Figure 4.1. □

Interesting about this example is something that we will explore in more detail: Apparently with *linear* matrix inequalities we can describe *quadratic* curves, like circles. This may well be the most important feature of LMIs. On the negative side this rules out any hope for a Simplex type method for SDPs as we have no vertex results here.

In Example 4.1.3 we derived the formula for the circle through computation of eigenvalues. For large LMIs computation of eigenvalues is not an easy task, and so we'd better find some other means to connect LMIs with quadratic inequalities. This can be done with the trick of Schur complements.

Definition 4.1.4 (Schur complement). Let M be a partitioned matrix

$$M \triangleq \begin{bmatrix} P & Q \\ R & S \end{bmatrix}.$$

The *Schur complement* of P in M is defined as $S - RP^{-1}Q$. Similarly the Schur complement of S (in M) is defined as $P - QS^{-1}R$. □

We could of course give representation independent definition of Schur complement, but the matrix formulae will be enough for our purposes[1]. Schur complements enter quite naturally in equations with partitioned matrices. For example, if we know that S is nonsingular, then z in the equation

$$\begin{bmatrix} P & Q \\ R & S \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} z \\ 0 \end{bmatrix}$$

satisfies $z = (P - QS^{-1}R)x$. In particular this shows that if S is nonsingular that then M—assuming it is square—is nonsingular iff the Schur complement of S is nonsingular. In fact we have that

$$\det(M) = \det(S) \cdot \det(P - QS^{-1}P).$$

The proof is simple: Take determinants on both sides of the following identity:

$$\begin{bmatrix} P & Q \\ R & S \end{bmatrix} \begin{bmatrix} I & 0 \\ -S^{-1}R & I \end{bmatrix} = \begin{bmatrix} P - QS^{-1}R & -QS^{-1}R \\ 0 & S \end{bmatrix}.$$

What we need is this:

[1] A representation independent definition of Schur complement: Let $M : W \to W$ be a bounded operator on some Hilbert space W. The Schur complement of M with respect to a subspace $V \subset W$ is the map $L : V \to V$ defined as $Lx = PMx$ where P denotes the projection onto V along $\overline{MV^{\perp}}$.

Lemma 4.1.5 (Schur complements for symmetric matrices). *Let P and S be symmetric matrices. The following are equivalent.*

1. $M := \begin{bmatrix} P & Q \\ Q^{\mathrm{T}} & S \end{bmatrix} > 0$.
2. $P > 0$ *and its Schur complement* $S - Q^{\mathrm{T}} P^{-1} Q > 0$.
3. $S > 0$ *and its Schur complement* $P - Q S^{-1} Q^{\mathrm{T}} > 0$.

Moreover, if $S > 0$ then $M \geq 0$ iff $P - QS^{-1}Q^{\mathrm{T}} \geq 0$. Similarly, if $P > 0$ then $M \geq 0$ iff $S - Q^{\mathrm{T}} P^{-1} Q \geq 0$.

Proof. ($1 \Leftrightarrow 2$):

$$\begin{aligned} M > 0 \quad &\Leftrightarrow \quad \begin{bmatrix} x^{\mathrm{T}} & y^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} P & Q \\ Q^{\mathrm{T}} & S \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} > 0 \quad \forall \begin{bmatrix} x \\ y \end{bmatrix} \neq 0 \\ &\Leftrightarrow \quad x^{\mathrm{T}} P x + x^{\mathrm{T}} Q y + y^{\mathrm{T}} Q^{\mathrm{T}} x + y^{\mathrm{T}} S y > 0 \quad \forall \begin{bmatrix} x \\ y \end{bmatrix} \neq 0 \\ &\Leftrightarrow \quad \det P \neq 0 \text{ and} \\ &\qquad (x^{\mathrm{T}} + y^{\mathrm{T}} Q^{\mathrm{T}} P^{-1}) P (x + P^{-1} Q y) + y^{\mathrm{T}} (S - Q^{\mathrm{T}} P^{-1} Q) y > 0 \quad \forall \begin{bmatrix} x \\ y \end{bmatrix} \neq 0 \\ &\Leftrightarrow \quad P > 0 \text{ and } S - Q^{\mathrm{T}} P^{-1} Q > 0. \end{aligned}$$

($1 \Leftrightarrow 3$) goes exactly the same. A another, matrix proof uses the fact that $M > 0$ iff $V^{\mathrm{T}} M V > 0$ if V is nonsingular (see appendix). Then ($1 \Leftrightarrow 2$) follows from the fact that

$$\begin{bmatrix} I & 0 \\ -Q^{\mathrm{T}} P^{-1} & I \end{bmatrix} \begin{bmatrix} P & Q \\ Q^{\mathrm{T}} & S \end{bmatrix} \begin{bmatrix} I & -P^{-1} Q \\ 0 & I \end{bmatrix} = \begin{bmatrix} P & 0 \\ 0 & S - Q^{\mathrm{T}} P^{-1} Q \end{bmatrix}.$$

∎

Consider again Example 4.1.3. The Schur complement of the upper left $\left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ block in

$$\begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & x_2 \\ x_1 & x_2 & 1 \end{bmatrix} \tag{4.4}$$

is

$$1 - \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 1 - (x_1^2 + x_2^2),$$

and so we see right away that (4.4) is nonnegative definite iff (x_1, x_2) lies in the unit disc. That's easy. The good thing about the Schur complement formula is that it shows how to go the other way, we mean, it shows how to transform certain quadratic feasible sets into *linear* matrix inequalities. Some examples from control theory are given in the next section. Two academic examples:

Example 4.1.6. The set $D := \{x : x^{\mathrm{T}}Qx + 2x^{\mathrm{T}}b + c < 0\}$ is a convex set if $Q > 0$. To see this consider the LMI

$$F(x) := \begin{bmatrix} Q & Qx \\ x^{\mathrm{T}}Q & -2x^{\mathrm{T}}b - c \end{bmatrix} > 0.$$

By the Schur complement formula the above holds iff $Q > 0$ and $x^{\mathrm{T}}Qx + 2x^{\mathrm{T}}b + c < 0$, that is, $D = \{x : F(x) > 0\}$. Convexity of D is therefore established. With a bit more effort it can be shown that D is convex if and only if $Q \geq 0$. □

The following example is copied from from Vandenberghe & Boyd (1996). It shows how to "linearize" a difficult optimization problem using the Schur complement.

Example 4.1.7. Suppose we are given the problem

$$\begin{array}{ll} \text{minimize} & \frac{(c^{\mathrm{T}}x)^2}{d^{\mathrm{T}}x} \\ \text{subject to} & \hat{F}(x) \geq 0 \end{array} \tag{4.5}$$

where x, c, d are column vectors and $\hat{F}(x) \geq 0$ is some LMI. Assume $d^{\mathrm{T}}x > 0$ on the feasible set. The objective function is not linear so it is not directly an SDP, but with help of the Schur complement we see that (4.5) is the same as the SDP in the variables x and $t \in \mathbb{R}$

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} \hat{F}(x) & 0 & 0 \\ 0 & t & c^{\mathrm{T}}x \\ 0 & c^{\mathrm{T}}x & d^{\mathrm{T}}x \end{bmatrix} \geq 0. \end{array}$$

□

4.2 Examples of LMIs and SDPs

The IPM for SDPs that we will describe later on in this chapter show that the LMI problem is easily solved, that is, that there are efficient algorithms that check whether or not there is an an x such that $F(x) > 0$, and that will find one such x if any exist. In this section we give examples—most of then from control theory—of LMI problems and SDPs.

Example 4.2.1 (Quadratic stabilization). Suppose that a signal x satisfies $\dot{x} = Ax + Bu$, but that $A(t)$ and $B(t)$ are uncertain matrices in that they are piecewise constant in time, taking one of the L values

$$\{\, [A_1\ B_1], \ldots, [A_L\ B_L]\,\}$$

but we do not no which value they take and how often and when they change. Our objective is to find matrices K and P such that

$$(A_i + B_iK)^{\mathrm{T}}P + P(A_i + B_iK) < 0, \ (i = 1, \ldots, L), \ \text{ and } P > 0. \tag{4.6}$$

If this can be achieved then from standard Lyapunov theory it follows that $u = Kx$ stabilizes the uncertain system. The matrix inequalities (4.6) are not all convex in the variables P and K. With Y defined as $Y = P^{-1}$ and W defined as $W = KP^{-1}$, the inequalities (4.6) may be rewritten as

$$(A_i + B_i W Y^{-1})^{\mathrm{T}} Y^{-1} + Y^{-1}(A_i + B_i W Y^{-1}) < 0, \ (i = 1, \dots, L), \quad \text{and } Y > 0.$$

These are still not LMIs in the variables (Y, W). Multiplying the L inequalities on the left and right by Y we get an equivalent

$$Y A_i + W^{\mathrm{T}} B_i^{\mathrm{T}} + A_i Y + B_i W < 0, \ (i = 1, \dots, L), \quad \text{and } Y > 0.$$

All $L + 1$ inequalities are now LMIs in the variables (Y, W), and, hence, this is a feasibility problem of LMIs and we can consider the problem solved. Once W and Y are found, the controller gain K follows by back-substitution $K = WY^{-1}$.

Khargonekar & Rotea (1991) used a similar trick to show that a certain mixed H_∞/H_2 problem can be cast as a convex optimization problem. □

Example 4.2.2 (Passivity). The first LMI that became popular in control (although certainly not under the name "LMI") is the Lyapunov inequality

$$AP + PA^{\mathrm{T}} < 0, \quad P = P^{\mathrm{T}} > 0.$$

The most famous LMI in control theory is the one associated with the Kalman-Yakubovich-Popov lemma. One version of the KYP lemma says that the following three are equivalent.

1. The system $\dot{x} = Ax + Bu, \ y = Cx + Du$ is stable and for some $\epsilon > 0$ we have

   $$\int_0^t y(t)^{\mathrm{T}} u(t)\, dt \geq \epsilon \int_0^t y(t)^{\mathrm{T}} y(t) + u(t)^{\mathrm{T}} u(t)\, dt \quad \forall t \geq 0$$

   for all inputs u and initial condition $x(0) = 0$. Such systems are sometimes called *very strictly passive*.

2. A has all its eigenvalues in the open left-half plane and $G(s) \triangleq C(sI - A)^{-1}B + D$ satisfies

   $$G(s) + G(s)^{\mathrm{T}} > 0, \quad \forall s \in \{s : \operatorname{Re} s \geq 0\} \cup \infty.$$

3. There is a symmetric matrix $P > 0$ such that

   $$\begin{bmatrix} PA + A^{\mathrm{T}}P & PB - C^{\mathrm{T}} \\ -C + B^{\mathrm{T}}P & -D - D^{\mathrm{T}} \end{bmatrix} < 0.$$

The condition in the third item is a feasibility problem of an LMI. □

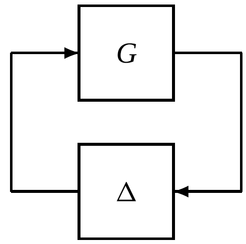

Figure 4.2: A μ problem.

Example 4.2.3 (Robustness problem). One of the most challenging problems in robust control is to determine whether or not a closed loop system remains stable and retains satisfactory performance specifications under uncertainties and variations of some sort. A canonical form of this problem is shown in Figure 4.2. Here the plant G is known and stable, and the block Δ represents the uncertainty which may be nonlinear or time-varying. The *robust stability problem* is to check if the closed loop is stable for all Δs in a given class of passive operators $\mathfrak{D}$.

There is well known result that says that if Δ is passive and G is very strictly passive, then the closed loop is stable (Desoer & Vidyasagar 1975). Since the whole class $\mathfrak{D}$ is assumed passive, we see that very strict passivity of G is enough to ensure robust stability of the closed loop. This passivity result, however, is overly pessimistic as it does not exploit specific properties (such as structure) which $\mathfrak{D}$ usually has. We can often do a lot better by exploiting these properties of $\mathfrak{D}$.

Suppose $\mathcal{E}$ is a set of constant nonsingular matrices such that $\mathfrak{D}E^{-1}$ is passive for every $E \in \mathcal{E}$. It will be clear from Figure 4.3 that robust stability is still ensured if we can find *one*

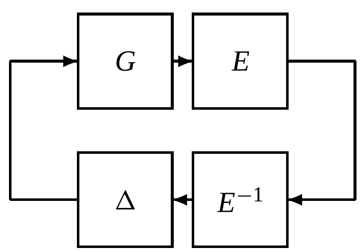

Figure 4.3: A μ problem.

E in this class $\mathcal{E}$ such that EG is very strictly passive. More explicitly, suppose

$$\mathfrak{D} = \{\Delta : \Delta = \begin{bmatrix} \Delta_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Delta_n \end{bmatrix}\},$$

where Δ_i is any passive operator of some fixed dimension. It is easy to verify that ΔE^{-1} is passive for any $\Delta \in \mathfrak{D}$ if

$$E \in \mathbf{E} := \{ \begin{bmatrix} e_1 I & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & e_n I \end{bmatrix} : e_i \in \mathbb{R},\, e_i > 0\}.$$

As a result, the closed loop is robustly stable if there is an $E \in \mathbf{E}$ such that EG is strictly passive. The condition that EG be very strictly passive for some $E \in \mathbf{E}$ is an LMI problem. To see this, let $G(s) = C(sI - A)^{-1}B + D$ be a realization of G. Then EG has realization $EC(sI - A)^{-1}B + ED$, and by Example 4.2.2, EG is very strictly passive iff for some P we have that

$$\begin{bmatrix} PA + A^{\mathrm{T}}P & PB - C^{\mathrm{T}}E^{\mathrm{T}} \\ -EC + B^{\mathrm{T}}P & -ED - D^{\mathrm{T}}E^{\mathrm{T}} \end{bmatrix} < 0, \quad P = P^{\mathrm{T}} > 0. \tag{4.7}$$

This is an LMI problem in the variables P and $E \in \mathbf{E}$. (Note that $\mathbf{E}$ has to be a convex set in order for this to be an LMI problem. In our example $\mathbf{E}$ is indeed convex.)

□

For many classes of operators $\mathfrak{D}$ the robust stability problem is known to be NP-hard (Braatz, Young, Doyle & Morari 1994) (Poljak & Rohn 1993). Then SDP or LMI sufficiency conditions like (4.7) are often about the best we can do. One interesting problem is to find out how conservative the SDP/LMI sufficiency conditions are.

Example 4.2.4 (The largest singular value). Suppose matrix $Z(x)$ depends affinely on x. It is not clear at first sight that the spectral norm (=the largest singular value) $\|Z(x)\|$ of $Z(x)$ is convex in x, but it is, as a Schur complement argument will show. The claim is that $\|Z(x)\| < \gamma$ iff

$$\begin{bmatrix} \gamma I_n & Z(x) \\ Z(x)^{\mathrm{T}} & \gamma I_m \end{bmatrix} > 0. \tag{4.8}$$

This is because the Schur complement of γI_n in (4.8) is

$$\gamma I_m - \frac{1}{\gamma} Z^{\mathrm{T}}(x)Z(x) = \frac{1}{\gamma} \cdot [\gamma^2 I_m - Z(x)^{\mathrm{T}} Z(x)],$$

which is > 0 iff $\|Z(x)\| < \gamma$. The inequality (4.8) is an LMI in x and, as a consequence, $\|Z(x)\|$ is convex in x and finding the minimal $\|Z(x)\|$ over x is an SDP in x and γ:

$$\min_{\gamma \in \mathbb{R},\, x} \gamma, \quad \text{subject to (4.8).}$$

□

Example 4.2.5 (Minimize maximum eigenvalue). Let $A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n$ be a symmetric $m \times m$ matrix depending on x (we assume all A_j are symmetric). The problem

$$\min_{x \in \mathbb{R}^n} \lambda_{\max} A(x),$$

where $\lambda_{\max}$ denotes the maximal eigenvalue, can be cast as the SDP

$$\min_{t \in \mathbb{R},\, x \in \mathbb{R}^n} t \quad \text{subject to } tI_m - A(x) \geq 0.$$

This tells us that the maximal eigenvalue of $A(x)$ is convex in x and that its minimum over x is easy to compute. By the way, $\lambda_{\max} A(x)$ is generally not differentiable with respect to x. □

Example 4.2.6 (Bounded Real Lemma). Another application of the the Kalman-Yakubovich-Popov lemma, is the so-called *Bounded Real Lemma (BRL)*. The standard BRL is in terms of Riccati equations. Here we present (without proof) an inequality version. Let $\dot{x} = Ax + Bu,\ y = Cx + Du$ be an initially at rest, stable system and define γ^* as

$$\gamma^* \triangleq \sup_{u \in L_2,\, t > 0} \frac{\int_0^t \|y(\tau)\|_2^2 \, d\tau}{\int_0^t \|u(\tau)\|_2^2 \, d\tau}.$$

The following holds.

1. $$\gamma^* = \sup_{\omega \in \mathbb{R}} \|G(j\omega)\|$$

   where $\|G(j\omega)\|$ denotes the largest singular value of $G(j\omega)$, and $G(s) \triangleq C(sI - A)^{-1}B + D$.

2. γ^* equals the solution of the SDP

$$\begin{aligned} &\underset{P \in \mathbb{R}^{n \times n},\, \gamma \in \mathbb{R}}{\text{minimize}} && \gamma \\ &\text{subject to} && \begin{bmatrix} A^{\mathrm{T}}P + PA & PB & C^{\mathrm{T}} \\ B^{\mathrm{T}}P & -\gamma I & D^{\mathrm{T}} \\ C & D & -\gamma I \end{bmatrix} < 0, \quad P = P^{\mathrm{T}} > 0. \end{aligned}$$

The expression $\sup_{\omega \in \mathbb{R}} \|G(j\omega)\|$ for stable rational matrices G is the ubiquitous H_∞-norm. □

Example 4.2.7 (H_∞ control). In H_∞ controller synthesis we need a solution $X > 0$ of the Riccati *in*equality

$$A^{\mathrm{T}}X + XA + X(\frac{1}{\gamma^2}B_1 B_1^{\mathrm{T}} - B_2 B_2^{\mathrm{T}})X + C_1^{\mathrm{T}}C_1 < 0. \tag{4.9}$$

This expression is quadratic in the variable X and thus not directly convex in X. We can try to use the Schur complement formula to transform it into an LMI. If $\frac{1}{\gamma^2}B_1B_1^{\mathrm{T}} - B_2B_2^{\mathrm{T}} \geq 0$ we have that (4.9) holds iff the LMI below is satisfied.

$$\begin{bmatrix} -A^{\mathrm{T}}X - XA - C_1^{\mathrm{T}}C_1 & X(\frac{1}{\gamma^2}B_1B_1^{\mathrm{T}} - B_2B_2^{\mathrm{T}})^{1/2} \\ (\frac{1}{\gamma^2}B_1B_1^{\mathrm{T}} - B_2B_2^{\mathrm{T}})^{1/2}X & I \end{bmatrix} > 0.$$

Unfortunately $\frac{1}{\gamma^2}B_1B_1^{\mathrm{T}} - B_2B_2^{\mathrm{T}}$ is generally not ≥ 0, so the square root need not exist.

Second attempt: Define $Y := X^{-1}$ and note that $X > 0$ iff $Y > 0$. Multiplying (4.9) from the left and right by Y gives the equivalent inequality

$$YA^{\mathrm{T}} + AY + (\frac{1}{\gamma^2}B_1B_1^{\mathrm{T}} - B_2B_2^{\mathrm{T}}) + YC_1^{\mathrm{T}}C_1Y < 0.$$

Using a Schur complement argument we, the above inequality is satisfied iff the LMI below holds.

$$\begin{bmatrix} -YA^{\mathrm{T}} - AY - (\frac{1}{\gamma^2}B_1B_1^{\mathrm{T}} - B_2B_2^{\mathrm{T}}) & YC_1^{\mathrm{T}} \\ C_1Y & I \end{bmatrix} > 0.$$

Summarizing: There is an $X > 0$ such that (4.9) holds iff there is a Y such that

$$\begin{bmatrix} -YA^{\mathrm{T}} - AY - (\frac{1}{\gamma^2}B_1B_1^{\mathrm{T}} - B_2B_2^{\mathrm{T}} & YC_1^{\mathrm{T}} \\ C_1Y & I \end{bmatrix}, \quad Y > 0,$$

which is a feasibility problem of an LMI and once Y is found X follows as $X = Y^{-1}$. □

4.3 Properties of SDPs

Recall that the semidefinite program is defined as the minimization problem of the form

$$\begin{array}{ll} \text{minimize} & c^{\mathrm{T}}x \\ \text{subject to} & F(x) \geq 0, \end{array} \tag{4.10}$$

where $F(x)$ is some LMI in x. In this section we look at equivalent formulation called the conic formulation, and we have look at duality for SDPs in conic form. In a way these are preparatory results for the IPM of the next section, but while the IPM is mainly technical, the conic formulation and its dual are very elegant. This section is largely a copy of (Vandenberghe & Boyd 1995), which in turn is influenced by (Nesterov & Nemirovsky 1993) and (Alizadeh 1992).

4.3.1 The conic formulation

In this section we introduce the conic formulation. In this somewhat more abstract formulation the duality results and algorithms based on it are more transparent than with formulations we used so far.

In what follows S denotes the subspace of symmetric matrices in $\mathbb{R}^{n\times n}$, L denotes a subspace of S, and $\mathbb{P}$ is the set of nonnegative definite matrices in S. The matrices C, D, X are elements of S and $\langle\cdot,\cdot\rangle$ is an inner product on S.

Definition 4.3.1. The optimization problem

$$\min_X \langle C, X\rangle \quad \text{subject to } X \in \mathbb{P} \cap (L + D) \tag{4.11}$$

is called an SDP in *conic form*. □

Any SDP can be written in conic form. For example the SDP (4.10) can be written in conic form as follows. Let

$$L = F(\mathbb{R}^n) - F_0; \quad D := F_0; \quad X := F(x).$$

If $F(x)$ is injective we can find an affine left inverse G of F. As a result the objective function $c^{\mathrm{T}}x$ is then also a function of X:

$$c^{\mathrm{T}}x = c^{\mathrm{T}}GX,$$

and as $c^{\mathrm{T}}G$ is an affine linear functional we can find a $C \in \mathbb{R}^{n\times n}$ such that

$$c^{\mathrm{T}}x = \langle C, X\rangle + a.$$

The term a is constant, hence, it does not contribute to the minimization problem and can as well be left out. The SDP (4.10) is thus essentially the same as (4.11). There is one minor assumption that we used: We required F to be injective. If $F(x)$ is not injective then we have a somewhat degenerate SDP that can be simplified. Namely, if $F(x)$ is not injective, then the feasible set $\{x : F(x) \geq 0\}$ contains a line, and hence $c^{\mathrm{T}}x$ is then either unbounded from below on the feasible set (i.e., a degenerate SDP) or c is perpendicular to this line in which case this linear dependency can be removed from the description (4.10) leaving an equivalent SDP in fewer variables.

Example 4.3.2. The usual inner product on $\mathbb{R}^{n\times n}$ with respect to some basis is $\langle C, X\rangle = \operatorname{tr} C^{\mathrm{T}}X$ where tr is the *trace*, the sum of the diagonal entries. It will now be clear that the following is an SDP, although it is not yet in any of the standard forms.

$$\min_{X\in\mathbb{R}^{n\times n}} \operatorname{tr} C^{\mathrm{T}}X, \quad \text{subject to } XA^{\mathrm{T}} + AX \leq -I_n,$$

□

4.3.2 The dual SDP

The dual of the primal SDP

$$\min_X \langle C, X\rangle \quad \text{subject to } X \in \mathbb{P} \cap (\mathcal{L} + D)$$

is

$$\min_Z \langle Z, D\rangle \quad \text{subject to } Z \in \mathbb{P} \cap (\mathcal{L}^\perp + C).$$

Before we can prove that this is indeed the dual we have to introduce an assumption.

Assumption 4.3.3. We assume that primal and dual are *strictly feasible*, that is, that there are $X \in \mathbb{P} \cap (\mathcal{L} + D)$ and $Z \in \mathbb{P} \cap (\mathcal{L}^\perp + C)$ such that $X > 0$ and $Z > 0$. □

Strictly feasible thus implies that the dimension of the primal and dual feasible sets equal that of $\mathcal{L}$ and $\mathcal{L}^\perp$ respectively. A consequence is that the primal and dual feasible sets have nonempty relative interior[2]. These are quite natural assumptions and only very specific SDPs are not strictly feasible. Under this assumption the distinction between $F(x) > 0$ and $F(x) \geq 0$ is essentially irrelevant, the difference is a "thin" set.

Lemma 4.3.4. *Suppose the primal and dual SDP are both strictly feasible and define the duality gap as*

$$gap(X, Z) := \langle Z, X\rangle.$$

The following holds.

1. *The duality gap satisfies*

$$gap(X, Z) := \langle Z, X\rangle = \langle C, X\rangle + \langle Z, D\rangle - \langle C, D\rangle, \tag{4.12}$$

   and the duality gap is nonnegative for feasible (X, Z).

2. *The primal and dual objective functions are bounded from below on their respective feasible sets, and optimal X^* and Z^* exist.*

3. *Optimal X^* and Z^* satisfy*

$$\langle C, X^*\rangle + \langle Z^*, D\rangle - \langle C, D\rangle = 0.$$

   In other words, (X^*, Z^*) is optimal iff $gap(X^*, Z^*) = 0$.

4. *For any feasible (X, Z) there holds*

$$\langle C, X\rangle - gap(X, Z) \leq \langle C, X^*\rangle \leq \langle C, X\rangle,$$

$$\langle Z, D\rangle - gap(X, Z) \leq \langle Z^*, D\rangle \leq \langle Z, X\rangle.$$

[2]nonempty interior relative to $\mathcal{L} + D$ and $\mathcal{L}^\perp + C$ respectively.

Proof. The inner product is $\langle X, Z\rangle := \operatorname{tr} XZ$.

1. By feasibility we have that $X \in (\mathcal{L} + D)$, $Z \in (\mathcal{L}^{\perp} + C)$ which yields (4.12).

   Further by feasibility X and Z are both nonnegative definite. Therefore $\operatorname{gap}(X, Z) := \langle X, Z\rangle := \operatorname{tr} XZ = \operatorname{tr} X^{1/2} Z X^{1/2} = \sum \lambda_k(X^{1/2} Z X^{1/2}) \geq 0$.

2. By assumption there is a feasible Z. Then boundedness from below of $\langle C, X\rangle$ follows from (4.12) and the fact that $\operatorname{gap}(X, Z) \geq 0$. The dual $\langle Z, D\rangle$ is bounded from below for the same reasons.

   By strict feasibility there is a positive definite feasible Z. Take one such $Z > 0$. Let $\{X_k\}$ be a sequence of feasible elements such that $\langle C, X_k\rangle$ converges to the optimal cost. Then from (4.12) it follows that $\langle X_k, Z\rangle$ is bounded from above, and as $Z > 0$, this means that X_k remains bounded so it has a converging subsequence whose limit is optimal by construction. Similarly Z^* exists.

3. Let $\mathcal{S}$ denote the set of symmetric matrices. Take an arbitrary $\alpha \in \mathbb{R}$. Then

$$\begin{aligned} \alpha \leq \langle C, X^*\rangle \quad &\Leftrightarrow \quad \nexists X \in \overset{\circ}{\mathbb{P}} \cap (\mathcal{L} + D),\ \langle C, X\rangle < \alpha \\ &\Leftrightarrow \quad \{X \oplus \langle C, X\rangle \,:\, X \in \overset{\circ}{\mathbb{P}}\} \cap \{(\mathcal{L} + D) \oplus (-\infty, \alpha)\} = \emptyset \end{aligned}$$

   This is an intersection of two convex sets of which the first is a cone and has a non-empty interior (assuming nondegeneracy $C \neq 0$). By a separating hyper-plane argument—in fact a variation of the geometric Hahn-Banach theorem (Luenberger 1969, page 133, Thm.3)—the intersection is empty iff there is a separating hyper-plane, that is, iff a functional $\langle Y \oplus y, \cdot\rangle$ exists that is non-positive on one convex set and positive on the other:

$$\begin{aligned} \alpha \leq \langle C, X^*\rangle \quad &\Leftrightarrow \quad \exists\, Y \oplus y \in \mathcal{S} \oplus \mathbb{R} \text{ such that} \\ &\qquad \langle Y \oplus y, \{X \oplus \langle C, X\rangle \,:\, X \in \overset{\circ}{\mathbb{P}}\}\rangle > 0 \\ &\qquad \text{and } \ \langle Y \oplus y, \{(\mathcal{L} + D) \oplus (-\infty, \alpha)\}\ \rangle \leq 0 \\ &\Leftrightarrow \quad \exists Y \in \mathcal{S}, y \in \mathbb{R} \text{ such that } \langle Y, X\rangle + y\langle C, X\rangle > 0 \\ &\qquad \forall X \in \overset{\circ}{\mathbb{P}} \ \text{ and } \ \langle Y, \mathcal{L} + D\rangle + y(-\infty, \alpha) \leq 0 \end{aligned} \tag{4.13}$$

   We may scale (Y, y) with a positive constant to achieve $y \in \{-1, 0, 1\}$. The case $y = -1$ is impossible as that contradicts (4.13). If $y = 0$ then (4.13) states that $\langle Y, \overset{\circ}{\mathbb{P}}\rangle > 0$ and $\langle Y, \mathcal{L} + D\rangle \leq 0$ but that contradicts the strict feasibility assumption that $\overset{\circ}{\mathbb{P}} \cap (\mathcal{L} + D) \neq \emptyset$. So only the case that $y = 1$ remains, and we can proceed as follows:

$$\alpha \leq \langle C, X^*\rangle \quad \Leftrightarrow \quad \exists Y \in \mathcal{S} \text{ such that } \ \langle Y + C, \overset{\circ}{\mathbb{P}}\rangle > 0 \tag{4.14}$$

$$\text{and } \langle Y, \mathcal{L} + D\rangle + (-\infty, \alpha) \leq 0. \tag{4.15}$$

Now we are almost there. The condition that $\langle Y + C, \overset{\circ}{\mathbb{P}}\rangle > 0$ is equivalent to that $Y + C \in \mathbb{P}$, $Y + C \neq 0$ (this is easy to verify), and boundedness from above of $\langle Y, L + D\rangle$ in the inequality (4.15) necessarily means that $Y \in L^{\perp}$. Therefore:

$$\begin{aligned}\alpha \leq \langle C, X^*\rangle \quad &\Leftrightarrow \quad \exists Y \in S,\ Y + C \in \mathbb{P}, Y + C \neq 0,\ Y \in L^{\perp},\ \langle Y, D\rangle + \alpha \leq 0\\ &\Leftrightarrow \quad \exists Z := Y + C \in \mathbb{P},\ Z \neq 0,\ Z \in L^{\perp} + C,\ \langle Z - C, D\rangle + \alpha \leq 0\\ &\Leftrightarrow \quad \langle Z^*, D\rangle - \langle C, D\rangle + \alpha \leq 0.\end{aligned}$$

As α is arbitrary this proves that

$$\langle Z^*, D\rangle - \langle C, D\rangle + \langle C, X^*\rangle \leq 0.$$

In Item 2 we showed that $\langle Z, D\rangle - \langle C, D\rangle + \langle C, X\rangle \geq 0$. Hence it must be equal to zero for the optimal (X^*, Z^*). Conversely, if $\langle Z, D\rangle - \langle C, D\rangle + \langle C, X\rangle$ is zero then from the fact that the duality gap is nonnegative it follows that (X, Z) must be optimal.

4. Easy.

∎

4.4 A primal-dual IPM for SDPs

Under assumption 4.3.3 the relative boundary[3] points of the feasible set are those elements of the feasible set that are nonnegative definite but not positive definite. In particular the relative boundary points are singular. So a candidate barrier function on the feasible set could be

$$\phi(X) := \log\det X^{-1}.$$

It has the properties of a barrier that we need:

Lemma 4.4.1. *The barrier function $\phi(X) := \log\det X^{-1}$ is convex on $\mathbb{P} \cap (L + D)$. It is well defined for strictly feasible X and grows to $+\infty$ as X approaches the (relative) boundary of the feasible set. Furthermore ϕ has a second order Taylor expansion on $\mathbb{P}$ given as*

$$\phi(X + \Delta) = \phi(X) - \langle \Delta, X^{-1}\rangle + \langle X^{-1}\Delta X^{-1}, \Delta\rangle + o(\|\Delta\|^2). \tag{4.16}$$

Proof. Convexity follows from the fact that the second order term in the Taylor series is nonnegative:

$$\begin{aligned}\langle X^{-1}\Delta X^{-1}, \Delta\rangle &= \operatorname{tr} X^{-1}\Delta X^{-1}\Delta\\ &= \operatorname{tr} X^{-1/2}\Delta X^{-1}\Delta X^{-1/2}\\ &= \operatorname{tr} X^{-1/2}\Delta X^{-1/2}X^{-1/2}\Delta X^{-1/2}\\ &= \|X^{-1/2}\Delta X^{-1/2}\|_{\mathrm{F}}^2 \geq 0.\end{aligned}$$

[3]Those are the points of the feasible set that are not in the *relative* interior.

The only nontriviality is to show that (4.16) holds. Note that X and $X+\Delta$ both feasible implies that Δ is symmetric.

$$\begin{aligned}\phi(X+\Delta) &= \log\det(X+\Delta)\\ &= \log\det(X(I+X^{-1}\Delta))^{-1}\\ &= \phi(X)+\log\det(I+X^{-1}\Delta)^{-1}\\ &= \phi(X)+\log\det\big(I-X^{-1}\Delta+X^{-1}\Delta X^{-1}\Delta+o(\|\Delta\|^2)\big)\\ &= \phi(X)-\langle\Delta, X^{-1}\rangle+\langle X^{-1}\Delta X^{-1}, \Delta\rangle+o(\|\Delta\|^2).\end{aligned}$$

■

We summarize the main ingredients of an IPM for the combined primal and dual SDP

$$\begin{array}{ll}\text{minimize} & \langle X, Z\rangle\\ \text{subject to} & X\in\mathbb{P}\cap(L+D),\ Z\in\mathbb{P}\cap(L^{\perp}+C).\end{array}$$

As with LPs, the use of a combined primal-dual SDP is that we know the optimal cost is zero. For this SDP we define the primal-dual potential function

$$G(X, Z) := \nu\sqrt{n}\log\langle Z, X\rangle+\phi(X)+\phi(Z)-n\log n.$$

The first term, $\nu\sqrt{n}\log\langle X, Z\rangle$, measures the duality gap: The more negative it is the closer we are to optimal zero duality gap. The second term, $\phi(x)+\phi(Z)-n\log n$, measures the distance to the boundary of the feasible set: The more positive it is, the closer we are to the boundary. The potential function thus exhibits a trade of between the size of the duality gap and distance to the boundary.

Lemma 4.4.2. $\phi(X)+\phi(Z)-n\log n\geq 0$.

Proof. There is an interesting proof in (Vandenberghe & Boyd 1995). ■

This explains why we included a constant term $-n\log n$ in the potential function. A consequence is that

$$\text{gap}(X, Z) := \langle Z, X\rangle \leq e^{\frac{G(X,Z)}{\nu\sqrt{n}}},$$

so to obtain a duality gap of less than ϵ it is sufficient to know that

$$G(X, Z)\leq \nu\sqrt{n}\log\epsilon.$$

The primal-dual IPM is a method to generate strictly feasible iterates (X_k, Z_k) such that G decreases each iteration by some fixed positive amount δ. Then to achieve a gap of at most ϵ we need at most

$$\frac{G(X_0, Z_0)-\nu\sqrt{n}\log\epsilon}{\delta}$$

iterations. Vandenberghe & Boyd (1995) show that such iterates can indeed be found, and that each iteration requires a polynomial number of operations.

Primal-dual IPM by Vandenberghe & Boyd (1995), based on results by Nesterov & Nemirovsky (1993) and Ye (1991)

---

input: Strictly feasible (X, Z).
output: A feasible pair (X, Z) such that the duality gap is at most ϵ.
begin

Take any $\theta \in (0, 0.35)$ and set $\delta = \theta - \log(1+\theta)$.
Set $k = 0$, $\nu = 1$.

repeat

1) *Find suitable search direction*
Let $W = \left[\begin{smallmatrix} X & 0 \\ 0 & Z \end{smallmatrix}\right]$. Compute a $\Delta_W := \left[\begin{smallmatrix} \Delta_X & 0 \\ 0 & \Delta_Z \end{smallmatrix}\right] \in \left[\begin{smallmatrix} L & 0 \\ 0 & L^\perp \end{smallmatrix}\right]$ such that

$$\frac{\langle \Delta_W, \nabla G(W) \rangle}{\|W^{-1/2} \Delta_W W^{-1/2}\|_{\mathrm{F}}} \geq \theta.$$

2) *Plane search*
Find $\lambda_X, \lambda_Z \in \mathbb{R}^+$ such that $X - \lambda_X \Delta_X \in \mathbb{P}$, $Z - \lambda_Z \Delta_Z \in \mathbb{P}$ and

$$G(X - \lambda_X \Delta_X, Z - \lambda_Z \Delta_Z) \;\leq\; G(X, Z) - \delta \tag{4.17}$$

.

3) *Update*
Set $X := X - \lambda_X \Delta_X$ and $Z := Z - \lambda_Z \Delta_Z$.

until $\langle Z, X \rangle \leq \epsilon$.
end

A few words on this algorithm: The problem of maximizing the expression

$$\frac{\langle \Delta_W, \nabla G(W) \rangle}{\|W^{-1/2} \Delta_W W^{-1/2}\|_{\mathrm{F}}}$$

is as a least squares problem, and its maximum can be shown to exceed θ which is all we need in Step 1. The conditions in Step 2 can also be met. In fact for

$$\lambda_X := \lambda_Z := \frac{\theta}{(1+\theta)\|W^{-1/2} \Delta_W W^{-1/2}\|_{\mathrm{F}}}$$

it can be shown that the update is strictly feasible and that it satisfies (4.17). In principle then there is no need to include an approximate plane search, but including one dramatically speeds up the algorithm.

4.5 Back to the original SDP

What can be derived for SDPs in conic form can also be derived for SPDs in the original form

$$\begin{array}{ll} \text{minimize} & c^{\mathrm{T}}x \\ \text{subject to} & F_0 + \sum_{i=1}^{n} x_i F_i \geq 0. \end{array} \tag{4.18}$$

In particular it can be shown that:

1. x is strictly feasible if $F(x) := F_0 + \sum x_k F_k > 0$.
2. The dual of (4.18) is

$$\left\{ \begin{array}{ll} \text{maximize}_Z & -\operatorname{tr} F_0 Z \\ \text{subject to} & Z = Z^{\mathrm{T}} \geq 0, \quad \operatorname{tr} F_i Z = c_i, \ (i = 1, \ldots, n) \end{array} \right. \tag{4.19}$$

3. Z is strictly feasible in the dual if it is feasible and $Z > 0$.

4.6 Phase I

So far we have assumed that we can always find an strict feasible point. For specific cases finding an initial strict feasible point may be straightforward. For example, a feasible point (t, x) of the SDP

$$\min_{t \in \mathbb{R},\, x \in \mathbb{R}^n} t \quad \text{subject to } tI_m - (A_0 + \sum x_i A_i) \geq 0.$$

is $(x = 0, t = 2\|A_0\|)$. For general SDPs there are some tricks. Vandenberghe & Boyd (1996) review a “big-M” procedure. There are three versions:

1. We have a strict feasible $x^{(0)}$ but no strict feasible $Z^{(0)}$ for the dual.
2. We have a strict feasible $Z^{(0)}$ for the dual but no strict feasible $x^{(0)}$ for the primal.
3. We have no strict feasible $x^{(0)}$ and no strict feasible $Z^{(0)}$ for the dual.

We will not go into much detail. Consider Item 1, that is, suppose we have an initial strict feasible point $x^{(0)}$ for the SDP

$$\left\{ \begin{array}{ll} \text{minimize}_x & c^{\mathrm{T}}x \\ \text{subject to} & F(x) \geq 0, \end{array} \right.$$

To obtain a strict feasible Z (some variation of it actually) we consider a modified SDP

$$\left\{ \begin{array}{ll} \text{minimize}_x & c^{\mathrm{T}}x \\ \text{subject to} & F(x) \geq 0, \quad \operatorname{tr} F(x) \leq M. \end{array} \right.$$

For M big enough, $M > \operatorname{tr} F(x^{(0)})$, this SDP will be equivalent to the original because the extra constraint $\operatorname{tr} F(x) \leq M$ will be inactive. The dual of this modified SDP is easily seen to be

$$\left\{ \begin{array}{rl} \text{maximize}_Z & -\operatorname{tr} F_0(Z - zI) - Mz \\ \text{subject to} & Z = Z^{\mathrm{T}} \geq 0,\ z \geq 0,\ \operatorname{tr} F_i(Z - zI) = c_i,\ (i = 1, \ldots, n). \end{array} \right. \tag{4.20}$$

For this dual we can compute a strict feasible (Z, z). Let U be any matrix of the same dimension as Z such that $\operatorname{tr} F_i U = c_i$ for all $i = \{1, \ldots, m\}$. This is a linear equation in U and can hence be solved by standard techniques. Take any $Z^{(0)}$ and $z^{(0)}$ such that

$$\frac{1}{2}(U + U^{\mathrm{T}}) = Z^{(0)} - z^{(0)} I, \quad Z^{(0)} > 0,\ z^{(0)} > 0.$$

Such $Z^{(0)}$ and $z^{(0)}$ obviously exist, and they are strict feasible points for the dual (4.20). Note that we do not derive a strict feasible Z for the dual of the original SDP, instead we derive one for the dual of the modified SDP. The modified is equivalent to the original, so this is allowed.

For the other two cases something similar can be done (Vandenberghe & Boyd 1996).

4.7 Notes and references

The text of this chapter based mostly on (Vandenberghe & Boyd 1995) and a survey paper by Vandenberghe & Boyd (1996).

The book by Nesterov & Nemirovsky (1993) contains a detailed treatment of SDP and more general convex optimization.

Semidefinite programming became popular in the western control community under the name *eigenvalue problem* (EVP). The book by Boyd, Ghaoui, Feron & Balakrishnan (1994) is large responsible for this. A variation of SDPs which is advocated in that book is what is called the *generalized eigenvalue problem (GEVP)*. The GEVP is to find an $x \in \mathbb{R}^m$ and real number γ that solve the minimization problem

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \gamma B(x) - A(x) > 0,\ B(x) > 0,\ C(x) > 0. \end{array}$$

Here $A(x)$, $B(x)$ and $C(x)$ are symmetric matrices that are affine in x (i.e, that have the same form as $F(x)$ in (4.2)). This is a *quasi* convex problem and can be solved in polynomial time (Boyd et al. 1994).

Software is available, such as the Matlab toolbox LMI-LAB by Gahinet & Nemirovsky (1993), and there is also some Matlab software available via anonymous ftp[4].

[4] *Software for semidefinite programming* by L. Vandenberghe and S. Boyd. (see `isl.stanford.edu/pub/boyd/`)

Exercise 4.7.1. Consider LMI $F(x) \geq 0$. Find an SDP, with easily computable strict feasible point, whose solution is less than or equal to zero iff the LMI is feasible.

(The point of this exercise is that since SDPs can be solved in polynomial time also feasibility can be checked in polynomial time. We did not mention this in the notes.) □

Exercise 4.7.2.

1. Write $\min_x \|A(x)\|$ as an SDP of the form (7.3). (Here $\|A\|$ denotes the spectral norm, and $A(x)$ is assumed affine in $x \in \mathbb{R}^n$.)

2. Derive the dual SDP of this SDP.

3. Show that strictly feasible x and Z exist. (Hence from Lemma 7.3.3 we know that optimal solutions exist with zero duality gap. Good.)

4. Write Matlab code that will solve this SDP using `sdpye.m`. (This macro is listed in the appendix.) Note that `sdpye.m` assumes you have strictly feasible $(x^{(0)}, Z^{(0)})$, this you have to generate.

   The macro `reshape.m` may be of use.

5. Try your macro on

$$A(x) = \begin{bmatrix} 2-x & x \\ 0 & 1 \end{bmatrix}.$$

□

A

Appendix

This appendix contains some standard definitions and results that we use in the notes but did not bother to explain. It is in telegram style.

A.1 Subspaces, hyper-planes, convex sets and cones

A *subspace* of $\mathbb{R}^n$ is a subset of $\mathbb{R}^n$ that is closed under addition and scalar multiplication. Any subspace in $\mathbb{R}^n$ can be expressed as $\{Cy : y \in \mathbb{R}^m\}$ for some $C \in \mathbb{R}^{n\times m}$. A set $\Omega \subset \mathbb{R}^n$ is *affine* if $\Omega - d_0$ is a subspace for some $d_0 \in \mathbb{R}^n$. An affine set (or affine subspace as it is often called) is so to say a shifted, or translated subspace.

An affine set in $\mathbb{R}^n$ of dimension $n-1$ is called a *hyper-plane* of $\mathbb{R}^n$. In other words, a set Ω is a hyper-plane if-and-only if for some $d \in \mathbb{R}$ and $a \in \mathbb{R}^n$ we have $\Omega = \{x \in \mathbb{R}^n : a^{\mathrm{T}}x = d\}$.

The set of points on one side of a hyper-plane is called a *half-space*. *Closed half-spaces* include the hyper-plane and *open half-spaces* do not include the hyper-plane. In other words, Ω is a closed half-space of $\mathbb{R}^n$ if-and-only-if for some $d \in \mathbb{R}$ and $a \in \mathbb{R}^n$ we have $\Omega = \{x \in \mathbb{R}^n : a^{\mathrm{T}}x \geq d\}$.

A subset of $\mathbb{R}^n$ is a *convex polyhedron* if it is the intersection of finitely many closed half-spaces of $\mathbb{R}^n$.

An element x of a convex polyhedron is a *vertex* or *extreme point* of the polyhedron if it can not be written as the average of two other elements from the polyhedron.

A subset Ω of a vector space is *convex* if for any two elements x, y in Ω, we have that

$$\lambda x + (1-\lambda)y \in \Omega$$

for all $\lambda \in [0,1]$. Ω is a *cone* if $x \in \Omega$ implies $\lambda x \in \Omega$ for all $\lambda > 0$. A cone Ω is called *pointed* if $\Omega \cap (-\Omega) = 0$ or $\emptyset$. Subspaces are cones, but they are not pointed. The ice cream cone is pointed. A typical pointed, convex cone is the set of positive definite matrices $\{P \in \mathbb{R}^{n\times n} : P = P^{\mathrm{T}} > 0\}$.

A.2 Matrix formulae

This section contains a collection of some basic definitions, properties and results about constant real-valued matrices.

The *transpose* of a matrix $A \in \mathbb{R}^{m\times n}$ is denoted as A^{T}. A is called *symmetric* if $A = A^{\mathrm{T}}$. A matrix U is *unitary* if it is square, invertible and $U^{-1} = U^{\mathrm{T}}$. A symmetric matrix A is said to be *nonnegative definite* (denoted $A \geq 0$) if for all $x \in \mathbb{R}^n$ we have $x^{\mathrm{T}}Ax \geq 0$. A symmetric[1] A is called *positive definite* (denoted $A > 0$) if $x^{\mathrm{T}}Ax > 0$ for all *nonzero* $x \in \mathbb{R}^n$.

(a) *Eigenvalue decomposition of symmetric matrices:* If $A = A^{\mathrm{T}}$ then $A = VDV^{\mathrm{T}}$ for some unitary V and diagonal, real-valued D. In this case the columns of V are eigenvectors of A and the diagonal entries of D are the eigenvalues of A.

(b) A symmetric A is nonnegative definite iff all eigenvalues $\lambda_i(A)$ are nonnegative.

(c) A symmetric A is positive definite iff all eigenvalues $\lambda_i(A)$ are strictly positive.

(d) If W is invertible then $A > 0$ iff $W^{\mathrm{T}}AW > 0$, and $A \geq 0$ iff $W^{\mathrm{T}}AW \geq 0$.

(e) The trace, $\operatorname{tr} A$, of a square matrix A is defined as the sum of the diagonal entries of A. We have that $\operatorname{tr} A = \sum \lambda_i(A)$.

(f) If A and B are square and have the same dimension then $\det AB = \det A \det B$.

(g) *Caley Hamilton:* For every square A and with characteristic polynomial χ_A defined as $\chi_A(\lambda) := \det(\lambda I - A)$ we have that $\chi_A(A) = 0$.

(h) The transformation $A \to TAT^{-1}$ is called a *similarity transformation*. It leaves the eigenvalues invariant, that is, $\chi_A = \chi_{TAT^{-1}}$.

An implication of Item (a) is that $A \geq 0$ iff $A = R^{\mathrm{T}}R$ for some R. Given $A \geq 0$ many matrices R can be found such that $A = R^{\mathrm{T}}R$. $A = R^{\mathrm{T}}R$ is a *Cholesky factorization* if R is square upper triangular with nonnegative diagonal elements, and such a factorization exists for every $A \geq 0$. Cholesky factors are unique and they are easy to compute if $A > 0$. Theoretically of interest is the fact that any $A \geq 0$ can be written as

$$A = R^2, \quad R = R^{\mathrm{T}} \geq 0,$$

so with the factor R itself some nonnegative definite matrix. Such a factor $R = R^{\mathrm{T}} \geq 0$ is called the *square root* of A. The square root is denoted as $A^{1/2}$, it is unique (see (Golub & Loan 1983)) and can be constructed in the following manner from the eigenvalue decomposition of A. Suppose

$$A = VDV^{\mathrm{T}}, \quad V \text{ unitary, } D \text{ diagonal,}$$

[1]Positive definiteness and nonnegative definiteness is only defined for real symmetric (and Hermitian) matrices.

is an eigenvalue decomposition of A. All diagonal entries of D are nonnegative if $A \geq 0$, which means that D has a diagonal square root $D^{1/2}$. Since $V^{\mathrm{T}}V = I$ we can take $A^{1/2}$ as defined below.

$$A = \underbrace{VD^{1/2}V^{\mathrm{T}}}_{A^{1/2}} \cdot \underbrace{VD^{1/2}V^{\mathrm{T}}}_{A^{1/2}}.$$

Next we prove a useful formula. Suppose that A and B^{T} have the same dimensions $m \times n$. We form the square $(n+m) \times (n+m)$ matrix M depending on scalar λ

$$M := \begin{bmatrix} \lambda I_n & B \\ A & I_m \end{bmatrix}.$$

The determinant of M can be found by examining

$$\underbrace{\begin{bmatrix} \lambda I_n & B \\ A & I_m \end{bmatrix}}_{M} \begin{bmatrix} I_n & -\frac{1}{\lambda}B \\ 0 & I_m \end{bmatrix} = \begin{bmatrix} \lambda I_n & 0 \\ A & I_m - \frac{1}{\lambda}AB \end{bmatrix}.$$

This reveals that $\det M = \lambda^{n-m} \det(\lambda I_m - AB)$. On the other hand we also have

$$\underbrace{\begin{bmatrix} \lambda I_n & B \\ A & I_m \end{bmatrix}}_{M} \begin{bmatrix} I_n & 0 \\ -A & I_m \end{bmatrix} = \begin{bmatrix} \lambda I_n - BA & B \\ 0 & I_m \end{bmatrix},$$

so that $\det M = \det(\lambda I_n - BA)$. Combining the two formulae we see that

Lemma A.2.1. *Let A and B^{T} be matrices of dimension $m \times n$. Then*

1. $\det(\lambda I_n - BA) = \lambda^{n-m} \det(\lambda I_m - AB)$.

2. $\operatorname{tr} AB = \operatorname{tr} BA$.

3. $\det(I_m - AB) = \det(I_n - BA)$.

Proof. Item 3 is a special case or Item 1. Item 1 is proved above. It shows in particular that the nonzero eigenvalues of AB and BA are the same even if AB and BA have different dimension. This implies Item 2. ■

We end this section with one of the most useful matrix formula for optimization (and control and identification theory as well). The formula is the so-called *Sherman-Morrison-Woodbury* formula:

$$(A + UV^{\mathrm{T}})^{-1} = A^{-1} - A^{-1}U(I + V^{\mathrm{T}}A^{-1}U)^{-1}V^{\mathrm{T}}A^{-1}.$$

It is sometimes referred to as the *matrix inversion lemma*. It is readily confirmed by multiplying both sides with $(A + UV^{\mathrm{T}})$. For the rank-one case (the case that U and V are column vectors) we get a simplified

$$(A + uv^{\mathrm{T}})^{-1} = A^{-1} - A^{-1}\frac{uv^{\mathrm{T}}}{1 + v^{\mathrm{T}}A^{-1}u}A^{-1}.$$

This is the version that is most often used.

A.3 Euclidean, Frobenius and spectral norm

Definition A.3.1.

1. Let $1 \leq p \leq \infty$. The *p-norm* of a vector $a \in \mathbb{R}^n$ is defined as

$$\|a\|_p = \begin{cases} \sqrt[p]{\sum_{i=1}^{n} |a_i|^p} & \text{if } 1 \leq p < \infty \\ \max_i |a_i| & \text{if } p = \infty \end{cases}$$

   The 2-norm is the familiar *Euclidean norm*.

2. The *Frobenius norm*, $\|A\|_{\mathrm{F}}$, of a matrix $A \in \mathbb{R}^{m\times n}$ is defined as $\|A\|_{\mathrm{F}} = \sqrt{\operatorname{tr} A^{\mathrm{T}} A}$.

3. The *spectral norm*, $\|A\|$, of a matrix is defined as the largest singular value of A, i.e., $\|A\| = \sqrt{\lambda_{\max}(A^{\mathrm{T}} A)}$.

□

The Frobenius norm $\|A\|_{\mathrm{F}}$ equals the square root of $\sum a_{ij}^2$.

A.3.1 Least squares in $\mathbb{R}^n$

Lemma A.3.2. *If $A \in \mathbb{R}^{m\times n}$ has full column rank then $A^{\mathrm{T}} A$ is nonsingular.*

Proof. If not, then $A^{\mathrm{T}} Ax = 0$ for some $0 \neq x \in \mathbb{R}^n$. But then also $\|Ax\|_2^2 = x^{\mathrm{T}}(A^{\mathrm{T}} Ax) = 0$, which implies $Ax = 0$. That is impossible because A has full column rank. ■

Lemma A.3.3 (Least squares). *Given $A \in \mathbb{R}^{m\times n}$ of full column rank, and a vector $b \in \mathbb{R}^m$, there is a unique x that minimizes*

$$\|Ax - b\|_2,$$

and it is given by

$$x = (A^{\mathrm{T}} A)^{-1} A^{\mathrm{T}} b. \tag{A.1}$$

If instead A has full row rank there are many solutions x of $Ax = b$, but there is a unique x that solves

$$\min \|x\|_2 \quad \textit{subject to } Ax = b$$

and it is given by

$$x = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1} b. \tag{A.2}$$

Proof. (A.1) follows directly by solving the linear equation $\nabla\|Ax - b\|_2^2 = 0$. To prove (A.2), define y as $y = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}b - x$. The condition $Ax = b$ implies that $Ay = 0$, but then $\|x\|_2^2 = (A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}b - y)^{\mathrm{T}}(A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}b - y) = \|A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}b\|_2^2 + \|y\|_2^2$, and so $\|x\|_2$ is the smallest for $y = 0$. ■

A.4 Matlab macros

A.4.1 LPs

```
function [xb,sb,Ap]=ye(A,b,c,xb,sb,yb)
%
%  [xb,sb]=ye(A,b,c,xb,sb)
%
%  Ye’s 1991 interior point algorithm for linear programming
%
%  Finds an xb in { x : Ax=b, x>=0 } such that
%  c’*xb < 1e-7+ min{ c’*x : Ax=b, x>=0 }.
%
%  No error handlings are included.
%
mn=size(A);
m=mn(1);
n=mn(2);
epsy=1e-7;
E=eye(n);
q=n+sqrt(n);
tel=0;
while ((sb’*xb)>epsy) & (tel<500)
  T=diag(xb);
  sp=T*sb;
  Ap=A*T;
  xp=T\xb;
  e=xp;
  g=q/(sp’*xp)*sp-e;
  d=(E-Ap’/(Ap*Ap’)*Ap)*g;
  dnorm=sqrt(d’*d);
  if (dnorm <0.4)
    sp=(sp’*e)/q*(d+e);
  else
    xp=e-(1/4/dnorm)*d;
  end
  sb=T\sp;
  xb=T*xp;
  tel=tel+1;
end
```

This is Ye’s IPM algorithm for LPs as discussed in Chapter 3. For example:

```
A=[1 2 1 0; 1 0 0 1];
b=[4; 2]’;
c=[-1 -1 0 0]’;
s=[1 1 1 1]’;
x=[1 1 1 1]’;
[xx,ss]=ye(A,b,c,x,s)
```

(That $s = \begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix}^{\mathrm{T}}$ is feasible is easy to verify.) The output is

```
xx =

    2.0000
    1.0000
    0.0000
    0.0000


ss =

    0.0000
    0.0000
    0.5000
    0.5000
```

A.4.2 SDPs

Lieven Vandenberghe emailed a Matlab macro for SDP. The input of the LMI $F_0 + \sum_{k=1}^{n} x_k F_n$ is a bit awkward and may require the use of the Matlab function `reshape`.

```
function [x,Z,ul] = sdp_ye(F,c,x0,Z0,nu,tol,maxiters)
%
%
% [x,Z,ul] = sdp_ye(F,c,x0,Z0,nu,tol,maxiters);
%
% Ye's potential reduction algorithm with plane search.
%
% Solves semidefinite program
%
%    minimize    c'*x
%    subject to  F_0 + x_1*F_1 + ... + x_m*F_m  >= 0
%
% and its dual
%
%    maximize    -Tr F_0*Z
%    subject to  Tr F_i*Z = c_i, i=1,...,m
%                Z >= 0
%
% Inputs arguments:
% - F:         matrix with m+1 columns
%              F = [ F0(:)  F_1(:) ... F_m(:) ],
%              F1, ..., Fm must be linearly independent.
% - c:         m-vector, specifies primal objective.
% - x0:        m-vector, strictly primal feasible.
% - Z0:        Z0(:). must be strictly dual feasible.
% (the code does not check feasibility of x0 and Z0)
% - nu:        nu >= 1.0.  Controls rate of convergence.
```

```
% - tol:       quit if duality gap is less than tol
% - maxiters:  maximum number of iterations
%
% Output arguments:
% - x,Z:       last primal and dual iterate
% - ul:        ul(1) is c'*x;  ul(2) is -Tr F_0*Z

n = sqrt(size(F,1));  m=length(c);
if (size(c,2) > size(c,1)), c = c';  end;

x = x0;  Z = Z0;  q = n+nu*sqrt(n);
X = F*[1;x];     % X=F0+x1*F1+...+xm*Fm
dg = X'*Z;       % duality gap is Tr X*Z
ul(1) = c'*x;  ul(2) = -F(:,1)'*Z;   % primal and dual objective

FF=zeros(n*n,m);      % to store scaled F_i's
disp(['        Primal obj       Dual obj       Duality gap']);

for iters=1:maxiters,

  disp([sprintf('% 19.5e',ul(1)), sprintf('% 17.5e', ul(2)), ...
        sprintf('% 16.5e',dg)]);

  [V,sig] = eig(reshape(X,n,n));    % X=V*sig*V'

  % least-squares problem.
  %   dx = argmin_v \| rhs - sum_{i=1}^m Fsc_i*v_i \|_F
  % with
  % - Fsc_i = sig^{-1/2}*V'*F_i*V*sig^{-1/2},
  % - rhs = I - (q/dg)*sig^{1/2}*V'*Z*V*sig^{1/2}

  sqrtsig = sqrt(diag(sig));  invsqrtsig = ones(n,1)./sqrtsig;
  for i=1:m       % Fsc_i = sig^{-1/2}*V'*F_i*V*sig^{-1/2}
     FF(:,i) = reshape( invsqrtsig(:,ones(1,n))' .*  ...
        ( V'*reshape(F(:,i+1),n,n)*V ) .* invsqrtsig(:,ones(1,n)), ...
        n*n, 1);
  end;
  dx = FF \ reshape( eye(n) - (q/dg)* sqrtsig(:,ones(1,n))' ...
       .* (V'*reshape(Z,n,n)*V) .*sqrtsig(:,ones(1,n)), n*n, 1);
  dX = F(:,2:m+1)*dx;

  % dual direction dZ = (dg/q) * (X^{-1} - X^{-1}*dX*X^{-1}) - Z;
  Xinv = V*diag(ones(n,1)./diag(sig))*V';   % inverse of X
  dZ = (dg/q) * reshape(Xinv - Xinv*reshape(dX,n,n)*Xinv, n*n, 1) ...
       - Z;     % dual search direction

  % plane search: minimize
  %  phi(alpha) = q*log(dg + alphax*ZdX + alphaz*XdZ)
  %   - sum_i(log(1 + alphax*sigx_i)) - sum_i(log(1+alphaz*sigz_i))
  sigxz = real([ eig(reshape(dX,n,n), reshape(X,n,n)), ...
                 eig(reshape(dZ,n,n), reshape(Z,n,n)) ]);
  ddg=[c'*dx, F(:,1)'*dZ];
  lam = 1;  alpha=zeros(1,2);
```

```
  while (lam > 1e-4) & (dg > tol)
     grad = (q/dg)*ddg - ...
            sum(sigxz./(ones(n,2)+alpha(ones(n,1),:).*sigxz));
     hess = sum((sigxz./(ones(n,2)+alpha(ones(n,1),:).*sigxz)).^2);
     lam = sqrt(sum((grad.^2)./hess));  dalpha = -(grad./hess)/(1+lam);
     alpha = alpha+dalpha;  dg = dg+dalpha*ddg';
  end

  x=x+alpha(1)*dx;  X=X+alpha(1)*dX;  Z=Z+alpha(2)*dZ;
  ul(1) = c'*x;  ul(2) = -F(:,1)'*Z;

  if (dg < tol)
     disp([sprintf('% 19.5e',ul(1)), sprintf('% 17.5e', ul(2)), ...
           sprintf('% 16.5e',dg)]);
     return;
  end;

end;

disp([sprintf('% 19.5e', ul(1)), sprintf('% 17.5e', ul(2)), ...
      sprintf('% 16.5e',dg)]);
disp('max. number of iterations exceeded');
```

Bibliography

Alizadeh, F. (1992), Optimization over the positive-definite cone: Interior point methods and combinatorial applications, *in* P. Pardalos, ed., 'Advances in Optimization and Parallel Computing', North-Holland, Amsterdam.

Anstreicher, K. M. & Bosch, R. A. (1992), 'Long steps in an $O(n^3 L)$ algorithm for linear programming', *Mathematical Programming* **54**, 251–265.

Barnes, E. R. (1986), 'A variation of karmakar's algorithm for solving linear programming problems', *Mathematical Programming* **36**, 174–182.

Bland (1977), '....'. No reference found.

Blum, L., Shub, M. & Smale, S. (1989), 'On a theory of computation over the real numbers: Np completeness, recursive functions and universal machines', *Bull. Amer. Math. Soc. (N.S.)* **21**(1), 1–46.

Boyd, S., Ghaoui, L. E., Feron, E. & Balakrishnan, V. (1994), *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia.

Boyd, S. & Vandenberghe, L. (1995), 'Introduction to convex optimization with engineering applications', Lecture notes EE392x, Stanford University. (Available at `isl.stanford.edu/pub/boyd/`).

Braatz, R. D., Young, P. M., Doyle, J. C. & Morari, M. (1994), 'Computational complexity of μ calculation', *IEEE Trans. Aut. Control* **39**(5), 1000–1002.

Dantzig, G. (1951), Maximization of a linear function of variables subject to linear inequalities, *in* T. Koopmans, ed., 'Activity Analysis of Production and Allocation', John Wiley & Sons, Inc., pp. 339–347.

den Hertog, D. (1993), *Interior Point Approach to Linear, Quadratic and COnvex Programming*, Kluwer.

Desoer, C. A. & Vidyasagar, M. (1975), *Feedback Systems: Input-Output Properties*, Academic Press, New York.

Flecther, R. (1987), *Practical Methods of Optimization, 2nd ed.*, John Wiley & Sons, New York.

Freund, R. M. (1991), 'Polynomial-time algorithm for linear programming based only on primal scaling and the project gradients of a potential function', *Mathematical Programming* **51**, 203–222.

Frisch, K. R. (1955*a*), The logarithmic potential method for solving linear programming problems, Technical report, Institute of Economics, Oslo, Norway. Memorandum.

Frisch, K. R. (1955*b*), The logarithmic potential method of convex programming, Technical report, Institute of Economics, Oslo, Norway. Memorandum.

Gahinet, P. & Nemirovsky, A. (1993), 'Lmi lab—a package for manipulating and solving lmis'. Preprint.

Goemans, M. X. (1994), Linear programming. part of lecture notes for 18.415/6.854 "Advanced Algorithms", MIT, Cambridge, MA, `http://theory.lcs.mit.edu/~goemans/`.

Goldfarb, D. & Todd, M. (1989), linear programming, *in* 'Handbook in Operations Research and Management Science', Vol. 1, Elsevier Science Publishers, pp. 73–170.

Golub, G. M. & Loan, C. V. (1983), *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland.

Huard, P. (1967), Resolution of mathematical programming with nonlinear constraints by the method of centers, *in* J. Abadie, ed., 'Nonlinear programming', North-Holland Publishing Company, Amsterdam, pp. 207–219.

Karmakar, N. K. (1984), 'A new polynomial-time algorithm for linear programming', *Combinatorica* pp. 373–395.

Khachian, L. G. (1979), 'A polynomial algorithm in linear programming', *Doklady Akademiia Nauk SSSR* **244**(5), 1093–1096. Translated into English in *Soviet Mathematics Doklady 20, 191-194*.

Khargonekar, P. P. & Rotea, M. A. (1991), 'Mixed H_2/H_∞ control: a convex optimization approach', *IEEE Trans. on Aut. Control* **36**(7), 824–837.

Klee, V. & Minty, G. J. (1972), How good is the simplex algorithm?, *in* O. Shisha, ed., 'Inequalities III', Academic Press, New York, pp. 159–175.

Luenberger, D. G. (1969), *Optimization by Vector Space Methods*, John Wiley, New York.

Nesterov, Y. & Nemirovsky, A. (1993), *Interior Point Polynomial Methods in Convex Programming*, Vol. 13 of *Studies in Applied Mathematics*, SIAM, Philadelphia.

Poljak, S. & Rohn, J. (1993), 'Checking robust nonsingularity is np-hard', *Math. Control, Signals and Systems* **6**, 1–9.

Todd, M. J. & Ye, Y. (1990), 'A centered projective algorithm for linear programming', *Mathematics of Operations Research* **15**, 508–529.

Vandenberghe, L. & Boyd, S. (1995), 'A primal-dual potential reduction method for problems involving matrix inequalities', *Mathematical Programming, Series B* **69**(1), 205–236.

Vandenberghe, L. & Boyd, S. (1996), 'Semidefinite programming', *Siam Review* **38**, 49–95.

Vanderbei, R. J., Meketon, M. S. & Freedman, B. A. (1986), 'A modification of karmakar's linear programming algorithm', *Algorithmica* **1**, 395–407.

Ye, Y. (1991), 'An $O(n^3L)$ potential reduction algorithm for linear programming', *Mathematical Programming* **50**, 239–258.