Preface

Books on a technical topic – like linear programming – *without* exercises ignore the principal beneficiary of the endeavor of writing a book, namely the student – who learns best by doing exercises, of course. Books *with* exercises – if they are challenging or at least to some extent so – need a solutions manual so that students can have recourse to it when they need it. Here we give solutions to all exercises and case studies of M. Padberg's *Linear Optimization and Extensions* (second edition, Springer-Verlag, Berlin, 1999). In addition we have included several new exercises and taken the opportunity to correct and change some of the exercises of the book. Here and in the main text of the present volume the terms "book", "text" etc. designate the second edition of Padberg's LP book and the page and formula references refer to that edition as well. All new and changed exercises are marked by a star * in this volume. The changes that we have made in the original exercises are inconsequential for the main part of the original text where several of the exercises (especially in Chapter 9) are used on several occasions in the proof arguments. None of the exercises that are used in the estimations, etc. have been changed. Quite a few exercises instruct the students to write a program in a computer language of their own choice. We have chosen to do that in most cases in MATLAB without *any* regard to efficiency, etc. Our prime goal here is to use a macro-language that resembles as closely as possible the mathematical statement of the respective algorithms. Once students master this first level, they can then go ahead and discover the pleasures and challenges of writing efficient computer code on their own.

To make the present volume as self-contained as possible, we have provided here summaries of each chapter of Padberg's LP book. While there is some overlap with the text, we think that this is tolerable. The summaries are –in almost all cases– without proofs, thus they provide a "mini-version" of the material treated in the text. Indeed, we think that having such summaries without the sometimes burdensome proofs is an advantage to the reader who wants to acquaint herself/himself with the material treated at length in the text. To make the cross-referencing with the text easy for the reader, we have numbered all chapters (and most sections and subsections) as well as the formulas in these summaries exactly like in the text. Moreover, we have reproduced here most of the illustrations of the text as we find these visual aids very helpful in communicating the material. Finally, we have reproduced here the appendices of the text as the descriptions of the cases contained therein would have taken too much space anyway.

We have worked on the production of this volume over several years and did so quite frequently at the Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB) in Berlin, Germany, where Alevras was a research fellow during some of this time. We are most grateful to ZIB's vice-president, Prof. Dr. Martin Grötschel, for his hospitality and tangible support of our endeavor. Padberg's work was also supported in part through an ONR grant and he would like to thank Dr. Donald Wagner of the Office of Naval Research, Arlington, VA, for his continued support.

New York City, January, 2001

Dimitris Alevras
Manfred Padberg

9. Ellipsoid Algorithms

Divide et impera![1]
Niccolo Machiavelli (1469-1527 A.D.)

Here we summarize the essentials of Chapter 9 of the text. We consider the linear optimization problem over a rational polyhedron $P \subseteq \mathbb{R}^n$ of facet complexity ϕ

$$\max\{\boldsymbol{cx} : \boldsymbol{x} \in P\} .$$

In Chapter 7.5.3 we reduced the problem of polynomial solvability of this problem to the question of the existence of subroutines FINDXZ(P,n,ϕ,Φ,$\boldsymbol{c}$,z^k,$\boldsymbol{x}$,FEAS) or FINDZX(P,n,ϕ,$\boldsymbol{c}$,z^k,$\boldsymbol{x}$,FEAS) that solve a **feasibility problem** in polynomial time. The ellipsoid algorithm settles this existence question in a theoretically satisfactory way for any rational polyhedron $P \subseteq \mathbb{R}^n$.

By point 7.5(d), we can replace P by a rational *polytope* P_Φ of equal dimension without changing the optimization problem. We assume that we have a linear description $\boldsymbol{Ax} \le \boldsymbol{b}$ of P_Φ with rational data $\boldsymbol{A}$, $\boldsymbol{b}$ and initially, that either $P_\Phi = \emptyset$ or $dim P_\Phi = n$. The case of flat polyhedra is discussed separately. It follows from point 7.5(d) that the ball $B(\boldsymbol{0}, R)$ contains P_Φ, where $R = \sqrt{n}2^\Phi$ and

$$\Phi = \langle \boldsymbol{c} \rangle + 8n\phi + 2n^2\phi + 2 .$$

The center of $B(\boldsymbol{0}, R)$ is $\boldsymbol{x}^0 = \boldsymbol{0}$. Checking $\boldsymbol{x}^0 \in P_\Phi$ we either find an inequality $\boldsymbol{a}^0\boldsymbol{x} \le a_0$ of the linear description of P_Φ such that $\boldsymbol{a}^0\boldsymbol{x}^0 > a_0$ or $\boldsymbol{x}^0 \in P_\Phi$ and we are done. If $\boldsymbol{a}^0\boldsymbol{x}^0 > a_0$ then $P_\Phi \subseteq B(\boldsymbol{0}, R) \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^0\boldsymbol{x} \le a_0\}$. Replacing a_0 by $\boldsymbol{a}^0\boldsymbol{x}^0$ we have that

$$P_\Phi \subseteq S_1 = B(\boldsymbol{0}, R) \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^0\boldsymbol{x} \le \boldsymbol{a}^0\boldsymbol{x}^0\} \subseteq E_1 ,$$

where E_1 is an ellipsoid (of minimum volume) that contains S_1. Let $\boldsymbol{x}^1$, the center of E_1, be the next "trial" solution: if $\boldsymbol{x}^1 \in P_\Phi$ we are done. Otherwise, we find an inequality $\boldsymbol{a}^1\boldsymbol{x} \le a_1$ from the linear description of P_Φ such that $\boldsymbol{a}^1\boldsymbol{x}^1 > a_1$ and iterate. At the k^{th} iteration of this algorithm we have the center $\boldsymbol{x}^k$ of an ellipsoid $E_k = E_{\boldsymbol{Q}_k}(\boldsymbol{x}^k, 1)$, where $\boldsymbol{Q}_k = \boldsymbol{F}_k\boldsymbol{F}_k^T$ is a positive definite matrix defining E_k. By construction $P_\Phi \subseteq E_k$. Either $\boldsymbol{x}^k \in P_\Phi$ – in which case we are done – or we find an inequality $\boldsymbol{a}^T\boldsymbol{x} \le a_0$ belonging to the linear description of P or P_Φ such that $\boldsymbol{a}^T\boldsymbol{x}^k > a_0$. In this case we set

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \frac{1}{n+1}\boldsymbol{F}_k\boldsymbol{d} \quad \text{where} \quad \boldsymbol{d} = \frac{\boldsymbol{F}_k^T\boldsymbol{a}}{\|\boldsymbol{F}_k^T\boldsymbol{a}\|} , \tag{9.1}$$

$$\boldsymbol{F}_{k+1} = \sqrt{\frac{n^2}{n^2-1}}\boldsymbol{F}_k\left(\boldsymbol{I}_n - \left(1 - \sqrt{\frac{n-1}{n+1}}\right)\boldsymbol{d}\boldsymbol{d}^T\right) . \tag{9.2}$$

We get an ellipsoid $E_{k+1} = E_{\boldsymbol{Q}_{k+1}}(\boldsymbol{x}^{k+1}, 1)$ with center $\boldsymbol{x}^{k+1}$ and positive definite matrix $\boldsymbol{Q}_{k+1} = \boldsymbol{F}_{k+1}\boldsymbol{F}_{k+1}^T$ defining E_{k+1}. As shown in Chapter 9.2, $E_{k+1} \supseteq P_\Phi$ and

$$vol(E_{k+1}) \le e^{-1/2n} vol(E_k) . \tag{9.3}$$

[1]Divide and conquer!

Iterating k times, we get $vol(E_k) \leq V_0 e^{-k/2n}$ for $k \geq 0$, where V_0 is the volume of $B(\mathbf{0}, R)$. Unless the algorithm stops with $\boldsymbol{x}^k \in P_\Phi$ for some k, it suffices to iterate at most

$$k_E = \lceil 2n(\log V_0 - \log V_{P_\Phi}) \rceil$$

times to conclude that $P_\Phi = \emptyset$, where V_{P_Φ} is the volume of P_Φ. By assumption P is either empty or full-dimensional. If $P \neq \emptyset$, we can bound V_{P_Φ} from below because P is a rational polyhedron.

In the left part of Figure 9.1 we show the iterative application of the ellipsoid algorithm when applied to the data of Exercise 8.2 (ii) without the objective function. We start at the point $\boldsymbol{x}^0 = (60, 60)$ as the center of the initial ball with radius $\|\boldsymbol{x}^0\|$ which contains all of the feasible set. We select as the "next" inequality $\boldsymbol{a}^T\boldsymbol{x} \leq a_0$ the one for which the slack $\boldsymbol{a}^T\boldsymbol{x}^0 - a_0$ is largest, to get $\boldsymbol{x}^1$; etc.

The above formulas yield polynomial *step complexity*, but not polynomial *time complexity* of the calculation. For the latter it is necessary that the digital sizes of $\langle \boldsymbol{x}^k \rangle$ and $\langle \boldsymbol{F}_k \rangle$ of the iterates stay bounded by a polynomial function of n, ϕ and $\langle \boldsymbol{c} \rangle$. It must also be shown that all necessary calculations can be carried out in *approximate* arithmetic, i.e. on a computer with limited word size. In the left part of Figure 9.1 we pretended that we can compute (9.1) and (9.2) "perfectly" – even though we divide, take square roots and calculate on a computer with a word size of merely 64 bits. To be correct, we have to replace the equality signs in (9.1) and (9.2) by the $\approx$ sign and specify the *precision* with which we need to calculate the corresponding numbers.

The geometric idea for the way to deal with the problem of approximate calculations is shown in the right part of Figure 9.1 for the ellipsoid E_1: since we cannot compute the center $\boldsymbol{x}^1$ of E_1 by formula (9.1) *exactly*, we get an approximate center $\boldsymbol{x}^A$ by committing round-off errors. To approximate the matrix $\boldsymbol{F}_1$ given by (9.2) we multiply the right-hand side by some factor $\beta \geq 1$, i.e. we scale all elements of it *up* to make them bigger. An approximate computation with round-off errors yields a matrix $\boldsymbol{F}_A$ that is used in lieu of $\boldsymbol{F}_1$. This corresponds to "blowing up" the perfect arithmetic ellipsoid E_1 concentrically to the ellipsoid E_β of Figure 9.1, i.e. E_β is a homothetic image of E_1 with a factor $\beta \geq 1$ of dilatation. The approximate calculation of $\boldsymbol{F}_A$ is then carried out with a sufficient precision to guarantee that the ellipsoid E_A with center $\boldsymbol{x}^A$ and defining matrix $\boldsymbol{Q}_A = \boldsymbol{F}_A\boldsymbol{F}_A^T$ contains the ellipsoid E_1 completely. In Chapter 9.2 we show that a blow-up factor of

$$\beta = 1 + 1/12n^2$$

works, where n is the number of variables of the optimization problem. Our graphical illustration in Figure 9.1 is "artistic". We used $\beta \approx \sqrt{1.5}$ to produce the figure and not $\beta = 49/48$, which is a lot smaller and works in $\mathbb{R}^2$.

In every iteration the ellipsoid algorithm needs only one inequality that is violated *or* the message that a violated inequality does not exist, i.e., at every iteration we have to solve a **separation problem** (or constraint identification problem) for the polyhedron P like the one we discussed in point 7.5(h). So far we have assumed that the number of constraints of the linear description of P does not matter and that the problem of finding a violated inequality can e.g. be done by *listing and checking* every single one of them. We call this method LIST-and-CHECK. LIST-and-CHECK does not give a polynomial algorithm for the linear optimization over rational polyhedra if the number of constraints of the linear description of P is exponential in n. But assuming the existence of *some* polynomial-time algorithm for the separation problem we get the existence of a polynomial-time algorithm for the optimization problem over rational polyhedra – namely the ellipsoid algorithm. It is a nontrivial result that the reverse statement holds as well:

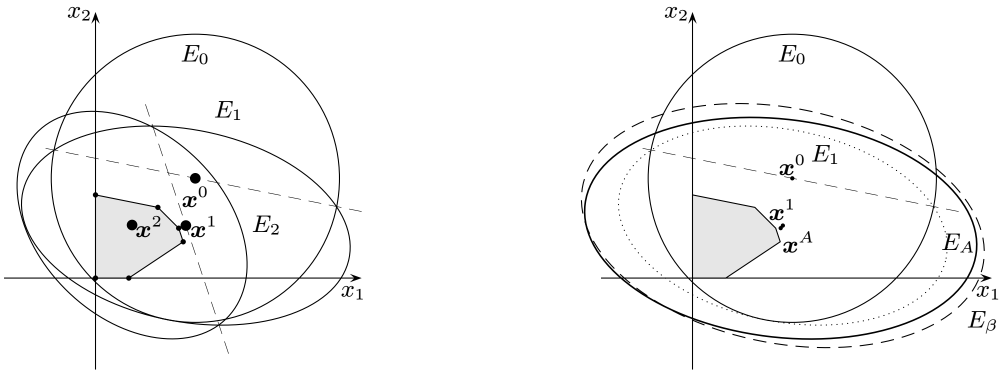


Fig. 9.1. The ellipsoid algorithm: "perfect" and approximate arithmetic

if for some rational polyhedron P the optimization problem can be solved in polynomial time, then the separation problem for P can be solved in polynomial time as well. This equivalence of optimization and separation for rational polyhedra is of fundamental importance in itself and particularly important for the field of combinatorial optimization: it constitutes the theoretical backbone of the algorithmic approach to combinatorial optimization problems called branch-and-cut.

9.1 Matrix Norms, Approximate Inverses, Matrix Inequalities

We have to "truncate" numbers in the ellipsoid algorithm and thus to replace e.g. a matrix $\boldsymbol{F}$ by some matrix $\boldsymbol{F}_A$, say, satisfying $\boldsymbol{F}_A \approx \boldsymbol{F}$. To analyze such an approximation we need a *norm*. The *Frobenius norm* of an $m \times n$ matrix $\boldsymbol{F}$ of real numbers f^i_j is

$$\|\boldsymbol{F}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(f^i_j)^2} \; . \tag{9.4}$$

Other norms that are frequently encountered in numerical linear algebra are

$$\|\boldsymbol{F}\|_1 = \max_{1\le j\le n}\sum_{i=1}^{m}|f^i_j| \; , \quad \|\boldsymbol{F}\|_2 = \max\{\|\boldsymbol{F}\boldsymbol{x}\| : \|\boldsymbol{x}\| = 1\} \; , \quad \|\boldsymbol{F}\|_\infty = \max_{1\le i\le m}\sum_{j=1}^{n}|f^i_j| \; .$$

$\|\boldsymbol{F}\|_2$ is the *spectral norm* and its value equals the square root of the largest eigenvalue of $\boldsymbol{F}^T\boldsymbol{F}$. If $\boldsymbol{F}$ is nonsingular, then $\|\boldsymbol{F}^{-1}\|_2 = \lambda^{-1}(\boldsymbol{F})$, i.e., it is the reciprocal of the square root of the smallest eigenvalue of $\boldsymbol{F}^T\boldsymbol{F}$. There are a number of relationships between these various matrix norms, e.g.

$$\|\boldsymbol{F}\|_2 \le \|\boldsymbol{F}\|_F \le \sqrt{n}\|\boldsymbol{F}\|_2 \; .$$

The spectral norm yields probably the most elegant proofs for what is to follow, but it is also the hardest one to compute. To avoid issues of computation we will not use it. We use the Frobenius norm and simply drop the subscript F for notational convenience. In Exercise 9.2 most of the properties of the Frobenius norm that we need are stated. In particular, for any two $m \times n$ real matrices we have

$$\|\boldsymbol{F}\boldsymbol{R}\| \le \|\boldsymbol{F}\|\|\boldsymbol{R}\| \ , \quad \|\boldsymbol{F}(\boldsymbol{I}_n - \alpha \boldsymbol{r}\boldsymbol{r}^T)\|^2 = \|\boldsymbol{F}\|^2 - \alpha(2-\alpha\|\boldsymbol{r}\|^2)\|\boldsymbol{F}\boldsymbol{r}\|^2 \ . \tag{9.5}$$

Remark 9.1 *Let $\boldsymbol{R}$ be any $n \times n$ matrix of reals with $\|\boldsymbol{R}\| < 1$. Then $(\boldsymbol{I}_n - \boldsymbol{R})^{-1}$ exists and*

$$\|(\boldsymbol{I}_n - \boldsymbol{R})^{-1}\| \le \frac{1}{1-\|\boldsymbol{R}\|} \ . \tag{9.6}$$

Let the elements of $\boldsymbol{F}_A$ be obtained by truncating the elements of some nonsingular matrix $\boldsymbol{F}$, i.e., $\boldsymbol{F}_A \approx \boldsymbol{F}$, and $\boldsymbol{F}_A = \boldsymbol{F} + \boldsymbol{R}$. Thus $\boldsymbol{R}$ is the matrix of "errors" due to the rounding or the truncation and the Frobenius norm is the sum of the squared errors for each element of $\boldsymbol{F}_A$. We need to know when $\boldsymbol{F}_A$ is nonsingular and how the errors "propagate" into the inverse of $\boldsymbol{F}_A$.

Remark 9.2 *Let $\boldsymbol{F}$ be any nonsingular matrix of size $n \times n$ and $\boldsymbol{R}$ be any $n \times n$ matrix with $\|\boldsymbol{F}^{-1}\boldsymbol{R}\| < 1$. Then $(\boldsymbol{F}+\boldsymbol{R})^{-1}$ exists and satisfies the inequalities*

$$\|(\boldsymbol{F}+\boldsymbol{R})^{-1}\| \le \frac{\|\boldsymbol{F}^{-1}\|}{1-\|\boldsymbol{F}^{-1}\boldsymbol{R}\|} \ , \tag{9.7}$$

$$\|(\boldsymbol{F}+\boldsymbol{R})^{-1} - \boldsymbol{F}^{-1}\| \le \frac{\|\boldsymbol{R}\| \, \|\boldsymbol{F}^{-1}\|^2}{1-\|\boldsymbol{F}^{-1}\boldsymbol{R}\|} \ . \tag{9.8}$$

To carry out the analysis of the ellipsoid algorithm using approximate arithmetic, we need the following two inequalities for the determinant and the norm of the inverse of a nonsingular matrix repeatedly.

Remark 9.3 *(i) Let $\boldsymbol{F}$ be any $n \times n$ matrix of reals. Then we have the inequality*

$$|det\boldsymbol{F}| \le n^{-n/2}\|\boldsymbol{F}\|^n \ . \tag{9.9}$$

(ii) If $\boldsymbol{F}$ is nonsingular and $n \ge 2$, then we have the inequality

$$\|\boldsymbol{F}^{-1}\| \le n(n-1)^{-\frac{n-1}{2}} \frac{\|\boldsymbol{F}\|^{n-1}}{|det\boldsymbol{F}|} \ . \tag{9.10}$$

9.2 Ellipsoid "Halving" in Approximate Arithmetic

To carry out the various constructions analytically, we drop the index k and denote by $\boldsymbol{F}$ the nonsingular matrix defining the "current" ellipsoid E_0 and by $\boldsymbol{x}^0$ its center, i.e.,

$$E_0(\boldsymbol{x}^0, 1) = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{F}^{-1}(\boldsymbol{x}-\boldsymbol{x}^0)\| \le 1\} \ . \tag{9.11}$$

Let $\boldsymbol{a}^T\boldsymbol{x} \le a_0$ be the linear inequality that the algorithm identifies and denote by $\boldsymbol{x}^P$ and $\boldsymbol{F}_P$ the updates (9.1) and (9.2) that result if calculated in *perfect* arithmetic, i.e. with an infinite precision:

$$\boldsymbol{x}^P = \boldsymbol{x}^0 - \frac{1}{n+1}\boldsymbol{F}\boldsymbol{d} \quad \text{where} \quad \boldsymbol{d} = \frac{\boldsymbol{F}^T\boldsymbol{a}}{\|\boldsymbol{F}^T\boldsymbol{a}\|} \tag{9.12}$$

$$\boldsymbol{F}_P = \sqrt{\frac{n^2}{n^2-1}}\boldsymbol{F}\left(\boldsymbol{I}_n - \left(1-\sqrt{\frac{n-1}{n+1}}\right)\boldsymbol{d}\boldsymbol{d}^T\right) \ . \tag{9.13}$$

Assuming that $\boldsymbol{F}_P$ is nonsingular, we get in perfect arithmetic an ellipsoid E_P with center $\boldsymbol{x}^P$ which in the iterative scheme of the introduction is the "next" ellipsoid, i.e.

$$E_P(\boldsymbol{x}^P, 1) = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{F}_P^{-1}(\boldsymbol{x}-\boldsymbol{x}^P)\| \leq 1\} \ . \tag{9.14}$$

For any "blow-up" factor $\beta \geq 1$ denote by E_β the enlarged ellipsoid with center $\boldsymbol{x}^P$, i.e.,

$$E_\beta = E_P(\boldsymbol{x}^P, \beta) = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{F}_P^{-1}(\boldsymbol{x}-\boldsymbol{x}^P)\| \leq \beta\} \ . \tag{9.15}$$

Thus the enlarged ellipsoid E_β is defined with respect to the matrix

$$\boldsymbol{F}_\beta = \beta \boldsymbol{F}_P \ . \tag{9.16}$$

Because of the finite wordlength of the computer we commit round-off errors and compute approximately

$$\boldsymbol{x}^A \approx \boldsymbol{x}^P \quad , \qquad \boldsymbol{F}_A \approx \boldsymbol{F}_\beta \ . \tag{9.17}$$

Assume that the error in the approximate calculation satisfies

$$\|\boldsymbol{x}^A - \boldsymbol{x}^P\| \leq \delta \quad \text{and} \quad \|\boldsymbol{F}_A - \boldsymbol{F}_\beta\| \leq \delta \quad \text{where} \quad \delta \leq p(n)\frac{|det\boldsymbol{F}|}{\|\boldsymbol{F}\|^{n-1}} \ , \ p(n) = 10^{-4}n^{-2} \ . \tag{9.18}$$

From inequality (9.9) it follows that the error of the approximation is less than $\|\boldsymbol{F}\|$ because by (9.9) e.g. $\|\boldsymbol{x}^A - \boldsymbol{x}^P\| \leq p(n)n^{-n/2}\|\boldsymbol{F}\|$. Condition (9.18) is translated in Chapter 9.3 into the number of digits of each component of $\boldsymbol{x}^A$ and $\boldsymbol{F}_A$ that need to be calculated *correctly* – before and after the "decimal" point in binary arithmetic. The calculations (9.17) yield an approximation E_A

$$E_A(\boldsymbol{x}^A, 1) = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{F}_A^{-1}(\boldsymbol{x}-\boldsymbol{x}^A)\| \leq 1\} \ , \tag{9.19}$$

to E_β which is an ellipsoid if $\boldsymbol{F}_A$ is "close enough" to $\boldsymbol{F}_\beta$ so as to guarantee the nonsingularity of $\boldsymbol{F}_A$.

The "battle-plan" of the proof is as follows:

- Establish that $\boldsymbol{F}_P$ is nonsingular if $\boldsymbol{F}$ is nonsingular and that (9.3) remains correct.
- Show that if E_0 contains P or P_Φ then so does the ellipsoid E_P.
- Show that if (9.18) is satisfied neither $\boldsymbol{x}^A$ nor $\boldsymbol{F}_A$ "grow" too much in size.
- Assuming (9.18) show that $\boldsymbol{F}_A$ is nonsingular and that $E_A \supseteq E_P$.
- Assuming (9.18) show that $\frac{vol(E_A)}{vol(E_0)}$ satisfies a relation like (9.3) to conclude a polynomial running time of the approximate calculations.
- Establish a lower bound on the volume V_{P_Φ} if $dimP_\Phi = n$ *and* prove inductively that (9.18) is satisfied for all necessary iterations. This is done in the next section.

The first step is easy because

$$det\boldsymbol{F}_P = \left(1+\frac{1}{n}\right)^{-\frac{n+1}{2}}\left(1-\frac{1}{n}\right)^{-\frac{n-1}{2}} det\boldsymbol{F} \ . \tag{9.20}$$

The factor appearing in (9.20) satisfies

$$\left(1+\frac{1}{n}\right)^{-\frac{n+1}{2}}\left(1-\frac{1}{n}\right)^{-\frac{n-1}{2}} \le e^{-\frac{1}{2n}} \qquad \text{for all } n \ge 1. \tag{9.21}$$

Computing the volumina of E_0 and E_P we get from (7.23) and (9.20)

$$\frac{vol(E_P)}{vol(E_0)} = \left(1+\frac{1}{n}\right)^{-\frac{n+1}{2}}\left(1-\frac{1}{n}\right)^{-\frac{n-1}{2}} \le e^{-\frac{1}{2n}}$$

by (9.21) and (9.3) follows. By (9.20) $\boldsymbol{F}_P$ is nonsingular and its inverse in terms of $\boldsymbol{F}^{-1}$ is

$$\boldsymbol{F}_P^{-1} = \sqrt{\frac{n^2-1}{n^2}}\left(\boldsymbol{I}_n - \left(1-\sqrt{\frac{n+1}{n-1}}\right)\boldsymbol{d}\boldsymbol{d}^T\right)\boldsymbol{F}^{-1} \ . \tag{9.22}$$

For notational convenience define $\boldsymbol{Q} = \boldsymbol{F}\boldsymbol{F}^T$ and $\boldsymbol{Q}_P = \boldsymbol{F}_P\boldsymbol{F}_P^T$, i.e. $\boldsymbol{Q}$ and $\boldsymbol{Q}_P$ are the positive definite matrices defining E_0 and E_P.

$$\boldsymbol{Q}_P = \frac{n^2}{n^2-1}\boldsymbol{Q}\left(\boldsymbol{I}_n - \frac{2}{n+1}\frac{\boldsymbol{a}\boldsymbol{a}^T\boldsymbol{Q}}{\boldsymbol{a}^T\boldsymbol{Q}\boldsymbol{a}}\right) \ , \quad \boldsymbol{Q}_P^{-1} = \frac{n^2-1}{n^2}\left(\boldsymbol{Q}^{-1} + \frac{2}{n-1}\frac{\boldsymbol{a}\boldsymbol{a}^T}{\boldsymbol{a}^T\boldsymbol{Q}\boldsymbol{a}}\right) \ . \tag{9.23}$$

Remark 9.4 *(i) For all $\boldsymbol{x} \in E_0$ and $\boldsymbol{a} \in \mathbb{R}^n$, $\boldsymbol{a} \neq \boldsymbol{0}$, $-\sqrt{\boldsymbol{a}^T\boldsymbol{Q}\boldsymbol{a}} \le \boldsymbol{a}^T(\boldsymbol{x}-\boldsymbol{x}^0) \le \sqrt{\boldsymbol{a}^T\boldsymbol{Q}\boldsymbol{a}}$.*
(ii) Let $\boldsymbol{a}^T\boldsymbol{x} \le a_0$ with $\boldsymbol{a} \neq \boldsymbol{0}$ be any inequality such that $\boldsymbol{a}^T\boldsymbol{x}^0 \ge a_0$ and $\mathcal{X} \subseteq E_0$ be any subset of E_0 such that $\boldsymbol{a}^T\boldsymbol{x} \le a_0$ for all $\boldsymbol{x} \in \mathcal{X}$. Then $\mathcal{X} \subseteq E_P$.

By inductive reasoning it follows that $P_\Phi \subseteq E_k$ if $B(\boldsymbol{0}, R)$ has a large enough radius to contain P_Φ initially.

To justify the approximate calculations of $\boldsymbol{x}^A$ and $\boldsymbol{F}_A$, we compute like in Exercise 9.2 (iii) from (9.16), (9.13) and (9.22)

$$\|\boldsymbol{F}_\beta\| \le \beta\sqrt{\frac{n^2}{n^2-1}}\|\boldsymbol{F}\| \ , \quad \|\boldsymbol{F}_\beta^{-1}\| \le \beta^{-1}\frac{n+1}{n}\|\boldsymbol{F}^{-1}\| \ , \tag{9.24}$$

where we have used $\|\boldsymbol{d}\| = 1$, see (9.12). In the following we use the inequality

$$2^x \ge 1 + \frac{2}{3}x + \frac{2}{9}x^2 \qquad \text{for all } x \ge 0 \ . \tag{9.25}$$

Remark 9.5 *If (9.18) is satisfied, then for $\beta = 1 + 1/12n^2$ and all $n \ge 2$*

$$\|\boldsymbol{x}^A\| \le \|\boldsymbol{x}^0\| + \frac{1}{n}\|\boldsymbol{F}\| \ , \quad \|\boldsymbol{F}_A\| \le 2^{1/n^2}\|\boldsymbol{F}\| \ . \tag{9.26}$$

The following inequality is readily verified for all $n \geq 2$

$$1 + 2(n+1)(n-1)^{-\frac{n-1}{2}} n^{-2} 10^{-4} \leq 1 + \frac{1}{12n^2} \ . \tag{9.27}$$

Remark 9.6 *If (9.18) is satisfied and* $det\boldsymbol{F} \neq 0$*, then for* $\beta = 1 + 1/12n^2$ *and all* $n \geq 2$ *the matrix* $\boldsymbol{F}_A$ *is nonsingular and* $E_A(\boldsymbol{x}^A, 1) \supseteq E_P(\boldsymbol{x}^P, 1)$*.*

The main point in the proof that $E_P(\boldsymbol{x}^P, 1) \subseteq E_A(\boldsymbol{x}^A, 1)$ is the estimation

$$\|\boldsymbol{F}_A^{-1}(\boldsymbol{x} - \boldsymbol{x}^A)\| \leq (1 + \|\boldsymbol{F}_A^{-1}\| \, \|\boldsymbol{F}_\beta - \boldsymbol{F}_A\|)(\|\boldsymbol{F}_\beta^{-1}(\boldsymbol{x} - \boldsymbol{x}^P)\| + \|\boldsymbol{F}_\beta^{-1}(\boldsymbol{x}^P - \boldsymbol{x}^A)\|) \ . \tag{9.28}$$

By Remark 9.4 the feasible set is contained in the approximate ellipsoid if (9.18) is satisfied. The verification of (9.18) will be done by induction.

To estimate next $\frac{vol(E_0)}{vol(E_A)}$ we calculate using (9.4), (9.13), (9.16) and (9.22)

$$\|\boldsymbol{F}^{-1}\boldsymbol{F}_\beta\| = \beta\sqrt{n\left(1 - \frac{1}{(n+1)^2}\right)} \ , \quad \|\boldsymbol{F}_\beta^{-1}\boldsymbol{F}\| = \frac{\sqrt{n^3+n+2}}{\beta n} \ . \tag{9.29}$$

Remark 9.7 *If* $det\boldsymbol{F} \neq 0$ *and (9.18) is satisfied, then for* $\beta = 1 + 1/12n^2$ *and for all* $n \geq 2$

$$2^{-1/n}|det\boldsymbol{F}| \leq |det\boldsymbol{F}_A| \leq 2^{-1/4n}|det\boldsymbol{F}| \ . \tag{9.30}$$

If (9.18) is satisfied at every iteration we can calculate the number of iterations required by the ellipsoid algorithm using approximate arithmetic, i.e. when (9.1) and (9.2) are replaced by (9.17). From (7.23) we compute

$$\frac{vol(E_A)}{vol(E_0)} = \frac{|det\boldsymbol{F}_A|}{|det\boldsymbol{F}|} \leq 2^{-1/4n} < e^{-1/6n}$$

using Remark 9.7 and (9.25). Denoting the ellipsoid at the k^{th} iteration again by E_k, we get $vol(E_k) < V_0 e^{-k/6n}$ and consequently, after at most

$$k_A = \lceil 6n(\log V_0 - \log V_{P_\Phi}) \rceil$$

iterations the ellipsoid algorithm with approximate arithmetic stops with the message that $P_\Phi = \emptyset$ – unless for some $k < k_A$ the corresponding iterate $\boldsymbol{x}^k$ belongs to P_Φ. Here we have assumed that (9.18) is satisfied at every iteration and that $P_\Phi \neq 0$ implies $dimP_\Phi = n$ and that $vol(P_\Phi) \geq V_{P_\Phi}$ – assumptions that we will have to remove later.

9.3 Polynomial-Time Algorithms for Linear Programming

Having settled the *analytical* problem of calculating the "perfect" arithmetic ellipsoid approximately by permitting round-off errors let us consider the linear program

(LP) $$\max\{\boldsymbol{cx} : \boldsymbol{Ax} \leq \boldsymbol{b}\} \ ,$$

where $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ all have **integer** components only, $m \geq 1$, $n \geq 2$ and $\boldsymbol{A} \neq \mathbf{O}$ to rule out trivialities. The problem (LP) has only inequalities, i.e. any equations have been eliminated or replaced by their corresponding pairs of inequalities.

From Chapter 7.5.1 we know that the integrality assumption is polynomially equivalent to assuming that the data are rational. So the integrality assumption is convenient, but not necessary. Other than this one we make no assumption. The rank of $\boldsymbol{A}$ can be anywhere between 1 and $\min\{m, n\}$ and the feasible set

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\} \tag{9.31}$$

can be a solid, flat, pointed or blunt polyhedron of $\mathbb{R}^n$. Nonnegativity requirements are part of $(\boldsymbol{A}, \boldsymbol{b})$.

To demonstrate the polynomial solvability of (LP) – polynomial in terms of m, n, the digital size $\langle \boldsymbol{c} \rangle$ of the vector $\boldsymbol{c}$ and the facet complexity ϕ of $\mathcal{X}$ – we proceed in three steps:

- We assume that $\mathcal{X}$ is *either* empty *or* bounded and full dimensional. By running the "basic" ellipsoid algorithm we decide whether or not $\mathcal{X} = \emptyset$ by producing a rational vector $\boldsymbol{x} \in \mathcal{X}$ if $\mathcal{X} \neq \emptyset$.
- We remove the assumptions and show that by embedding any $\mathcal{X}$ into a somewhat larger polyhedron $\mathcal{X}_h$ we can answer the question in polynomial time for any rational $\mathcal{X}$.
- We use binary search like in Chapter 7.5.3 to prove the existence of a polynomial-time algorithm for the linear program (LP).

To carry out the first step we make the following assumption.

Assumption A: *There exists $R \geq 1$, i.e.* some *radius, such that $\mathcal{X} \subseteq B(\boldsymbol{0}, R)$. Moreover, if $\mathcal{X} \neq \emptyset$ then there exist $\boldsymbol{x} \in relint\mathcal{X}$ and a radius $r > 0$ such that $B(\boldsymbol{x}, r) \subseteq \mathcal{X}$.*

To state the basic ellipsoid algorithm, we denote by x_j^P, x_j^k the components of $\boldsymbol{x}^P$, $\boldsymbol{x}^k$ and by ${}_\beta f_j^i$, ${}_k f_j^i$ the elements of $\boldsymbol{F}_\beta$, $\boldsymbol{F}_k$ for $1 \leq i, j \leq n$ and $k \geq 0$. The input consists of m, n, $\boldsymbol{A}$ and $\boldsymbol{b}$, the parameters R, T and p and if $\mathcal{X} \neq \emptyset$, $\boldsymbol{x}$ is the output of the algorithm.

Basic Ellipsoid Algorithm (m, n, R, T, p, $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{x}$)

Step 0: Set $\boldsymbol{x}^0 := 0$, $\boldsymbol{F}_0 := R\boldsymbol{I}_n$, $k := 0$.

Step 1: **if** $k \geq T$, **stop** "$\mathcal{X}$ is empty".
if $\boldsymbol{a}^i\boldsymbol{x}^k \leq b_i$ for all $1 \leq i \leq m$, **stop** "$\boldsymbol{x} := \boldsymbol{x}^k$".
Let $(\boldsymbol{a}^i, b_i)$ for some $i \in \{1, \ldots, m\}$ be such that $\boldsymbol{a}^i\boldsymbol{x}^k > b_i$ and set $\boldsymbol{a}^T := \boldsymbol{a}^i$.

Step 2: Calculate approximately $\boldsymbol{x}^{k+1} \approx \boldsymbol{x}^P$ and $\boldsymbol{F}_{k+1} \approx \boldsymbol{F}_\beta$ where

$$\boldsymbol{x}^P := \boldsymbol{x}^k - \frac{1}{n+1}\frac{\boldsymbol{F}_k\boldsymbol{F}_k^T\boldsymbol{a}}{\|\boldsymbol{F}_k^T\boldsymbol{a}\|}\ , \tag{9.32}$$

$$\boldsymbol{F}_\beta := \frac{n + 1/12n}{\sqrt{n^2-1}}\boldsymbol{F}_k\left(\boldsymbol{I}_n - \frac{1-\sqrt{(n-1)/(n+1)}}{\boldsymbol{a}^T\boldsymbol{F}_k\boldsymbol{F}_k^T\boldsymbol{a}}(\boldsymbol{F}_k^T\boldsymbol{a})(\boldsymbol{a}^T\boldsymbol{F}_k)\right)\ , \tag{9.33}$$

such that the binary representation of each component of $\boldsymbol{x}^{k+1}$ and $\boldsymbol{F}_{k+1}$ satisfies $|x_j^{k+1} - x_j^P| \leq 2^{-p}$ and $|{}_{k+1}f_j^i - {}_\beta f_j^i| \leq 2^{-p}$ for $1 \leq i, j \leq n$. Replace $k+1$ by k and **go to** Step 1.

Remark 9.8 (Correctness and finiteness) *If Assumption A is true, then the basic ellipsoid algorithm finds a rational vector $\boldsymbol{x} \in \mathcal{X}$ or concludes correctly that $\mathcal{X} = \emptyset$ when it is executed with the parameters*

$$T = \lceil 6n^2 \log\frac{R}{r}\rceil \quad , \quad p = 14 + n^2 + \lceil 15n \log\frac{R}{r}\rceil\ . \tag{9.34}$$

The proof uses Remarks 9.5, 9.6 and 9.7 to estimate *inter alia*

$$\|\boldsymbol{F}_k\| \le \sqrt{n}R2^{\frac{k}{n^2}} \ , \ \|\boldsymbol{x}^k\| \le kR2^{\frac{k}{n^2}} \ , \ R^n 2^{-\frac{k}{n}} \le |det\boldsymbol{F}_k| \le R^n 2^{-\frac{k}{4n}} \ . \tag{9.35}$$

We are now ready to drop Assumption A. Denote by ϕ the facet complexity of $\mathcal{X}$, i.e.

$$\phi \ge \max_{1\le i\le m} \{\langle \boldsymbol{a}^i\rangle + \langle b_i\rangle\} \ , \quad \phi_A = \max_{1\le i\le m} \langle \boldsymbol{a}^i\rangle \ . \tag{9.36}$$

It follows that $n+1 \le \phi_A < \phi$ since $\boldsymbol{A} \ne \mathbf{O}$ and moreover, we can choose ϕ such that $\phi \le \phi_A + \langle \boldsymbol{b}\rangle$, where $\langle \boldsymbol{b}\rangle$ is the digital size of $\boldsymbol{b}$. For any integer $h \ge 1$ we denote by $\boldsymbol{h}^{-1} \in \mathbb{R}^m$ the vector having m components equal to $1/h$ and let

$$\mathcal{X}_h = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} + \boldsymbol{h}^{-1}\} \ , \tag{9.37}$$

which corresponds to "perturbing" the feasible set $\mathcal{X}$ of (LP).

Remark 9.9 *(i)* $\mathcal{X} \ne \emptyset$ *if and only if* $\mathcal{X}_h \ne \emptyset$ *for all* $h \ge p2^{p\phi_A}$ *where* $p = 1 + \min\{m, n\}$.
(ii) If $\mathcal{X} \ne \emptyset$*, then for all* $u \ge 2^{n\phi}$ *and all integers* $h \ge 1$ *the set* $\mathcal{X}_h^u$ *of solutions to*

$$\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} + \boldsymbol{h}^{-1} \ , \ -u - 1/h \le x_j \le u + 1/h \ , \ \textit{for } 1 \le j \le n \tag{9.38}$$

is bounded, full dimensional and $B(\boldsymbol{x}, r_h) \subseteq \mathcal{X}_h^u$ *for all* $\boldsymbol{x} \in \mathcal{X}$ *where*

$$r_h = h^{-1}2^{-\phi_A+n} \ . \tag{9.39}$$

To visualize the construction of Remark 9.9 take for instance

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^2 : 1 \le x_1 \le 1, \, 1 \le x_2 \le 1\}$$

and bring it into the form (9.37). Graphing the corresponding solution set, one sees that the introduction of the perturbation $1/h$ in each inequality corresponds to a "tearing apart" of the solution set to obtain a full dimensional set of solutions; see also Figure 9.8 below. If $\mathcal{X}$ is empty then there is nothing to tear apart and as the first part of Remark 9.9 shows, the emptiness of $\mathcal{X}_h$ is preserved if the perturbation is "small enough". Running the basic ellipsoid algorithm with $R = 2^{n\phi}$, $T = 20n^3\phi$ and $p = 55n^2\phi$ we conclude like in point 7.6(g):

Remark 9.10 *Every* $m \times n$ *system of linear inequalities* $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}$ *with rational data can be "solved" in time that is polynomial in the digital size of its input.*

There are several ways to deal with the optimization aspect of the linear program (LP). The simplest way is to use linear programming duality and to reduce the optimization problem to the problem of finding a solution to a system of linear inequalities – like we did in Remark 6.5. The second way uses binary search and a "sliding objective" function; see the text.

Remark 9.11 *Every linear program with rational data can be "optimized" in time that is polynomial in the digital size of its input.*

Neither the radius R of the ball circumscribing $\mathcal{X}$ or $\mathcal{X}_h^u$ nor the radius r of the ball that is inscribed into $\mathcal{X}$ or $\mathcal{X}_h^u$, see (9.39), depend on the *number* m of linear inequalities of (LP). Consequently, none of the other two parameters T and p of the basic ellipsoid algorithm depends on the number m. They are polynomial functions of ϕ, n and $\langle \boldsymbol{c} \rangle$ only. The dependence of the basic ellipsoid algorithm on the number m of the inequalities of (LP) enters in Step 1 when we have to find a violated inequality for the system (9.43) or prove that none exists. The same is true for the auxiliary computations. For the time being we assume that we *find* violated inequalities by the "algorithm" LIST-and-CHECK that we discussed in the introduction. If m is of the order of n, i.e. $m = \mathcal{O}(n)$, then the total effort to solve (LP) becomes a polynomial function of n, ϕ and $\langle \boldsymbol{c} \rangle$ only, whereas in the general case we need, of course, note the dependence on m explicitly. Before coming back to the question of how to deal with the case of possibly exponentially many constraints defining $\mathcal{X}$ we first discuss some "practical" variants of the basic ellipsoid algorithm.

9.4 Deep Cuts, Sliding Objective, Large Steps, Line Search

Going back to Figure 9.1 we see that instead of cutting E_k with $\boldsymbol{a}^T\boldsymbol{x} \le a_0$ – where the right-hand side equals a_0 and which is valid for all $\boldsymbol{x} \in \mathcal{X}$ – we replaced a_0 by the larger quantity $\boldsymbol{a}^T\boldsymbol{x}^k$. This replacement forces the cut to pass through the *center* of the current ellipsoid and the resulting algorithm is therefore called **central cut** ellipsoid algorithm. It is not overly difficult to work out the formulas corresponding to (9.1) and (9.2) when instead of $\boldsymbol{a}^T\boldsymbol{x} \le \boldsymbol{a}^T\boldsymbol{x}^k$ we use the **deep cut** $\boldsymbol{a}^T\boldsymbol{x} \le a_0$. They are given below.

A less obvious modification of the basic algorithmic idea concerns the optimization aspect of (LP). Let

$$z = \max\{\boldsymbol{c}\boldsymbol{x}^k : \boldsymbol{x}^k \text{ feasible}\} ,$$

where initially $z = -\infty$. Then we can use the objective function as a "sliding constraint" of the form $\boldsymbol{c}\boldsymbol{x} \ge z$ where the value of z increases during the course of the calculations. This gives rise to a **sliding objective** and thereby to a device that speeds the convergence of the procedure considerably.

A third modification to the basic idea goes as follows. Suppose the current iterate $\boldsymbol{x}^k$ is feasible. Then the point $\boldsymbol{x}^* = \boldsymbol{x}^k + \boldsymbol{F}_k\boldsymbol{F}_k^T\boldsymbol{c}^T/\|\boldsymbol{c}\boldsymbol{F}_k\|$ maximizes the linear function $\boldsymbol{c}\boldsymbol{x}$ over the current ellipsoid $E_k = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{F}_k^{-1}(\boldsymbol{x} - \boldsymbol{x}^k)\| \le 1\}$; see Remark 9.4(i). Consequently, we can determine by a *least ratio test* the largest $\lambda \ge 0$ such that

$$\boldsymbol{x}(\lambda) = \boldsymbol{x}^k + \lambda(\boldsymbol{x}^* - \boldsymbol{x}^k) \text{ is feasible}$$

and thereby make a **large step** towards optimality by "shooting" through the interior of the feasible set. We calculate the largest λ, the corresponding feasible $\boldsymbol{x}(\lambda)$ and its objective function value z_λ, say. If $z_\lambda > z$, then we update the current best solution to be $\boldsymbol{x}(\lambda)$, replace z by z_λ and use in one of the subsequent iterations the objective function as a cut to reduce the volume of the ellipsoid.

The fourth modification of the basic algorithmic idea is aimed at improving the chances of the algorithm to find *feasible* solutions to (LP). The algorithm generates a sequence of individual points and their probability to fall into the feasible set is rather small. Consider two consecutive centers $\boldsymbol{x}^k$ and $\boldsymbol{x}^{k+1}$ of the ellipsoids generated by the algorithm. They determine a line

$$\boldsymbol{x}(\mu) = (1 - \mu)\boldsymbol{x}^k + \mu\boldsymbol{x}^{k+1} \text{ where } -\infty < \mu < +\infty .$$

We can decide the question of whether or not $\boldsymbol{x}(\mu)$ *meets* the feasible set by a **line search** that involves again a simple least ratio test. If the test is negative, we continue as we would do without it. If the test comes out positive, then we get an interval $[\mu_{min}, \mu_{max}]$ such that $\boldsymbol{x}(\mu)$ is feasible for all μ in the interval. Computing the objective function we find $\boldsymbol{cx}(\mu) = \boldsymbol{cx}^k + \mu(\boldsymbol{cx}^{k+1} - \boldsymbol{cx}^k)$. Consequently, if $\boldsymbol{cx}^{k+1} > \boldsymbol{cx}^k$ then $\overline{\mu} = \mu_{max}$ yields the best possible solution vector while $\overline{\mu} = \mu_{min}$ does so in the opposite case. The rest is clear: we proceed like we did in the case of large steps.

We are now ready to state an ellipsoid algorithm for linear programs in canonical form

$$(\mathrm{LP}_C) \qquad \max\{\boldsymbol{cx} : \widetilde{\boldsymbol{A}}\boldsymbol{x} \le \widetilde{\boldsymbol{b}}\,,\ \boldsymbol{x} \ge \boldsymbol{0}\}\ ,$$

where $(\widetilde{\boldsymbol{A}}, \widetilde{\boldsymbol{b}})$ is an $m \times (n+1)$ matrix of rationals. We assume that $\widetilde{\boldsymbol{A}}$ contains no zero row and denote by $(\boldsymbol{a}^i, b_i)$ for $1 \le i \le m+n$ the rows of the matrix $(\boldsymbol{A}, \boldsymbol{b}) = \begin{pmatrix} \widetilde{\boldsymbol{A}} & \widetilde{\boldsymbol{b}} \\ -\boldsymbol{I}_n & \boldsymbol{0} \end{pmatrix}$.

The DCS ellipsoid algorithm takes m, n, $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{c}$ as inputs. z_L is a lower bound, z_U an upper bound on the optimal objective function value. R is a common upper bound on the variables of (LP$_C$) and ε a perturbation parameter to ensure full dimensionality of the feasible set when intersected with the sliding objective function constraint $\boldsymbol{cx} \ge z$. Since we are perturbing the constraint set of (LP$_C$) by a parameter $\varepsilon > 0$ we shall call solutions to the perturbed constraint set nearly feasible or ε-feasible solutions and correspondingly, we shall utilize the term ε-optimal solution to denote a nearly feasible, nearly optimal solution to (LP$_C$). V_F is a positive lower bound on the volume of a full dimensional, ε-optimal set. In other words, if the current ellipsoid has a volume less than V_F we shall conclude that either ε-optimality is attained – if a feasible solution $\overline{\boldsymbol{x}}$ with objective function value z was obtained – or else that the feasible set of (LP$_C$) is empty. As we know from Chapters 7.5 and 9.3 we can *always* find theoretical values for ε and R and by consequence for z_L, z_U and V_F as well that the algorithm needs to converge.

In practice, we set the perturbation parameter e.g. $\varepsilon = 10^{-4}$ and use a rough data dependent estimate for the common upper bound R on the variables. Similarly, we use e.g. $V_F = 10^{-2}$ to fix the stopping criterion and from R we estimate z_L and z_U e.g. as follows

$$z_L = -1 + nc^- R \quad , \qquad z_U = 1 + nc^+ R\ ,$$

where $c^- = \min\{c_j : 1 \le j \le n\}$ and $c^+ = \max\{c_j : 1 \le j \le n\}$.

"DCS" stands for deep cut, sliding objective, large steps and line search, i.e. all of the devices that we discussed above to speed the empirical rate of convergence of the underlying basic algorithmic idea. For this "practical" version of the ellipsoid algorithm we ignore the blow-up factor $\beta \ge 1$ that is necessary to obtain the theoretical result since $\beta - 1 = 1/12n^2$ is a horribly small positive number for reasonably sized n.

In the DCS ellipsoid algorithm we assume $\boldsymbol{c} \ne \boldsymbol{0}$. If $\boldsymbol{c} = \boldsymbol{0}$ then some modifications and simplifications impose themselves the details of which we leave as an exercise for *you* to figure out.

DCS Ellipsoid Algorithm (m, n, z_L, z_U, ε, R, V_F, $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\overline{\boldsymbol{x}}$, z)

Step 0: Set $k := 0$, $x_j^0 := R/2$ for $1 \le j \le n$, $z := z_L$, $z_0 := z_L$,
$R_0 := \sqrt{n}(1 + R/2)$, $\boldsymbol{H}_0 := R_0 \boldsymbol{I}_n$, $f_0 := \left(1 + \frac{1}{n}\right)^{-\frac{n+1}{2}} \left(1 - \frac{1}{n}\right)^{-\frac{n-1}{2}}$, $V_0 := R_0^n \pi^{n/2}/\Gamma(1 + n/2)$.

Step 1: Set $mxv := b_j + \varepsilon - \boldsymbol{a}^j \boldsymbol{x}^k$ where $b_j - \boldsymbol{a}^j \boldsymbol{x}^k \le b_i - \boldsymbol{a}^i \boldsymbol{x}^k$ for all $1 \le i \le n+m$.
if $mxv < 0$ **go to** Step 2.
Set $\boldsymbol{x}^* := \boldsymbol{x}^k + \boldsymbol{H}_k \boldsymbol{H}_k^T \boldsymbol{c}^T / \|\boldsymbol{c}\boldsymbol{H}_k\|$, $\lambda := \max\{\lambda : \lambda \boldsymbol{a}^i(\boldsymbol{x}^* - \boldsymbol{x}^k) \le b_i + \varepsilon - \boldsymbol{a}^i \boldsymbol{x}^k,\ 1 \le i \le n+m\}$.
if $\lambda \ge 1$ **stop** "(LP$_C$) is unbounded."

if $\boldsymbol{c}(\boldsymbol{x}^k + \lambda(\boldsymbol{x}^* - \boldsymbol{x}^k)) \leq z$ **go to** Step 2.
Set $\overline{\boldsymbol{x}} := \boldsymbol{x}^k + \lambda(\boldsymbol{x}^* - \boldsymbol{x}^k)$, $z := \boldsymbol{c}\overline{\boldsymbol{x}}$.

Step 2: **if** $(\boldsymbol{c}\boldsymbol{x}^k > z$ or $(mxv < 0$ and $z_0 - z > mxv))$ **then**
Choose Θ so that $b_j + \varepsilon \leq \Theta \leq \boldsymbol{a}^j\boldsymbol{x}^k$. Set $\alpha_k := \dfrac{\boldsymbol{a}^j\boldsymbol{x}^k - \Theta}{\|\boldsymbol{a}^j\boldsymbol{H}_k\|}$, $\boldsymbol{r}^T := \boldsymbol{a}^j$.
else
Set $\alpha_k := (z - \boldsymbol{c}\boldsymbol{x}^k)/\|\boldsymbol{c}\boldsymbol{H}_k\|$, $\boldsymbol{r}^T := -\boldsymbol{c}$, $z_0 := z$.
endif

Step 3: **if** $\alpha_k < 1$ and $V_k \geq V_F$ **go to** Step 4.
if $z_L < z < z_U$ **stop** "$\overline{\boldsymbol{x}}$ is an ε-optimal solution to (LP$_C$)."
stop "(LP$_C$) is infeasible or unbounded."

Step 4: Set

$$\boldsymbol{x}^{k+1} := \boldsymbol{x}^k - \frac{1 + n\alpha_k}{(n+1)\|\boldsymbol{H}_k^T\boldsymbol{r}\|}\boldsymbol{H}_k\boldsymbol{H}_k^T\boldsymbol{r} \;, \tag{9.44}$$

$$\boldsymbol{H}_{k+1} := n\sqrt{\frac{1-\alpha_k^2}{n^2-1}}\boldsymbol{H}_k\left(\boldsymbol{I}_n - \left(1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right)\frac{(\boldsymbol{H}_k^T\boldsymbol{r})(\boldsymbol{r}^T\boldsymbol{H}_k)}{\|\boldsymbol{H}_k^T\boldsymbol{r}\|^2}\right) \;, \tag{9.45}$$

$$V_{k+1} := (1-\alpha_k^2)^{\frac{n-1}{2}}(1-\alpha_k)f_0V_k \;. \tag{9.46}$$

Let $I := \{\mu \in \mathbb{R} : \mu\boldsymbol{a}^i(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \leq b_i + \varepsilon - \boldsymbol{a}^i\boldsymbol{x}^k$ for $1 \leq i \leq n+m\}$.
if $I \neq \emptyset$ and $\boldsymbol{c}\boldsymbol{x}^k \neq \boldsymbol{c}\boldsymbol{x}^{k+1}$ **then**
if $\boldsymbol{c}\boldsymbol{x}^{k+1} > \boldsymbol{c}\boldsymbol{x}^k$ **then** set $\overline{\mu} := \max\{\mu : \mu \in I\}$ **else** $\overline{\mu} := \min\{\mu : \mu \in I\}$.
if $|\overline{\mu}| = \infty$ **stop** "(LP$_C$) is unbounded."
if $\boldsymbol{c}\boldsymbol{x}^k + \overline{\mu}(\boldsymbol{c}\boldsymbol{x}^{k+1} - \boldsymbol{c}\boldsymbol{x}^k) > z$ **then** set $\overline{\boldsymbol{x}} := \boldsymbol{x}^k + \overline{\mu}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)$, $z := \boldsymbol{c}\overline{\boldsymbol{x}}$.
endif
Replace $k+1$ by k and **go to** Step 1.

9.4.1 Linear Programming the Ellipsoidal Way: Two Examples

In Figures 9.2, 9.3, 9.4 we show the first nine iterations that result when we use the data of Exercise 8.2 (ii) and minimize x_2 *without* the use of the line search, i.e. we assume in Step 4 that always $I = \emptyset$. In Figures 9.5, 9.6, 9.7 we show the corresponding 12 first iterations when we maximize x_2 *with* line search. To make the corresponding pictures more readable we have depicted every *third* ellipse by a "dashed" curve, whereas all the others are drawn solidly. The first ellipse shown in Figures 9.3, 9.4 and Figures 9.6, 9.7, respectively, are the "last" ellipse of the respective preceding picture. "Dashed" lines correspond to using the original constraints to cut the ellipse in half, while "dotted" lines correspond to cuts using the sliding objective. In Figure 9.3 the arrows show the "large" steps that the algorithm takes, while there are none in Figures 9.5, 9.6, 9.7. Note that in Step 2 we use – like in Step 1 – a "most violated" constraint: a sliding objective cut is executed only if the objective cut is a most violated constraint. Note the different convergence behavior of the algorithm that results from the existence of alternative optima; see the text for a detailed discussion of the computer runs.

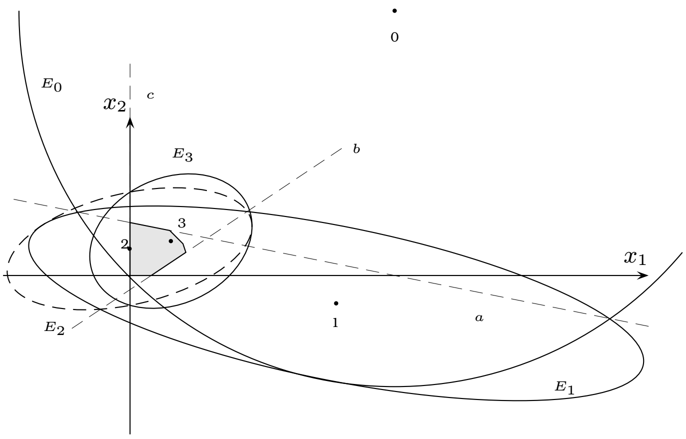


Fig. 9.2. Deep cuts, sliding objective, large steps (minimize x_2)

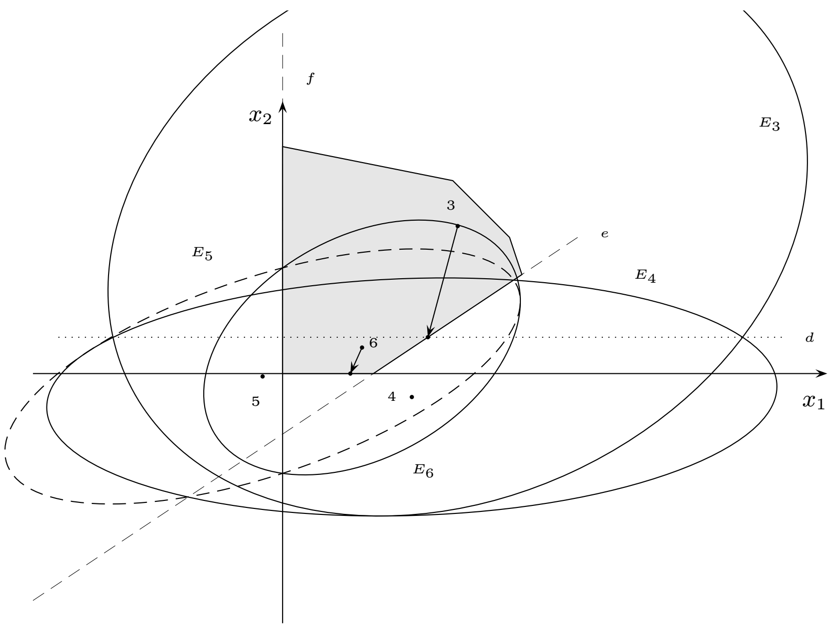


Fig. 9.3. Deep cuts, sliding objective, large steps for iterations $3, \ldots, 6$

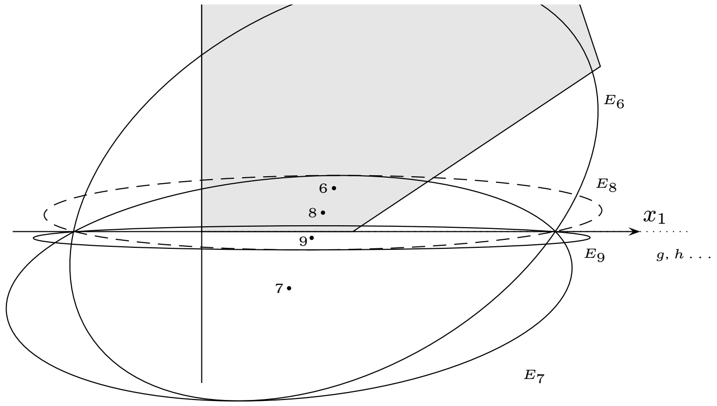


Fig. 9.4. Proving optimality of a face of dimension 1 in $\mathbb{R}^2$ the ellipsoidal way

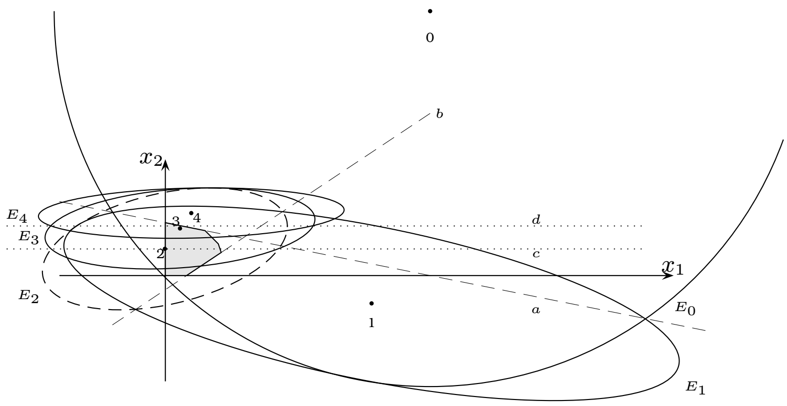


Fig. 9.5. Deep cuts, sliding objective, line search (maximize x_2)

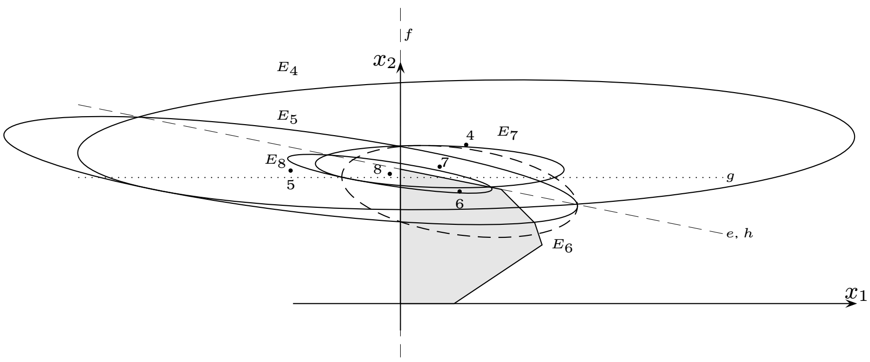


Fig. 9.6. Deep cuts, sliding objective, line search for iterations $4, \ldots, 8$

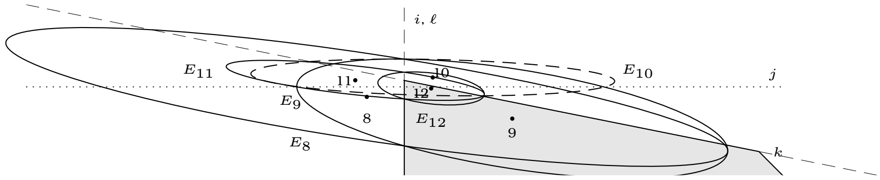


Fig. 9.7. Proving optimality of a face of dimension 0 in $\mathbb{R}^2$ the ellipsoidal way

9.4.2 Correctness and Finiteness of the DCS Ellipsoid Algorithm

By assumption the parameter R is a common upper bound on the variables of (LP$_C$) that is large enough so that the hypercube

$$\{\boldsymbol{x} \in \mathbb{R}^n : 0 \leq x_j \leq R \text{ for } 1 \leq j \leq n\}$$

contains all of the feasible set of (LP$_C$) if (LP$_C$) is bounded and enough of the unbounded portion of the feasible set to permit us to conclude unboundedness via the value of the objective function; see point 7.5(d) and the discussion of binary search in Chapter 9.3. We start the algorithm at the center $\boldsymbol{x}^0 = \frac{1}{2}R\boldsymbol{e}$ of this hypercube and by choice of R_0 in Step 0 the initial ball $B(\boldsymbol{x}^0, R_0)$ does the job.

The validity of the DCS ellipsoid algorithm is established inductively like in Chapter 9.2. Using formula (9.45) for the update $\boldsymbol{H}_{k+1}$ we compute its determinant in terms of the determinant $\boldsymbol{H}_k$

$$det\boldsymbol{H}_{k+1} = \left(1 + \frac{1}{n}\right)^{-\frac{n+1}{2}} \left(1 - \frac{1}{n}\right)^{-\frac{n-1}{2}} (1 - \alpha_k^2)^{\frac{n-1}{2}} (1 - \alpha_k)\, det\boldsymbol{H}_k \ . \tag{9.47}$$

To establish the containment of the feasible set in the updated ellipsoid, we form the positive definite matrix $\boldsymbol{G}_{k+1} = \boldsymbol{H}_{k+1}\boldsymbol{H}_{k+1}^T$ and compute its inverse

$$\boldsymbol{G}_{k+1}^{-1} = \frac{n^2 - 1}{n^2(1 - \alpha_k^2)} \left(\boldsymbol{G}_k^{-1} + \frac{2(1 + n\alpha_k)}{(n-1)(1 - \alpha_k)} \frac{\boldsymbol{r}\boldsymbol{r}^T}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}\right) \ ; \tag{9.48}$$

see Exercise 9.6. Let $E_k = E_k(\boldsymbol{x}^k, 1)$ be the ellipsoid that the DCS algorithm constructs at iteration k

$$E_k(\boldsymbol{x}^k, 1) = \{\boldsymbol{x} \in \mathbb{R}^n : (\boldsymbol{x} - \boldsymbol{x}^k)^T \boldsymbol{G}_k^{-1} (\boldsymbol{x} - \boldsymbol{x}^k) \leq 1\} \ . \tag{9.49}$$

It follows using (7.23) from (9.47) that

$$\frac{vol(E_{k+1})}{vol(E_k)} = \left(\frac{1 - \alpha_k}{1 + 1/n}\right)^{\frac{n+1}{2}} \left(\frac{1 + \alpha_k}{1 - 1/n}\right)^{\frac{n-1}{2}} \leq e^{-\alpha_k - \frac{1}{2n}}$$

for all $k \geq 0$. Setting $V_{k+1} = vol(E_{k+1})$ and $f_0 = (1 - 1/n)^{-\frac{n+1}{2}}(1 + 1/n)^{-\frac{n-1}{2}}$ we get

$$V_{k+1} = V_0 f_0 \prod_{\ell=0}^{k} (1 - \alpha_\ell^2)^{\frac{n-1}{2}} (1 - \alpha_\ell) \ , \tag{9.50}$$

which shows that the DCS algorithm updates the volume of the current ellipsoid correctly in formula (9.46). Moreover, it shows the "deflating" effect of the deep cuts on the volume of the ellipsoid quite clearly.

It is shown in the text that the DCS ellipsoid algorithm is correct if the bound R is "large enough" and the perturbation ε is "small enough". From (9.50) and the formula for the ratio of the volumina it follows that the stopping criterion $V_k < V_F$ is satisfied after at most

$$\lceil 2n \log \frac{V_0}{V_F} \rceil$$

iterations. The α_ℓ's introduce a data-dependency into the stopping criterion that does, however, not change the theoretical worst-case behavior of the algorithm.

9.5 Optimal Separators, Most Violated Separators, Separation

Ἥξεις ἀφίξεις οὐ θνήξεις ἐν πολέμῳ. [2]
Pythia, High priestess of Delphi.

Throughout the rest of this chapter we will deal mostly with rational polytopes $P \subseteq \mathbb{R}^n$ rather than with polyhedra. From Chapter 7.5 we know that we can always do so while preserving polynomiality of the facet complexity and vertex complexity in terms of the original parameters. We are interested in the linear optimization problem $\max\{\boldsymbol{cx} : \boldsymbol{x} \in P\}$ where the vector $\boldsymbol{c}$ is some row vector with rational coefficients and P has a linear description with possibly exponentially many linear inequalities. To approach this problem we start with a *partial* linear description of the rational polytope P having $\mathcal{O}(n)$ constraints, which gives us a larger polytope P_0 that contains P. Solving the linear program

$$\max\{\boldsymbol{cx} : \boldsymbol{x} \in P_0\}$$

we either conclude that P_0 and thus P is empty or we get an optimal solution $\boldsymbol{x}^0 \in P_0$, e.g. an optimal extreme point of P_0. Now we check the constraint set of P by *some* "separation algorithm" - other than LIST-and-CHECK – to find a violated constraint, i.e. we solve something like the separation problem of Chapter 7.5.4 *algorithmically*. If the separation problem does not produce a violated constraint, then $\boldsymbol{x}^0$ is an optimal solution to the problem $\max\{\boldsymbol{cx} : \boldsymbol{x} \in P\}$ – see the *outer inclusion principle* of Chapter 7.5.4 which works with a "local" description of P in the neighborhood of a maximizer $\boldsymbol{x}^{max}$, say, of $\boldsymbol{cx}$ over P rather than the *complete* linear description of P.

To stress the point that we wish to make once again, suppose that the feasible set of our linear program is given by a *convex polygon* in $\mathbb{R}^2$ with, say, $10^{10^{10}}$ or more "corners" and just as many facets: all you need are at most **three** of its facet defining constraints to *prove* optimality of some corner that maximizes your *linear* function in $\mathbb{R}^2$, see Figure 9.8. The problem is to find the "right" ones and almost nothing else matters.

Suppose that the separation algorithm finds a constraint $\boldsymbol{h}^1\boldsymbol{x} \le h_0^1$, say, such that $\boldsymbol{h}^1\boldsymbol{x}^0 > h_0^1$. Then

$$P \subseteq P_1 = P_0 \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{h}^1\boldsymbol{x} \le h_0^1\} \subset P_0$$

[2] "You will depart, you will arrive, you will not die in the war" or is it "you will depart, you will not arrive, you will die in the war"?

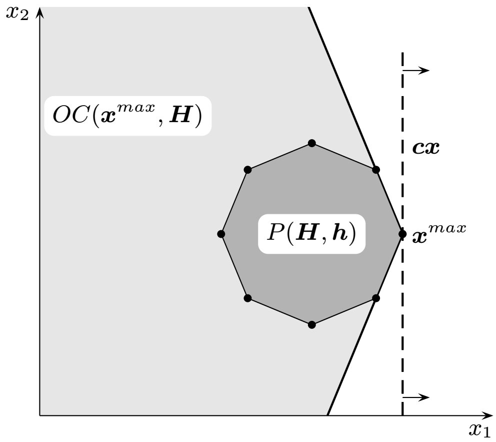


Fig. 9.8. The outer inclusion principle in $\mathbb{R}^2$

and we can iterate. The question: does this iterative scheme converge "fast enough" to permit linear optimization over any rational polytope in polynomial time?

This is where the (basic) ellipsoid algorithm enters: neither its running time T nor the required precision p, see e.g. (9.34), depend on the number m of the constraints of the linear program. So if we can find a constraint defining P that is violated by the *current* iterate $\boldsymbol{x}^k$ or prove that no such constraint exists in time that is polynomial in n, ϕ, $\langle \boldsymbol{x}^k \rangle$ and $\langle \boldsymbol{c} \rangle$, then the polynomiality of the entire iterative scheme follows from the polynomiality of the ellipsoid algorithm.

Let us discuss first *what kind* of a violated constraint we wish to find *ideally* to obtain "fast" convergence of this iterative scheme. We shall forget what other authors have called "the separation problem" and first determine what it is that we really want.

Denote by P_k the polytope that we have after k iterations and by $\boldsymbol{x}^k$ the current optimizer. Denote by

$$SP = \{(\boldsymbol{h}, h_0) \in \mathbb{R}^{n+1} : P \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{hx} \leq h_0\}\} \tag{9.51}$$

the set of all candidates for a solution of the separation problem e.g. as defined in Chapter 7.5.4. The set SP is the h_0-polar of the polytope P, see (7.13) in Chapter 7.4 where $\boldsymbol{Y}$ is void because P by assumption is a polytope. Ideally, we wish to find a constraint that "moves" the objective function "down" as fast as possible because we know that $P \subseteq P_k$. So we want ideally a solution to the problem

$$\min_{(\boldsymbol{h}, h_0) \in SP} \max_{\boldsymbol{x} \in \mathbb{R}^n} \{\boldsymbol{cx} : \boldsymbol{x} \in P_k \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{hx} \leq h_0\}\} \ . \tag{9.52}$$

Since P is a polytope this min-max problem has a solution if $P \neq \emptyset$ and if $P = \emptyset$ we simply declare an arbitrary "violated" inequality to be the solution, e.g. $\boldsymbol{hx} = \boldsymbol{cx} \leq h_0 = \boldsymbol{cx}^k - 100$. If the objective function value of (9.52) is greater than or equal to $\boldsymbol{cx}^k$, then by the outer inclusion principle the current iterate $\boldsymbol{x}^k$ solves the linear optimization problem $\max\{\boldsymbol{cx} : \boldsymbol{x} \in P\}$ and we are done. Otherwise, the objective function value is less than $\boldsymbol{cx}^k$ and let us call any $(\boldsymbol{h}, h_0) \in \mathbb{R}^{n+1}$ that solves (9.52) an **optimal separator** for P with respect to the objective function $\boldsymbol{cx}$

that we wish to maximize over P. It follows that $\boldsymbol{hx}^k > h_0$ and we can iterate. What we like to have ideally is not necessarily what we can do in computational practice and indeed, we are not aware of any linear optimization problem for which a solution to the min-max problem (9.52) is known. We refer to this problem sometimes as the problem of "finding the right cut" because we are evidently cutting off a portion of the polytope P_k by an optimal separator $\boldsymbol{hx} \leq h_0$ and the cut is "right" because it moves the objective function value as much as possible. A general solution to (9.52) does not seem possible, but for certain *classes* of optimization problems an answer to this problem may be possible.

Since a solution to (9.52) seems elusive, we have to scale down our aspirations somewhat and approximate the problem of finding the right cut. The next best objective that comes to one's mind is to ask for $(\boldsymbol{h}, h_0) \in SP$ such that the amount of *violation* $\boldsymbol{hx}^k - h_0$ is maximal. So we want to solve

$$\max\{\boldsymbol{hx}^k - h_0 : (\boldsymbol{h}, h_0) \in SP\,,\ \|\boldsymbol{h}\|_\infty = 1\}\ , \tag{9.53}$$

where $\|\boldsymbol{h}\|_\infty = \max\{|h_j| : 1 \leq j \leq n\}$ is the ℓ_∞-norm. Because we normalized by $\|\boldsymbol{h}\|_\infty = 1$ the maximum in (9.53) exists if $P \neq \emptyset$. If $P = \emptyset$, then any inequality $\boldsymbol{hx} \leq h_0$ with $\boldsymbol{hx}^k - h_0 > 0$ is simply declared to be a "solution" to (9.53). If the objective function value in (9.53) is less than or equal to zero, we stop: the current iterate $\boldsymbol{x}^k$ maximizes $\boldsymbol{cx}$ over P. Otherwise, any solution to (9.53) is

a **most violated separator** for P.

It follows that $\boldsymbol{hx}^k > h_0$ and we can iterate. Indeed, in all of the computational work that preceded as well as that followed the advent of the ellipsoid algorithm the constraint identification (or separation) problem was approached in this and no other way. Posing the separation problem that we need to solve iteratively the way we have done it *solves* the separation problem of Chapter 7.5.4: the objective function value of (9.53) provides a "proof" that all constraints of P are satisfied by the current iterate $\boldsymbol{x}^k$ of the overall iterative scheme. Of course, problem (9.53) has one nonlinear constraint, but we can get around the nonlinearity easily.

We consider in the **separation step** of the basic ellipsoid algorithm, i.e. in Step 1, only *most violated* separators. This agrees not only with computational practice, but it also alleviates certain theoretical difficulties that arise when the separation step is not treated as an optimization problem on its own.

If one merely asks for *some* separator as we do in the first part of point 7.5(j), then – in case that $dimP = k < n$ – it can happen that the separation subroutine always returns a hyperplane that is *parallel* to the affine hull of P, see e.g. Figure 9.4, and, of course, the volumina of the ellipsoids tend to zero. Thus if the volume falls below a certain value V_F, say, we can no longer conclude that $P = \emptyset$ because – even though $P \neq \emptyset$ – the n-dimensional volume of P is zero if $dimP < n$.

9.6 ε-Solidification of Flats, Polytopal Norms, Rounding

For any $\varepsilon > 0$ define the ε-**solidification** P_ε^∞ of a polytope $P \subseteq \mathbb{R}^n$ with respect to the ℓ_∞-norm by

$$P_\varepsilon^\infty = \{\boldsymbol{z} \in \mathbb{R}^n : \exists\, \boldsymbol{x} \in P \textit{ such that } \|\boldsymbol{x} - \boldsymbol{z}\|_\infty \leq \varepsilon\}\ , \tag{9.54}$$

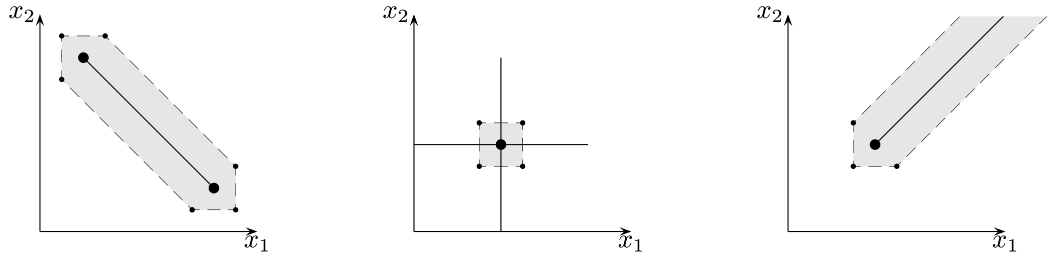


Fig. 9.9. ε-Solidification (9.54) with $\varepsilon = 0.5$ of three rational flats in $\mathbb{R}^2$

where $\|\boldsymbol{x} - \boldsymbol{z}\|_\infty = \max\{|x_j - z_j| : 1 \le j \le n\}$ is the ℓ_∞-norm.

Remark 9.12 *(i) For every $\varepsilon > 0$ and nonempty polytope $P \subseteq \mathbb{R}^n$ the set $P_\varepsilon^\infty \subseteq \mathbb{R}^n$ is a full dimensional polytope.*
(ii) If $\boldsymbol{hx} \le h_0$ for all $\boldsymbol{x} \in P$, then $\boldsymbol{hx} \le h_0 + \varepsilon \|\boldsymbol{h}\|_1$ for all $\boldsymbol{x} \in P_\varepsilon^\infty$. If $\boldsymbol{hx} \le h_0$ for all $\boldsymbol{x} \in P_\varepsilon^\infty$, then $\boldsymbol{hx} \le h_0 - \varepsilon \|\boldsymbol{h}\|_1$ for all $\boldsymbol{x} \in P$, where $\|\boldsymbol{h}\|_1 = \sum_{j=1}^n |h_j|$ is the ℓ_1-norm.
(iii) Let $\boldsymbol{Hx} \le \boldsymbol{h}$ be any linear description of P_ε^∞. Then $\boldsymbol{Hx} \le \boldsymbol{h} - \varepsilon\, \boldsymbol{d}$ is a linear description of P where $\boldsymbol{d}$ is the vector of the ℓ_1-norms of the rows of $\boldsymbol{H}$.

Suppose P is a rational flat. Then $P \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{hx} = h_0\}$ for some $(\boldsymbol{h}, h_0) \in \mathbb{R}^{n+1}$. Consequently, $\boldsymbol{hx} \le h_0$ and $-\boldsymbol{hx} \le -h_0$ for all $\boldsymbol{x} \in P$. Thus

$$P_\varepsilon^\infty \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{hx} \le h_0 + \varepsilon \|\boldsymbol{h}\|_1 \,,\; -\boldsymbol{hx} \le -h_0 + \varepsilon \|\boldsymbol{h}\|_1\} \;,$$

which corresponds to "tearing apart" the equations defining the affine hull of P. Since for every $\boldsymbol{x} \in P$ the hypercube $\boldsymbol{x} + \{\boldsymbol{z} \in \mathbb{R}^n : |z_j| \le \varepsilon \text{ for } 1 \le j \le n\}$ is contained in P_ε^∞ and this hypercube contains $B(\boldsymbol{x}, r = \varepsilon)$ the n-dimensional volume of P_ε^∞ satisfies

$$\textit{if } P \ne \emptyset \textit{ then } vol(P_\varepsilon^\infty) \ge 2^n \varepsilon^n > \frac{\varepsilon^n \pi^{n/2}}{\Gamma(1 + n/2)} \textit{ for all } \varepsilon > 0 \;. \tag{9.55}$$

Since $dimP_\varepsilon^\infty = n$ if $P \ne \emptyset$ the polytope P_ε^∞ has a linear description that is unique *modulo* multiplication by positive scalars. Thus if $\boldsymbol{Hx} \le \boldsymbol{h}$ is *some* linear description of P then the set $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{Hx} \le \boldsymbol{h} + \varepsilon\, \boldsymbol{d}\}$ contains P_ε^∞ by Remark 9.12 (ii), where $\boldsymbol{d}$ is defined in part (iii). But the containment can be proper.

Let $\boldsymbol{x}^* \in \mathbb{R}^n$ be arbitrary and suppose that we have a **separation subroutine** for the polytope P that finds a most violated separator $\boldsymbol{hx} \le h_0$. If $\boldsymbol{hx}^* \le h_0$, then $\boldsymbol{x}^* \in P_\varepsilon^\infty$ where $\varepsilon > 0$ is arbitrary. From Remark 9.12 it follows that for a point outside of P_ε^∞ there exists at least one of the representations of some facet of P that is violated by it; see the text for more detail.

Remark 9.13 *Let $\boldsymbol{Hx} \le \boldsymbol{h}$ and $\phi \ge n + 1$ be such that $\langle \boldsymbol{h}^i \rangle + \langle h_i \rangle \le \phi$ for all rows $(\boldsymbol{h}^i, h_i)$ of $(\boldsymbol{H}\ \boldsymbol{h})$. Let $\boldsymbol{x}^a \in \mathbb{R}^n$ and $(\boldsymbol{H}_1\ \boldsymbol{h}^1)$, $(\boldsymbol{H}_2\ \boldsymbol{h}^2)$ be a partitioning of $(\boldsymbol{H}\ \boldsymbol{h})$ such that*

$$\boldsymbol{h}^1 - \varepsilon \boldsymbol{d}^1 \le \boldsymbol{H}_1 \boldsymbol{x}^a \le \boldsymbol{h}^1 + \varepsilon \boldsymbol{d}^1 \;, \quad \boldsymbol{H}_2 \boldsymbol{x}^a < \boldsymbol{h}^2 - \varepsilon \boldsymbol{d}^2 \;, \tag{9.56}$$

where $\boldsymbol{d}^1$, $\boldsymbol{d}^2$ are the vectors of the ℓ_1-norms of the corresponding rows of $\boldsymbol{H}_1$, $\boldsymbol{H}_2$. If $0 \le \varepsilon \le 2^{-5(n+1)\phi}$ then the system $\boldsymbol{H}_1 \boldsymbol{x} = \boldsymbol{h}^1$, $\boldsymbol{H}_2 \boldsymbol{x} \le \boldsymbol{h}^2$ is solvable.

In Figure 9.9 we have illustrated the ε-solidification of three flats in $\mathbb{R}^2$. Exercise 9.7 shows that solidification works for polyhedra and states various facts about P_ε^∞. It shows also that we can replace the ℓ_∞-norm in the definition of the ε-solidification of a polyhedron P by the ℓ_1-norm without changing the basic properties.

The ℓ_1-norm and the ℓ_∞-norm are special **polytopal norms** on $\mathbb{R}^n$ in the sense that their respective "unit spheres" $\{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_p \leq 1\}$ where $p \in \{1, \infty\}$ are full dimensional polytopes in $\mathbb{R}^n$. Moreover, the polytopes $\{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_1 \leq 1\}$ and $\{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_\infty \leq 1\}$ are **dual polytopes** in the sense that there is an one-to-one correspondence between the facets of either of them and the extreme points of the other. More generally, let $\|\cdot\|_P$ be any polytopal norm on $\mathbb{R}^n$, i.e. the unit sphere with respect to $\|\cdot\|_P$ is a polytope. Given $\|\cdot\|_P$ define for $\boldsymbol{y} \in \mathbb{R}^n$ the "length" of $\boldsymbol{y}$ in the **dual norm** $\|\cdot\|_P^*$ by

$$\|\boldsymbol{y}\|_P^* = \max\{\boldsymbol{y}^T\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^n\,,\ \|\boldsymbol{x}\|_P \leq 1\}\ . \tag{9.57}$$

The maximum exists because $\{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_P \leq 1\}$ is a polytope, i.e. a compact convex subset of $\mathbb{R}^n$, and $\boldsymbol{y}^T\boldsymbol{x}$ is continuous in $\boldsymbol{x}$. *You* prove that $\|\cdot\|_1$ and $\|\cdot\|_\infty$ is a pair of dual norms. Exercise 9.8 shows that $\|\cdot\|_P^*$ is a polytopal norm on $\mathbb{R}^n$ and that it satisfies **Hölder's inequality**

$$\boldsymbol{y}^T\boldsymbol{x} \leq \|\boldsymbol{y}\|_P^*\|\boldsymbol{x}\|_P \text{ for all } \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n\ . \tag{9.58}$$

It also shows that an ε-solidification of polyhedra can be defined with respect to any polytopal norm on $\mathbb{R}^n$. Moreover, if we have a separation subroutine for any polyhedron P that finds a most violated separator then we can separate points from it using the corresponding ε-solidification of P.

9.6.1 Rational Rounding and Continued Fractions

Rational rounding is the process of approximating a real number $\Theta \in \mathbb{R}$ as a ratio of two integer numbers p/q with $q \geq 1$. If the denominator $q = 1$ then $p = \lfloor\Theta\rfloor$ or $p = \lceil\Theta\rceil$ are the only "reasonable" choices with a maximum error of $1/2$. We denote by

$$[\Theta] = \min\{\Theta - \lfloor\Theta\rfloor, \lceil\Theta\rceil - \Theta\} \tag{9.59}$$

the smaller of the two fractional parts of Θ and pronounce $[\Theta]$ as "frac of Θ."

Remark 9.14 *Let Θ and $Q > 1$ be any real numbers. Then there exists an integer q such that $1 \leq q \leq Q$ and $[q\Theta] \leq Q^{-1}$.*

A ratio p/D with integers p and $D \geq 1$ is a **best approximation** to $\Theta \in \mathbb{R}$ if

$$[D\Theta] = |D\Theta - p| \textit{ and } [q\Theta] > [D\Theta] \textit{ for all } 1 \leq q < D\ . \tag{9.60}$$

Two sequences of integers $q_1 = 1 < q_2 < q_3 < \cdots$ and $p_1, p_2, p_3, \ldots$ such that p_n/q_n is a best approximation to Θ for all $1 \leq q \leq D = q_n$ can be constructed inductively as follows. Initially, we let $q_1 = 1$ and find an integer p_1 such that $|q_1\Theta - p_1| = [\Theta] \leq 1/2$. By definition p_1/q_1 is a best approximation of Θ with $D = 1$. If $p_1 = q_1\Theta$, i.e. if Θ is an integer, the inductive process stops. So suppose that we have constructed the $n \geq 1$ first, best approximations to Θ and that the process does not stop. Then we have integers p_n and q_n such that $p_n \neq q_n\Theta$, i.e. $[q_n\Theta] > 0$. From Remark 9.14 when applied with $Q > [q_n\Theta]^{-1}$ it follows that there exist integer numbers $q \geq 1$ such that $[q\Theta] < [q_n\Theta]$. Let q_{n+1} be the *smallest* integer and p_{n+1} be a corresponding integer number such that

$$[q_{n+1}\Theta] = |q_{n+1}\Theta - p_{n+1}| \tag{9.61}$$
$$[q_{n+1}\Theta] < [q_n\Theta] \tag{9.62}$$
$$[q\Theta] \geq [q_n\Theta] \text{ for all } 1 \leq q < q_{n+1}. \tag{9.63}$$

Since by the inductive hypothesis p_n/q_n is a best approximation to Θ with $D = q_n$ we have from (9.62) that $q_{n+1} > q_n$ and from (9.63) and (9.62) that the ratio p_{n+1}/q_{n+1} is a best approximation to Θ with $D = q_{n+1}$. Consequently, the induction works, the numbers p_n and q_n have the stated properties and either the process continues or it stops.

Several important properties of this sequence of best approximations to Θ can be established that lead to a polynomially bounded algorithm for finding best approximations to rational numbers when the denominator is bounded by a prescribed number. Since $q_n < q_{n+1}$ and applying Remark 9.14 with $Q = q_{n+1}$ it follows from (9.63) that

$$q_n[q_n\Theta] < q_{n+1}[q_n\Theta] \leq 1 \ . \tag{9.64}$$

Moreover, the signs of $q_n\Theta - p_n$ and $q_{n+1}\Theta - p_{n+1}$ alternate, i.e.

$$(q_n\Theta - p_n)(q_{n+1}\Theta - p_{n+1}) \leq 0 \ . \tag{9.65}$$

All numbers in the sequence are integers and

$$p_nq_{n+1} - p_{n+1}q_n = q_n(q_{n+1}\Theta - p_{n+1}) - q_{n+1}(q_n\Theta - p_n) \ , \tag{9.66}$$
$$p_nq_{n+1} - p_{n+1}q_n = \pm 1 \ , \tag{9.67}$$
$$\text{sign}(p_nq_{n+1} - p_{n+1}q_n) = -\text{sign}(q_n\Theta - p_n) \tag{9.68}$$
$$p_nq_{n+1} - p_{n+1}q_n = -(p_{n-1}q_n - p_nq_{n-1}) \tag{9.69}$$

for all p_n, q_n that the inductive process generates. From (9.69) $p_n(q_{n+1} - q_{n-1}) = q_n(p_{n+1} - p_{n-1})$. From (9.67) $g.c.d.(p_n, q_n) = 1$ and since the coprime representation of a rational is unique, there exists some positive integer a_n such that $q_{n+1} - q_{n-1} = a_nq_n$ and $p_{n+1} - p_{n-1} = a_np_n$. So for all $n \geq 2$ of the inductive process there exist integers $a_n \geq 1$ such that

$$q_{n+1} = a_nq_n + q_{n-1} \ , \qquad p_{n+1} = a_np_n + p_{n-1} \ . \tag{9.70}$$

Multiply the first part of (9.70) by Θ, subtract the second and use the alternating signs (9.65). We get

$$|q_{n-1}\Theta - p_{n-1}| = a_n|q_n\Theta - p_n| + |q_{n+1}\Theta - p_{n+1}| \ . \tag{9.71}$$

Now $|q_{n+1}\Theta - p_{n+1}| = [q_{n+1}\Theta] < [q_n\Theta] = |q_n\Theta - p_n|$ by (9.62). Hence

$$a_n = \left\lfloor \frac{|q_{n-1}\Theta - p_{n-1}|}{|q_n\Theta - p_n|} \right\rfloor \ , \tag{9.72}$$

which together with (9.70) gives a procedure to calculate p_{n+1}, q_{n+1} once the values of p_n and q_n are known for $k \leq n$ where $n \geq 2$.

To start the procedure to calculate p_n and q_n iteratively from (9.70) and (9.72) we have to know what the first two iterations of the inductive process produce in terms of p_n and q_n or prescribe a start that is consistent with the inductive hypothesis that we have made to derive the above properties.

If $|\Theta| \geq 1$ and Θ is not an integer, then we can always write $\Theta = \lfloor \Theta \rfloor + \Theta'$ with $0 < \Theta' < 1$ and a best approximation to Θ' yields instantaneously a best approximation to Θ. So we can assume WROG that $0 < \Theta < 1$. If $0 < \Theta \leq 1/2$ then $p_1 = 0$, $q_1 = 1$ yields a best approximation to Θ for $D = 1$ and by Exercise 9.9 (ii) $p_2 = 1$, $q_2 = \lfloor \Theta^{-1} \rfloor$ does the same for $D = \lfloor \Theta^{-1} \rfloor$. So if we initialize

$$p_0 = 1\,,\ q_0 = 0\,,\ p_1 = 0\,,\ q_1 = 1\,, \tag{9.73}$$

then formulas (9.70) and (9.72) produce precisely the respective best approximations to Θ for $n \leq 2$, the inductive hypothesis applies and so we can continue to use the formulas until the process stops, i.e. $\Theta = p_{n+1}/q_{n+1}$, if it stops at all. If $1/2 < \Theta < 1$ then the initialization (9.73) produces $p_1 = 0$, $q_1 = 1$ which is, of course, not a best approximation to $\Theta > 1/2$. Calculating we get from (9.70) and (9.72) $p_2 = 1$, $q_2 = 1$ because $a_1 = 1$, which gives a best approximation of $\Theta > 1/2$ for $D = 1$. Carrying out one more step in the iterative application of (9.70) and (9.72) with the initialization (9.73) we get $a_2 = \lfloor \frac{\Theta}{1-\Theta} \rfloor$, $p_3 = a_2$ and $q_3 = a_2 + 1$. By Exercise 9.9 (iii) p_3/q_3 is a best approximation to Θ for all $1 \leq q < D = \lfloor \frac{\Theta}{1-\Theta} \rfloor + 1$. Now the inductive hypothesis applies to p_2, q_2 and p_3, q_3, we ignore the first iteration and thus we can continue to use the formulas like in the first case.

If the number Θ *equals* r/s with integers $r, s \geq 1$ and $g.c.d(r, s) = 1$, then r/s is itself a best approximation to Θ for all $1 \leq q < s$, the q_n are strictly increasing and thus $q_n = s$, $p_n = r$ at some point and the process stops. If Θ is irrational then $\lim_{n\to\infty} \frac{p_n}{q_n} = \Theta$ because by (9.64) we have $|\Theta - p_n/q_n| < q_n^{-2}$ and $1 = q_1 \leq q_2 < q_3 < \cdots$ for any $0 < \Theta < 1$. With the initialization (9.73) it follows from (9.65) and (9.67) that for all $n \geq 0$

$$(-1)^{n+1}(q_n\Theta - p_n) \geq 0\,, \qquad p_n q_{n+1} - p_{n+1} q_n = (-1)^n\ . \tag{9.74}$$

Consider now the **best approximation problem** for $\Theta \in \mathbb{R}$ relative to a prescribed integer number $D \geq 2$: *we wish to find integer numbers p and $1 \leq q \leq D$ such that $|\Theta - p/q|$ is as small as possible.*

Best Approximation Algorithm (Θ, D)

Step 0: Set $a_0 := \lfloor \Theta \rfloor$, $\Theta := \Theta - a_0$, $p_0 := 1$, $q_0 := 0$, $p_1 := 0$, $q_1 := 1$, $n := 1$.
Step 1: **if** $q_n\Theta = p_n$ **stop** "p_n/q_n is a best approximation".
if $q_n > D$ **go to** Step 3. Set $a_n := \left\lfloor \dfrac{|q_{n-1}\Theta - p_{n-1}|}{|q_n\Theta - p_n|} \right\rfloor$.
Step 2: Set $p_{n+1} := a_n p_n + p_{n-1}$, $q_{n+1} := a_n q_n + q_{n-1}$. Replace $n+1$ by n, **go to** Step 1.
Step 3: Set $k := \lfloor \frac{D - q_{n-1}}{q_n} \rfloor$, $p'_n := p_{n-1} + k p_n$, $q'_n := q_{n-1} + k q_n$.
if $|\Theta - p_n/q_n| \leq |\Theta - p'_n/q'_n|$ **stop** "p_n/q_n is a best approximation".
stop "p'_n/q'_n is a best approximation".

Remark 9.15 (Correctness and finiteness) *For rational Θ and integer $D \geq 2$ the best approximation algorithm's run time is polynomial in the digital size of its input.*

$$\Theta = \Theta_1 = \frac{1}{a_1 + \Theta_2} = \frac{1}{a_1 + \frac{1}{a_2 + \Theta_3}} = \cdots = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots \, a_{N-1} + \cfrac{1}{a_N}}}}$$

Fig. 9.10. Continued fractions for a rational number

To relate the preceding to the **continued fraction** process define

$$\Theta_n = \frac{|q_n\Theta - p_n|}{|q_{n-1}\Theta - p_{n-1}|} \text{ for } n \geq 1 \ ,$$

where we assume again like in the algorithm that the integer part of the original data has been cleared away, i.e. $0 < \Theta < 1$. It follows from the initialization (9.73) that $\Theta_1 = \Theta$ and $0 \leq \Theta_n < 1$ for $n \geq 1$ from (9.62). From (9.71) we get

$$\Theta_n^{-1} = a_n + \Theta_{n+1} \textit{ for all } n \geq 1 \textit{ with } \Theta_n > 0 \ . \qquad (9.75)$$

Now suppose that Θ is a rational number. Then by the above $\Theta = p_{N+1}/q_{N+1}$, say, so that $\Theta_{N+1} = 0$ and thus by (9.75) $\Theta_N^{-1} = a_N$. Consequently we can write Θ like in Figure 9.10, which explains the term "continued fraction." If Θ is irrational then $\Theta_n > 0$ for all $n \geq 1$ and the continued fraction goes on "forever", which permits one to find high-precision rational approximations of irrational numbers.

Rational rounding can e.g. be used in the context of establishing the polynomiality of linear programming via the combination of the binary search algorithm 2 and the basic ellipsoid algorithm, see Chapter 9.3.1, to find the optimal objective function value z_P exactly; see the text.

9.7 Optimization and Separation

Here the fundamental *polynomial-time equivalence* of optimization and separation for rational polyhedra is established. It is shown that for any rational polyhedron $P \subseteq \mathbb{R}^n$ the linear optimization problem

$$\max\{\boldsymbol{cx} : \boldsymbol{x} \in P\}$$

can be solved in time that is polynomial in n, the facet complexity ϕ of P and $\langle \boldsymbol{c} \rangle$ if and only if the separation problem (9.53) is solvable in time that is polynomial in n, ϕ and $\langle \boldsymbol{x}^k \rangle$.

The geometric idea of our construction is simple and illustrated in Figures 9.11 and 9.12 for a full dimensional polyhedron in $\mathbb{R}^2$. In Figure 9.11 we depict the situation when $\boldsymbol{cx}$ with $\boldsymbol{c} = (-1, -1)$ is maximized, while Figure 9.12 illustrates the basic idea for finding direction vectors in the asymptotic cone of a polyhedron. For any integer $u \geq 1$ denote

$$P_u = P \cap \{\boldsymbol{x} \in \mathbb{R}^n : -u \leq x_j \leq u \text{ for } 1 \leq j \leq n\} \, . \qquad (9.76)$$

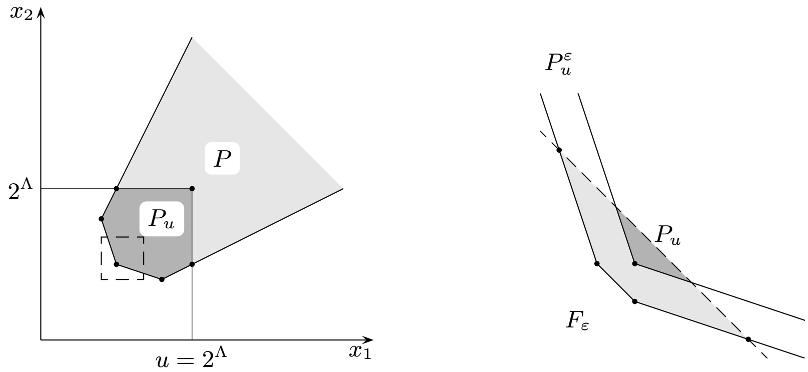


Fig. 9.11. Locating the optimum and proving optimality

Its ε-solidification P_u^ε with respect to the ℓ_1-norm is either empty or a full dimensional polytope in $\mathbb{R}^n$. To find direction vectors in the asymptotic cone we define

$$\Lambda = \phi + 5n\phi + 4n^2\phi + 1 \ . \tag{9.77}$$

Setting $u = 2^\Lambda > 2^{4n\phi}$ it follows from point 7.5(b) that all extreme points of P are properly contained in P_u. Moreover, we get a sufficiently large portion of the "unbounded" region of P by *doubling* this value of u which permits us to find direction vectors in the asymptotic cone of P by solving *two* linear optimization problems rather than one.

At the outset the polyhedron $P \subseteq \mathbb{R}^n$ may be empty, it may have none or several optimal extreme points or the objective function may be unbounded. To encompass all possibilities, we perturb the original objective function so as to achieve uniqueness of the maximizer over the *larger* polytope P_u where $u = 2^{\Lambda+1}$. To locate the unique maximizer $\boldsymbol{x}^{max}$ of the perturbed objective function over P_u where $u \in \{2^\Lambda, 2^{\Lambda+1}\}$ we let the algorithm run until $\boldsymbol{x}^{max}$ is the only *rational* point in the remaining ε-optimal set with components that have denominators $q_j \geq 1$ and $q_j \leq 2^{6n\phi}$. We have illustrated the basic idea in the second part of Figure 9.11 which "zooms" into the area of the first part depicted by a square and that contains $\boldsymbol{x}^{max}$. Running the algorithm "long enough" we find an ε-optimal rational vector $\overline{\boldsymbol{x}} \in P_u^\varepsilon$ "close" to $\boldsymbol{x}^{max}$. Using the best approximation algorithm of Chapter 9.6 we can round $\overline{\boldsymbol{x}}$ *componentwise* to obtain $\boldsymbol{x}^{max}$, see Remark 9.16. If the maximizer $\boldsymbol{x}^{max}$ obtained this way satisfies $|x_j^{max}| < 2^\Lambda$ for all $1 \leq j \leq n$, then we are done and an optimal extreme point of P for the original objective function has been located. If the maximizer $\boldsymbol{x}^{max}$ satisfies $|x_j^{max}| = 2^\Lambda$ for some $1 \leq j \leq n$, then either the linear optimization problem over P has an unbounded objective function value or an optimal extreme point of P may still exist. In either case, we execute the algorithm a second time to optimize the same perturbed objective function over P_u where $u = 2^{\Lambda+1}$. We get a second, unique maximizer $\boldsymbol{y}^{max}$, say, over the larger polytope and – since we have used *identical* objective functions – the difference vector $\boldsymbol{y} = \boldsymbol{y}^{max} - \boldsymbol{x}^{max}$ belongs to the asymptotic cone of the polyhedron P. This will let us decide whether the objective function is unbounded or bounded and in the latter case we find an optimizing point as well. Exercise 9.10 reviews the perturbation technique of Chapter 7.5.4 and summarizes part of what we need to establish the validity of our construction. The last part of Exercise 9.10 shows in particular that we can assume WROG that every nontrivial linear inequality $\boldsymbol{hx} \leq h_0$ belonging to a linear description of a rational polyhedron P of facet complexity ϕ satisfies $\|\boldsymbol{h}\|_\infty = 1$.

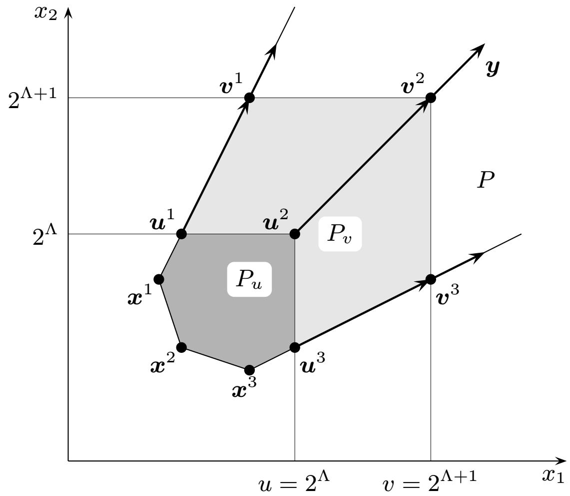


Fig. 9.12. Finding a direction vector in the asymptotic cone of P

9.7.1 ε-Optimal Sets and ε-Optimal Solutions

The following remark makes the first part of the construction precise and gives an analytical meaning to the terms "ε-optimal set" and "ε-optimal solution": the set F_ε defined in (9.78) is an ε-optimal set and any rational $\overline{\boldsymbol{x}} \in F_\varepsilon$ is an ε-optimal solution to the linear optimization problem over P_u.

Remark 9.16 *Let $P \subseteq \mathbb{R}^n$ be a rational polytope of facet complexity ϕ, P_u be as defined in (9.76) with integer $u \geq 2^{4n\phi}$ and let P_u^ε be the ε-solidification of P_u with respect to the ℓ_1-norm. Let $\boldsymbol{x}^{max} \in P_u$ be the* unique *maximizer of $\boldsymbol{dx}$ over P_u and $z_P = \boldsymbol{dx}^{max}$, where $\boldsymbol{d}$ has rational components and $\|\boldsymbol{d}\|_\infty = 1$. Define*

$$F_\varepsilon = P_u^\varepsilon \cap \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{dx} \geq z_P - \varepsilon\} \ , \tag{9.78}$$

where $0 < \varepsilon \leq 2^{-\Psi}$ and $\Psi = 9n\phi + 12n^2\phi + \langle \boldsymbol{d} \rangle$. Then F_ε is a full dimensional polytope, $vol(F_\varepsilon) \geq 2^n\varepsilon^n/n!$ and every extreme point $\boldsymbol{y} \in F_\varepsilon$ satisfies $|y_j - x_j^{max}| < 2^{-6n\phi-1}$ for $1 \leq j \leq n$. Moreover, rounding any rational $\overline{\boldsymbol{x}} \in F_\varepsilon$ componentwise by the best approximation algorithm with $\Theta = \overline{x}_j$ and $D = 2^{6n\phi}$ we obtain x_j^{max} and thus the maximizer $\boldsymbol{x}^{max}$ in time polynomial in n, ϕ and $\langle \overline{\boldsymbol{x}} \rangle$.

9.7.2 Finding Direction Vectors in the Asymptotic Cone

The following remark makes the second part of the construction precise; see also Exercise 9.11 (i) below.

Remark 9.17 *Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron of facet complexity ϕ, let $v > u \geq 2^\Lambda$ be any integers where Λ is defined in (9.77), let P_v and P_u be defined as in (9.76) with respect to v and u, respectively, and let C_∞ be the asymptotic cone of P. Then every extreme point $\boldsymbol{x}^v \in P_v$ can be written as $\boldsymbol{x}^v = \boldsymbol{x} + v\boldsymbol{t}$ where $\boldsymbol{x}, \boldsymbol{t} \in \mathbb{R}^n$ are rational vectors, $\boldsymbol{t} \in C_\infty$, $\langle \boldsymbol{t} \rangle \leq 4n^2\phi$ and moreover,*

$\boldsymbol{x}^u = \boldsymbol{x} + u\boldsymbol{t} \in P_u$ is an extreme point of P_u. Likewise, if $\boldsymbol{x}^u \in P_u$ is an extreme point of P_u and $\boldsymbol{x}^u = \boldsymbol{x} + u\boldsymbol{t}$, say, then $\boldsymbol{x}^v = \boldsymbol{x}^u + (v-u)\boldsymbol{t} \in P_v$ is an extreme point of P_v.

9.7.3 A CCS Ellipsoid Algorithm

The CCS ellipsoid algorithm is a central cut, sliding objective version of the basic ellipsoid algorithm. It takes the number of variables n, a rational vector $\boldsymbol{d}$ with $\|\boldsymbol{d}\|_\infty = 1$, the facet complexity ϕ and P as input. "P" is an identifier of the polyhedron over which we optimize the linear function $\boldsymbol{dx}$ and used to communicate with the separation subroutine SEPAR$(\boldsymbol{x}, \boldsymbol{h}, h_0, \phi, P)$. u specifies the hypercube with which we intersect P; see (9.76). ε, p and T are the parameters for the ε-solidification of P in the ℓ_1-norm, the required precision for the approximate calculation in terms of binary positions and the number of steps of the algorithm, respectively.

The subroutine SEPAR$(\boldsymbol{x}^k, \boldsymbol{h}, h_0, \phi, P)$ of Step 1 returns a most violated separator, i.e., a solution $(\boldsymbol{h}, h_0)$ to (9.53). The normalization requirement $\|\boldsymbol{h}\|_\infty = 1$ is no serious restriction at all; see Exercise 9.10 (v). It shows why we use the ε-solidification of P_u or P in the ℓ_1-norm: by Exercise 9.7(vi) we get $\boldsymbol{h}^T\boldsymbol{x} \le h_0 + \varepsilon\,\|\boldsymbol{h}\|_\infty = h_0 + \varepsilon$ as the corresponding inequality for P_u^ε or P^ε if $\boldsymbol{h}^T\boldsymbol{x} \le h_0$ for all $\boldsymbol{x} \in P$ and $(\boldsymbol{h}, h_0)$ was returned by the separation subroutine. So, we "perturb" the feasible set like in Chapter 9.3 by adding a "small enough" $\varepsilon > 0$ to the right-hand side h_0 of every most violated separator $(\boldsymbol{h}, h_0)$ that the separation subroutine returns. $\boldsymbol{h}$ is a column vector and $\boldsymbol{u}^j \in \mathbb{R}^n$ is the j-th unit vector.

CCS Ellipsoid Algorithm (n, $\boldsymbol{d}$, ϕ, P, u, ε, p, T)

Step 0: Set $k := 0$, $\boldsymbol{x}^0 := 0$, $\boldsymbol{F}_0 := nu\boldsymbol{I}_n$, $z_L := -nu\|\boldsymbol{d}\| - 1$, $\overline{z} := z_L$.

Step 1: **if** $|x_j^k| > u + \varepsilon$ for some $j \in \{1, \ldots, n\}$ **then**
  set $\boldsymbol{h} := \boldsymbol{u}^j$ if $x_j^k > 0$, $\boldsymbol{h} := -\boldsymbol{u}^j$ otherwise.
 else
  call SEPAR$(\boldsymbol{x}^k, \boldsymbol{h}, h_0, \phi, P)$.
  if $\boldsymbol{h}^T\boldsymbol{x}^k \le h_0 + \varepsilon$ **then**
   Set $\boldsymbol{h} := -\boldsymbol{d}^T$. **if** $\boldsymbol{dx}^k > \overline{z}$ **then** set $\overline{\boldsymbol{x}} := \boldsymbol{x}^k$, $\overline{z} := \boldsymbol{dx}^k$.
  endif
 endif

Step 2: **if** $k = T$ **go to** Step 3. Set

$$\boldsymbol{x}^{k+1} \approx \boldsymbol{x}^k - \frac{1}{n+1}\frac{\boldsymbol{F}_k\boldsymbol{F}_k^T\boldsymbol{h}}{\|\boldsymbol{F}_k^T\boldsymbol{h}\|},$$

$$\boldsymbol{F}_{k+1} \approx \frac{n + 1/12n}{\sqrt{n^2-1}}\boldsymbol{F}_k\left(\boldsymbol{I}_n - \frac{1 - \sqrt{(n-1)/(n+1)}}{\boldsymbol{h}^T\boldsymbol{F}_k\boldsymbol{F}_k^T\boldsymbol{h}}(\boldsymbol{F}_k^T\boldsymbol{h})(\boldsymbol{h}^T\boldsymbol{F}_k)\right),$$

where $\approx$ means that componentwise the round-off error is at most 2^{-p}.
Replace $k+1$ by k and **go to** Step 1.

Step 3: **if** $\overline{z} = z_L$ **stop** "$z_P = -\infty$. P is empty."
Round $\overline{\boldsymbol{x}}$ componentwise to the nearest rational $\boldsymbol{x}^*$ such that each component of $\boldsymbol{x}^*$ has a positive denominator less than or equal to $2^{6n\phi}$. **stop** "$\boldsymbol{x}^*$ is an optimal extreme point of P_u."

Remark 9.18 (Correctness and finiteness) *Let $P \subseteq \mathbb{R}^n$ be a rational polyhedron of facet complexity ϕ. Let $\boldsymbol{d} \in \mathbb{R}^n$ be a rational vector with $\|\boldsymbol{d}\|_\infty = 1$ such that* $\max\{\boldsymbol{dx} : \boldsymbol{x} \in P_u\}$ *has a* unique *maximizer $\boldsymbol{x}^{max} \in P_u$ with objective function value $z_P = \boldsymbol{dx}^{max}$ if $P \neq \emptyset$, where P_u is defined in (9.76) and $u \geq 2^{4n\phi}$ is any integer. If the subroutine* SEPAR($\boldsymbol{x}^k, \boldsymbol{h}, h_0, \phi, P$) *returns a most violated separator $\boldsymbol{hx} \leq h_0$ for $\boldsymbol{x}^k$ and P satisfying $\|\boldsymbol{h}\|_\infty = 1$, then the* CCS *ellipsoid algorithm concludes correctly that $P_u = P = \emptyset$ or it finds $\boldsymbol{x}^{max}$ if it is executed with the parameters*

$$\varepsilon = 2^{-\Psi}\,,\ T = \lceil 6n^2 \log \frac{n^2 u}{\varepsilon} \rceil\,,\ p = 14 + n^2 + \lceil 15n \log \frac{n^2 u}{\varepsilon} \rceil\,,$$

where $\Psi = 9n\phi + 12n^2\phi + \langle \boldsymbol{d} \rangle$.

9.7.4 Linear Optimization and Polyhedral Separation

For any polyhedron $P \subseteq \mathbb{R}^n$ with a linear description $\boldsymbol{Hx} \leq \boldsymbol{h}$, say, we denote by

$$P_\infty = \{\boldsymbol{y} \in \mathbb{R}^n : \boldsymbol{Hy} \leq \boldsymbol{0}\,,\ \|\boldsymbol{y}\|_\infty \leq 1\}$$

the intersection of the asymptotic cone C_∞ of P with the unit sphere in the ℓ_∞-norm.

Linear optimization problem: *Given a rational polyhedron $P \subseteq \mathbb{R}^n$ of facet complexity ϕ and a rational vector $\boldsymbol{c} \in \mathbb{R}^n$ (i) conclude that P is empty or (ii) find $\boldsymbol{x}^{max} \in P$ with $\boldsymbol{cx}^{max} \geq \boldsymbol{cx}$ for all $\boldsymbol{x} \in P$ or (iii) find $\boldsymbol{t} \in P_\infty$ with $\boldsymbol{ct} > 0$ and $\boldsymbol{ct} \geq \boldsymbol{cy}$ for all $\boldsymbol{y} \in P_\infty$.*

Polyhedral separation problem: *Given a rational polyhedron $P \subseteq \mathbb{R}^n$ of facet complexity ϕ and a rational vector $\boldsymbol{z} \in \mathbb{R}^n$ (i) conclude that $\boldsymbol{z} \in P$ or (ii) find a most violated separator for $\boldsymbol{z}$ and P, i.e. find a rational vector $(\boldsymbol{h}, h_0) \in \mathbb{R}^{n+1}$ that solves the problem*

$$\max\{\boldsymbol{hz} - h_0 : (\boldsymbol{h}, h_0) \in SP\,,\ \|\boldsymbol{h}\|_\infty = 1\}\,,$$

where $SP = \{(\boldsymbol{h}, h_0) \in \mathbb{R}^{n+1} : P \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{hx} \leq h_0\}\}$.

Neither problem specifies the *way* in which the polyhedron $P \subseteq \mathbb{R}^n$ is given: all we need is the information that the polyhedron is a subset of $\mathbb{R}^n$, that its facet complexity is at most ϕ and some "identifier" for P that permits us to communicate to some subroutine for instance.

Remark 9.19 *Let $P \subseteq \mathbb{R}^n$ be any rational polyhedron of facet complexity ϕ. If there exists an algorithm A, say, that solves the polyhedral separation problem in time that is bounded by a polynomial in n, ϕ and $\langle \boldsymbol{z} \rangle$, then the linear optimization problem can be solved in time that is bounded by a polynomial in n, ϕ and $\langle \boldsymbol{c} \rangle$.*

To outline the proof that the statement of Remark 9.19 can be reversed as well let

$$S = \{\boldsymbol{x}^1, \dots, \boldsymbol{x}^p\} \text{ and } T = \{\boldsymbol{y}^1, \dots, \boldsymbol{y}^r\}$$

be any minimal generator of P and denote by $\boldsymbol{X}$ the $n \times p$ matrix with columns $\boldsymbol{x}^i$, by $\boldsymbol{Y}$ the $n \times r$ matrix with columns $\boldsymbol{y}^i$. Either $\boldsymbol{X}$ or $\boldsymbol{Y}$ or both may be void. The set $SP = \{(\boldsymbol{h}, h_0) \in \mathbb{R}^{n+1} : P \subseteq \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{hx} \leq h_0\}\}$ of separators for P satisfies

$$SP = \{(\boldsymbol{h}, h_0) \in \mathbb{R}^{n+1} : \boldsymbol{hX} - h_0\boldsymbol{g} \leq \boldsymbol{0}\,,\ \boldsymbol{hY} \leq \boldsymbol{0}\}\,, \tag{9.80}$$

where $\boldsymbol{g} \in \mathbb{R}^p$ is a row vector with p components equal to 1. It follows that SP is a polyhedral cone in $\mathbb{R}^{n+1}$ of facet complexity at most $\phi^* = 4n^2\phi + 3$. Denote by

$$SP_\infty = \{(\boldsymbol{h}, h_0) \in \mathbb{R}^{n+1} : \boldsymbol{hX} - h_0\boldsymbol{g} \leq \boldsymbol{0}\,,\ \boldsymbol{hY} \leq \boldsymbol{0}\,,\ -\boldsymbol{e} \leq \boldsymbol{h} \leq \boldsymbol{e}\} \tag{9.81}$$

the polyhedron in $\mathbb{R}^{n+1}$ over which we need to maximize $\boldsymbol{hz} - h_0$ in order to find a most violated separator for $\boldsymbol{z}$ and P, where $\boldsymbol{e} \in \mathbb{R}^n$ is a row vector with n components equal to 1. The polyhedron SP_∞ contains the halfline defined by $(\boldsymbol{0}, 1) \in \mathbb{R}^n$ if $\boldsymbol{X}$ is nonvoid, it contains the line defined by $(\boldsymbol{0}, \pm 1) \in \mathbb{R}^{n+1}$ *if and only if* $\boldsymbol{X}$ is void and every nonzero extreme point $(\boldsymbol{h}, h_0)$ of SP_∞ satisfies $\|\boldsymbol{h}\|_\infty = 1$. By Remark 9.19 we can optimize the linear function $\boldsymbol{hz} - h_0$ over SP_∞ in polynomial time provided that the polyhedral separation problem for SP_∞ and any rational $(\boldsymbol{f}, f_0) \in \mathbb{R}^{n+1}$, say, can be solved in time that is bounded by a polynomial in n, ϕ and $\langle \boldsymbol{f} \rangle + \langle f_0 \rangle$. So we need to identify the set of separators for SP_∞. We will do so in two steps: first we identify the set SP^*, say, of separators for SP.

Since SP is a polyhedral cone in $\mathbb{R}^{n+1}$, its set SP^* of separators is a subset of all halfspaces of $\mathbb{R}^{n+1}$ that contain the origin, i.e.

$$SP^* = \{(\boldsymbol{x}, x_{n+1}) \in \mathbb{R}^{n+1} : SP \subseteq \{(\boldsymbol{h}, h_0) \in \mathbb{R}^{n+1} : \boldsymbol{hx} - h_0 x_{n+1} \le 0\}\} \ ,$$

because SP contains the origin of $\mathbb{R}^{n+1}$ and with every nonzero point the cone SP contains the entire halfline defined by it. It follows that

$$\begin{aligned} SP^* &= \{(\boldsymbol{x}, x_{n+1}) \in \mathbb{R}^{n+1} : \boldsymbol{hx} - h_0 x_{n+1} \le 0 \text{ for all } (\boldsymbol{h}, h_0) \in SP\} \\ &= \{(\boldsymbol{x}, x_{n+1}) \in \mathbb{R}^{n+1} : \boldsymbol{Hx} - \boldsymbol{h} x_{n+1} \le \boldsymbol{0}\, ,\ -x_{n+1} \le 0\} \ , \end{aligned}$$

where $\boldsymbol{Hx} \le \boldsymbol{h}$ is any linear description of the polyhedron $P \subseteq \mathbb{R}^n$. The inequality $x_{n+1} \ge 0$ follows because SP contains the halfline noted above. If $P \ne \emptyset$ and $\boldsymbol{X}$ is void, i.e. if P contains lines, then we must replace $x_{n+1} \ge 0$ by the equation $x_{n+1} = 0$, because SP contains the line defined by $(\boldsymbol{0}, \pm 1) \in \mathbb{R}^{n+1}$ in this case. So if $P \ne \emptyset$ and $\boldsymbol{X}$ is nonvoid, then the set SP^* of separators for SP is the *homogenization* of the polyhedron P – see (7.5). If $P \ne \emptyset$ and $\boldsymbol{X}$ is void, then the set SP^* of separators in question is simply the asymptotic cone of P – see (7.3). Define

$$SP^*_\infty = \{(\boldsymbol{x}, x_{n+1}) \in \mathbb{R}^{n+1} : \boldsymbol{Hx} - \boldsymbol{h} x_{n+1} \le \boldsymbol{0}\, ,\ -\boldsymbol{e} \le \boldsymbol{x} \le \boldsymbol{e}\, ,\ 0 \le x_{n+1} \le 1\} \ , \tag{9.82}$$

where $\boldsymbol{e} \in \mathbb{R}^n$ has n components equal to 1. So the set SP^*_∞ of all separators is a nonempty polytope in $\mathbb{R}^{n+1}$, every nonzero extreme point of which has an ℓ_∞-norm of 1.

Consider now $(\boldsymbol{x}^0, x^0_{n+1}) \ne (\boldsymbol{x}^1, x^1_{n+1}) \in SP^*_\infty$ with $x^0_{n+1} > 0$ and $x^1_{n+1} > 0$. Then $(x^1_{n+1}\boldsymbol{x}^0, x^0_{n+1}x^1_{n+1}) \in SP^*_\infty$ and $(x^0_{n+1}\boldsymbol{x}^1, x^0_{n+1}x^1_{n+1}) \in SP^*_\infty$. If for some $(\boldsymbol{f}, f_0) \in \mathbb{R}^{n+1}$ we have

$$\boldsymbol{f}(x^1_{n+1}\boldsymbol{x}^0) - f_0 x^0_{n+1}x^1_{n+1} > \boldsymbol{f}(x^0_{n+1}\boldsymbol{x}^1) - f_0 x^0_{n+1}x^1_{n+1} > 0 \ ,$$

then the point $(\boldsymbol{x}^0, x^0_{n+1})$ is a "more violated" separator for $(\boldsymbol{f}, f_0)$ and SP than $(\boldsymbol{x}^1, x^1_{n+1})$, because for all $\lambda \ge 0$ $(\lambda x^1_{n+1}\boldsymbol{x}^0, \lambda x^0_{n+1}x^1_{n+1}) \in SP^*$, $(\lambda x^0_{n+1}\boldsymbol{x}^1, \lambda x^0_{n+1}x^1_{n+1}) \in SP^*$ and the previous inequalities remain true for the entire open halfline, i.e. for all $\lambda > 0$.

Thus even if $\boldsymbol{fx}^1 - f_0 x^1_{n+1} > \boldsymbol{fx}^0 - f_0 x^0_{n+1} > 0$, rather than $\boldsymbol{fx}^0 - f_0 x^0_{n+1} > \boldsymbol{fx}^1 - f_0 x^1_{n+1}$, for the "original" points $(\boldsymbol{x}^0, x^0_{n+1})$, $(\boldsymbol{x}^1, x^1_{n+1}) \in SP^*_\infty$ it can happen that $(\boldsymbol{x}^0, x^0_{n+1})$ defines a more violated separator. It follows that we have to scale the "homogenizing" components of two violated separators $(\boldsymbol{x}^0, x^0_{n+1})$ and $(\boldsymbol{x}^1, x^1_{n+1})$ with $x^0_{n+1} > 0$ and $x^1_{n+1} > 0$ to have *equal* values if we want to decide which one of the two is more violated than the other; see also Exercise 9.12 (iii) below.

Suppose that $(\boldsymbol{f}, f_0) \in \mathbb{R}^{n+1}$ is given, that $P \ne \emptyset$ and that algorithm B, say, solves the linear optimization problem over $P \subseteq \mathbb{R}^n$. We run the algorithm with the objective function vector $\boldsymbol{c} = \boldsymbol{f}$.

Assume that algorithm B finds $\boldsymbol{x}^{max} \in P$ with $\boldsymbol{fx}^{max} \ge \boldsymbol{fx}$ for all $\boldsymbol{x} \in P$. If $\boldsymbol{fx}^{max} \le f_0$ then we conclude that $(\boldsymbol{f}, f_0) \in SP$. Otherwise, $\boldsymbol{fx}^{max} - f_0 > 0$ and $\boldsymbol{fx}^{max} - f_0 \ge \boldsymbol{fx} - f_0$ for all $\boldsymbol{x} \in P$.

Then

$$\boldsymbol{x}^0 = \alpha \boldsymbol{x}^{max}\,,\; x^0_{n+1} = \alpha \qquad \text{where } \alpha^{-1} = \left\| \begin{pmatrix} \boldsymbol{x}^{max} \\ 1 \end{pmatrix} \right\|_\infty \tag{9.83}$$

solves the polyhedral separation problem for $(\boldsymbol{f}, f_0)$ and SP; see the text.

Suppose that algorithm B finds $\boldsymbol{t} \in P_\infty$ with $\boldsymbol{f}\boldsymbol{t} > 0$ and $\boldsymbol{f}\boldsymbol{t} \geq \boldsymbol{f}\boldsymbol{y}$ for all $\boldsymbol{y} \in P_\infty$. Then

$$\boldsymbol{x}^0 = \boldsymbol{t}\,,\; x^0_{n+1} = 0 \tag{9.84}$$

solves the polyhedral separation problem for $(\boldsymbol{f}, f_0) \in \mathbb{R}^{n+1}$ and SP; see the text.

It follows that if $P \neq \emptyset$ we can solve the polyhedral separation problem for any rational $(\boldsymbol{f}, f_0) \in \mathbb{R}^{n+1}$ and the cone SP in time that is polynomially bounded in n, ϕ and $\langle \boldsymbol{f} \rangle + \langle f_0 \rangle$ by solving $\max\{\boldsymbol{f}\boldsymbol{x} : \boldsymbol{x} \in P\}$. If we conclude that $(\boldsymbol{f}, f_0) \in SP$ define $(\boldsymbol{x}^0, x^0_{n+1}) = (\boldsymbol{0}, 0)$; otherwise the most violated separator $(\boldsymbol{x}^0, x^0_{n+1})$ for $(\boldsymbol{f}, f_0) \in \mathbb{R}^{n+1}$ and SP is given by (9.83) or (9.84), respectively.

It remains to show that we can solve the separation problem for $(\boldsymbol{f}, f_0) \in \mathbb{R}^{n+1}$ and SP_∞.

The polyhedron SP_∞ differs from the cone SP by exactly $2n$ constraints of the form $-1 \leq h_j \leq 1$ for all $1 \leq j \leq n$, which we can check by LIST-and-CHECK in polynomial time. Given $(\boldsymbol{x}^0, x^0_{n+1})$ define

$$\alpha = \boldsymbol{f}\boldsymbol{x}^0 - f_0 x^0_{n+1}\,,\; \beta = \max_{1\leq j\leq n}\{0, 1 - f_j\}\,,\; \gamma = \max_{1\leq j\leq n}\{0, 1 + f_j\}\,,$$

where f_j is the j-th component of $\boldsymbol{f} \in \mathbb{R}^n$. If $\max\{\alpha, \beta, \gamma\} = 0$ we conclude that $(\boldsymbol{f}, f_0) \in SP_\infty$. Otherwise, if $\max\{\alpha, \beta, \gamma\} = \alpha$ then $(\boldsymbol{x}^0, x^0_{n+1})$ is a most violated separator for $(\boldsymbol{f}, f_0)$ and SP_∞. If $\max\{\alpha, \beta, \gamma\} = \beta$ then $(\boldsymbol{x}^0, x^0_{n+1}) = (\boldsymbol{u}^k, 1) \in \mathbb{R}^{n+1}$ is a most violated separator for $(\boldsymbol{f}, f_0)$ and SP_∞ where $k \in \{1, \ldots, n\}$ is such that $\beta = 1 - f_k$. Likewise, if $\max\{\alpha, \beta, \gamma\} = \gamma$ then $(\boldsymbol{x}^0, x^0_{n+1}) = (-\boldsymbol{u}^k, 1) \in \mathbb{R}^{n+1}$ is a most violated separator for $(\boldsymbol{f}, f_0)$ and SP_∞ where $k \in \{1, \ldots, n\}$ is such that $\gamma = 1 + f_k$.

The preceding combination of *some* algorithm B for the linear optimization problem and of LIST-and-CHECK yields a separation routine SEPAR*$(\boldsymbol{f}, f_0, \boldsymbol{x}^0, x^0_{n+1}, \phi, SP_\infty)$ that solves the polyhedral separation problem for rational $(\boldsymbol{f}, f_0) \in \mathbb{R}^{n+1}$ and SP_∞ if the underlying polyhedron $P \subseteq \mathbb{R}^n$ is nonempty. Moreover, if the running time for algorithm B is bounded by a polynomial in n, ϕ and $\langle \boldsymbol{c} \rangle$, then the running time of SEPAR* is evidently bounded by a polynomial in n, ϕ and $\langle \boldsymbol{f} \rangle + \langle f_0 \rangle$.

To complete the outline of the proof that we can solve the problem

$$\max\{\boldsymbol{h}\boldsymbol{z} - h_0 : (\boldsymbol{h}, h_0) \in SP_\infty\}$$

for any rational $\boldsymbol{z} \in \mathbb{R}^n$ in polynomial time if a polynomial-time algorithm B for the linear optimization problem is known, we proceed as follows.

We run the algorithm B a first time with the objective function $\boldsymbol{c} = \boldsymbol{0}$. If algorithm B concludes that $P = \emptyset$ then we declare $\boldsymbol{h}\boldsymbol{z} = z_1 > h_0 = z_1 - 1$ to be a solution to the polyhedral separation problem. Since P is empty, any inequality that is violated by $\boldsymbol{z}$ is evidently a most violated inequality for $\boldsymbol{z}$ and P.

So suppose that we conclude that $P \neq \emptyset$. Now the separation subroutine SEPAR* applies and thus by Remark 9.19 we can solve the linear optimization problem

$$\max\{\boldsymbol{h}\boldsymbol{z} - h_0 : (\boldsymbol{h}, h_0) \in SP_\infty\}$$

in time that is polynomially bounded in n, ϕ and $\langle \boldsymbol{z} \rangle$.

If $(\boldsymbol{h}^{max}, h_0^{max})$ with $\boldsymbol{h}^{max}\boldsymbol{x} - h_0^{max} \geq \boldsymbol{h}\boldsymbol{z} - h_0$ for all $(\boldsymbol{h}, h_0) \in SP_\infty$ is obtained, then we conclude that $\boldsymbol{z} \in P$ if $\boldsymbol{h}^{max}\boldsymbol{z} \leq h_0^{max}$ and otherwise, a most violated separator for $\boldsymbol{z}$ and P has been obtained.

If a finite maximizer $(\boldsymbol{h}^{max}, h_0^{max}) \in SP_\infty$ does not exist, then the solution to the above linear optimization problem provides a direction vector in the asymptotic cone of SP_∞. The polyhedron SP_∞ contains the halfline defined by $(\boldsymbol{0}, 1) \in \mathbb{R}^{n+1}$ along which the objective function tends to $-\infty$. Consequently, if the unbounded case arises, then SP_∞ contains the line defined by $(\boldsymbol{0}, \pm 1) \in \mathbb{R}^{n+1}$ and the finite generator of P consists only of halflines, i.e. the matrix $\boldsymbol{X}$ in the definition of SP_∞ is void.

So we solve the linear optimization problem

$$\max\{\boldsymbol{h}\boldsymbol{z} - h_0 : (\boldsymbol{h}, h_0) \in SP_\infty\,,\ h_0 = 0\}$$

using the separation subroutine SEPAR*, i.e. we iterate the whole procedure a second time. Now the unbounded case cannot arise and we find a new $(\boldsymbol{h}^{max}, 0) \in \mathbb{R}^{n+1}$ with $\|\boldsymbol{h}^{max}\|_\infty = 1$ such that $\boldsymbol{h}^{max}\boldsymbol{z} \geq \boldsymbol{h}\boldsymbol{z}$ for all $(\boldsymbol{h}, 0) \in SP_\infty$. If $\boldsymbol{h}^{max}\boldsymbol{z} \leq 0$ then we conclude that $\boldsymbol{z} \in P$, whereas otherwise $(\boldsymbol{h}^{max}, 0)$ is a most violated separator for $\boldsymbol{z}$ and P since every separator $(\boldsymbol{h}, h_0)$ for P satisfies $h_0 = 0$ in this case.

The concatenation of polynomials in some variables yields a polynomial in the same variables. Thus the entire procedure can be executed in time that is bounded by some polynomial in n, ϕ and $\langle \boldsymbol{z} \rangle$.

Remark 9.20 *For any rational polyhedron $P \subseteq \mathbb{R}^n$ and $n \geq 2$ the linear optimization problem and the polyhedral separation problem are polynomial-time equivalent problems.*

So if either one of the two problems above is solvable in polynomial time, then so is the other. From a theoretical point of view we may thus concentrate on anyone of the two problems to study the algorithmic "tractability" of linear optimization problems over rational polyhedra P in $\mathbb{R}^n$.

Remark 9.19 has several important implications which we do not prove in detail. Among these are that if either problem is polynomially solvable for some rational polyhedron $P \subseteq \mathbb{R}^n$, then we can find

- *the dimension $dimP$ of P,*
- *a linear description of the affine hull $aff(P)$ of P,*
- *a linear description of the lineality space L_P of P,*
- *extreme points and extreme rays of P if there are any,*
- *facet-defining linear inequalities for P,* etc

in polynomial time. The latter is of particular importance for the **branch-and-cut** approach to combinatorial optimization problems which rely on finding (parts of) ideal descriptions of the corresponding polyhedra.

9.8 Exercises

Exercise 9.1

Show that for $n = 1$ the ellipsoids E_k are intervals and that the updating formulas (9.1) and (9.2) become $x^{k+1} = x^k - \frac{1}{2}F_k\,\text{sign}(a)$, $F_{k+1} = \frac{1}{2}F_k$ for $k \geq 0$ where $ax \leq b$ is any inequality that is violated by x^k and $\text{sign}(a) = 1$ if $a \geq 0$, -1 otherwise. (Hint: *Note that $\boldsymbol{d}\boldsymbol{d}^T = 1$, $\boldsymbol{I}_n = 1$ and thus the terms $n-1$ cancel.*)

The inequality $\|\boldsymbol{F}^{-1}(\boldsymbol{x} - \boldsymbol{x}^0)\| \leq 1$ in one dimension becomes $|x - x^0| \leq |F|$, since $\boldsymbol{x}$, $\boldsymbol{x}^0$ and $\boldsymbol{F}$ are scalars. WROG we assume that F is positive. Then the inequality corresponds to the interval $[x^0 - F, x^0 + F]$. Let $ax \leq b$ be a violated inequality by x^0. We can assume that $a = \pm 1$ (the case $a = 0$ is meaningless), since every inequality can be brought in that form. Then the vector $\boldsymbol{d}$ becomes a scalar $Fa/|Fa|$, i.e. $d = \text{sign}(a) = \pm 1$. Thus the updating formula for the center is given by

$$x^{k+1} = x^k - \frac{1}{2}F_k\text{sign}(a)\ .$$

Note that $\boldsymbol{d}\boldsymbol{d}^T$ becomes $d^2 = 1$ and $\boldsymbol{I}_n$ becomes $I_1 = 1$. Then we calculate from the updating formula for F

$$F_{k+1} = \sqrt{\frac{n^2}{n^2-1}}F_k\left(1 - 1 + \sqrt{\frac{n-1}{n+1}}\right) = \sqrt{\frac{n^2(n-1)}{(n-1)(n+1)^2}}F_k = \frac{n}{n+1}F_k = \frac{1}{2}F_k\ .$$

Exercise 9.2

Let $\boldsymbol{F}$, $\boldsymbol{R}$ be two $m \times n$ matrices of reals.

(i) Show $\|\boldsymbol{F}\| = 0$ if and only if $\boldsymbol{F} = \mathbf{O}$, $\|\alpha\boldsymbol{F}\| = |\alpha|\|\boldsymbol{F}\|$ for all $\alpha \in \mathbb{R}$, $\|\boldsymbol{F} + \boldsymbol{R}\| \leq \|\boldsymbol{F}\| + \|\boldsymbol{R}\|$.

(ii) Show $\|\boldsymbol{I}_n\| = \sqrt{n}$ and $\|\boldsymbol{F}\|^2 = trace(\boldsymbol{F}\boldsymbol{F}^T)$.

(iii) Show $\|\boldsymbol{F}\|_2 \leq \|\boldsymbol{F}\| \leq \sqrt{n}\|\boldsymbol{F}\|_2$. (Hint: *Use $trace(\boldsymbol{F}^T\boldsymbol{F}) = \sum \lambda_i$.*)

(iv) For $\boldsymbol{F}$ as before, $\boldsymbol{R}$ of size $n \times p$, $\boldsymbol{r} \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$ show

$$\|\boldsymbol{F}\boldsymbol{R}\| \leq \|\boldsymbol{F}\|\|\boldsymbol{R}\|\ , \quad \|\boldsymbol{F}(\boldsymbol{I}_n - \alpha\boldsymbol{r}\boldsymbol{r}^T)\|^2 = \|\boldsymbol{F}\|^2 - \alpha(2 - \alpha\|\boldsymbol{r}\|^2)\|\boldsymbol{F}\boldsymbol{r}\|^2\ .$$

(i) We have $\|\boldsymbol{F}\| = \sqrt{\sum_{i=1}^m \sum_{j=1}^n (f_j^i)^2} = 0$, which is equivalent to $f_j^i = 0$ for all $1 \leq i \leq m$ and $1 \leq j \leq n$. Thus $\boldsymbol{F} = \mathbf{O}$.

To prove that $\|\alpha \boldsymbol{F}\| = |\alpha|\|\boldsymbol{F}\|$ we calculate

$$\|\alpha \boldsymbol{F}\| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(\alpha f_j^i)^2} = \sqrt{\alpha^2 \sum_{i=1}^{m}\sum_{j=1}^{n}(f_j^i)^2} = |\alpha|\|\boldsymbol{F}\|$$

To prove that $\|\boldsymbol{F}+\boldsymbol{R}\| \leq \|\boldsymbol{F}\| + \|\boldsymbol{R}\|$ we prove the equivalent inequality (since both parts are nonnegative) $\|\boldsymbol{F}+\boldsymbol{R}\|^2 \leq (\|\boldsymbol{F}\| + \|\boldsymbol{R}\|)^2$. We calculate

$$\begin{aligned}
\|\boldsymbol{F}+\boldsymbol{R}\|^2 &= \sum_{i=1}^{m}\sum_{j=1}^{n}(f_j^i)^2 + \sum_{i=1}^{m}\sum_{j=1}^{n}(r_j^i)^2 + 2\sum_{i=1}^{m}\sum_{j=1}^{n} f_j^i r_j^i \\
(\|\boldsymbol{F}\| + \|\boldsymbol{R}\|)^2 &= \sum_{i=1}^{m}\sum_{j=1}^{n}(f_j^i)^2 + \sum_{i=1}^{m}\sum_{j=1}^{n}(r_j^i)^2 + 2\sqrt{(\sum_{i=1}^{m}\sum_{j=1}^{n}(f_j^i)^2)(\sum_{i=1}^{m}\sum_{j=1}^{n}(r_j^i)^2)}
\end{aligned}$$

and thus the inequality to prove becomes

$$\sum_{i=1}^{m}\sum_{j=1}^{n} f_j^i r_j^i \leq \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(f_j^i)^2}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(r_j^i)^2}$$

which follows from the Cauchy-Schwarz inequality.

(ii) The first part follows directly from the definition of the Frobenius norm:

$$\|\boldsymbol{I}_n\| = \sqrt{\sum_{k=1}^{n}\sum_{j=1}^{n}\delta_j^k} = \sqrt{\sum_{i=1}^{n} 1} = \sqrt{n}\ ,$$

where $\delta_j^j = 1$, $\delta_j^k = 0$. To prove the second part, let $\boldsymbol{G} = \boldsymbol{F}\boldsymbol{F}^T$. Then $g_k^k = \sum_{j=1}^{n}(f_j^k)^2$. Now we have

$$trace(\boldsymbol{F}\boldsymbol{F}^T) = \sum_{k=1}^{m} g_k^k = \sum_{k=1}^{m}\sum_{j=1}^{n}(f_j^k)^2 = \|\boldsymbol{F}\|^2\ .$$

(iii) The matrix $\boldsymbol{F}^T\boldsymbol{F}$ is symmetric. Thus $trace(\boldsymbol{F}^T\boldsymbol{F}) = \sum_{i=1}^{n}\lambda_i$ where λ_i are the eigenvalues of $\boldsymbol{F}^T\boldsymbol{F}$. Since $\boldsymbol{F}^T\boldsymbol{F}$ is also positive semi-definite we have $\lambda_i \geq 0$ for all $1 \leq i \leq n$. Let $\Lambda = \max\{\lambda_i : 1 \leq i \leq n\}$. Then we calculate

$$\|\boldsymbol{F}\|_2 = \sqrt{\Lambda} \leq \sqrt{\sum_{i=1}^{n}\lambda_i} = \|\boldsymbol{F}\|$$

where we have used the second part of (ii) for the last equality. Moreover,

$$\|\boldsymbol{F}\| = \sqrt{\sum_{i=1}^{n}\lambda_i} \leq \sqrt{n\Lambda} = \sqrt{n}\|\boldsymbol{F}\|\ .$$

(iv) From the Cauchy-Schwarz inequality we have $(\boldsymbol{f}^i \boldsymbol{r}_j)^2 \leq \|\boldsymbol{f}^i\|^2 \|\boldsymbol{r}_j\|^2$, where $\boldsymbol{f}^i$ is the i-th row of $\boldsymbol{F}$ and $\boldsymbol{r}_j$ the j-th column of $\boldsymbol{R}$. Summing over j first and over i next, we get

$$\sum_{i=1}^{m} \sum_{j=1}^{p} (\boldsymbol{f}^j \boldsymbol{r}_j)^2 \leq \sum_{i=1}^{m} \|\boldsymbol{f}^i\|^2 \sum_{j=1}^{p} \|\boldsymbol{r}_j\|^2$$

which using $\boldsymbol{f}^i \boldsymbol{r}_j = \sum_{k=1}^n f_k^i r_j^k$ gives

$$\sum_{i=1}^{m} \sum_{j=1}^{p} (\sum_{k=1}^{n} f_k^i r_j^k)^2 \leq \left(\sum_{i=1}^{m} \sum_{j=1}^{n} (f_j^i)^2 \right) \left(\sum_{j=1}^{p} \sum_{i=1}^{n} (r_j^i)^2 \right) \ .$$

Using the definition of the Frobenius norm we get $\|\boldsymbol{FR}\|^2 \leq \|\boldsymbol{F}\|^2 \|\boldsymbol{R}\|^2$ which proves $\|\boldsymbol{FR}\| \leq \|\boldsymbol{F}\| \|\boldsymbol{R}\|$, since $\| \cdot \| \geq 0$. To prove the second relation, we calculate

$$\begin{aligned}
\|\boldsymbol{F}(\boldsymbol{I}_n - \alpha \boldsymbol{r}\boldsymbol{r}^T)\|^2 &= trace(\boldsymbol{F}(\boldsymbol{I}_n - \alpha \boldsymbol{r}\boldsymbol{r}^T)(\boldsymbol{I}_n - \alpha \boldsymbol{r}\boldsymbol{r}^T)\boldsymbol{F}^T) = trace((\boldsymbol{F} - \alpha \boldsymbol{F}\boldsymbol{r}\boldsymbol{r}^T)(\boldsymbol{F}^T - \alpha \boldsymbol{r}\boldsymbol{r}^T \boldsymbol{F}^T)) \\
&= trace(\boldsymbol{F}\boldsymbol{F}^T - \alpha \boldsymbol{F}\boldsymbol{r}\boldsymbol{r}^T \boldsymbol{F} - \alpha \boldsymbol{F}\boldsymbol{r}\boldsymbol{r}^T \boldsymbol{F}^T + \alpha^2 \boldsymbol{F}\boldsymbol{r}\boldsymbol{r}^T \boldsymbol{r}\boldsymbol{r}^T \boldsymbol{F}^T) \\
&= trace(\boldsymbol{F}\boldsymbol{F}^T) - 2\alpha \|\boldsymbol{F}\boldsymbol{r}\|^2 + \alpha^2 \|\boldsymbol{r}\|^2 \|\boldsymbol{F}\boldsymbol{r}\|^2 = \|\boldsymbol{F}\|^2 - \alpha(2 - \alpha\|\boldsymbol{r}\|^2)\|\boldsymbol{F}\boldsymbol{r}\|^2 \ .
\end{aligned}$$

Exercise 9.3

Let $\boldsymbol{Q} = \boldsymbol{F}\boldsymbol{F}^T$ and $\boldsymbol{Q}_P = \boldsymbol{F}_P \boldsymbol{F}_P^T$ where

$$\boldsymbol{F}_P = \sqrt{\frac{n^2}{n^2-1}} \boldsymbol{F} \left(\boldsymbol{I}_n - \left(1 - \sqrt{\frac{n-1}{n+1}} \right) \boldsymbol{d}\boldsymbol{d}^T \right) \ .$$

Show that

$$\begin{aligned}
\boldsymbol{Q}_P &= \frac{n^2}{n^2-1} \boldsymbol{Q} \left(\boldsymbol{I}_n - \frac{2}{n+1} \frac{\boldsymbol{a}\boldsymbol{a}^T \boldsymbol{Q}}{\boldsymbol{a}^T \boldsymbol{Q} \boldsymbol{a}} \right) \ , \\
\boldsymbol{Q}_P^{-1} &= \frac{n^2-1}{n^2} \left(\boldsymbol{Q}^{-1} + \frac{2}{n-1} \frac{\boldsymbol{a}\boldsymbol{a}^T}{\boldsymbol{a}^T \boldsymbol{Q} \boldsymbol{a}} \right) \ .
\end{aligned}$$

For $\boldsymbol{Q}_P$ we calculate using (9.13)

$$\begin{aligned}
\boldsymbol{Q}_P = \boldsymbol{F}_P \boldsymbol{F}_P^T &= \frac{n^2}{n-1} \boldsymbol{F} \left(\boldsymbol{I}_n - \left(1 - \sqrt{\frac{n-1}{n+1}} \boldsymbol{d}\boldsymbol{d}^T \right) \right) \left(\boldsymbol{I}_n - \left(1 - \sqrt{\frac{n-1}{n+1}} \boldsymbol{d}\boldsymbol{d}^T \right) \right) \boldsymbol{F}^T \\
&= \frac{n^2}{n^2-1} \left[\boldsymbol{F}\boldsymbol{F}^T - 2 \left(1 - \sqrt{\frac{n-1}{n+1}} \right) \boldsymbol{F}\boldsymbol{d}\boldsymbol{d}^T \boldsymbol{F}^T + \left(1 - \sqrt{\frac{n-1}{n+1}} \right)^2 \boldsymbol{F}\boldsymbol{d}\boldsymbol{d}^T \boldsymbol{d}\boldsymbol{d}^T \boldsymbol{F}^T \right]
\end{aligned}$$

or factoring out the scalar $\boldsymbol{d}^T\boldsymbol{d} = \|\boldsymbol{d}\|^2 = 1$ from the last term

$$= \frac{n^2}{n^2-1}\left[\boldsymbol{Q} - \left(2 - 2\sqrt{\frac{n-1}{n+1}} - 1 - \frac{n-1}{n+1} + 2\sqrt{\frac{n-1}{n+1}}\right)\boldsymbol{F}\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{F}^T\right]$$

and since $\boldsymbol{F}\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{F} = \frac{\boldsymbol{Q}\boldsymbol{a}\boldsymbol{a}^T\boldsymbol{Q}}{\boldsymbol{a}^T\boldsymbol{Q}\boldsymbol{a}}$ because $\boldsymbol{d} = \boldsymbol{F}^T\boldsymbol{a}/\|\boldsymbol{F}^T\boldsymbol{a}\|$ and $\|\boldsymbol{d}\| = 1$

$$= \frac{n^2}{n^2-1}\boldsymbol{Q}\left(\boldsymbol{I}_n - \frac{2}{n+1}\frac{\boldsymbol{a}\boldsymbol{a}^T\boldsymbol{Q}}{\boldsymbol{a}^T\boldsymbol{Q}\boldsymbol{a}}\right) .$$

For $\boldsymbol{Q}_P^{-1}$ we calculate using the formula (9.22) for $\boldsymbol{F}_P^{-1}$

$$\boldsymbol{Q}_P^{-1} = (\boldsymbol{F}_P^{-1})^T\boldsymbol{F}_P^{-1} = \frac{n-1}{n^2}(\boldsymbol{F}^{-1})^T\left(\boldsymbol{I}_n - \left(1 - \sqrt{\frac{n+1}{n-1}}\boldsymbol{d}\boldsymbol{d}^T\right)\right)\left(\boldsymbol{I}_n - \left(1 - \sqrt{\frac{n+1}{n-1}}\boldsymbol{d}\boldsymbol{d}^T\right)\right)\boldsymbol{F}^{-1}$$

$$= \frac{n^2-1}{n^2}\left[(\boldsymbol{F}^{-1})^T\boldsymbol{F}^{-1} - 2\left(1 - \sqrt{\frac{n+1}{n-1}}\right)(\boldsymbol{F}^{-1})^T\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{F}^{-1} + \left(1 - \sqrt{\frac{n+1}{n-1}}\right)^2(\boldsymbol{F}^{-1})^T\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{F}^{-1}\right]$$

or factoring out the scalar $\boldsymbol{d}^T\boldsymbol{d} = \|\boldsymbol{d}\|^2 = 1$ from the last term

$$= \frac{n^2-1}{n^2}\left[\boldsymbol{Q}^{-1} - \left(2 - 2\sqrt{\frac{n+1}{n-1}} - 1 - \frac{n+1}{n-1} + 2\sqrt{\frac{n+1}{n-1}}\right)(\boldsymbol{F}^{-1})^T\boldsymbol{d}\boldsymbol{d}^T\boldsymbol{F}^{-1}\right]$$

$$= \frac{n^2-1}{n^2}\boldsymbol{Q}^{-1}\left(\boldsymbol{I}_n + \frac{2}{n+1}\frac{\boldsymbol{a}\boldsymbol{a}^T}{\boldsymbol{a}^T\boldsymbol{Q}\boldsymbol{a}}\right)$$

where we have used $\|\boldsymbol{d}\| = 1$ and $\boldsymbol{d}^T\boldsymbol{F}^{-1} = \boldsymbol{a}^T/\|\boldsymbol{F}^T\boldsymbol{a}\|$ which follows from the definition of $\boldsymbol{d}$; see (9.12).

Exercise 9.4

Let $\boldsymbol{Q}_k = \boldsymbol{F}_k\boldsymbol{F}_k^T$ be the positive definite matrix that defines the ellipsoid E_k. Denote by λ^k_{min} the smallest and by λ^k_{max} the largest eigenvalue of $\boldsymbol{Q}_k$. Prove that $\lambda^k_{min} \leq R^2 2^{-k/2n^2}$ and $\lambda^k_{max} \geq R^2 2^{-2k/n^2}$ for all k of the basic ellipsoid algorithm. (Hint: *Use (9.35).*)

Since $\boldsymbol{Q}_k$ is positive definite and nonsingular we have $\lambda^k_{min} > 0$. From $\boldsymbol{Q}_k = \boldsymbol{F}_k\boldsymbol{F}_k^T$ we have that the eigenvalues of $\boldsymbol{F}_k$ are $|\mu_i| = \sqrt{\lambda_i}$ and thus $|det\boldsymbol{F}| = \prod_{i=1}^n \sqrt{\lambda_i}$. It follows that

$$(\lambda^k_{min})^{n/2} \leq |det\boldsymbol{F}| \leq (\lambda^k_{max})^{n/2} .$$

From (9.35) we have

$$(\lambda^k_{min})^{n/2} \leq |det\boldsymbol{F}| \leq R^n 2^{-\frac{k}{4n}} \Rightarrow \lambda^k_{min} \leq R^2 2^{-\frac{k}{2n^2}}$$

and similarly

$$(\lambda^k_{max})^{n/2} \geq |det\boldsymbol{F}| \geq R^n 2^{-\frac{k}{n}} \Rightarrow \lambda^k_{max} \geq R^2 2^{-\frac{2k}{n^2}} \ .$$

Exercise 9.5

(i) Write a computer program of the DCS ellipsoid algorithm in a computer language of your choice for the linear programming problem (LP).

(ii) Solve the problems of Exercises 5.1, 5.9 and 8.2(ii) using your computer program.

(i) The following listing is an implementation of the algorithm in MATLAB. The required input is as in the simplex algorithm; see Exercise 5.2.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% This is the implementation of the DCS Ellipsoid algorithm
%% as found on pages 309-310.
%%
%% NAME    : dcsel
%% PURPOSE: Solve the LP: max {cx: a~x <= b, x >=0}
%% INPUT   : The matrix a~and the vectors c and b.
%% OUTPUT : z : the optimal value
%%          x : the optimal solution
%%          k : the number of iterations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[m,n]=size(A);
A = [A;-eye(n)];
b = [b zeros(1,n)];
[m,n]=size(A);
zl=-10000;
zu=10000;
eps=10^(-9);
R=500;
Vf=10^(-6);

R0=sqrt(n)*(1+R/2);
veps=eps*ones(m,1);
z=zl+1;
z0=z;
x = (R/2)*ones(n,1);
H=R0*eye(n);
```

```
k=0;
f0=(1+1/n)^(-(n+1)/2) * (1-1/n)^(-(n-1)/2);
V=R0^n * pi^(n/2) / gamma(1+n/2);

while (1<2)
   slack=b'+veps- A*x ;
   [mxv,j]=min(slack);
   if (mxv >= 0)
   xstar=x+H*H'*c'/norm(c*H);
   help=A*(xstar-x);
   if (all(help <= 0)), error('Unbounded'), end;
   lambda=10000;
   for i=1:m,
      if (help(i) > 0)
         if (slack(i) /help(i) <= lambda)
           lambda=slack(i)/help(i);
         end;
      end;
   end;
   if (lambda >= 1), error('Unbounded'), end;
      if (c*(x+lambda*(xstar-x)) > z)
         xbar=x+lambda*(xstar-x);
         z=c*xbar;
      end;
   end;
   if ((c*x > z & z0 > zl+1) | (mxv <0 & z0-z > mxv))
      theta=b(j)+eps;
      alpha=(A(j,:)*x-theta)/norm(A(j,:)*H);
      r=A(j,:)';
   else
      alpha=(z-c*x)/norm(c*H);
      r=-c';
      z0=z;
   end;
   if ((alpha >=1 | V < Vf) & (z < zu & z > zl))
      fprintf('Optimal solution found in %d iterations, Vol= %8.4f\n',k-1,V);
      fprintf('%8.4f',xbar);
      fprintf('\nz=%8.4f\n',z);
      return;
   elseif (z <= zl)
      fprintf('Infeasible');
      return;
   elseif (z >= zu)
      fprintf('Unbounded');
      return;
   end;
```

```
   xold=x;
   x=x-(1+n*alpha)*H*H'*r/((n+1)*norm(H'*r));
   quan1=sqrt(((n-1)*(1-alpha))/((n+1)*(1+alpha)));
   quan2=n*sqrt((1-alpha^2)/(n^2-1));
   H=quan2*H*(eye(n)-(1-quan1)*H'*r*r'*H/norm(H'*r)^2);
   V=(1-alpha)*f0*V*(1-alpha^2)^((n-1)/2);
   fprintf('%10.5f ',x);
   fprintf(' cx=%10.5f, z=%10.5f\n',c*x,z);
   z1=c*(x-xold);
   help1=A*(x-xold);
   if (all(help1 < 0) & z1 > 0), error('Unbounded'), end;
   slack1=b'+veps-A*xold;
   mumax=-10000;
   mumin=10000;
   substep=1;
   for i=1:m,
      if (help1(i) > 0 )
         if (slack1(i)/help1(i) <=mumin)
            mumin=slack1(i)/help1(i);
         end;
      elseif (help1(i) < 0)
         if (slack1(i)/help1(i) >= mumax)
            mumax=slack1(i)/help1(i);
         end;
      else
         if (slack(i) < 0)
            substep=0;
         end;
      end;
   end;
   if (mumin < mumax & substep > 0)
      substep=0;
   end;
   if (substep > 0 )
      if (z1 < 0)
         mubar=mumax;
      else
         mubar=mumin;
      end;
      if (c*xold+mubar*z1 > z)
         xbar=xold+mubar*(x-xold);
         z=c*xbar;
      end;
   end;
   k=k+1;
end;
```

(ii) For the data of Exercise 5.1 we get

```
>> clear
>> psdat
>> dcsel
  -4.03134  165.32289  -88.70846   80.64577  cx= 294.36367, z=-9999.00000
 -10.02109   30.06256  -30.06291   10.02074  cx= -30.06467, z=-9999.00000
 -52.62810   42.08574   12.86073   -5.27118  cx=  61.90156, z=-9999.00000
   9.55652   35.42522   -3.34090  -46.70324  cx=  18.61864, z=-9999.00000
   3.54364   -1.94823  -14.64155    9.78702  cx= -37.74956, z=-9999.00000
  -3.83235   -9.72340    3.63685    4.58637  cx= -13.11476, z=-9999.00000
  -8.02842    7.99811    2.99137   -2.24117  cx=  15.42061, z=-9999.00000
   3.97518    4.26213    1.43981   -7.85914  cx=  10.77769, z=-9999.00000
   2.39358   -1.88535   -0.05693    4.51311  cx=   7.92960, z=-9999.00000
   2.06213    2.37324    0.89350    4.29667  cx=  23.41133, z=   8.96852
  -0.41061    5.66114   -1.79996    2.86110  cx=  14.68454, z=   8.96852
  -2.33922    3.31667    2.20371    1.33717  cx=  16.76076, z=   8.96852
   2.51569    1.14168    1.38714   -0.54868  cx=  12.90763, z=   8.96852
   2.19221    5.93326    1.48589   -1.14267  cx=  25.84241, z=  14.90422
.......................................................................
   0.00350    9.34385    0.65380    0.00101  cx=  30.65576, z=  30.64666
   0.00065    9.33560    0.66299   -0.00319  cx=  30.65366, z=  30.64666
  -0.00233    9.32698    0.66508    0.00744  cx=  30.65146, z=  30.64666
   0.00564    9.32288    0.66461    0.00532  cx=  30.64897, z=  30.64666
   0.00410    9.33853    0.65410    0.00306  cx=  30.64632, z=  30.65073
   0.00258    9.34019    0.65755    0.00082  cx=  30.65757, z=  30.65073
   0.00066    9.33448    0.66385   -0.00201  cx=  30.65616, z=  30.65073
Optimal solution found in 119 iterations, Vol=   0.0000
  0.0021  9.3386  0.6593  0.0000
z= 30.6572
>>
```

For Exercise 5.9 with $n = 3$, $a = b = 2$ and $c = 5$, we get

```
>> clear
>> psdat
>> dcsel
 -85.49883   82.25058  208.06265  cx=  30.56848, z=-9999.00000
  11.12853  -64.29617  171.42596  cx=  87.34772, z=-9999.00000
   1.20391   12.52250  126.74039  cx= 156.60104, z=-9999.00000
 -10.39740    2.76268   74.50551  cx=  38.44128, z=-9999.00000
   1.90067   -6.57926   24.50713  cx=  18.95129, z=-9999.00000
   1.38161    3.95126   -6.60592  cx=   6.82304, z=-9999.00000
   0.58555    2.21549   17.22113  cx=  23.99432, z=-9999.00000
  -0.03957    0.85243    9.95152  cx=  11.49810, z=-9999.00000
  -0.53385   -0.22534   29.23529  cx=  26.64921, z=  19.92770
   1.33403   -1.70743   24.00886  cx=  25.93012, z=  19.92770
```

```
   0.98365   -3.93039   26.10890  cx=  22.18273, z=  19.92770
   0.15010    1.90863   13.56085  cx=  17.97850, z=  19.92770
  -0.29774    1.49264   22.51259  cx=  24.30692, z=  20.09956
  -0.92891    0.90634   24.64934  cx=  22.74638, z=  20.09956
   0.71750   -0.20129   18.03809  cx=  20.50551, z=  20.09956
   0.55850   -0.93332   23.94083  cx=  24.30821, z=  21.48207
   0.29328    1.24644   19.78129  cx=  23.44728, z=  21.48207
............................................................
  -0.00120   -0.00039   25.00568  cx=  25.00008, z=  24.99545
   0.00141   -0.00352   25.00029  cx=  24.99888, z=  24.99545
   0.00057    0.00211   24.99013  cx=  24.99661, z=  24.99545
  -0.00020    0.00127   24.99280  cx=  24.99452, z=  24.99545
Optimal solution found in 68 iterations, Vol=   0.0000
  0.0002  0.0017 24.9913
z= 24.9957
>>
```

For the data of Exercise 8.2(ii) we get

```
>> clear
>> psdat
>> dcsel
 194.74376  -26.28120  cx=  26.28120, z=-9999.00000
  -0.42637   25.28755  cx= -25.28755, z=-9999.00000
 -20.12962   -1.44852  cx=   1.44852, z= -14.24470
  34.80955    1.12224  cx=  -1.12224, z= -14.24470
  11.15640    9.73883  cx=  -9.73883, z=   0.00000
  -8.55908   -8.48696  cx=   8.48696, z=   0.00000
  14.71900   -5.87632  cx=   5.87632, z=   0.00000
   6.88480   -1.39700  cx=   1.39700, z=   0.00000
  14.14111    5.56176  cx=  -5.56176, z=   0.00000
   6.40835   -1.85392  cx=   1.85392, z=   0.00000
   8.98593    0.61797  cx=  -0.61797, z=   0.00000
   8.12674   -0.20599  cx=   0.20599, z=   0.00000
   8.41314    0.06866  cx=  -0.06866, z=   0.00000
   8.31767   -0.02289  cx=   0.02289, z=   0.00000
   8.34949    0.00763  cx=  -0.00763, z=   0.00000
   8.33889   -0.00254  cx=   0.00254, z=   0.00000
   8.34242    0.00085  cx=  -0.00085, z=   0.00000
   8.34124   -0.00028  cx=   0.00028, z=   0.00000
   8.34164    0.00009  cx=  -0.00009, z=   0.00000
   8.34151   -0.00003  cx=   0.00003, z=   0.00000
   8.34155    0.00001  cx=  -0.00001, z=   0.00000
   8.34153   -0.00000  cx=   0.00000, z=   0.00000
   8.34154    0.00000  cx=  -0.00000, z=   0.00000
   8.34154   -0.00000  cx=   0.00000, z=   0.00000
   8.34154    0.00000  cx=  -0.00000, z=   0.00000
```

```
   8.34154   -0.00000  cx=   0.00000, z=   0.00000
   8.34154    0.00000  cx=  -0.00000, z=   0.00000
   8.34154   -0.00000  cx=   0.00000, z=   0.00000
Optimal solution found in 27 iterations, Vol=   0.0000
 10.8264 -0.0000
z=  0.0000
>>
```

Exercise 9.6

(i) Let $\boldsymbol{H}_{k+1}$ be as defined in (9.45), i.e.,

$$\boldsymbol{H}_{k+1} = n\sqrt{\frac{1-\alpha_k^2}{n^2-1}}\boldsymbol{H}_k\left(\boldsymbol{I}_n - \left(1 - \sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right)\frac{(\boldsymbol{H}_k^T\boldsymbol{r})(\boldsymbol{r}^T\boldsymbol{H}_k)}{\|\boldsymbol{H}_k^T\boldsymbol{r}\|^2}\right) .$$

Like we proved (9.20) show that

$$det\boldsymbol{H}_{k+1} = \left(1+\frac{1}{n}\right)^{-\frac{n+1}{2}}\left(1-\frac{1}{n}\right)^{-\frac{n-1}{2}}(1-\alpha_k^2)^{\frac{n-1}{2}}(1-\alpha_k)\, det\boldsymbol{H}_k .$$

(ii) Define $\boldsymbol{G}_k = \boldsymbol{H}_k\boldsymbol{H}_k^T$ and show using (9.45) that

$$\boldsymbol{G}_{k+1} = \frac{n^2(1-\alpha_k^2)}{n^2-1}\boldsymbol{G}_k\left(\boldsymbol{I}_n - \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)}\frac{\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{G}_k}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}\right) ,$$

$$\boldsymbol{G}_{k+1}^{-1} = \frac{n^2-1}{n^2(1-\alpha_k^2)}\left(\boldsymbol{G}_k^{-1} + \frac{2(1+n\alpha_k)}{(n-1)(1-\alpha_k)}\frac{\boldsymbol{r}\boldsymbol{r}^T}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}\right) .$$

(iii) Show $\Gamma(1+n/2) = (n/2)!$ for all even $n \geq 1$, $\Gamma(1+n/2) = \dfrac{n!\sqrt{\pi}}{\lfloor\frac{n}{2}\rfloor!2^n}$ for all odd $n \geq 1$.

(i) Letting $\alpha = 1 - \sqrt{\dfrac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}$ the proof of (9.20) applies unchanged since $0 \leq \alpha_k < 1$. Thus replacing $\boldsymbol{F}_P$ by $\boldsymbol{H}_{k+1}$, $\boldsymbol{F}$ by $\boldsymbol{H}_k$ and $\boldsymbol{d}$ by $\boldsymbol{H}_k^T\boldsymbol{r}/\|\boldsymbol{H}_k^T\boldsymbol{r}\|$, since $\|\boldsymbol{d}\| = 1$ we get

$$det\boldsymbol{H}_{k+1} = \left(\frac{n^2(1-\alpha_k^2)}{n^2-1}\right)^{n/2}\sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\, det\boldsymbol{H}_k .$$

To bring this to the required form we have

$$\sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}} = \sqrt{\frac{(n-1)n(1-\alpha_k)^2}{(n+1)n(1-\alpha_k^2)}} = \left(1-\frac{1}{n}\right)^{1/2}\left(1+\frac{1}{n}\right)^{-1/2}(1-\alpha_k)(1-\alpha_k^2)^{-1/2}$$

and

$$\frac{n^2}{n^2-1}(1-\alpha_k^2) = \frac{n}{n+1}\frac{n}{n-1}(1-\alpha_k^2) = \left(1+\frac{1}{n}\right)^{-1}\left(1-\frac{1}{n}\right)^{-1}(1-\alpha_k^2)\ .$$

Thus after grouping of terms we get

$$det\boldsymbol{H}_{k+1} = \left(1+\frac{1}{n}\right)^{-\frac{n+1}{2}}\left(1-\frac{1}{n}\right)^{-\frac{n-1}{2}}(1-\alpha_k^2)^{\frac{n-1}{2}}(1-\alpha_k)det\boldsymbol{H}_k\ .$$

(ii) Using the definition of $\boldsymbol{H}_{k+1}$ we calculate:

$$\boldsymbol{G}_{k+1} = \boldsymbol{H}_{k+1}\boldsymbol{H}_{k+1}^T = n^2\frac{1-\alpha_k^2}{n^2-1}\left(\boldsymbol{H}_k - \left(1-\sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right)\frac{\boldsymbol{H}_k\boldsymbol{H}_k^T\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{H}_k}{\|\boldsymbol{H}_k^T\boldsymbol{r}\|^2}\right)$$
$$\left(\boldsymbol{H}_k^T - \left(1-\sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right)\frac{\boldsymbol{H}_k^T\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{H}_k\boldsymbol{H}_k^T}{\|\boldsymbol{H}_k^T\boldsymbol{r}\|^2}\right)$$
$$= n^2\frac{1-\alpha_k^2}{n^2-1}\left[\boldsymbol{H}_k\boldsymbol{H}_k^T - 2\left(1-\sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right)\left(\frac{\boldsymbol{H}_k\boldsymbol{H}_k^T\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{H}_k\boldsymbol{H}_k^T}{\|\boldsymbol{H}_k^T\boldsymbol{r}\|^2}\right)\right.$$
$$\left.+\left(1-\sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right)^2\frac{\boldsymbol{H}_k\boldsymbol{H}_k^T\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{H}_k\boldsymbol{H}_k^T\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{H}_k\boldsymbol{H}_k^T}{\|\boldsymbol{H}_k^T\boldsymbol{r}\|^4}\right]$$

and factoring out the term $\boldsymbol{r}^T\boldsymbol{H}_k\boldsymbol{H}_k^T\boldsymbol{r} = \|\boldsymbol{H}_k^T\boldsymbol{r}\|^2$ from the numerator of the last fraction

$$= \frac{n^2(1-\alpha_k^2)}{n^2-1}\left(\boldsymbol{G}_k - \left(2\left(1-\sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right) - \left(1-\sqrt{\frac{(n-1)(1-\alpha_k)}{(n+1)(1+\alpha_k)}}\right)^2\right)\frac{\boldsymbol{H}_k\boldsymbol{H}_k^T\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{H}_k\boldsymbol{H}_k^T}{\|\boldsymbol{H}_k^T\boldsymbol{r}\|^2}\right)$$
$$= \frac{n^2(1-\alpha_k^2)}{n^2-1}\left(\boldsymbol{G}_k - \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)}\frac{\boldsymbol{G}_k\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{G}_k}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}\right) = \frac{n^2(1-\alpha_k^2)}{n^2-1}\boldsymbol{G}_k\left(\boldsymbol{I}_n - \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)}\frac{\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{G}_k}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}\right)$$

For the inverse $\boldsymbol{G}_{k+1}^{-1}$ we have

$$\boldsymbol{G}_{k+1}^{-1} = \frac{n^2-1}{n^2(1-\alpha_k^2)}\left(\boldsymbol{I}_n - \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)}\frac{\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{G}_k}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}\right)^{-1}\boldsymbol{G}_k^{-1}\ .$$

Using Exercise 4.1 (ii) with $\boldsymbol{u} = -\frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)(\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r})}\boldsymbol{r}$ and $\boldsymbol{v} = \boldsymbol{G}_k\boldsymbol{r}$ we calculate $\boldsymbol{v}^T\boldsymbol{u} \neq -1$ because $\alpha_k \neq 1$ and thus

$$\left(\boldsymbol{I}_n - \frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)}\frac{\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{G}_k}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}\right)^{-1} = \boldsymbol{I}_n + \frac{1}{1-\frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)}\frac{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}}\frac{2(1+n\alpha_k)}{(n+1)(1+\alpha_k)}\frac{\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{G}_k}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}$$
$$= \boldsymbol{I}_n + \frac{2(1+n\alpha_k)}{n+n\alpha_k+1+\alpha_k-2-2n\alpha_k}\frac{\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{G}_k}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}$$
$$= \boldsymbol{I}_n + \frac{2(1+n\alpha_k)}{(n-1)(1-\alpha_k)}\frac{\boldsymbol{r}\boldsymbol{r}^T\boldsymbol{G}_k}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}\ .$$

It follows that
$$\boldsymbol{G}_{k+1}^{-1} = \frac{n^2-1}{n^2(1-\alpha_k^2)}\left(\boldsymbol{G}_k^{-1} + \frac{2(1+n\alpha_k)}{(n-1)(1-\alpha_k)}\frac{\boldsymbol{r}\boldsymbol{r}^T}{\boldsymbol{r}^T\boldsymbol{G}_k\boldsymbol{r}}\right) .$$

(iii) For $n > 1$ and even, i.e. $n = 2k$ with $k \geq 1$ is integer we have
$$\Gamma(1+n/2) = \Gamma(1+k) = k\Gamma(k) = k! = \left(\frac{n}{2}\right)!$$
where we have used elementary properties of the gamma function; see Chapter 7.7. For $n \geq 1$ and odd, i.e. $n = 2k+1$ where $k \geq 0$ is integer we have
$$\Gamma\left(\frac{2k+1}{2}\right) = \Gamma\left(\frac{2k-1}{2}+1\right) = \frac{2k-1}{2}\Gamma\left(\frac{2k-1}{2}\right) ,$$
where the last equality follows from the fact that $2k-1$ is even. Thus we get
$$\Gamma\left(1+\frac{n}{2}\right) = \Gamma\left(\frac{2k+3}{2}\right) = \frac{1}{2^{k+1}}\Gamma\left(\frac{1}{2}\right)\prod_{\ell=0}^{k}(2\ell+1) .$$
Using the identity $\prod_{\ell=0}^{k}(2\ell+1) = \frac{(2k+1)!}{k!2^k}$ and that $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ we get $\Gamma\left(1+\frac{n}{2}\right) = \frac{n!}{\lfloor\frac{n}{2}\rfloor!2^n}\sqrt{\pi}$.

Exercise 9.7

Let $P \subseteq \mathbb{R}^n$ be a polyhedron *and define $P_\varepsilon^\infty = \{\boldsymbol{z} \in \mathbb{R}^n : \exists \boldsymbol{x} \in P$ such that $\|\boldsymbol{x}-\boldsymbol{z}\|_\infty \leq \varepsilon\}$.*

(i) Show that P_ε^∞ is a polyhedron in $\mathbb{R}^n$ and that $P \neq \emptyset$ implies $dim P_\varepsilon^\infty = n$ for $\varepsilon > 0$.

(ii) Show that if $\boldsymbol{z} \in P_\varepsilon^\infty$ is an extreme point then there exists an extreme point $\boldsymbol{x} \in P$ such that $z_i - x_i = \pm\varepsilon$ for all $1 \leq i \leq n$.

(iii) Let $\boldsymbol{y} \in \mathbb{R}^n$. Show that $\boldsymbol{x} + (\boldsymbol{y}) \in P_\varepsilon^\infty$ for some $\boldsymbol{x} \in P_\varepsilon^\infty$ if and only if there exists $\widetilde{\boldsymbol{x}} \in P$ such that $\widetilde{\boldsymbol{x}} + (\boldsymbol{y}) \in P$. Show that the asymptotic cone C_∞ of P is the asymptotic cone of P_ε^∞.

(iv) Show that P_ε^∞ has at most $p2^n$ extreme points where p is the number of extreme points of P.

(v) Suppose that the facet complexity of P is ϕ and its vertex complexity is ν. Show that for rational $\varepsilon \geq 0$ the polyhedron P_ε^∞ has a vertex complexity of $2\nu + 2n\langle\varepsilon\rangle$ and a facet complexity of $3\phi + 2\langle\varepsilon\rangle$.

(vi) Define P_ε^1 by
$$P_\varepsilon^1 = \{\boldsymbol{z} \in \mathbb{R}^n : \exists \boldsymbol{x} \in P \text{ such that } \|\boldsymbol{x}-\boldsymbol{z}\|_1 \leq \varepsilon\} .$$
Show that P_ε^1 is a polyhedron, $\boldsymbol{hx} \leq h_0 + \varepsilon\|\boldsymbol{h}\|_\infty$ for all $\boldsymbol{x} \in P_\varepsilon^1$ if $\boldsymbol{hx} \leq h_0$ for all $\boldsymbol{x} \in P$ and $\boldsymbol{hx} \leq h_0 - \varepsilon\|\boldsymbol{h}\|_\infty$ for all $\boldsymbol{x} \in P$ if $\boldsymbol{hx} \leq h_0$ for all $\boldsymbol{x} \in P_\varepsilon^1$. (Hint: *Use that $\sum_{j=1}^n |x_j| \leq \varepsilon$ if and only if $\sum_{j=1}^n \delta_j x_j \leq \varepsilon$ for the 2^n vectors $(\delta_1, \ldots, \delta_n)$ with $\delta_j \in \{+1,-1\}$ for all $1 \leq j \leq n$.*)

(vii) *Show that if $\boldsymbol{z} \in P^1_\varepsilon$ is an extreme point then $\boldsymbol{z} = \boldsymbol{x} \pm \varepsilon \boldsymbol{u}^i$ where $\boldsymbol{x} \in P$ is an extreme point of P and $\boldsymbol{u}^i \in \mathbb{R}^n$ is the i-th unit vector for some $1 \le i \le n$. Show that P^1_ε has at most $2np$ extreme points if P has p extreme points.*

(iix) *Suppose P has a facet complexity of ϕ and a vertex complexity of ν. Show that P^1_ε has vertex complexity of $2\nu + 2\langle\varepsilon\rangle$ and a facet complexity of $3\phi + 2\langle\varepsilon\rangle$.*

(ix) *Prove that*

$$\text{if } P \neq \emptyset \text{ then } vol(P^1_\varepsilon) \ge \frac{2^n \varepsilon^n}{n!} > \frac{\varepsilon^n \pi^{n/2}}{n^n \Gamma(1+n/2)} \text{ for all } \varepsilon > 0\ .$$

(Hint: *Use that $P^1_\varepsilon \supseteq \boldsymbol{x} + \{\boldsymbol{z} \in \mathbb{R}^n : \|\boldsymbol{z}\|_1 \le 1\}$ and $P^1_\varepsilon \supseteq B(\boldsymbol{x}, r = \varepsilon/n)$ for every $\boldsymbol{x} \in P$.*)

(x) *Show that Remark 9.13 remains correct if we replace the ℓ_1-norm by the ℓ_∞-norm.*

(i) Let $S = \{\boldsymbol{x}^1, \dots, \boldsymbol{x}^p\}$, $T = \{\boldsymbol{r}^1, \dots, \boldsymbol{r}^t\}$ be any finite generator of P. Then every $\boldsymbol{x} \in P$ can be written as $\boldsymbol{x} = \boldsymbol{X}\boldsymbol{\mu} + \boldsymbol{R}\boldsymbol{\nu}$ where $\boldsymbol{X} = (\boldsymbol{x}^1 \dots \boldsymbol{x}^p)$ is an $n \times p$ matrix, $\boldsymbol{R} = (\boldsymbol{r}^1 \dots \boldsymbol{r}^t)$ an $n \times t$ matrix, $\boldsymbol{\mu} \ge \boldsymbol{0}$, $\boldsymbol{f}\boldsymbol{\mu} = 1$, $\boldsymbol{\nu} \ge \boldsymbol{0}$ and $\boldsymbol{f} = (1, \dots, 1) \in \mathbb{R}^p$. If P is pointed, then $\boldsymbol{x}^1, \dots, \boldsymbol{x}^p$ are the extreme points and $\boldsymbol{r}^1, \dots, \boldsymbol{r}^t$ are the direction vectors of the extreme rays of P. Let $\boldsymbol{e}^T = (1, \dots, 1) \in \mathbb{R}^n$. Then P^∞_ε is the image of the polyhedron

$$PP_\varepsilon = \{(\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\nu}) \in \mathbb{R}^{n+p+t} : -\varepsilon\, \boldsymbol{e} \le \boldsymbol{X}\boldsymbol{\mu} + \boldsymbol{R}\boldsymbol{\nu} - \boldsymbol{z} \le \varepsilon\, \boldsymbol{e}\,,\ \boldsymbol{f}\boldsymbol{\mu} = 1\,,\ \boldsymbol{\mu} \ge \boldsymbol{0}\,,\ \boldsymbol{\nu} \ge \boldsymbol{0}\}$$

under the linear transformation with the matrix $\boldsymbol{L} = (\boldsymbol{I}_n\ \mathbf{O}_p\ \mathbf{O}_t)$, i.e. P^∞_ε is the image of PP_ε when we project out the variables $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. By point 7.3(g) it follows that P^∞_ε is a polyhedron. To prove the dimension of P^∞_ε, suppose that $P \neq \emptyset$ and let $\boldsymbol{x} \in P$. Then the points $\boldsymbol{x}$, $\boldsymbol{x} + \varepsilon\, \boldsymbol{u}^1, \dots, \boldsymbol{x} + \varepsilon \boldsymbol{u}^n$ are $n+1$ affinely independent points in P^∞_ε where $\boldsymbol{u}^i$ is the i-th unit vector in $\mathbb{R}^n$. Thus $dim P^\infty_\varepsilon = n$.

(ii) Let $\boldsymbol{z} \in P^\infty_\varepsilon$ and $\boldsymbol{x} \in P$ be such that $z_i - x_i = \pm\varepsilon$ for all $1 \le i \le n$ and suppose that $\boldsymbol{z}$ is an extreme point of P^∞_ε but $\boldsymbol{x}$ is not an extreme point of P. It follows that $\boldsymbol{x} = \mu\boldsymbol{x}^1 + (1-\mu)\boldsymbol{x}^2$ where $0 < \mu < 1$, $\boldsymbol{x}^1, \boldsymbol{x}^2 \in P$ and $\boldsymbol{x}^1 \neq \boldsymbol{x}^2$, $\boldsymbol{x}^1 \neq \boldsymbol{x} \neq \boldsymbol{x}^2$. We then get $\boldsymbol{z} = \boldsymbol{x} \pm \varepsilon\, \boldsymbol{e} = \mu\boldsymbol{x}^1 + (1-\mu)\boldsymbol{x}^2 \pm (\mu + 1 - \mu)\varepsilon\, \boldsymbol{e} = \mu(\boldsymbol{x}^1 \pm \varepsilon\, \boldsymbol{e}) + (1-\mu)(\boldsymbol{x}^2 \pm \varepsilon\, \boldsymbol{e})$. From the definition of P_ε it follows that $\boldsymbol{z}^1 = \boldsymbol{x}^1 \pm \varepsilon\, \boldsymbol{e} \in P^\infty_\varepsilon$ and $\boldsymbol{z}^2 = \boldsymbol{x}^2 \pm \varepsilon\, \boldsymbol{e} \in P^\infty_\varepsilon$. Moreover, $\boldsymbol{z}^1 \neq \boldsymbol{z}^2$ and thus $\boldsymbol{z} = \mu\boldsymbol{z}^1 + (1-\mu)\boldsymbol{z}^2$ with $0 < \mu < 1$ contradicts the assumption that $\boldsymbol{z}$ is an extreme point of P^∞_ε.

(iii) From Remark 9.12 (ii) we have that the asymptotic cones of P and P^∞_ε are the same, since every *valid inequality* $\boldsymbol{h}\boldsymbol{x} \le h_0$ for one of the polyhedra gives rise to a valid inequality of the form $\boldsymbol{h}\boldsymbol{x} \le h'_0$ for the other and vice versa. An inequality is *valid* for P, if it is satisfied by every $\boldsymbol{x} \in P$. To prove the first part suppose that $(\boldsymbol{y}) \in P^\infty_\varepsilon$ is a halfline of P^∞_ε, i.e. $\boldsymbol{h}\boldsymbol{y} \le 0$ for all $\boldsymbol{h}$ such that $\boldsymbol{h}\boldsymbol{x} \le h_0$ is valid for P^∞_ε. By part (ii) of Remark 9.12 we have that $\boldsymbol{h}\boldsymbol{x} \le h_0 - \varepsilon\, \|\boldsymbol{h}\|_1$ is valid for P for all such $\boldsymbol{h}$. Suppose that $(\boldsymbol{y})$ is not a halfline of P, i.e., there exists a valid inequality $\boldsymbol{h}\boldsymbol{x} \le h_0$ for P^∞_ε such that for some $\boldsymbol{x} \in P$ there exists $\alpha > 0$ such that $\boldsymbol{h}(\boldsymbol{x} + \alpha\boldsymbol{y}) > h_0 - \varepsilon\, \|\boldsymbol{h}\|_1$. Then $\alpha\boldsymbol{h}\boldsymbol{y} > 0$ for some $\alpha > 0$, i.e., $\boldsymbol{h}\boldsymbol{y} > 0$ which contradicts the assumption that $(\boldsymbol{y})$ is a halfline of P^∞_ε. On the other hand, suppose that $(\boldsymbol{y})$ is a halfline of P. Then $\boldsymbol{h}\boldsymbol{y} \le 0$ for all $\boldsymbol{h}$ such that $\boldsymbol{h}\boldsymbol{x} \le h_0$ is valid for P. By point (ii) of Remark 9.12 we have that $\boldsymbol{h}\boldsymbol{x} \le h_0 + \varepsilon\, \|\boldsymbol{h}\|_1$ for all $\boldsymbol{x} \in P^1_\varepsilon$ for all such $(\boldsymbol{h}, h_0)$. Suppose that $(\boldsymbol{y})$ is not a halfline of P^∞_ε. Then there exists a valid inequality $\boldsymbol{h}\boldsymbol{x} \le h_0$ for P such that for some $\boldsymbol{x} \in P^\infty_\varepsilon$ there exists $\alpha > 0$ such that $\boldsymbol{h}(\boldsymbol{x} + \alpha\boldsymbol{y}) > h_0 + \varepsilon\, \|\boldsymbol{h}\|_1$. Consequently we have $h_0 \le h_0 + \varepsilon\,\|\boldsymbol{h}\|_1 < \boldsymbol{h}(\boldsymbol{x}+\alpha\boldsymbol{y}) \le h_0 + \alpha\boldsymbol{hy}$, i.e. $\alpha\boldsymbol{hy} > 0$ for some $\alpha > 0$ which contradicts the assumption that $(\boldsymbol{y})$ is a halfline of P.

(iv) From part (ii) we have that for each extreme point $\boldsymbol{x}$ of P there exists an extreme point $\boldsymbol{z}$ of P_ε^∞ such that $z_i = x_i \pm \varepsilon$, for $1 \le i \le n$. That is each component of $\boldsymbol{x}$ can be either increased or decreased by ε. So for each extreme point of P there are at most 2^n different points z that can be constructed as above. Thus if P has p extreme points, then P_ε^∞ has at most $p2^n$ extreme points.

(v) From Remark 9.12 (ii) we have that $\boldsymbol{hz} \le h_0 + \varepsilon$ is a valid inequality for P_ε^∞ if $\boldsymbol{hx} \le h_0$ is valid for P. Thus for the facet complexity of P_ε^∞ we calculate

$$\langle\boldsymbol{h}\rangle+\langle h_0+\varepsilon\,\|\boldsymbol{h}\|_1\rangle = \langle\boldsymbol{h}\rangle+2\langle h_0\rangle+2\varepsilon+2\langle\|\boldsymbol{h}\|_1\rangle \le 3\langle\boldsymbol{h}\rangle+2\langle h_0\rangle+2\langle\varepsilon\rangle+2(1-n) \le 3(\langle\boldsymbol{h}\rangle+\langle h_0\rangle)+2\langle\varepsilon\rangle \le 3\phi+2\langle\varepsilon\rangle$$

where we have used $\langle\|\boldsymbol{h}\|_1\rangle \le \langle\boldsymbol{h}\rangle - n + 1$; see Chapter 7.5. For the vertex complexity, from part (ii) of this exercise we have that if $\boldsymbol{z}$ is an extreme point of P_ε^∞ then $z_i = x_i \pm \varepsilon$ for $1 \le i \le n$ where $\boldsymbol{x}$ is an extreme point of P. We calculate

$$\langle\boldsymbol{z}\rangle = \sum_{i=1}^n\langle z_i\rangle = \sum_{i=1}^n\langle x_i \pm \varepsilon\rangle \le 2\sum_{i=1}^n(\langle x_i\rangle + \langle\varepsilon\rangle) \le 2\sum_{i=1}^n\langle x_i\rangle + 2n\langle\varepsilon\rangle = 2\langle\boldsymbol{x}\rangle + 2n\langle\varepsilon\rangle \le 2\nu + 2n\langle\varepsilon\rangle\ .$$

(vi) Let $\boldsymbol{\Delta} = (\boldsymbol{\delta}^k)_{k=1}^{2^n}$ be the matrix with rows the 2^n vectors with components ± 1. The constraint $\|\boldsymbol{x}-\boldsymbol{z}\|_1 \le \varepsilon$ is written as $\|\boldsymbol{x}-\boldsymbol{z}\|_1 = \sum_{i=1}^n |x_i - z_i| \le \varepsilon$ and it is equivalent to the constraints $\boldsymbol{\Delta}(\boldsymbol{x}-\boldsymbol{z}) \le \varepsilon\,\boldsymbol{e}$; see also Exercise 2.2(ii). So we have that

$$P_\varepsilon^1 = \{\boldsymbol{z} \in \mathbb{R}^n : \exists\boldsymbol{x} \in P \text{ with } \boldsymbol{\Delta}(\boldsymbol{x}-\boldsymbol{z}) \le \varepsilon\,\boldsymbol{e}\}\ .$$

Since P is a polyhedron, let like in part (i) $\boldsymbol{X}$, $\boldsymbol{R}$ be the matrices corresponding to a finite generator of P. Then

$$\boldsymbol{x} = \boldsymbol{X\mu} + \boldsymbol{R\nu} \text{ where } \boldsymbol{\mu} \ge \boldsymbol{0}\,,\ \boldsymbol{f\mu} = 1\,,\ \boldsymbol{\nu} \ge \boldsymbol{0}$$

and $\boldsymbol{f} = (1,\dots,1)^T \in \mathbb{R}^p$. P_ε^1 is the image of the polyhedron

$$PP_\varepsilon^1 = \{(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\nu}) \in \mathbb{R}^{n+p+t} : \boldsymbol{\Delta}(\boldsymbol{X\mu}+\boldsymbol{R\nu}-\boldsymbol{z}) \le \varepsilon\,\boldsymbol{e}\,,\ \boldsymbol{f\mu} = 1\,,\ \boldsymbol{\mu} \ge \boldsymbol{0}\,,\ \boldsymbol{\nu} \ge \boldsymbol{0}\}$$

under the linear transformation $\boldsymbol{L} = (\boldsymbol{I}_n\ \mathbf{O}_p\ \mathbf{O}_t)$ where $\mathbf{O}_p$ is $n \times p$ and $\mathbf{O}_t$ is $n \times t$, i.e. when we project out $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$. It follows by point 7.3(g) that P_ε^1 is a polyhedron.

Suppose now that $\boldsymbol{z} \in P_\varepsilon^1$. Then there exists $\boldsymbol{x} \in P$ such that $\sum_{i=1}^n |x_i - z_i| \le \varepsilon$. We calculate $\boldsymbol{hz}-\boldsymbol{hx} = \boldsymbol{h}(\boldsymbol{x}-\boldsymbol{z}) \le \sum_{i=1}^n |h_i||z_i - x_i| \le \sum_{i=1}^n \|\boldsymbol{h}\|_\infty|z_i - x_i| \le \varepsilon\,\|\boldsymbol{h}\|_\infty$. Since for every $\boldsymbol{x} \in P$ we have $\boldsymbol{hx} \le h_0$ it follows that $\boldsymbol{hz} \le h_0 + \varepsilon\,\|\boldsymbol{h}\|_\infty$ for all $\boldsymbol{z} \in P_\varepsilon^1$. On the other hand, suppose that $\boldsymbol{hz} \le h_0$ for all $\boldsymbol{z} \in P_\varepsilon^1$ but there exists $\boldsymbol{x} \in P$ such that $\boldsymbol{hx} > h_0 - \varepsilon\,\|\boldsymbol{h}\|_\infty$. Let $\boldsymbol{z}$ be such that $z_i = x_i$ for all $i \ne k$ and $z_k = x_k + \varepsilon$ if $h_k \ge 0$, $z_k = x_k - \varepsilon$ if $h_k < 0$, where $1 \le k \le n$ is such that $|h_k| = \|\boldsymbol{h}\|_\infty$ and $1 \le i \le n$. Thus if $h_k \ge 0$ we have

$$\boldsymbol{hz} = \boldsymbol{hx} + h_k\varepsilon = \boldsymbol{hx} + \varepsilon\,\|\boldsymbol{h}\|_\infty > h_0 - \varepsilon\,\|\boldsymbol{h}\|_\infty + \varepsilon\,\|\boldsymbol{h}\|_\infty = h_0$$

and if $h_k < 0$ that

$$\boldsymbol{hz} = \boldsymbol{hx} - \varepsilon\,h_k = \boldsymbol{hx} + \varepsilon\,\|\boldsymbol{h}\|_\infty > h_0 - \varepsilon\,\|\boldsymbol{h}\|_\infty + \varepsilon\,\|\boldsymbol{h}\|_\infty = h_0\ ,$$

i.e., $\boldsymbol{hz} > h_0$ for some $\boldsymbol{z} \in P$ which is a contradiction.

(vii) Suppose that $\boldsymbol{z} = \boldsymbol{x} \pm \varepsilon\, \boldsymbol{u}^i$ is an extreme point of P^1_ε but $\boldsymbol{x} \in P$ is not an extreme point of P. Then there exist $0 < \mu < 1$ and $\boldsymbol{x}^1 \neq \boldsymbol{x} \neq \boldsymbol{x}^2 \neq \boldsymbol{x}^1$ such that $\boldsymbol{x} = \mu\boldsymbol{x}^1 + (1-\mu)\boldsymbol{x}^2$. But then we have

$$\begin{aligned}\boldsymbol{z} &= \boldsymbol{x} \pm \varepsilon\, \boldsymbol{u}^i = \mu\boldsymbol{x}^1 + (1-\mu)\boldsymbol{x}^2 \pm \varepsilon\, \boldsymbol{u}^i = \mu\boldsymbol{x}^1 + (1-\mu)\boldsymbol{x}^2 \pm \mu\varepsilon\, \boldsymbol{u}^i \pm (1-\mu)\varepsilon\, \boldsymbol{u}^i \\ &= \mu(\boldsymbol{x}^1 \pm \varepsilon\, \boldsymbol{u}^i) + (1-\mu)(\boldsymbol{x}^2 \pm \varepsilon\, \boldsymbol{u}^i) = \mu\boldsymbol{z}^1 + (1-\mu)\boldsymbol{z}^2\end{aligned}$$

where $\boldsymbol{z}^k = \boldsymbol{x}^k \pm \varepsilon\boldsymbol{u}^i$ for $k = 1, 2$. From the definition of P^1_ε it follows that $\boldsymbol{z}^1, \boldsymbol{z}^2 \in P^1_\varepsilon$ and since $\boldsymbol{z}^1 \neq \boldsymbol{z}^2$, because $\boldsymbol{x}^1 \neq \boldsymbol{x}^2$, we have that $\boldsymbol{z}$ is the convex combination of two points in P^1_ε which contradicts the assumption that $\boldsymbol{z}$ is an extreme point.

For each extreme point $\boldsymbol{x} \in P$ there are $2n$ distinct points of the form $\boldsymbol{z} = \boldsymbol{x} \pm \varepsilon\, \boldsymbol{u}^i$. Thus if P has p extreme points, P^1_ε has at most $2np$ extreme points.

(iix) From the previous part we have that the extreme points of P^1_ε are of the form $\boldsymbol{z} = \boldsymbol{x} \pm \varepsilon\, \boldsymbol{u}^i$ where $\boldsymbol{x}$ is an extreme point of P. Since the vertex complexity of P is $\langle \boldsymbol{x} \rangle \leq \nu$, we compute

$$\langle \boldsymbol{z} \rangle = \langle \boldsymbol{x} \pm \varepsilon\, \boldsymbol{u}^i \rangle = \langle \sum_{\substack{j=1 \\ j\neq i}}^{n} x_j \rangle + \langle x_i \pm \varepsilon \rangle \leq 2\sum_{j=1}^{n} \langle x_j \rangle + 2\langle \varepsilon \rangle = 2\langle \boldsymbol{x} \rangle + 2\langle \varepsilon \rangle \leq 2\nu + 2\langle \varepsilon \rangle \ .$$

For the facet complexity we have that for any valid inequality $\boldsymbol{hz} \leq h_0 + \varepsilon\, \|\boldsymbol{h}\|_\infty$ we have $\langle \boldsymbol{h} \rangle + \langle h_0 \rangle \leq \phi$ and thus we compute

$$\begin{aligned}\langle \boldsymbol{h} \rangle + \langle h_0 + \varepsilon\, \|\boldsymbol{h}\|_\infty \rangle &\leq \langle \boldsymbol{h} \rangle + 2\langle h_0 \rangle + 2\langle \varepsilon \rangle + 2\langle \|\boldsymbol{h}\|_\infty \rangle \leq 2(\langle \boldsymbol{h} \rangle + \langle h_0 \rangle) + 2\langle \varepsilon \rangle + \langle \|\boldsymbol{h}\|_\infty \rangle \\ &\leq 2(\langle \boldsymbol{h} \rangle + \langle h_0 \rangle) + 2\langle \varepsilon \rangle + \langle \boldsymbol{h} \rangle \leq 3(\langle \boldsymbol{h} \rangle + \langle h_0 \rangle) + 2\langle \varepsilon \rangle \leq 3\phi + 2\langle \varepsilon \rangle \ .\end{aligned}$$

(ix) Since $P^1_\varepsilon \supseteq \boldsymbol{x} + \{\boldsymbol{z} \in \mathbb{R}^n : \|\boldsymbol{z}\|_1 \leq 1\}$ and $P^1_\varepsilon \supseteq B(\boldsymbol{x}, \varepsilon/n)$ we have like in (9.55)

$$vol(P^1_\varepsilon) \geq vol(B^1) > vol(B(\boldsymbol{x}, \varepsilon/n)) = \frac{\pi^{n/2}\varepsilon^n}{\Gamma(1+n/2)n^n} \ .$$

Thus, we have to show that the volume of the sphere in ℓ_1-norm, B^ε is given by $2^n\varepsilon^n/n!$. To this end, it suffices to show that $V_n = vol(B^1) = 2^n/n!$. We have

$$\begin{aligned}V_n &= \int_{-1}^{1} V_{n-1}(1-|x_n|)dx_n = \int_{-1}^{1} f_{n-1}(1-|x_n|)^{n-1}dx_n = f_{n-1}\sum_{j=0}^{n-1}(-1)^j\binom{n-1}{j}\int_{-1}^{1}|x_n|^j dx_n \\ &= f_{n-1}\sum_{j=0}^{n-1}(-1)^j\binom{n-1}{j}\frac{2}{j+1} = 2f_{n-1}\sum_{j=0}^{n-1}(-1)^j\frac{(n-1)!}{j!(n-1-j)!}\frac{1}{j+1} \\ &= 2f_{n-1}\sum_{j=0}^{n-1}(-1)^j\frac{n!}{(j+1)!(n-1-j)!}\frac{1}{n} = 2f_{n-1}\sum_{j=0}^{n-1}(-1)^j\binom{n}{j+1}\frac{1}{n} = 2\frac{f_{n-1}}{n}\sum_{\ell=1}^{n}(-1)^{\ell-1}\binom{n}{\ell} \\ &= 2\frac{f_{n-1}}{n}(-1)\sum_{\ell=1}^{n}(-1)^\ell\binom{n}{\ell} = -2\frac{f_{n-1}}{n}\left(\sum_{\ell=0}^{n}(-1)^\ell\binom{n}{\ell} - (-1)^0\binom{n}{0}\right) \\ &= -2\frac{f_{n-1}}{n}((1-1)^n - 1) = 2\frac{f_{n-1}}{n} \ .\end{aligned}$$

So we have $f_n = \frac{2}{n} f_{n-1}$ and thus $f_n = 2^n/n!$.

(x) The proof of Remark 9.13 goes through unchanged when the vectors $\boldsymbol{d}^1$, $\boldsymbol{d}^2$ are the vectors of the ℓ_∞-norms (rather than the ℓ_1-norms) of the corresponding rows of $\boldsymbol{H}_1$, $\boldsymbol{H}_2$. This follows because the estimation (7.17) applies as well to $\|\boldsymbol{h}^i\|_\infty$ and thus the assertion follows.

---

*Exercise 9.8

Let $\|\cdot\|_P$ be a polytopal norm on $\mathbb{R}^n$, i.e., $\|\cdot\|_P$ is a norm in $\mathbb{R}^n$ and its "unit sphere" $B_P = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_P \le 1\}$ is a polytope.

(i) Show that the "dual norm" $\|\boldsymbol{y}\|_P^* = \max\{\boldsymbol{y}^T\boldsymbol{x} : \|\boldsymbol{x}\|_P \le 1\}$ is a norm on $\mathbb{R}^n$.

(ii) Let $\widetilde{\boldsymbol{a}}^i\boldsymbol{x} \le b_i$ for $1 \le i \le m$ be any linear description of $B_P = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_P \le 1\}$ and $\boldsymbol{a}^i = \widetilde{\boldsymbol{a}}^i/\|\widetilde{\boldsymbol{a}}^i\|_P^*$ for $1 \le i \le m$. Show that $B_P = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^i\boldsymbol{x} \le 1\,,\ 1 \le i \le m\}$, that $\boldsymbol{0} \in \mathrm{relint} B_P$ and $\dim B_P = n$.

(iii) Show that $\|\cdot\|_P^*$ is a polytopal norm on $\mathbb{R}^n$.

(iv) Prove Hölder's inequality $\boldsymbol{y}^T\boldsymbol{x} \le \|\boldsymbol{y}\|_P^*\|\boldsymbol{x}\|_P$.

(v) Let P_ε^P be defined by $P_\varepsilon^P = \{\boldsymbol{z} \in \mathbb{R}^n : \exists \boldsymbol{x} \in P$ such that $\|\boldsymbol{z} - \boldsymbol{x}\|_P \le \varepsilon\}$. Show that P_ε^P is a polyhedron. Show that $\boldsymbol{h}\boldsymbol{x} \le h_0 + \varepsilon\,\|\boldsymbol{h}\|_P^*$ for all $\boldsymbol{x} \in P_\varepsilon^P$ if $\boldsymbol{h}\boldsymbol{x} \le h_0$ for all $\boldsymbol{x} \in P$. Show that $\boldsymbol{h}\boldsymbol{x} \le h_0 - \varepsilon\,\|\boldsymbol{h}\|_P^*$ for all $\boldsymbol{x} \in P$ if $\boldsymbol{h}\boldsymbol{x} \le h_0$ for all $\boldsymbol{x} \in P_\varepsilon^P$.

(i) For any $\boldsymbol{y} \in \mathbb{R}^n$, let $\boldsymbol{y}' = \boldsymbol{y}/\|\boldsymbol{y}\|_P$. Then $\boldsymbol{y} \in \mathbb{R}^n$, by the homogeneity of $\|\cdot\|_P$ $\|\boldsymbol{y}'\|_P \le 1$ and $\|\boldsymbol{y}'\|_P \le 1$ and thus from the definition of $\|\cdot\|_P$ we have that $\|\boldsymbol{y}\|_P^* \ge \boldsymbol{y}^T\boldsymbol{y}' = \boldsymbol{y}^T\boldsymbol{y}/\|\boldsymbol{y}\|_P = \|\boldsymbol{y}\|^2/\|\boldsymbol{y}\|_P$ where $\|\boldsymbol{y}\|$ is the Euclidean norm of $\boldsymbol{y}$, i.e. $\|\boldsymbol{y}\|_P^*$ is the ratio of two nonnegative numbers and thus it is nonnegative which is zero if and only if $\|\boldsymbol{y}\| = 0$, i.e. if and only if $\boldsymbol{y} = \boldsymbol{0}$. The homogeneity follows trivially since $\|\alpha\boldsymbol{y}\|_P^* = \max\{\alpha\boldsymbol{y}^T\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^n\,,\ \|\boldsymbol{x}\|_P \le 1\} = \alpha\max\{\boldsymbol{y}^T\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^n\,,\ \|\boldsymbol{x}\|_P \le 1\} = \alpha\|\boldsymbol{y}\|_P^*$ for all $\boldsymbol{y} \in \mathbb{R}^n$ and $\alpha \ge 0$. Finally we have $\|\boldsymbol{y}+\boldsymbol{z}\|_P^* = \max\{(\boldsymbol{y}+\boldsymbol{z})^T\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^n\,,\ \|\boldsymbol{x}\|_P \le 1\} \le \max\{\boldsymbol{y}^T\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^n\,,\ \|\boldsymbol{x}\|_P \le 1\} + \max\{\boldsymbol{z}^T\boldsymbol{x} : \boldsymbol{x} \in \mathbb{R}^n\,,\ \|\boldsymbol{x}\|_P \le 1\} = \|\boldsymbol{y}\|_P^* + \|\boldsymbol{z}\|_P^*$, i.e. the triangle inequality holds and thus $\|\cdot\|_P^*$ is a norm.

(ii) We are given that $B_P = \{\boldsymbol{x} \in \mathbb{R}^n : \widetilde{\boldsymbol{a}}^i\boldsymbol{x} \le b_i$ for $i = 1, \ldots, m\}$. From the definition (9.57) of the dual norm we have $\|\boldsymbol{a}^i\|_P^* = \max\{\boldsymbol{a}^i\boldsymbol{x} : \boldsymbol{x} \in B_P\} \le b_i$ and thus $\widetilde{\boldsymbol{a}}^i\boldsymbol{x} \le \|\widetilde{\boldsymbol{a}}^i\|_P^*$ for all $\boldsymbol{x} \in B_P$ and all $1 \le i \le m$. Since $\|\cdot\|_P^* \ge 0$ and $\widetilde{\boldsymbol{a}}^i \ne \boldsymbol{0}$ we have that $\|\widetilde{\boldsymbol{a}}^i\|_P^* > 0$ and thus dividing by $\|\widetilde{\boldsymbol{a}}^i\|_P^*$ we get $\boldsymbol{a}^i\boldsymbol{x} \le 1$ for all $\boldsymbol{x} \in B_P$ and $1 \le i \le m$, i.e. $B_P = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{a}^i\boldsymbol{x} \le 1$ for $1 \le i \le m\}$. Since $\boldsymbol{a}^i\boldsymbol{0} < 1$ for $1 \le i \le m$, $\boldsymbol{0} \in$ *relint*B_P. Let $\|\boldsymbol{a}\| = \max\{\|\boldsymbol{a}^i\| : 1 \le i \le m\}$ and $\varepsilon = 1/\|\boldsymbol{a}\|$. By the Cauchy-Schwarz inequality, $\boldsymbol{a}^i\boldsymbol{x} \le \|\boldsymbol{a}^i\|\|\boldsymbol{x}\| \le \|\boldsymbol{a}^i\|/\|\boldsymbol{a}\| \le 1$ for $1 \le i \le m$ and for all $\boldsymbol{x} \in \mathbb{R}^n$ with $\|\boldsymbol{x}\| \le \varepsilon$. Consequently, $B_P \supseteq B(\boldsymbol{0}, r = \varepsilon)$, i.e., B_P contains a ball with center $\boldsymbol{0} \in \mathbb{R}^n$ and radius $r = \varepsilon > 0$, and thus $\dim B_P = n$.

(iii) Since $\|\cdot\|_P$ is a polytopal norm the set $B_P = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_P \le 1\}$ is a polytope with $\dim B_P = n$ and $\boldsymbol{0} \in$ *relint*B_P. Let $S_P = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^p\}$ be the set of the extreme points of B_P and $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}$ be a linear description of B_P. Then we have that $\|\boldsymbol{y}\|_P^* = \max\{\boldsymbol{y}^T\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}\} = \boldsymbol{y}^T\boldsymbol{x}'$ where $\boldsymbol{x}' \in S_P$.

Thus $B_P^* = \{\boldsymbol{y} \in \mathbb{R}^n : \|\boldsymbol{y}\|_P^* \le 1\} = \{\boldsymbol{y} \in \mathbb{R}^n : \boldsymbol{y}^T\boldsymbol{x}^i \le 1$ for all $1 \le i \le p\}$. Since $\dim B_P = n$ and $\boldsymbol{0} \in$ *relint*B_P it follows that there exists $\boldsymbol{\mu} \in \mathbb{R}^p$, $\boldsymbol{\mu} \ge \boldsymbol{0}$ such that $\sum_{i=1}^p \mu_i \boldsymbol{x}^i = \pm\boldsymbol{u}^i$ where $\boldsymbol{u}^i$ is the i-th unit vector in $\mathbb{R}^n$ and $1 \le i \le n$. Consequently, by the duality theorem of linear programming $0 \le \max\{\boldsymbol{y}^T(\pm\boldsymbol{u}^i) : \boldsymbol{y}^T\boldsymbol{x}^i \le 1$ for $i = 1,\dots,p\} = \min\{\boldsymbol{e}^T\boldsymbol{\mu} : \sum_{i=1}^p \mu_i\boldsymbol{x}^i = \pm\boldsymbol{u}^i, \boldsymbol{\mu} \ge \boldsymbol{0}\} < \infty$ for $1 \le i \le n$ and thus B_P^* is a bounded set of $\mathbb{R}^n$, i.e., the "unit sphere" in the $\|\cdot\|_P^*$-norm is a polytope.

(iv) The inequality holds trivially as equality if $\boldsymbol{x} = \boldsymbol{0}$. From the homogeneity of $\|\cdot\|_P$ we have that for every $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{x} \ne \boldsymbol{0}$, $\boldsymbol{x}' = \boldsymbol{x}/\|\boldsymbol{x}\|_P$ satisfies $\boldsymbol{x}' \in \mathbb{R}^n$ and $\|\boldsymbol{x}'\|_P \le 1$. Thus from the definition (9.57) of the dual norm it follows that $\boldsymbol{y}^T\boldsymbol{x}' \le \|\boldsymbol{y}\|_P^*$, i.e. $\boldsymbol{y}^T\boldsymbol{x}/\|\boldsymbol{x}\|_P \le \|\boldsymbol{y}\|_P^*$ and since $\|\boldsymbol{x}\|_P > 0$ we get $\boldsymbol{y}^T\boldsymbol{x} \le \|\boldsymbol{y}\|_P^*\|\boldsymbol{x}\|_P$ for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^n$.

(v) Let $P_\varepsilon^P = \{\boldsymbol{z} \in \mathbb{R}^n : \exists \boldsymbol{x} \in P$ such that $\|\boldsymbol{z} - \boldsymbol{x}\|_P \le \varepsilon\}$. Since $\|\cdot\|_P$ is a polytopal norm there exists a linear description of the set $\{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_P \le \varepsilon\}$ and thus, like in the proof of Remark 9.12, P_ε^P is the image of a polyhedron and thus a polyhedron itself. Suppose that $\boldsymbol{h}\boldsymbol{x} \le h_0$ for all $\boldsymbol{x} \in P$ and let $\boldsymbol{z} \in P_\varepsilon^P$. Then there exists $\boldsymbol{x} \in P$ such that $\|\boldsymbol{z} - \boldsymbol{x}\|_P \le \varepsilon$. From Hölder's inequality we get $\boldsymbol{h}(\boldsymbol{z}-\boldsymbol{x}) \le \|\boldsymbol{h}\|_P^*\|\boldsymbol{z}-\boldsymbol{x}\|_P \le \varepsilon\|\boldsymbol{h}\|_P^*$ and thus $\boldsymbol{h}\boldsymbol{z} \le \boldsymbol{h}\boldsymbol{x} + \varepsilon\|\boldsymbol{h}\|_P^* \le h_0 + \varepsilon\|\boldsymbol{h}\|_P^*$ for all $\boldsymbol{z} \in P_\varepsilon^P$. Suppose now that $\boldsymbol{h}\boldsymbol{z} \le h_0$ for all $\boldsymbol{z} \in P_\varepsilon^P$ and let $\boldsymbol{x}^* \in \mathbb{R}^n$ be such that $\|\boldsymbol{h}\|_P^* = \max\{\boldsymbol{h}\boldsymbol{x} : \|\boldsymbol{x}\|_P \le 1\} = \boldsymbol{h}\boldsymbol{x}^*$. Assume that there exists $\boldsymbol{y} \in P$ such that $\boldsymbol{h}\boldsymbol{y} > h_0 - \varepsilon\|\boldsymbol{h}\|_P^*$. Let $\boldsymbol{z} = \boldsymbol{y} + \varepsilon\boldsymbol{x}^*$ and thus $\|\boldsymbol{z} - \boldsymbol{y}\|_P = \varepsilon\|\boldsymbol{x}^*\|_P \le \varepsilon$, i.e., $\boldsymbol{z} \in P_\varepsilon^P$. But then $\boldsymbol{h}\boldsymbol{z} = \boldsymbol{h}\boldsymbol{y} + \varepsilon\boldsymbol{h}\boldsymbol{x}^* > h_0 - \varepsilon\|\boldsymbol{h}\|_P^* + \varepsilon\|\boldsymbol{h}\|_P^* = h_0$ is a contradiction.

---

*Exercise 9.9

Let $0 < \Theta < 1$.

(i) Show that $\Theta \le \lfloor\Theta^{-1}\rfloor^{-1} < 2\Theta$ *and* $\Theta\lfloor\Theta^{-1}\rfloor > 1 - \Theta$.

(ii) Show that $\lfloor\Theta^{-1}\rfloor^{-1}$ *is a best approximation to* Θ *for all* $1 \le q < \lfloor\Theta^{-1}\rfloor$.

(iii) Suppose $1/2 < \Theta < 1$. *Show that* r/s *where* $r = \lfloor\frac{\Theta}{1-\Theta}\rfloor$ *and* $s = \lfloor\frac{\Theta}{1-\Theta}\rfloor + 1$ *is a best approximation to* Θ *for all* $1 \le q < s$.

(iv) Show that $q_n \ge 2^{(n-1)/2}$ *for* $n \ge 2$ *for the integers* q_n *generated by the inductive process.*

(v) Suppose $0 < \Theta \ne \Theta' < 1$ *and that the integers* p_n, q_n *and* p'_n, q'_n *generated by the respective inductive processes are such that* $p_n = p'_n$ *and* $q_n = q'_n$ *for* $1 \le n \le N$, *say. Show that* $|\Theta - \Theta'| \le 2^{-N+1}$.

(i) From the definition of the lower integer part of a number we have $\lfloor\Theta^{-1}\rfloor \le \Theta^{-1}$ and since $0 < \Theta < 1$, we get $\Theta \le \lfloor\Theta^{-1}\rfloor^{-1}$ and the left part of the first inequality follows. For the right part we have: $\Theta^{-1} < \lfloor\Theta^{-1}\rfloor + 1 \le \lfloor\Theta^{-1}\rfloor + \lfloor\Theta^{-1}\rfloor = 2\lfloor\Theta^{-1}\rfloor$ and thus $\lfloor\Theta^{-1}\rfloor^{-1} < 2\Theta$, where we have used the definition of the lower integer part of a number in the first inequality and the inequality $\lfloor\Theta^{-1}\rfloor \ge 1$ in the second. To prove that $\Theta\lfloor\Theta^{-1}\rfloor > 1 - \Theta$ we have from the definition of the lower integer part of a number that $\Theta^{-1} < \lfloor\Theta^{-1}\rfloor + 1$. Multiplying both sides by $\Theta > 0$ we get $1 < \Theta\lfloor\Theta^{-1}\rfloor + \Theta$ and thus $\Theta\lfloor\Theta^{-1}\rfloor > 1 - \Theta$.

(ii) Applying the definition (9.60) of the best approximation with $p = 1$ and $D = \lfloor\Theta^{-1}\rfloor$ we have to show that (a) $[\Theta\lfloor\Theta^{-1}\rfloor] = |\Theta\lfloor\Theta^{-1}\rfloor - 1|$ and (b) $[q\Theta] > [\Theta\lfloor\Theta^{-1}\rfloor]$ for all $1 \le q < \lfloor\Theta^{-1}\rfloor$. From the first inequality of part (i) we have that $\frac{1}{2} < \Theta\lfloor\Theta^{-1}\rfloor \le 1$ and thus $[\Theta\lfloor\Theta^{-1}\rfloor] = \lceil\Theta\lfloor\Theta^{-1}\rfloor\rceil - \Theta\lfloor\Theta^{-1}\rfloor < \Theta\lfloor\Theta^{-1}\rfloor - \lfloor\Theta\lfloor\Theta^{-1}\rfloor\rfloor$, i.e. $[\Theta\lfloor\Theta^{-1}\rfloor] = 1 - \Theta\lfloor\Theta^{-1}\rfloor$ and (a) follows. To prove (b) we have to show that $[q\Theta] > 1 - \Theta\lfloor\Theta^{-1}\rfloor$ for all $1 \le q < \lfloor\Theta^{-1}\rfloor$. First we note that $q\Theta < \lfloor\Theta^{-1}\rfloor\Theta$ for all $1 \le q < \lfloor\Theta^{-1}\rfloor$ since $\Theta > 0$. Thus, in particular, $0 < q\Theta < 1$ and $\lfloor q\Theta\rfloor = 0$, $\lceil q\Theta\rceil = 1$. If $[q\Theta] = 1 - q\Theta$ then we have $[q\Theta] = 1 - q\Theta > 1 - \lfloor\Theta^{-1}\rfloor\Theta$. If $[q\Theta] = q\Theta$, we have $[q\Theta] = q\Theta \ge \Theta > 1 - \Theta\lfloor\Theta^{-1}\rfloor$ where we have used the second inequality of part (i) and thus (b) follows, and the proof of (ii) is complete.

(iii) To show that $\frac{r}{s}$ with $r = \lfloor\frac{\Theta}{1-\Theta}\rfloor$, $s = \lfloor\frac{\Theta}{1-\Theta}\rfloor + 1$ is a best approximation to Θ for all $1 \le q < s$ if $1/2 < \Theta < 1$ we have to prove

(a) $[s\Theta] = |s\Theta - r|$ and (b) $[q\Theta] > [s\Theta]$ for all integer q with $1 \le q < s$.

We observe first, from $\frac{\Theta}{1-\Theta} \ge \lfloor\frac{\Theta}{1-\Theta}\rfloor$ and $1 - \Theta > 0$, that $s\Theta \ge \lfloor\frac{\Theta}{1-\Theta}\rfloor$ and thus $\lfloor s\Theta\rfloor \ge \lfloor\frac{\Theta}{1-\Theta}\rfloor$. On the other hand, $s\Theta < s = r + 1$ since $\Theta < 1$ and thus $\lfloor s\Theta\rfloor = r$. To prove (a) we have to show that $\min\{s\Theta - \lfloor s\Theta\rfloor, \lceil s\Theta\rceil - s\Theta\} = s\Theta - \lfloor s\Theta\rfloor$. This is equivalent to $s\Theta - \lfloor s\Theta\rfloor \le \lceil s\Theta\rceil - s\Theta$, which is true if $s\Theta$ is integer, and so we can assume that $s\Theta$ is not integer. Thus we need to show $s\Theta \le \lfloor s\Theta\rfloor + 1/2$, i.e., $\left(1 + \lfloor\frac{\Theta}{1-\Theta}\rfloor\right)\Theta \le \lfloor\frac{\Theta}{1-\Theta}\rfloor + 1/2$. Let $x = \frac{\Theta}{1-\Theta}$. Then $\Theta = \frac{x}{1+x}$ and the assertion reads $(1 + \lfloor x\rfloor)\frac{x}{1+x} \le \lfloor x\rfloor + 1/2$ or equivalently, $x \le 2\lfloor x\rfloor + 1$ which is trivially true for all $x \ge 0$. Consequently, part (a) follows.

To prove part (b), we first prove

(b1) $q\Theta - \lfloor q\Theta\rfloor > s\Theta - r$ for all integer q with $1 \le q \le r$.

We claim $q\Theta - \lfloor q\Theta\rfloor \ge (q+1)\Theta - \lfloor(q+1)\Theta\rfloor$ or equivalently, $\lfloor(q+1)\Theta\rfloor \ge \lfloor q\Theta\rfloor + \Theta$ for all integer $q \in [1, r]$. Since $q \le \lfloor\frac{\Theta}{1-\Theta}\rfloor$ we have $q \le \frac{\Theta}{1-\Theta}$ and thus $q \le (q+1)\Theta$. Since q is integer we get $q \le \lfloor(q+1)\Theta\rfloor$ and thus $\lfloor(q+1)\Theta\rfloor \ge q\Theta\Theta^{-1} \ge \lfloor q\Theta\rfloor\Theta^{-1} > \lfloor q\Theta\rfloor$ since $0 < \Theta < 1$ if $\lfloor q\Theta\rfloor > 0$. If $\lfloor q\Theta\rfloor = 0$ then $q = 1$ and the assertion is true as well because $\Theta > 1/2$. Consequently, the claim follows and thus, it suffices to prove (b1) for $q = r$, i.e., $r\Theta - \lfloor r\Theta\rfloor > (r+1)\Theta - r$ because $s = r + 1$. But $r > \Theta + \lfloor r\Theta\rfloor$ is trivially true since $r \ge 1$ is integer and $0 < \Theta < 1$. Hence (b1) follows. We are left with proving

(b2) $\lceil q\Theta\rceil - q\Theta > s\Theta - r$ for all integer q with $1 \le q \le r$.

Suppose first that $\lfloor\frac{\Theta}{1-\Theta}\rfloor = \frac{\Theta}{1-\Theta}$. Then $s\Theta - r = 0$ and assume(b2) is wrong. Let $q \in [1, r]$ be the smallest integer with $\lceil q\Theta\rceil - q\Theta = 0$. It follows that $\Theta = p/q$ where $1 \le p = \lceil q\Theta\rceil < q$. But $\Theta = \frac{1}{r+1}$ and thus $p(r+1) = rq$. Hence we get $r = \frac{p}{q-p}$ and $r + 1 = \frac{q}{q-p}$, i.e., both p and q are divisible by the integer $q - p \ge 1$. If $q - p = 1$, then $p = r$ and $q = r + 1$. Otherwise, $q - p \ge 2$ contradicts our assumption that q is the smallest integer with the required property. Consequently, such q does not exist in the range $1, \ldots, r$ and (b2) follows. Now suppose $\frac{\Theta}{1-\Theta}$ is not integer. Like in the proof of part (b1) we conclude $q \le \lfloor(q+1)\Theta\rfloor$. If $(q+1)\Theta$ is integer, then from $0 < \Theta < 1$, we get $q = (q+1)\Theta$, i.e., $q = \frac{\Theta}{1-\Theta}$, which contradicts the assumption that $\frac{\Theta}{1-\Theta}$ is not integer. Consequently, $(q+1)\Theta$ is not integer. Thus $\lceil q\Theta\rceil \le \lceil\Theta\lfloor(q+1)\Theta\rfloor\rceil \le \lfloor(q+1)\Theta\rfloor < \lceil(q+1)\Theta\rceil$ and hence from $0 < \Theta < 1$, $\lceil(q+1)\Theta\rceil - (q+1)\Theta \ge \lceil q\Theta\rceil - q\Theta$ for all integer $q \in [1, r]$. Consequently, it suffices to prove (b2) for $q = 1$, i.e., $1 - \Theta > \left(1 + \lfloor\frac{\Theta}{1-\Theta}\rfloor\right)\Theta - \lfloor\frac{\Theta}{1-\Theta}\rfloor$. Using $x = \frac{\Theta}{1-\Theta}$ the assertion is equivalent to $1 + \lfloor x\rfloor > x$ which is trivially true because x is a positive noninteger. Thus (b2) follows in this case as well, and hence the proof of part (iii) is complete.

(iv) We prove the assertion by induction. From the construction of the sequence of integers q_i we have $q_1 = 1 < q_2 < \cdots$. Thus since $q_2 > 1$ and integer, $q_2 \ge 2 > 2^{1/2}$ and the assertion is true for $n = 2$. Suppose that for $n = k \geq 2$ we have $q_k \geq 2^{(k-1)/2}$. By (9.70) $q_{k+1} = a_k q_k + q_{k-1} \geq q_k + q_{k-1} \geq 2^{(k-1)/2} + 2^{(k-2)/2} = 2^{(k-2)/2}(2^{1/2}+1) > 2^{(k-2)/2}2 = 2^{k/2}$ and thus the assertion follows for $n = k+1$ and the inductive proof is complete. (Note that here we use the inductive process *without* the particular initialization (9.73), which "shifts" the index n of the inductive process by 1 if $1 > \Theta > 1/2$.)

(v) Assume WROG that $0 < \Theta < \Theta' < 1$ and that the inductive process carries out at least $N \geq 1$ iterations. By (9.65), i.e., because the signs of $q_n\Theta - p_n$ and $q'_n\Theta' - p'_n$ alternate, we have either $\Theta' \leq p_N/q_N$ or $p_N/q_N \leq \Theta$. Since $\Theta' \neq \Theta$, the process continues for at least one more iteration for either Θ or Θ' or both. Suppose that $\Theta' \leq p_N/q_N$. Then $p_{N+1}/q_{N+1} \leq \Theta$ and $0 < \Theta' - \Theta \leq \frac{p_N}{q_N} - \frac{p_{N+1}}{q_{N+1}} = \frac{1}{q_N q_{N+1}} \leq 2^{-N+\frac{1}{2}}$, by part (iv). Suppose that $p_N/q_N \leq \Theta$. Then $p'_{N+1}/q'_{N+1} \geq \Theta'$ and $0 < \Theta' - \Theta \leq \frac{p'_{N+1}}{q'_{N+1}} - \frac{p'_N}{q'_N} = \frac{1}{q'_{N+1}q'_N} \leq 2^{-N+\frac{1}{2}}$ by part (iv) as well because $p'_N = p_N$ and $q'_N = q_N$ by assumption. Thus $|\Theta - \Theta'| \leq 2^{-N+\frac{1}{2}} \leq 2^{-N+1}$ as we have asserted.

---

*Exercise 9.10

Let $P \subseteq \mathbb{R}^n$ *be a rational polyhedron of facet complexity* ϕ, *let* $P_u = P \cap \{\boldsymbol{x} \in \mathbb{R}^n : -u \leq x_j \leq u \text{ for } 1 \leq j \leq n\}$ *for some integer* $u \geq 1$ *and let* $\boldsymbol{c} \in \mathbb{R}^n$ *be any rational vector.*

(i) Every extreme point $\boldsymbol{x} \in P_u$ *is a rational vector with components* $x_j = p_j/q_j$ *with integers* $0 \leq |p_j| < u2^{6n\phi+1}$ *and* $1 \leq q_j < 2^{6n\phi}$ *for* $1 \leq j \leq n$.

(ii) Any two extreme points $\boldsymbol{x}, \boldsymbol{y} \in P_u$ *with* $\boldsymbol{cx} > \boldsymbol{cy}$ *satisfy* $\boldsymbol{cx} > \boldsymbol{cy} + 2^{-12n^2\phi - \langle \boldsymbol{c} \rangle}$.

(iii) For $\Delta \geq 1 + u2^{6n\phi+12n^2\phi+\langle \boldsymbol{c} \rangle+1}$ *let* $\widetilde{d}_j = \Delta^n c_j + \Delta^{n-j}$ *for* $1 \leq j \leq n$ *and* $\widetilde{\boldsymbol{d}} = (\widetilde{d}_1, \ldots, \widetilde{d}_n)$. *Then the linear optimization problem* $\max\{\widetilde{\boldsymbol{d}}\boldsymbol{x} : \boldsymbol{x} \in P_u\}$ *has a* unique *maximizer* $\boldsymbol{x}^{max} \in P_u$ *and* $\boldsymbol{cx}^{max} = \max\{\boldsymbol{cx} : \boldsymbol{x} \in P_u\}$.

(iv) Define $\boldsymbol{d} = \widetilde{\boldsymbol{d}}/\|\widetilde{\boldsymbol{d}}\|_\infty$ *where* $\widetilde{\boldsymbol{d}}$ *is defined in part (iii). Then* $\langle \boldsymbol{d} \rangle \leq 3.5n(n-1)\lceil \log_2 \Delta \rceil + 2(n-1)\langle \boldsymbol{c} \rangle$ *and thus for* $u = 2^{\Lambda+1}$ *and the smallest* Δ *satisfying the condition of part (iii) we have* $\langle \boldsymbol{d} \rangle \leq 3.5n(n-1)\phi(16n^2 + 11n\phi + 1) + (3.5n+2)(n-1)\langle \boldsymbol{c} \rangle + 14n(n-1)$, *where* $\Lambda = \phi + 5n\phi + 4n^2\phi + 1$.

(v) Let $(\boldsymbol{h}, h_0)$ *belong to a linear description of* P, $\|\boldsymbol{h}\|_\infty > 0$ *and* $\langle \boldsymbol{h} \rangle + \langle h_0 \rangle \leq \phi$. *Show that* $\langle \widetilde{\boldsymbol{h}} \rangle + \langle \widetilde{h}_0 \rangle \leq n\phi + 2$ *where* $\widetilde{\boldsymbol{h}} = \boldsymbol{h}/\|\boldsymbol{h}\|_\infty$ *and* $\widetilde{h}_0 = h_0/\|\boldsymbol{h}\|_\infty$.

(i) If $\boldsymbol{x} \in P_u$ is an extreme point of P, then the assertion follows from point 7.5(b). So suppose that $\boldsymbol{x} \in P_u$ is an extreme point of P_u that is not an extreme point of P. Then by point 7.2(b) $\boldsymbol{x}$ is determined uniquely by a system of equations

$$\begin{pmatrix} \boldsymbol{I}_k & \mathbf{O} \\ \boldsymbol{F}_1 & \boldsymbol{F}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}^1 \\ \boldsymbol{x}^2 \end{pmatrix} = \begin{pmatrix} u\boldsymbol{g}^* \\ \boldsymbol{f}^* \end{pmatrix}$$

where $\boldsymbol{I}_k$ is the $k \times k$ identity matrix with $1 \leq k \leq n$, $\boldsymbol{F}_1$, $\boldsymbol{F}_2$ are $(n-k) \times k$ and $(n-k) \times (n-k)$ matrices, $\boldsymbol{g}^*$ is a vector with entries $+1$ if $x_j = u$, -1 if $x_j = -u$ and $\boldsymbol{f}^*$ has $n-k$ components.

Moreover, every row $(\boldsymbol{f}^i, f_i)$ of $\begin{pmatrix}\boldsymbol{F}_1 & \boldsymbol{F}_2 & \boldsymbol{f}^*\end{pmatrix}$ satisfies $\langle \boldsymbol{f}^i \rangle + \langle f_i \rangle \leq \phi$ and if $k < n$ then $\det F_2 \neq 0$. If $k = n$ then F_2 is empty and we define $\det \boldsymbol{F}_2 = 1$. Denote by $\boldsymbol{G}$ the $n \times n$ matrix of this equation system and suppose the components of $\boldsymbol{x}$ are indexed to agree with the above partitioning into $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$. By $\boldsymbol{g}_j$ we denote the j-th column of $\begin{pmatrix}\boldsymbol{I}_k & \mathbf{O}\end{pmatrix}$, by $\boldsymbol{f}_j$ the j-th column of $\begin{pmatrix}\boldsymbol{F}_1 & \boldsymbol{F}_2\end{pmatrix}$ and by $\boldsymbol{u}_j$ the j-th unit vector in $\mathbb{R}^n$. If we let

$$\boldsymbol{G}_j = \boldsymbol{G} + \boldsymbol{u}_j^T \left(\begin{pmatrix} u\boldsymbol{g}^* \\ \boldsymbol{f}^* \end{pmatrix} - \begin{pmatrix} \boldsymbol{g}_j \\ \boldsymbol{f}_j \end{pmatrix} \right) ,$$

then by Cramer's rule $x_j = \det \boldsymbol{G}_j / \det \boldsymbol{G}$ and we need to estimate the digital sizes of the determinants. From formula (7.18) we get $\langle \det \boldsymbol{G} \rangle \leq 2\langle \boldsymbol{G} \rangle - n^2 \leq 2n\phi - n^2$. Moreover, $\det \boldsymbol{G}$ is a rational number of digital size less than $2n\phi$ and thus there exist integers p, q with $0 \leq |p| < 2^{2n\phi}$, $1 \leq q_j < 2^{2n\phi}$ such that $\det \boldsymbol{G} = p/q$. Suppose that $1 \leq j \leq k$. We calculate $\det \boldsymbol{G}_j = \pm u \det \boldsymbol{F}_2$. Moreover, by the same reasoning as before $\langle \det \boldsymbol{F}_2 \rangle < 2n\phi$ is correct and thus there exist integers p_j, q_j with $0 \leq |p_j| < 2^{n\phi}$, $1 \leq q_j < 2^{2n\phi}$ such that $\det \boldsymbol{F}_2 = p_j/q_j$. It follows that $x_j = \pm u p_j q / q_j p$ satisfies $u p_j q$ integer, $0 \leq |u p_j q| < u 2^{4n\phi} < u 2^{6n\phi+1}$ and $1 \leq |q_j p| < 2^{4n\phi} < 2^{6n\phi}$ since u is integer and thus the assertion follows in this case. Suppose now that $k+1 \leq j \leq n$. From the formula for the determinant of a partitioned matrix of Chapter 2.2 we calculate

$$\begin{aligned} \det \boldsymbol{G}_j &= \det(\boldsymbol{F}_2 + \boldsymbol{v}_j^T(\boldsymbol{f}^* - \boldsymbol{f}_j) - u\boldsymbol{F}_1\boldsymbol{v}_j^T\boldsymbol{g}^*) \\ &= (\det \boldsymbol{F}_2)(\det(\boldsymbol{I}_k + \boldsymbol{F}_2^{-1}\boldsymbol{v}_j^T(\boldsymbol{f}^* - \boldsymbol{f}_j) - u\boldsymbol{F}_2^{-1}\boldsymbol{F}_1\boldsymbol{v}_j^T\boldsymbol{g}^*)) \\ &= (\det \boldsymbol{F}_2)(f_* - u g_*) , \end{aligned}$$

where $\boldsymbol{v}_j \in \mathbb{R}^{n-k}$ is the j-th unit vector, $f_* = \boldsymbol{v}_j^T \boldsymbol{F}_2^{-1} \boldsymbol{f}^*$ and $g_* = \boldsymbol{v}_j^T \boldsymbol{F}_2^{-1} \boldsymbol{F}_1 \boldsymbol{g}^*$. We calculate

$$f_* \det \boldsymbol{F}_2 = \det(\boldsymbol{F}_2 + \boldsymbol{v}_j^T(\boldsymbol{f}^* - \boldsymbol{f}_j))$$

by factoring out $\boldsymbol{F}_2$ and thus by (7.18) $\langle f_* \det \boldsymbol{F}_2 \rangle < 2n\phi$ is correct, i.e., there exist integer p_j, q_j with $0 \leq |p_j| < 2^{2n\phi}$, $1 \leq q_j < 2^{2n\phi}$ such that $f_* \det \boldsymbol{F}_2 = p_j/q_j$. We calculate also

$$-g_* \det \boldsymbol{F}_2 = \det \left(\begin{pmatrix} \boldsymbol{I}_k & \mathbf{O} \\ \boldsymbol{F}_1 & \boldsymbol{F}_2 \end{pmatrix} + \boldsymbol{u}_j^T \begin{pmatrix} \boldsymbol{g}^* \\ -\boldsymbol{f}_j \end{pmatrix} \right)$$

by applying the determinant formula for partitioned matrices first and then factoring out $\boldsymbol{F}_2$. Since $\phi \geq n+1$ it follows that the matrix on the right is a rational number of digital size less than $2n\phi$ and thus there exist integers r, s with $0 \leq |r| < 2^{2n\phi}$ and $1 \leq s < 2^{2n\phi}$ such that $-g_* \det \boldsymbol{F}_2 = r/s$. Consequently, since $u \geq 1$ is integer $x_j = q(sp_j + u + q_j)/psq_j$ satisfies the assertion and the proof of (i) is complete.

(ii) By part (i) of this exercise we have $x_j = p_j/q_j$ and $y_j = r_j/s_j$ with integer numbers p_j, q_j, r_j, s_j satisfying $0 \leq |p_j|, |r_j| < u2^{6n\phi+1}$ and $1 \leq q_j, s_j < 2^{6n\phi}$ for $1 \leq j \leq n$. Let $c_j = a_j/b_j$ with integer a_j and $b_j \geq 1$ for $1 \leq j \leq n$ since $\boldsymbol{c}$ is rational. Since $\boldsymbol{cx} > \boldsymbol{cy}$ it follows that $\boldsymbol{c}(\boldsymbol{x}-\boldsymbol{y}) \geq (\prod_{j=1}^n q_j s_j b_j)^{-1} > 2^{-12n^2\phi - \langle \boldsymbol{c} \rangle}$ because $\prod_{j=1}^n b_j < 2^{\langle \boldsymbol{c} \rangle}$.

(iii) The proof of this part of the exercise goes like the proof of Exercise 7.14(iii).

(iv) The proof follows from Exercise 7.14(iv) and by a simple substitution of the values for Δ, Λ, u and the rough estimation $1 + 2^{3+\alpha} < 2^{4+\alpha}$ for $\alpha \geq 0$.

(v) Let $h_j = \frac{p_j}{q_j}$ with integer p_j, q_j satisfying $0 \leq |p_j| < 2^\phi$, $1 \leq q_j < 2^\phi$ and $|\frac{p_\ell}{q_\ell}| \geq |\frac{p_j}{q_j}|$ for $1 \leq j \leq n$. Then $\widetilde{h}_j = \frac{p_j q_\ell}{q_j p_\ell}$, $\widetilde{h}_\ell = 1$, $\langle \widetilde{h}_j \rangle \leq \langle p_j \rangle + \langle q_j \rangle + \langle p_\ell \rangle + \langle q_\ell \rangle$ and $\langle \widetilde{h}_\ell \rangle = 2$ where $0 \leq j \neq \ell \leq n$. Consequently, $\langle \widetilde{\boldsymbol{h}} \rangle + \langle \widetilde{h}_0 \rangle \leq \langle \boldsymbol{h} \rangle + \langle h_0 \rangle + (n-1)(\langle p_\ell \rangle + \langle q_\ell \rangle) + 2 \leq n\phi + 2$.

---

*Exercise 9.11

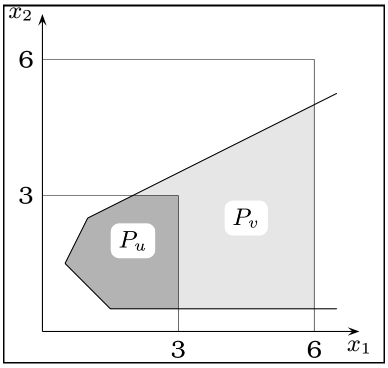
(i) Let $P = \{\boldsymbol{x} \in \mathbb{R}^2 : -4x_1 + 2x_2 \leq 1\,,\ x_1 + x_2 \geq 2\,,\ -x_1 + 2x_2 \leq 4\,,\ 2x_2 \geq 1\}$ and P_u be as defined in (9.76). Find the maximizer $\boldsymbol{x}^{max}$ of $\max\{x_1 + x_2 : \boldsymbol{x} \in P_u\}$ for $u = 3$ and the corresponding $\boldsymbol{y}^{max}$ for $v = 2u$. Does $\boldsymbol{y}^{max} - \boldsymbol{x}^{max}$ belong to the asymptotic cone of P? If not, what is the smallest possible value of u that works? What is the theoretical value that you get for u using $\Lambda = \phi + 5n\phi + 4n^2\phi + 1$?

(ii) Suppose that the direction vector $\boldsymbol{t} \in P_\infty$ of the proof of Remark 9.19 satisfies $\boldsymbol{ct} = 0$ and let $T_= = \{\boldsymbol{y} \in T : \boldsymbol{cy} = 0,\ \|\boldsymbol{y}\|_\infty = 1\}$ where (S, T) is a minimal generator of the polyhedron $P \subseteq \mathbb{R}^n$ such that $\langle x_j \rangle \leq 4n\phi$ for all j and $\boldsymbol{x} \in S \cup T$. Prove that $\boldsymbol{t} \succeq \boldsymbol{y}$ for all $\boldsymbol{y} \in T_=$, i.e. that $\boldsymbol{t}$ is lexicographically greater than or equal to every $\boldsymbol{y} \in T_=$.

(iii) Determine the facet and vertex complexity of the polytopes S_n and C_n of Exercise 7.2 and of H_n and O_n of Exercise 7.7.

(iv) Find polynomial-time algorithms that solve the polyhedral separation problem over S_n, C_n, H_n and O_n.

(i) The polytopes P_3 and P_6 are shown in the figure. Maximizing the function $x_1 + x_2$ over P_3



we get $\boldsymbol{x}^{max} = (3, 3)$ while maximizing it over P_6 we get $\boldsymbol{y}^{max} = (6, 5)$. The difference vector $\boldsymbol{y}^{max} - \boldsymbol{x}^{max} = (3, 2)$ and it is not in the asymptotic cone C_∞ of P, where $C_\infty = \{\boldsymbol{y} \in \mathbb{R}^2 : -4x_1 + 2x_2 \leq 0\,,\ x_1 + x_2 \geq 0\,,\ -x_1 + 2x_2 \leq 0\,,\ 2x_2 \geq 0\}$ since it violates the third inequality. Selecting $u \geq 4$ we have the maximizer lying on the extreme ray $-x_1 + 2x_2 = 4$ of the polyhedron P and thus $u = 4$ and $v = 2u = 8$ will give $\boldsymbol{x}^{max} = (4, 4)$ and $\boldsymbol{y}^{max} = (8, 6)$, and thus $\boldsymbol{y}^{max} - \boldsymbol{x}^{max} = (4, 2) \in C_\infty$. To compute the theoretical value of u, we first calculate ϕ for the polyhedron P. Since $\langle 0 \rangle = 1$, $\langle 1 \rangle = \langle -1 \rangle = 2$, $\langle 2 \rangle = 3$ and $\langle 4 \rangle = \langle -4 \rangle = 4$, we have that $\phi \geq \max\{\langle -4 \rangle + \langle 2 \rangle + \langle 1 \rangle, \langle 1 \rangle + \langle 1 \rangle + \langle 2 \rangle, \langle -1 \rangle + \langle 2 \rangle + \langle 4 \rangle, \langle 0 \rangle + \langle 2 \rangle + \langle 1 \rangle\} = 9$. So selecting $\phi = 9$ we get from $\Lambda = \phi + 5n\phi + 4n^2\phi + 1$ with $n = 2$ that $\Lambda = 244$ and thus $u = 2^{244}$ which is a horribly big number and evidently much bigger than required in this case.

(ii) By assumption the unique maximizer $\boldsymbol{x}^{max}$ of $\boldsymbol{dx}$ over the polytope P_u for $u = 2^\Lambda$ satisfies $x_j^{max} = \pm u$ for at least one $j \in \{1, \ldots, n\}$ and thus $\boldsymbol{x}^{max}$ is the unique solution to a system of equations of the form

$$\begin{pmatrix} \pm \boldsymbol{I}_k & \mathbf{O} \\ \boldsymbol{F}_1 & \boldsymbol{F}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_1^{max} \\ \boldsymbol{x}_1^{max} \end{pmatrix} = \begin{pmatrix} u\boldsymbol{e}_k \\ \boldsymbol{f}^* \end{pmatrix} , \tag{1}$$

where $\pm \boldsymbol{I}_k$ is some $k \times k$ matrix with $1 \leq k \leq n$ having $+1$ or -1 on its main diagonal (according to $x_j = u$ or $x_j = -u$), zeros elsewhere, $\boldsymbol{F}_1$, $\boldsymbol{F}_2$ are $(n-k) \times k$ and $(n-k) \times (n-k)$ matrices, $\boldsymbol{e}_k$ is a vector of k ones and $\boldsymbol{f}^*$ has $n-k$ components. Every row $(\boldsymbol{f}^i, f_i)$ of $\begin{pmatrix} \boldsymbol{F} & \boldsymbol{F}_2 & \boldsymbol{f}^* \end{pmatrix}$ satisfies $\langle \boldsymbol{f}^i \rangle + \langle f_i \rangle \leq \phi$ and $\det \boldsymbol{F}_2 \neq 0$, where by convention $\det \boldsymbol{F}_2 = 1$ if $\boldsymbol{F}_2$ is empty. Moreover, $\boldsymbol{y}^{max}$ satisfies (1) with u replaced by $v = 2^{\Lambda+1}$ and $\boldsymbol{t} = 2^{-\Lambda}(\boldsymbol{y}^{max} - \boldsymbol{x}^{max}) \in C_\infty$ satisfies $\langle t_j \rangle \leq 4n\phi$ for $1 \leq j \leq n$. (For more detail than given in the proof of Remark 9.17 on the estimation of $\langle t_j \rangle$ see the proof of Exercise 9.10(i).) From the uniqueness of the respective maximizers $\boldsymbol{x}^{max}$, $\boldsymbol{y}^{max}$ and the assumptions that $x_j = \pm u$ for at least one index j and $\Lambda > 4n\phi$, it follows that every basis defining $\boldsymbol{x}^{max}$ is of the form (1) and thus by the duality theory of linear programming

$$\boldsymbol{d} = \boldsymbol{\lambda} \begin{pmatrix} \pm \boldsymbol{I}_k & \mathbf{O} \\ \boldsymbol{F}_1 & \boldsymbol{F}_2 \end{pmatrix} \text{ with } \boldsymbol{\lambda} \geq 0 \tag{2}$$

for some such basis defining $\boldsymbol{x}^{max}$ and $\boldsymbol{y}^{max}$, respectively. Consequently, a basis satisfying (1) and (2) exists. Since by construction $\boldsymbol{t} \in C_\infty$ and $\boldsymbol{t} \neq \mathbf{0}$, it follows that $\widetilde{\boldsymbol{t}} = \boldsymbol{t}/\|\boldsymbol{t}\|_\infty \in P_\infty$ is an extreme point of P_∞. More precisely, by dropping some of the constraints of P_∞ we have

$$P_\infty \subseteq OC(\widetilde{\boldsymbol{t}}, \widetilde{\boldsymbol{H}}) = \left\{ \boldsymbol{y} \in \mathbb{R}^n : \begin{pmatrix} \pm \boldsymbol{I}_k & \mathbf{O} \\ \boldsymbol{F}_1 & \boldsymbol{F}_2 \end{pmatrix} \boldsymbol{y} \leq \begin{pmatrix} \boldsymbol{e}_k \\ \mathbf{0} \end{pmatrix} \right\} ,$$

i.e., $OC(\widetilde{\boldsymbol{t}}, \widetilde{\boldsymbol{H}})$ is the displaced outer cone with apex at $\widetilde{\boldsymbol{t}}$ containing all of P_∞ (see the end of Chapter 7.5.4), and $\widetilde{\boldsymbol{t}}$ is an extreme point of $OC(\widetilde{\boldsymbol{t}}, \widetilde{\boldsymbol{H}})$. From (2) it follows that $\widetilde{\boldsymbol{t}}$ maximizes $\boldsymbol{dy}$ over $OC(\widetilde{\boldsymbol{t}}, \widetilde{\boldsymbol{H}})$ and thus by the outer inclusion principle, $\widetilde{\boldsymbol{t}}$ maximizes $\boldsymbol{dy}$ over P_∞. The matrix $\widetilde{\boldsymbol{H}}$ defining P_∞ is given by $\boldsymbol{H}$, $\boldsymbol{I}_n$ and $-\boldsymbol{I}_n$ and thus P_∞ has a facet complexity of ϕ – just like the polyhedron P. Hence $\langle \widetilde{t_j} \rangle \leq 4n\phi$ for $1 \leq j \leq n$. Since $\boldsymbol{d}\widetilde{\boldsymbol{t}} \geq \boldsymbol{dy}$ for all $\boldsymbol{y} \in P_\infty$, it follows like in Exercise 7.14(iii) that $\widetilde{\boldsymbol{t}}$ is the unique maximizer of $\boldsymbol{d}$ over P_∞ because the number Δ that we use to prove Remark 9.19 is greater than the number $1 + 2^{4n\phi + 8n^2\phi + \langle \boldsymbol{c} \rangle + 1}$ that suffices to guarantee uniqueness. But then by Exercise 7.14(v) $\widetilde{\boldsymbol{t}} \succeq \boldsymbol{y}$ for all $\boldsymbol{y} \in T_=$ and the proof of part (ii) is complete.

(iii) The digital size of the inequality $-x_j \leq 0$ equals $n+2$ since we also store zero coefficients. Likewise the digital size of $x_j \leq 1$ is $n+3$ and of $\sum_{j=1}^n x_j \leq 1$ is $2(n+1)$. Consequently, the facet complexity of the polytope S_n is $\phi = 2(n+1)$ and that of C_n is $\phi = n+3$, where $n \geq 1$ is arbitrary. The polyhedron $\boldsymbol{H}_n$ of Exercise 7.7(i) has the same facet complexity as S_n, i.e., $\phi = 2(n+1)$, while the polytope O_n of Exercise 7.7(ii) has $\phi = 2n + 1 + \lceil \log_2 n \rceil$.

(iv) Since S_n has $n+1$ constraints and C_n has $2n$ constraints, the (trivial) algorithm LIST-and-CHECK is a polynomial-time separation algorithm for S_n and C_n, respectively, no matter what rational $\boldsymbol{y} \in \mathbb{R}^n$ is given as input. (LIST-and-CHECK is just that; you list all inequalities and check them one by one for violation.) Since both H_n and O_n have exponentially many inequalities, LIST-and-CHECK does not work in either case since it may, in the worst case, require exponential time in n to execute it. In the case of H_n every constraint is of the form $\boldsymbol{hx} = \sum_{j=1}^{n} \delta_j x_j \leq 1$ where $\delta_j \in \{0,1\}$ for $1 \leq j \leq n$ and $\|\boldsymbol{h}\|_\infty = 1$, except for the trivial constraint $\boldsymbol{0x} \leq 1$ which is never violated. The polyhedral separation problem for H_n is

$$\max\{\sum_{j=1}^{n} \delta_j z_j - 1 : \delta_j \in \{0,1\} \text{ for } 1 \leq j \leq n\}$$

where $\boldsymbol{z} \in \mathbb{R}^n$ is a rational vector. To solve the problem we scan the vector $\boldsymbol{z}$ and set $\delta_j = 1$ if $z_j > 0$, $\delta_j = 0$ otherwise. This separation algorithm is linear in n and $\langle \boldsymbol{z} \rangle$ and thus a polynomial-time algorithm for the polyhedral separation problem for H_n.

In the case of the polytope O_n we can check the $2n$ constraints $0 \leq x_j \leq 1$ for $1 \leq j \leq n$ by the algorithm LIST-and-CHECK in polynomial time. So we can assume WROG that the rational vector $\boldsymbol{z} \in \mathbb{R}^n$ for which we want to solve the polyhedral separation problem satisfies $0 \leq z_j \leq 1$. The separation problem for the remaining exponentially many constraints of O_n is

$$\max\{\sum_{j \in N_1} z_j - \sum_{j \in N-N-1} z_j - |N_1| + 1 : N_1 \subseteq N,\ |N_1| \text{ even}\}$$
$$= 1 - \min\{\sum_{j \in N_1} (1 - z_j) + \sum_{j \in N-N_1} z_j : N_1 \subseteq N,\ |N_1| \text{ even}\}\ ,$$

and a violated constraint is obtained if the objective function of the minimization problem is less than one. To solve the problem we order the components of $\boldsymbol{z}$ in decreasing order which requires time that is polynomial in n and $\langle \boldsymbol{z} \rangle$. E.g. the sorting algorithm HEAPSORT requires $\mathcal{O}(n \log n)$ operations in the worst case. So we can assume WROG that $1 \geq z_1 \geq z_2 \geq \cdots \geq z_k \geq 1/2 > z_{k+1} \geq \cdots \geq z_n \geq 0$, where $0 \leq k \leq n$. Finding the index k or verifying that $k = 0$ can be done by scanning the ordered vector $\boldsymbol{z}$ once, i.e., in time that is linear in n and $\langle \boldsymbol{z} \rangle$. If the index k is even we set $N_1^* = \{1, \ldots, k\}$. If k is odd we set $N_1^* = \{1, \ldots, k-1\}$ if $z_k + z_{k+1} < 1$, $N_1^* = \{1, \ldots, k+1\}$ otherwise. By construction $|N_1^*|$ is even, $z_{i_1} + z_{i_2} < 1$ for all $i_1 \neq i_2 \notin N_1^*$ and $z_{i_1} + z_{i_2} \geq 1$ for all i_1, $i_2 \in N_1^*$ in all cases. We claim that N_1^* solves the minimization problem. Suppose not and let $S \subseteq N$ be an optimal solution. Then $|S|$ is even, $S \neq N_1^*$ and

$$z(S) = \sum_{j \in S}(1 - z_j) + \sum_{j \in N-S} z_j < \sum_{j \in N_1^*} (1 - z_j) + \sum_{j \in N-N_1^*} z_j = z(N_1^*)\ .$$

If $|S| = |N_1^*|$ then by construction $\sum_{j \in S} z_j \leq \sum_{j \in N_1^*} z_j$ and $\sum_{j \in N-N_1^*} z_j \leq \sum_{j \in N-S} z_j$. But then $z(N_1^*) \leq z(S)$ and thus if $z(S) < z(N_1^*)$ then $|S| - |N_1^*|$ is an even number different from zero. Suppose first that $|S| \geq 2 + |N_1^*|$. Then there exists $i_1 \neq i_2 \in S$ such that i_1, $i_2 \notin N_1^*$. Let $S' = S - \{i_1, i_2\}$. We compute $z(S') = z(S) - 2(1 - z_{i_1} - z_{i_2}) \geq z(S)$ by the optimality of S and thus $z_{i_1} + z_{i_2} \geq 1$ which is a contradiction because $i_1 \neq i_2 \notin N_1^*$. Suppose now that $|S| + 2 \leq |N_1^*|$. Then there exists $i_1 \neq i_2 \in N_1^*$ such that $i_1 \notin S$, $i_2 \notin S$. Let $S' = S \cup \{i_1, i_2\}$. We compute $z(S') = z(S) + 2(1 - z_{i_1} - z_{i_2}) \geq z(S)$ by the optimality of S and thus $z_{i_1} + z_{i_2} \leq 1$. Since i_1, $i_2 \in N_1^*$ we get $z_{i_1} + z_{i_2} = 1$ and thus $z(S') = z(S)$. Consequently, S' is optimal as well as $|S| < |S'| \leq |N_1^*|$, we can reappply the reasoning and after finitely many steps we arrive at a contradiction because the cardinality of an optimal S is bounded by $|N_1^*|$ in this case. Consequently, the claim follows. The polyhedral separation problem for O_n can thus be solved in $\mathcal{O}(\langle \boldsymbol{z} \rangle n \log n)$ time for any rational $\boldsymbol{z} \in \mathbb{R}^n$.

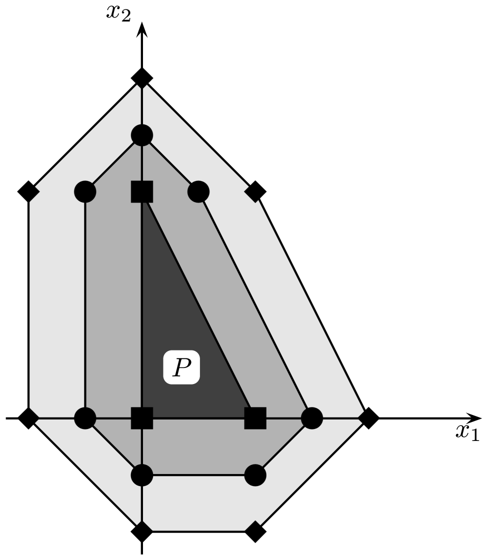


Fig. 9.13. ε-solidifications of P for $\varepsilon = 0$, $1/2$ and 1

---

*Exercise 9.12

(i) *Consider the polytope* $P = \{\boldsymbol{x} \in \mathbb{R}^2 : 2x_1 + x_2 \le 2\,,\ x_1 \ge 0\,,\ x_2 \ge 0\}$*. Find minimal generators for the corresponding* SP*,* SP_∞ *and* SP_∞^* *as defined in (9.80), (9.81) and (9.82). Show that every nonzero extreme point* $(\boldsymbol{h}, h_0)$ *of* SP_∞ *defines a facet* $\boldsymbol{hx} \le h_0 + \varepsilon$ *of the* ε*-solidification* P_ε^1 *of* P *in the* ℓ_1*-norm and vice versa, that every facet* $\boldsymbol{hx} \le h_0 + \varepsilon$ *with* $\|\boldsymbol{h}\|_\infty = 1$ *of* P_ε^1 *defines an extreme point* $(\boldsymbol{h}, h_0)$ *of* SP_∞ *where* $\varepsilon > 0$*.*

(ii) *Do the same as in part (i) of this exercise for the polyhedron* $P = \{\boldsymbol{x} \in \mathbb{R}^2 : 2x_1 - x_2 = 0\,,\ x_1 \ge 1\}$*.*

(iii) *Do the same as in part (i) of this exercise for the polyhedron* $P = \{\boldsymbol{x} \in \mathbb{R}^2 : 2x_1 + x_2 \ge 5\,,\ x_1 - x_2 \ge -2\,,\ x_2 \ge 1\}$*. In addition, let* $(\boldsymbol{f}, f_0) = (-1, -1, -6)$ *and solve the linear program* $\max\{-x_1 - x_2 + 6x_3 : (x_1, x_2, x_3) \in SP_\infty^*\}$*. Does its optimal solution yield a most violated separator for* $(\boldsymbol{f}, f_0)$ *and* SP*? If not, what is the most violated separator in this case?*

(iv) *Let* $P \subseteq \mathbb{R}^n$ *be any nonempty, line free polyhedron and* P_ε^1 *its* ε*-solidification with respect to the* ℓ_1*-norm where* $\varepsilon > 0$*. Show that the extreme points of* SP_∞ *as defined in (9.81) are in one-to-one correspondence with the facets of* P_ε^1*. What happens if* P *is permitted to have lines?*

(i) The polytope P, see Figure 9.13, has three extreme points $\boldsymbol{x}^1 = (1, 0)$, $\boldsymbol{x}^2 = (0, 2)$ and $\boldsymbol{x}^3 = (0, 0)$. Consequently, the h_0-polar SP of P is given by $SP = \{(h_1, h_2, h_0) \in \mathbb{R}^3 : h_1 - h_0 \le 0,\ 2h_2 - h_0 \le$

$0, \ -h_0 \le 0\}$. Running the double description algorithm (or by hand calculation) we find that SP is pointed and has three extreme rays $(-1,0,0)$, $(0,-1,0)$ and $(2,1,2)$. SP is the set of all separators for P and the set of normed separators SP_∞ for P is obtained from SP by intersecting SP with the constraints $-1 \le h_j \le 1$ for $j = 1,2$ as we are working with the ℓ_1-norm. Using the homogenization (7.5) and running the double description algorithm (or by hand calculation) we find that SP_∞ is pointed and has a minimal generator consisting of the eight extreme points

$$(0,0,0),\ (-1,0,0),\ (0,-1,0),\ (-1,-1,0),\ (1,1,2),\ (1,-1,1),\ (1,\tfrac{1}{2},1),\ (-1,1,2)$$

and the extreme ray given by $(0,0,1)$. The set of normed separators SP_∞^* for SP given by (9.82) is the polytope

$$SP_\infty^* = \{\boldsymbol{x} \in \mathbb{R}^3 : 2x_1 + x_2 - 2x_3 \le 0, -x_1 \le 0, -x_2 \le 0, -1 \le x_1 \le 1, -1 \le x_2 \le 1, 0 \le x_3 \le 1\} .$$

Using the double description algorithm (or by hand calculation) we find that a minimal generator has the following six extreme points

$$(0,0,0),\ (0,0,1),\ (0,1,\tfrac{1}{2}),\ (0,1,1),\ (1,0,1),\ (\tfrac{1}{2},1,1) .$$

To answer the second part of this problem we first calculate the ε-solidification P_ε^1 in the ℓ_1-norm. To do so we proceed like in the proof of Exercise 9.7(vi). To calculate P_ε^1 we thus have to project out the μ-variables from the polyhedron

$$\begin{aligned} PP_\varepsilon^1 =\{(\boldsymbol{z},\boldsymbol{\mu}) \in \mathbb{R}^4 : & -z_1 - z_2 + \mu_1 + 2\mu_2 \le \varepsilon, -z_1 + z_2 + \mu_1 - 2\mu_2 \le \varepsilon, z_1 - z_2 - \mu_1 + 2\mu_2 \le \varepsilon, \\ & z_1 + z_2 - \mu_1 - 2\mu_2 \le \varepsilon, \mu_1 + \mu_2 \le 1, \mu_1 \ge 0, \mu_2 \ge 0\} \end{aligned}$$

where we have used that $\boldsymbol{x} = \boldsymbol{0}$ is an extreme point of P. To do so we need a minimal generator of the cone (for the general definition see(7.8))

$$C = \{\boldsymbol{u} \in \mathbb{R}^7 : u_1 - u_2 + u_3 - u_4 + u_5 - u_6 = 0, 2u_1 + 2u_2 - 2u_3 - 2u_4 + u_5 - u_7 = 0, \boldsymbol{u} \ge \boldsymbol{0}\} .$$

Running the double description algorithm we find that C is pointed and has the following ten extreme rays

$$\begin{aligned} &(1,0,0,1,0,0,0),\ (0,1,1,0,0,0,0),\ (1,0,1,0,0,2,0),\ (0,0,1,0,2,3,0),\ (0,0,0,0,1,1,1), \\ &(0,0,0,1,2,1,0),\ (0,1,0,3,4,0,0),\ (0,1,0,0,1,0,3),\ (1,1,0,0,0,0,4),\ (1,0,0,0,0,1,2) . \end{aligned}$$

Consequently, we find (besides some trivial redundant inequalities) that P_ε^1 is given by

$$\begin{aligned} P_\varepsilon^1 =\{\boldsymbol{x} \in \mathbb{R}^2 : & \ 2x_1 + x_2 \le 2 + 2\varepsilon, x_1 + x_2 \le 2 + \varepsilon, -x_1 + x_2 \le 2 + \varepsilon, \\ & x_1 - x_2 \le 1 + \varepsilon, -x_1 - x_2 \le \varepsilon, -x_1 \le \varepsilon, -x_2 \le \varepsilon\} . \end{aligned}$$

From Figure 9.13, we see that every inequality of the linear description of P_ε^1 corresponds to a nonzero extreme point of SP_∞ and vice versa. Note, however, that you have to normalize the first inequality of P_ε^1 to get the correspondence. The extreme points of P_ε^1 for $\varepsilon \ge 0$ are

$$(-\varepsilon,0),\ (0,-\varepsilon),\ (1,-\varepsilon),\ (1+\varepsilon,0),\ (\varepsilon,2),\ (0,2+\varepsilon),\ (-\varepsilon,2) .$$

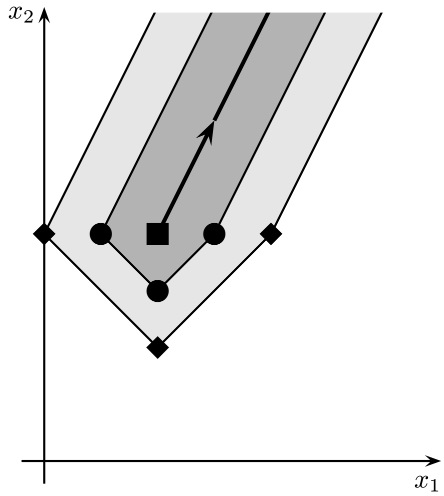


Fig. 9.14. ε-solidifications of P for $\varepsilon = 0$, $1/2$ and 1

(ii) The polyhedron P, see Figure 9.14, is a flat consisting of the extreme point $\boldsymbol{x} = (1,2)$ and the direction vector $\boldsymbol{y} = (1,2)$. Consequently, the h_0-polar SP of P is the cone $SP = \{(h_1,h_2,h_0) \in \mathbb{R}^3 : h_1 + 2h_2 - h_0 \le 0, h_1 + 2h_2 \le 0\}$. Running the double description algorithm (or by hand calculation) we find that SP is a blunt cone, the basis of the lineality space of SP is given by $(-2,1,0)$ and the conical part of SP is generated by $(0,0,1)$, $(-1,0,-1)$. So a minimal generator of SP is given by $\{(-2,1,0),(2,-1,0),(0,0,1),(-1,0,-1)\}$. The set SP_∞ of normed separators for P is obtained by intersecting SP with the constraints $-1 \le h_j \le 1$ for $j = 1,2$ as we are working with the ℓ_1-norm. Using the homogenization (7.5) and running the double description algorithm, we find that SP_∞ is a pointed polyhedron. Its minimal generator consists of the four extreme points $(-1,\frac{1}{2},0)$, $(1,-\frac{1}{2},0)$, $(-1,-1,-3)$, and $(1,-1,-1)$, and the extreme ray given by $(0,0,1)$. The set of normed separators SP^*_∞ for SP is the polytope

$$SP^*_\infty = \{\boldsymbol{x} \in \mathbb{R}^3 : 2x_1 - x_2 = 0, -x_1 + x_3 \le 0, 0 \le x_1 \le 1, 0 \le x_2 \le 1, 0 \le x_3 \le 1\} .$$

Using the double description algorithm we find that a minimal generator of SP^*_∞ has the following three extreme points $(0,0,0)$, $(\frac{1}{2},1,0)$, and $(\frac{1}{2},1,\frac{1}{2})$. To answer the second part of this problem we calculate the ε-solidification P^1_ε of P in the ℓ_1-norm. To do so we proceed like in part (i). To calculate P^1_ε we thus have to project out variable ν_1 from the polyhedron

$$\begin{aligned} PP^1_\varepsilon = &\{(\boldsymbol{z},\nu_1) \in \mathbb{R}^3 : -z_1 - z_2 + 3\nu_1 \le -3 + \varepsilon, z_1 - z_2 + \nu_1 \le -1 + \varepsilon, -z_1 + z_2 - \nu_1 \le 1 + \varepsilon, \\ &z_1 + z_2 - 3\nu_1 \le 3 + \varepsilon, \nu_1 \ge 0\} \end{aligned}$$

where we have simply eliminated the μ variable since it must equal one. We thus need a minimal generator of the cone

$$C = \{\boldsymbol{u} \in \mathbb{R}^5 : 3u_1 + u_2 - u_3 - 3u_4 - u_5 = 0, \boldsymbol{u} \ge \boldsymbol{0}\} .$$

Running the double description algorithm we get the following six extreme rays

$$(1,0,0,1,0),\ (1,0,3,0,0),\ (0,1,0,0,1),\ (0,3,0,1,0),\ (0,1,1,0,0),\ (1,0,0,0,3) .$$

Consequently, we find that (up to some redundant inequalities) P_ε^1 is given by

$$P_\varepsilon^1 = \{\boldsymbol{x} \in \mathbb{R}^2 : -2x_1 + x_2 \le 2\varepsilon, 2x_1 - x_2 \le 2\varepsilon, -x_1 - x_2 \le -3 + \varepsilon, x_1 - x_2 \le -1 + \varepsilon\} .$$

From Figure 9.14 we see that after normalization every inequality of the linear description of P_ε^1 corresponds to a nonzero extreme point of SP_∞ and vice versa. Note that as in part (i) you have to normalize the first and second inequalities of P_ε^1 to get the correspondence. The three extreme points of P_ε^1 for $\varepsilon \ge 0$ are $(1-\varepsilon, 2)$, $(1, 2-\varepsilon)$, $(1+\varepsilon, 2)$. In addition we need the direction vector $\boldsymbol{y} = (1, 2)$ of the extreme ray of P for a minimal pointwise description of P_ε^1.

(iii) The polyhedron P, see Figure 9.15, is an unbounded set having two extreme points $(1, 3)$ and $(2, 1)$, and two direction vectors $(1, 0)$, and $(1, 1)$ for its extreme rays. Consequently, the h_0-polar SP of P is the cone $SP = \{(h_1, h_2, h_0) \in \mathbb{R}^3 : h_1 - 3h_2 - h_0 \le 0, 2h_1 - h_2 - h_0 \le 0, h_1 \le 0, h_1 + h_2 \le 0\}$. Running the double description algorithm we find that SP is a pointed cone having four extreme rays $(0, 0, 1)$, $(0, -1, 3)$, $(-2, 1, -5)$ and $(-1, 1, -3)$. The set SP_∞ of the normed separators for P is obtained by intersecting SP with the constraints $-1 \le h_j \le 1$ for $j = 1, 2$. Using the homogenization (7.5) and running the double description algorithm we find that SP_∞ is a pointed polyhedron. Its minimal generator consists of the five extreme points

$$(0, 0, 0),\ (0, -1, 3),\ (-1, -1, 2),\ (-1, \tfrac{1}{2}, -\tfrac{5}{2}),\ (-1, 1, -3)$$

and the extreme ray given by the direction vector $(0, 0, 1)$.The set of normed separators SP_∞^* for SP is the polytope $SP_\infty^* = \{\boldsymbol{x} \in \mathbb{R}^3 : 2x_1 + x_2 - 5x_3 \ge 0, x_1 - x_2 + 2x_3 \ge 0, x_2 - x_3 \ge 0, 0 \le x_1 \le 1, 0 \le x_2 \le 1, 0 \le x_3 \le 1\}$. Running the double description algorithm we find that the (quasi-unique) minimal generator of SP_∞^* has the six extreme points

$$(0, 0, 0),\ (1, 0, 0),\ (1, 1, 0),\ (1, \tfrac{1}{2}, \tfrac{1}{2}),\ (1, 1, \tfrac{3}{5}),\ (\tfrac{1}{3}, 1, \tfrac{1}{3}) .$$

To answer the second part of this problem, we calculate the ε-solidification P_ε^1 of P in the ℓ_1-norm. To do so we proceed like in parts (i) and (ii). To calculate P_ε^1 we thus have to project out the μ and ν variables from the polyhedron

$$\begin{aligned} PP_\varepsilon^1 =& \{(\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\nu}) \in \mathbb{R}^6 : z_1 + z_2 - 4\mu_1 - 3\mu_2 - \nu_1 - 2\nu_2 \le \varepsilon, -z_1 + z_2 - 2\mu_1 + \mu_2 + \nu_1 \le \varepsilon, \\ & z_1 - z_2 + 2\mu_1 - \mu_2 - \nu_1 \le \varepsilon, -z_1 - z_2 + 4\mu_1 + 3\mu_2 + \nu_1 + 2\nu_2\varepsilon, \mu_1 + \mu_2 = 1, \boldsymbol{\mu} \ge \boldsymbol{0}, \boldsymbol{\nu} \ge \boldsymbol{0}\} . \end{aligned}$$

To carry out the projection we calculate the cone (7.8):

$$\begin{aligned} C =& \{(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^9 : -4u_1 - 2u_2 + 2u_3 + 4u_4 - u_5 + v_1 = 0, -3u_1 + u_2 - u_3 + 3u_4 - u_6 + v_1 = 0, \\ & - u_1 + u_2 - u_3 + u_4 - u_7 = 0, -2u_1 + 2u_4 - u_8 = 0, \boldsymbol{u} \ge \boldsymbol{0}\} . \end{aligned}$$

Running the double description algorithm we get the seven extreme rays

$$\begin{aligned} &(1, 0, 0, 1, 0, 0, 0, 0, 0), \quad (0, 1, 1, 0, 0, 0, 0, 0, 0), \quad (0, 0, 0, 0, 1, 1, 0, 0, 1), \quad (0, 1, 0, 0, 0, 3, 1, 0, 2), \\ &(0, 0, 0, 1, 1, 0, 1, 2, -3),\ (0, 1, 0, 3, 0, 0, 4, 6, -10),\ (0, 0, 1, 1, 4, 0, 0, 2, -2) , \end{aligned}$$

and thus we calculate

$$P_\varepsilon^1 = \{\boldsymbol{x} \in \mathbb{R}^2 : 2x_1 + x_2 \ge 5 - 2\varepsilon, x_1 - x_2 \ge -2 - \varepsilon, x_2 \ge 1 - \varepsilon, x_1 + x_2 \ge 3 - \varepsilon\} .$$

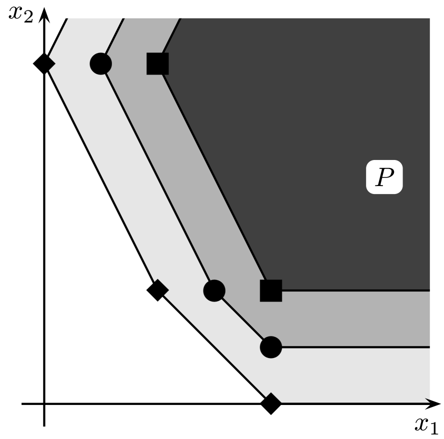


Fig. 9.15. ε-solidifications of P for $\varepsilon = 0$, $1/2$ and 1

From Figure 9.15 we see that after normalization every inequality of the linear description of P^1_ε corresponds to a nonzero extreme point of SP_∞ and vice versa. The extreme points of P^1_ε are $(1-\varepsilon, 3)$, $(2-\varepsilon, 1)$, $(2, 1-\varepsilon)$. In addition we need the two direction vectors of the extreme rays of P for a minimal pointwise description of P^1_ε.

From the above pointwise description of SP^*_∞ we find that $\widetilde{\boldsymbol{x}} = (1, 1, \frac{3}{5})$ is the unique optimal solution to $\max\{\boldsymbol{f}\boldsymbol{x} - f_0 x_{n+1} : (\boldsymbol{x}, x_{n+1}) \in SP^*_\infty\}$ and since $f_1x_1 + f_2x_2 - f_0x_3 = 1.6 > 0$ the point $\widetilde{\boldsymbol{x}}$ separates $(\boldsymbol{f}, f_0)$ from the cone SP. To find the most violated separator for $(\boldsymbol{f}, f_0)$ and SP, we apply the procedure described on pages 344-346; see (9.83) and (9.84). Solving $\max\{\boldsymbol{f}\boldsymbol{x} : \boldsymbol{x} \in P\}$ we find the (unique) optimizer $\boldsymbol{x}^{max} = (2, 1)$ and thus (9.83) applies. We get $\alpha = 1/2$ and thus $\boldsymbol{x}^0 = (1, \frac{1}{2}, \frac{1}{2})$ is a most violated separator for $(\boldsymbol{f}, f_0)$ and SP , i.e., a most violated separator for $(\boldsymbol{f}, f_0)$ and the cone SP cannot be found by solving the linear program $\max\{\boldsymbol{f}\boldsymbol{x} - f_0x_{n+1} : (\boldsymbol{x}, x_{n+1}) \in SP^*_\infty\}$.

(iv) Let $(\boldsymbol{h}, h_0)$ be an extreme point of SP_∞. Then $\boldsymbol{h}\boldsymbol{x} \le h_0$ for all $\boldsymbol{x} \in P$ and by Exercise 9.7(iv) $\boldsymbol{h}\boldsymbol{x} \le \widetilde{h}_0 = h_0 + \varepsilon$ for all $\boldsymbol{x} \in P^1_\varepsilon$ since $\|\boldsymbol{h}\|_\infty = 1$. Since $\varepsilon > 0$ and by Exercise 9.7(ix) $\dim P^1_\varepsilon = n$, there exists $(\boldsymbol{f}, \widetilde{f}_0) \in \mathbb{R}^{n+1}$ such that $\|\boldsymbol{f}\|_\infty = 1$, $\boldsymbol{f}\boldsymbol{x} \le \widetilde{f}_0$ defines a facet of P^1_ε and $A = \{\boldsymbol{x} \in P^1_\varepsilon : \boldsymbol{h}\boldsymbol{x} = \widetilde{h}_0\} \subseteq B = \{\boldsymbol{x} \in P^1_\varepsilon : \boldsymbol{f}\boldsymbol{x} = \widetilde{f}_0\}$. By Exercise 9.7(vi) $\boldsymbol{f}\boldsymbol{x} \le f_0 = \widetilde{f}_0 - \varepsilon$ for all $\boldsymbol{x} \in P$ and thus $(\boldsymbol{f}, f_0) \in SP_\infty$. From $A \subseteq B$ it follows that $\boldsymbol{h}\boldsymbol{x}^i = h_0$ implies $\boldsymbol{f}\boldsymbol{x}^i = f_0$ for $1 \le i \le p$ and likewise $\boldsymbol{h}\boldsymbol{y}^i = 0$ implies $\boldsymbol{f}\boldsymbol{y}^i = 0$ for $1 \le i \le r$. Suppose $h_j = \pm 1$ for some $j \in \{1, \ldots, n\}$. Since P is pointed, $\boldsymbol{h}\boldsymbol{x}^i = h_0$ for some $i \in \{1, \ldots, p\}$. But $\boldsymbol{x}^i \pm \varepsilon \boldsymbol{u}_j \in P^1_\varepsilon$, $\boldsymbol{h}(\boldsymbol{x}^i \pm \varepsilon \boldsymbol{u}_j) = h_0 \pm \varepsilon h_j = \widetilde{h}_0$ and thus $\boldsymbol{f}(\boldsymbol{x}^i \pm \varepsilon \boldsymbol{u}_j) = f_0 \pm \varepsilon f_j = \widetilde{f}_0 = f_0 + \varepsilon$ implies $f_j = h_j$. Since $(\boldsymbol{h}, h_0)$ is an extreme point it follows that $(\boldsymbol{h}, h_0) = (\boldsymbol{f}, f_0)$, i.e., $\boldsymbol{h}\boldsymbol{x} \le h_0 + \varepsilon$ defines a facet of P^1_ε. To show the reverse statement, suppose $(\boldsymbol{h}, \widetilde{h}_0) \in \mathbb{R}^{n+1}$ defines a facet of P^1_ε. We can assume WROG that $\|\boldsymbol{h}\|_\infty = 1$ and thus $(\boldsymbol{h}, h_0) \in SP_\infty$ where $h_0 = \widetilde{h}_0 - \varepsilon$. Denote by $(\boldsymbol{h}^i, h^i_0)$ for $1 \le i \le s$ the extreme points of SP_∞. Since SP_∞ has exactly one halfline, it follows that $(\boldsymbol{h}, h_0) = \sum_{i=1}^s \mu_i(\boldsymbol{h}^i, h^i_0) + \lambda(\boldsymbol{0}, 1)$ with $\mu_i \ge 0$, $\sum_{i=1}^s \mu_i = 1$ and $\lambda \ge 0$. Suppose $\lambda > 0$. Then $\boldsymbol{h}\boldsymbol{x} \le h_0 - \lambda$ for all $\boldsymbol{x} \in P$, because $(\boldsymbol{h}, h_0 - \lambda)$ is a nonnegative combination of $\boldsymbol{h}^i\boldsymbol{x} \le h^i_0$ – which by the first part define facets of P. But then $\boldsymbol{h}\boldsymbol{x} \le h_0 - \lambda + \varepsilon < \widetilde{h}_0$ for all $\boldsymbol{x} \in P^1_\varepsilon$ shows the contradiction. Consequently, $\lambda = 0$ and thus $(\boldsymbol{h}, \widetilde{h}_0) = \sum_{i=1}^s \mu_i(\boldsymbol{h}^i, \widetilde{h}^i_0)$ with

$\mu_i \geq 0$, $\sum_{i=1}^{s} \mu_i = 1$, where $\widetilde{h}_0^i = h_0^i + \varepsilon$ for $1 \leq i \leq s$. Since $\dim P_\varepsilon^1 = n$ it follows that $(\boldsymbol{h}, \widetilde{h}_0) = (\boldsymbol{h}^i, \widetilde{h}_0^i)$ for some $i \in \{1, \ldots, s\}$ since the linear description of a full dimensional polyhedron by its facets is unique *modulo* the multiplication by positive scalars; see page 129 of the book. Consequently, $(\boldsymbol{h}, h_0)$ defines an extreme point of SP_∞ and the proof is complete.

As we did not utilize the extremality of $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^p$ in the above argument it follows that the statement about the correspondence remains correct if P contains lines. If $p = 0$, then the feasible $\boldsymbol{x}^i$ needed to prove that $f_j = h_j$ can be chosen to equal $\mathbf{0}$ since $\mathbf{0} \in P$ in this case.