IMM
DEPARTMENT OF MATHEMATICAL MODELLING

Technical University of Denmark
DK-2800 Lyngby – Denmark

J. No. ALO
1.2.1999
HBN/ms

ALGORITHMS FOR LINEAR OPTIMIZATION

An Introduction

Hans Bruun Nielsen

2nd Edition

IMM

Preface

These notes were written for the students in the course *04412 Optimization and Data Fitting 2* at the Technical University of Denmark. Hopefully, they can also be useful to others.

Suggestions for improvement are very welcome.

Lyngby, December 1997

Hans Bruun Nielsen

email: hbn@imm.dtu.dk
url: http://www.imm.dtu.dk/~hbn

I am grateful to professor C. Roos for pointing out some errors, which have been corrected in this version.

Lyngby, February 1998. Hans Bruun Nielsen

More errors have been found and corrected in this second edition.

Lyngby, January 1999. Hans Bruun Nielsen

Contents

Notation

We use the following notation.

- $\lceil x \rceil$ = smallest integer such that $\lceil x \rceil \geq x$.
- $a \sim b$: a is of the same order of magnitude as b.
- $\mathbb{R}_+$ ($\mathbb{R}_{++}$): The set of nonnegative (strictly positive) real numbers.
- $\mathbb{R}^n$: The n-dimensional vector space over $\mathbb{R}$.
- $\mathbb{R}^{m\times n}$: Set of $m\times n$ matrices with real elements.
- $\mathcal{A}$: Set of indices, $\mathcal{A} = \{i_1, \ldots, i_\nu\} \subseteq \{1, \ldots, n\}$, $\nu = \#\mathcal{A}$.
- $\#\mathcal{A}$: Number of elements in the set $\mathcal{A}$.
- $\mathbf{x} = [x_i]$: Typical name for a vector. Normally, $\mathbf{x}$ is a *column vector,* i.e. $\mathbf{x} \in \mathbb{R}^n$ is equivalent with $\mathbf{x} \in \mathbb{R}^{n\times 1}$. The *transpose* of $\mathbf{x}$ is the row vector $\mathbf{x}^T \in \mathbb{R}^{1\times n}$.
- $\mathbf{e}$ ($\boldsymbol{0}$) : Vector of all ones (all zeros).
- $\mathbf{e}_i$: Vector with $(\mathbf{e}_i)_i = 1$ and all other components equal to zero.
- $\mathbf{x}_{\mathcal{A}}$: The subvector $[x_{i_1} \ \ldots \ x_{i_\nu}]^T$ with $i_k \in \mathcal{A}$.
- $|\mathbf{x}|$ = $[|x_i|]$. Similarly, $\mathbf{x}^{-1} = [x_i^{-1}]$ and $\mathbf{x}^{1/2} = \sqrt{\mathbf{x}} = [\sqrt{x_i}]$.
- $\mathbf{x} \leq \boldsymbol{y}$ $\Leftrightarrow$ $x_i \leq y_i$, $i = 1, \ldots, n$.
- $\mathbf{x}^T\boldsymbol{y}$ = $x_1y_1 + \cdots + x_ny_n$. Inner produt.
- $\mathbf{x} \odot \boldsymbol{y}$ = $[x_iy_i]$, elementwise (or *Hadamard*) product. (In MATLAB: `x .* y`).
- $\|\mathbf{x}\|_p$ = $(x_1^p + \cdots x_n^p)^{1/p}$. Vector norm. Special cases:
  $\|\mathbf{x}\|_1 = |x_1| + \cdots + |x_n| = \mathbf{e}^T|\mathbf{x}|$, $\quad \|\mathbf{x}\|_\infty = \max\{|x_1|, \cdots, |x_n|\}$,
  $\|\mathbf{x}\|_2 = \sqrt{x_1^2 + \cdots + x_n^2} = \sqrt{\mathbf{x}^T\mathbf{x}}$.

- $\boldsymbol{A} = [a_{ij}]$: Typical name for a matrix. The *transpose* of $\boldsymbol{A}$ is denoted $\boldsymbol{A}^T$ with $(\boldsymbol{A}^T)_{ij} = a_{ji}$.
- $\boldsymbol{A}^{-T}$: Short for $\left(\boldsymbol{A}^{-1}\right)^T = \left(\boldsymbol{A}^T\right)^{-1}$.
- $\mathrm{diag}(\mathbf{x})$: Diagonal matrix with x_i as the ith diagonal element.
- $\boldsymbol{I}$ = $\mathrm{diag}(\boldsymbol{e})$.
- $\boldsymbol{0}$: Matrix of all zeros. It should be clear from the context, whether $\boldsymbol{0}$ is a matrix or a vector.
- $\mathbf{X}$ = $\mathrm{diag}(\mathbf{x})$. Common usage in Interior Point environments.
- $\boldsymbol{A}_{:,j}$ ($\boldsymbol{A}_{i,:}$): Column (row) vector with elements from the jth column (ith row) of $\boldsymbol{A}$. Examples: $\boldsymbol{I}_{:,k} = \mathbf{e}_k$, $\boldsymbol{I}_{k,:} = \mathbf{e}_k^T$.
- $\boldsymbol{A}_{\mathcal{A}}$: The submatrix af $\boldsymbol{A}$ consisting of the columns $\boldsymbol{A}_{:,j_1}, \ldots, \boldsymbol{A}_{:,j_\nu}$ with $j_k \in \mathcal{A}$.
- $\|\boldsymbol{A}\|_F$ = $\left(\Sigma_{ij}\, a_{ij}\right)^{1/2}$. Frobenius (or Euclidean) matrix norm.
- $\|\boldsymbol{A}\|_p$ = $\max_{\|\mathbf{x}\|_p=1} \left\{\|\boldsymbol{A}\mathbf{x}\|_p\right\}$. Induced matrix norm. Special cases:
  $\|\boldsymbol{A}\|_1 = \max_j \left\{\Sigma_i\ |a_{ij}|\right\}$, $\qquad \|\boldsymbol{A}\|_\infty = \max_i \left\{\Sigma_j\ |a_{ij}|\right\}$,
  $\|\boldsymbol{A}\|_2 = \max_j \left\{\sigma_j(\boldsymbol{A})\right\} = \left\{\max_j \left\{\lambda_j(\boldsymbol{A}^T\boldsymbol{A})\right\}\right\}^{1/2}$.
  Here $\sigma_j(\boldsymbol{B})$ $\left(\lambda_j(\boldsymbol{B})\right)$ is the jth singular value (eigenvalue) of matrix $\boldsymbol{B}$.
- $\nabla_x\varphi$: Vector with jth component $\frac{\partial\varphi}{\partial x_j}$.
- $\Delta\varphi$: Matrix with (i,j)th component $\frac{\partial^2\varphi}{\partial x_i\partial x_j}$.
- $\mathcal{P}$ ($\mathcal{P}^+$) : Feasible domain for problem (P) (interior of this domain).
- $\mathcal{P}^*$: Set of optimal solutions for problem (P).
- ε_M : Machine accuracy (also called unit round off). In all examples we have used MATLAB, which follows the IEEE standard: $\varepsilon_M = 2^{-53} \simeq 1.11 \cdot 10^{-16}$. Results are shown rounded to an appropriate number of digits.

1. Introduction

A *linear optimization (LO) problem*[1] consists in finding a vector that optimizes (i.e. minimizes or maximizes) a linear *objective function* subject to a finite set of *linear constraints*. The constraints may be equality constraints or inequality constraints. As two examples consider the so-called *canonical form*

$$(P_c) \qquad \min\{\, \boldsymbol{c}^T\mathbf{x} \;:\; \boldsymbol{A}\mathbf{x} \geq \boldsymbol{b},\; \mathbf{x} \geq \boldsymbol{0} \,\} \;, \tag{1.1a}$$

and the *standard form*

$$(P_s) \qquad \min\{\, \boldsymbol{c}^T\mathbf{x} \;:\; \boldsymbol{A}\mathbf{x} = \boldsymbol{b},\; \mathbf{x} \geq \boldsymbol{0} \,\} \;. \tag{1.1b}$$

In both formulations the matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and the vectors $\boldsymbol{c}, \mathbf{x} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^m$. The two problems are different – but in Section 2.1 we shall see that it is easy to get from one formulation to the other.

For a problem (P) the *feasible domain* $\mathcal{P}$ is the set of points in $\mathbb{R}^n$, for which all the constraints are satisfied. If this set is empty (i.e. the constraints are inconsistent), then the problem is said to be infeasible, otherwise it is *feasible*. If the objective function is bounded on $\mathcal{P}$, then the problem is said to be *bounded*, otherwise it is unbounded. The *optimal set* $\mathcal{P}^*$ is the set of optimal solutions. This set is empty if the problem is infeasible or unbounded.

Assume that (P) is feasible, and let $\mathbf{x}$ and $\boldsymbol{y}$ denote two points in $\mathcal{P}$. Then

$$\boldsymbol{c}^T\boldsymbol{y} = \boldsymbol{c}^T\mathbf{x} - \boldsymbol{c}^T(\mathbf{x}-\boldsymbol{y}) \;.$$

If $\boldsymbol{c}^T(\mathbf{x}-\boldsymbol{y}) > 0$, then $\mathbf{x}$ cannot be an optimal solution. Put another way: If

1) This is often denoted a *linear programming (LP) problem*, but the name linear optimization is more descriptive and is becoming increasingly popular in the literature.

$$\boldsymbol{c}^T\boldsymbol{h} < 0 \;, \tag{1.2}$$

then $\boldsymbol{c}^T(\mathbf{x}+t\boldsymbol{h}) < \boldsymbol{c}^T\mathbf{x}$ for all $t > 0$, and we say that $\boldsymbol{h}$ is a *descent direction*. The relative gain

$$\left(\boldsymbol{c}^T\mathbf{x} - \boldsymbol{c}^T(\mathbf{x}+t\boldsymbol{h})\right) / \|t\boldsymbol{h}\|_2$$

is maximal if $\boldsymbol{h}$ is chosen as the *steepest descent direction,*

$$\boldsymbol{h}_{\mathrm{sd}} = -\boldsymbol{c} \;.$$

An immediate consequence of these observations is that a point in $\mathcal{P}^*$ must lie on the boundary of $\mathcal{P}$, since otherwise we can decrease the value of the objective function by following a descent direction to the boundary.

Example 1.1. Consider the problem

$$(P) \qquad \min\left\{\; 2x_1 + 3x_2 \;:\; \begin{array}{l} \frac{1}{2}x_1 + x_2 \geq 1 \\ \frac{2}{3}x_1 - x_2 \geq -2 \end{array} ,\; x_1, x_2 \geq 0 \;\right\} \;.$$

The feasible domain is shown in Figure 1.1, with the lines $x_1 + 2x_2 = 2$ and $x_1 - 1.5x_2 = -3$ indicated by thin line. Figure 1.2 shows the space of descent directions corresponding to the vector $\boldsymbol{c} = [\,2 \;\; 3\,]^T$

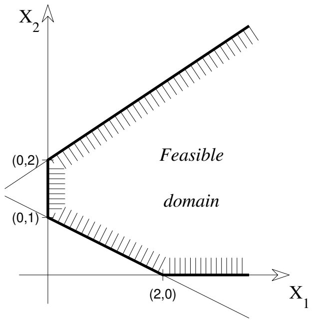


Figure 1.1. *Feasible domain*

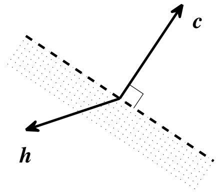


Figure 1.2. *Descent directions*

In the next figure we indicate a possible way of solving the optimization problem. The ideas behind this algorithm are generalized in the Simplex method, see Chapter 4.

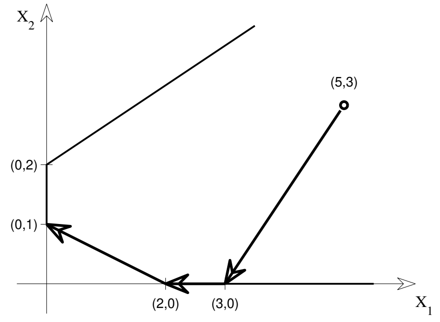


Figure 1.3.
Simplex idea

We start at the point[2] $\mathbf{x}=(5,\ 3)$, which is in the interior of the feasible domain. From that point we follow the steepest descent direction until we reach the boundary of $\mathcal{P}$. This happens at the point $\mathbf{x}=(3,\ 0)$. Using $\boldsymbol{h}=(-1,\ 0)$, we stay on the boundary and decrease $\boldsymbol{c}^T\mathbf{x}$ further. The next constraint is reached at the point $\mathbf{x}=(2,\ 0)$, and with $\boldsymbol{h}=(-2,\ 1)$ we get on to $\mathbf{x}=(0,\ 1)$. Now it is not possible to decrease the objective function further. Thus, the optimal set consists of one point, $\mathcal{P}^*=\{\mathbf{x}^*\}=\{(0,\ 1)\}$, and the minimum value is $\boldsymbol{c}^T\mathbf{x}^*=3$.

Example 1.2. For the problem of Example 1.1 it is easily seen that if we change the first two inequalities, then the constraints are inconsistent. This means that the problem

$$(P') \qquad \min\left\{\ 2x_1+3x_2\ :\ \begin{matrix}\frac{1}{2}x_1+x_2\le 1\\ \frac{2}{3}x_1-x_2\le -2\end{matrix}\ ,\ x_1,x_2\ge 0\ \right\}\ .$$

is infeasible.

If the problem is changed to

$$(Q) \qquad \min\left\{\ x_1+2x_2\ :\ \begin{matrix}\frac{1}{2}x_1+x_2\ge 1\\ \frac{2}{3}x_1-x_2\ge -2\end{matrix}\ ,\ x_1,x_2\ge 0\ \right\}\ .$$

then $\boldsymbol{c}=[\,1\ \ 2\,]^T$ is orthogonal to the line segment between $(2,0)$ and $(0,1)$. In this case $\mathcal{Q}^*$ consists of all the points in that segment, $\mathcal{Q}^*=\{\mathbf{x}^*\mid\mathbf{x}^*=(2,\ 0)+t(-2,\ 1),\ 0\le t\le 1\}$, with $\boldsymbol{c}^T\mathbf{x}^*=2$. This shows that the optimal solution to an LO problem may not be unique.

[2] In this example we identify vectors in $\boldsymbol{R}^2$ with geometric vectors, and e.g. write $\mathbf{x}=(5,\ 3)$ instead of the more appropriate $\mathbf{x}=[\,5\ \ 3\,]^T$.

Finally, consider

$$(U) \qquad \min\left\{\ -2x_1-3x_2\ :\ \begin{matrix}\frac{1}{2}x_1+x_2\ge 1\\ \frac{2}{3}x_1-x_2\ge -2\end{matrix}\ ,\ x_1,x_2\ge 0\ \right\}\ .$$

It is clear, that the points $\mathbf{x}=(5,\ 3)+t(3,\ 2)$ are feasible for all $t\ge 0$, and $\mathbf{c}^T\mathbf{x}=-19-12t\to-\infty$ as $t\to\infty$. This is an unbounded problem.

As mentioned in Example 1.1, Figure 1.3 can be said to indicate the basic idea in the *Simplex method:* The optimal solution is found as a vertex (corner point) of the feasible domain, and can be determined by going from one vertex to the next as long as the objective function decreases. This method was originally formulated by Dantzig in 1947 and has since been constantly improved.

For about 40 years the Simplex method was the only real workhorse for solving LO problems. Starting in 1984, however, the publication of Karmarkar's algorithm provoked an overwhelming interest in the socalled *Interior point methods* (IPM), and now a number of efficient variants exist.

The *logarithmic barrier function* is basic for most (if not all) IPMs. It is defined via the function[3]

$$\psi(t)=t-\log(1+t) \quad \text{for} \quad t\in(-1,\infty)\ . \tag{1.3}$$

The graph is shown below. Note, that $t=0$ is the unique minimizer of ψ with $\psi(0)=0$.

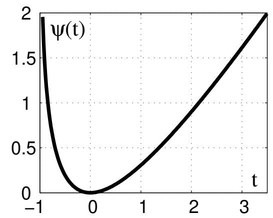


Figure 1.4. *Graph of ψ, (1.3)*

To see how this function can be used in the LO context, consider the formulation (1.1a), and introduce the *surplus vector* $\mathbf{s}(\mathbf{x})$ defined by

[3] log is the natural (or Naperian) logarithm. This implies that the derivative of $\log x$ is x^{-1}.

$$\boldsymbol{s}(\mathbf{x}) = \boldsymbol{A}\mathbf{x} - \boldsymbol{b}\ . \tag{1.4}$$

For a feasible point it follows that $\mathbf{x} \ge \boldsymbol{0}$ and $\boldsymbol{s}(\mathbf{x}) \ge \boldsymbol{0}$. We shall assume that in $\mathcal{P}^+$, the interior of the feasible domain, we have strict inequality both for $\mathbf{x}$ and for $\boldsymbol{s}$, i.e.

$$\mathbf{x} > \boldsymbol{0} \quad \text{and} \quad \boldsymbol{s}(\mathbf{x}) > \boldsymbol{0} \quad \text{for } \mathbf{x} \in \mathcal{P}^+\ . \tag{1.5}$$

Then we can define the *logarithmic barrier function*

$$\varphi_\mu(\mathbf{x}) = \boldsymbol{c}^T\mathbf{x} - \mu\left(\sum_{j=1}^{n} \log x_j + \sum_{i=1}^{m} \log s_i(\mathbf{x})\right)\ , \tag{1.6}$$

where the *barrier parameter* μ is a positive number. The logarithm terms are introduced to keep $\mathbf{x}$ in $\mathcal{P}^+$. For a fixed value of μ the minimizer $\mathbf{x}(\mu)$ of $\varphi_\mu(\mathbf{x})$ in $\mathcal{P}^+$ is found by solving a nonlinear system of equations (see Section 3.1), and letting $\mu \searrow 0$ we shall see in Section 3.2 that $\mathbf{x}(\mu) \to \mathbf{x}^* \in \mathcal{P}^*$.

Example 1.3. For the problem of Example 1.1 we have

$$\boldsymbol{A} = \begin{bmatrix} 1/2 & 1 \\ 2/3 & -1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}\ .$$

We have used the IPM sketched above and found the results for $x_j(\mu)$ and $s_i(\mu) = s_i(\mathbf{x}(\mu))$ shown in in the left part of Figure 1.5. In the right part we give these results in the usual x_1x_2-plane, cf. Figure 1.3.

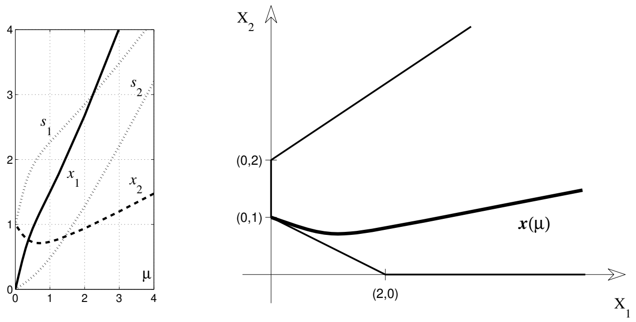


Figure 1.5. *Interior point method*

In Chapter 2 we give the theoretical background for LO algorithms. Chapters 3 and 4 discuss Interior Point and Simplex methods, respectively, and in Chapter 5 we introduce another class of methods, called continuation methods.

The present note is only a brief introduction to this vast field. For a more profound discussion an abundance of textbooks are available. We shall refer to [7] and [13] for the theory and programming of the Simplex method, while [39] is a thorough introduction to the theory seen from an IPM point of view. Mathematical models employing linear optimization are discussed in [4], and [19] is an introduction in Danish to both theory, the Simplex method and applications.

2. Theoretical Background

In this chapter we give the background for algorithms for solving LO problems. First, in Section 2.1 we show that any problem can be written in canonical or in standard form (1.1). In Section 2.2 we discuss the important aspect of *duality*, and show that with any problem

$$(P) \qquad \min\{\ f_P(\mathbf{x})\ :\ \text{P\_constraints}\ \}\ ,$$

there is an associated *dual problem*

$$(D) \qquad \max\{\ f_D(\mathbf{y})\ :\ \text{D\_constraints}\ \}\ .$$

In this connection (P) is called the *primal problem*. In Table 2.1, p. 20 we give rules for transformation between (P) and (D).

In Sections 2.3–4 we introduce *complementarity*, and give some fundamental results for LO problems. Finally, in Sections 2.5–7 we discuss *projection*, the *logarithmic barrier function* and the *central path*, which are basic for Interior Point methods.

2.1. Formulation

A linear optimization problem has the form

$$(P) \qquad \min\{\ f_P(\mathbf{x})\ :\ \text{Set of linear constraints}\ \} \tag{2.1a}$$

or

$$(P) \qquad \max\{\ f_P(\mathbf{x})\ :\ \text{Set of linear constraints}\ \}\ . \tag{2.1b}$$

Here, $\mathbf{x} \in \mathbb{R}^n$ and $f_P(\mathbf{x})$ is a linear objective function

$$f_P(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + f_0\ ,$$

where $\mathbf{c} \in \mathbb{R}^n$ and $f_0 \in \mathbb{R}$ are given.

We first note that the two versions of (2.1) are equivalent in the sense that $\max\{f_P(\mathbf{x})\ :\ \text{Constraints}\}$ has the same optimal solution(s) $\mathbf{x}^*$ as $\min\{-f_P(\mathbf{x}) : \text{Constraints}\}$, and that the constant f_0 has no influence on $\mathbf{x}^*$. Therefore, we need only discuss problems of the form

$$(P) \qquad \min\{\ f_P(\mathbf{x}) \equiv \mathbf{c}^T\mathbf{x}\ :\ \text{Set of linear constraints}\ \}\ . \tag{2.2}$$

There are the following possibilities for the constraints,

1° Equality constraints

$$\mathbf{a}_i^T\mathbf{x} = b_i\ . \tag{2.3a}$$

2° Inequality constraints

$$\mathbf{a}_i^T\mathbf{x} \ge b_i \quad \text{or} \quad \mathbf{a}_i^T\mathbf{x} \le b_i\ . \tag{2.3b}$$

3° Simple constraints

$$x_j \ge \ell_j \quad \text{or} \quad x_j \le u_j\ . \tag{2.3c}$$

Here, the vectors $\mathbf{a}_i \in \mathbb{R}^n$ and the scalars $b_i, \ell_j, u_j \in \mathbb{R}$ are given.

Generally, the set of constraints includes more than one of these types. To ease the discussion, however, we shall assume that the problem is given in (or rewritten to) either canonical or standard form, (1.1):

$$(P_c) \qquad \min\{\ \mathbf{c}^T\mathbf{x}\ :\ \boldsymbol{A}\mathbf{x} \ge \mathbf{b},\ \mathbf{x} \ge \boldsymbol{0}\ \}\ ,$$
$$(P_s) \qquad \min\{\ \mathbf{c}^T\mathbf{x}\ :\ \boldsymbol{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \ge \boldsymbol{0}\ \}\ .$$

Example 2.1. In this example we show how some frequently occuring constraints can be rewritten to conform with (1.1). The list is not exhaustive.

Given constraint	Canonical form	Standard form
$\mathbf{a}_i^T\mathbf{x} \le b_i$	$-\mathbf{a}_i^T\mathbf{x} \ge -b_i$	$\mathbf{a}_i^T\mathbf{x} + s_i = b_i,\ s_i \ge 0$
$\mathbf{a}_i^T\mathbf{x} \ge b_i$		$\mathbf{a}_i^T\mathbf{x} - s_i = b_i,\ s_i \ge 0$
$\mathbf{a}_i^T\mathbf{x} = b_i$	$\mathbf{a}_i^T\mathbf{x} \ge b_i$ and $-\mathbf{a}_i^T\mathbf{x} \ge -b_i$	

The extra variables $\{s_i\}$ are called *slack variables*. They are closely related to the surplus vector $\boldsymbol{s}(\mathbf{x})$ defined in (1.4). For each slack variable n is increased by one, $\boldsymbol{A}$ is augmented with a column from the unit matrix $\boldsymbol{I}$, and $\boldsymbol{c}$ is augmented with a zero element.

In both formulations simple constraints on the variables can be treated as follows,

$x_j \geq \ell_j$	$x_j = \ell_j + x'_j,$	$x'_j \geq 0$
$x_j \leq u_j$	$x_j = u_j - x'_j,$	$x'_j \geq 0$
$\ell_j \leq x_j \leq u_j$	$x_j = \ell_j + x'_j + x''_j$ with $x'_j + x''_j = u_j - \ell_j$,	$x'_j, x''_j \geq 0$
x_j free	$x'_j - x''_j,$	$x'_j, x''_j \geq 0$

In all cases x_j is replaced by x'_j (possibly augmented with x''_j), and the transformation may imply changes in $\boldsymbol{A}$, $\boldsymbol{b}$ and $\boldsymbol{c}$.

Example 2.2. Consider the problem

$$(P) \qquad \max\left\{ 13x_1 + 10x_2 \ : \ \begin{matrix} 3x_1 + 2x_2 \leq 24 \\ -x_1 + 7x_2 = 15 \end{matrix}, \ x_1 \geq 0 \right\} .$$

The variable x_2 is free, and we write it as $x_2 = x'_2 - x''_2$ with $x'_2, x''_2 \geq 0$. By further use of the tools of example 2.1 we see that (P) is equivalent with

$$(P_c) \qquad \min\left\{ \boldsymbol{c}^T\mathbf{x} \ : \ \boldsymbol{A}\mathbf{x} \geq \boldsymbol{b}, \ \mathbf{x} \geq \boldsymbol{0} \right\} ,$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x'_2 \\ x''_2 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} -13 \\ -10 \\ 10 \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} -3 & -2 & 2 \\ -1 & 7 & -7 \\ 1 & -7 & 7 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} -24 \\ 15 \\ -15 \end{bmatrix} .$$

Alternatively, we can reformulate the problem to

$$(P_s) \qquad \min\left\{ \boldsymbol{c}^T\mathbf{x} \ : \ \boldsymbol{A}\mathbf{x} = \boldsymbol{b}, \ \mathbf{x} \geq \boldsymbol{0} \right\} .$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x'_2 \\ x''_2 \\ s_1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} -13 \\ -10 \\ 10 \\ 0 \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 3 & 2 & -2 & 1 \\ -1 & 7 & -7 & 0 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 24 \\ 15 \end{bmatrix} .$$

Example 2.3. The problem of Example 1.1 can be written in standard form with

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ s_1 \\ s_2 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} 1/2 & 1 & -1 & 0 \\ -2/3 & 1 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} .$$

Example 2.4. Consider a problem in standard form with n variables and $\text{rank}(\boldsymbol{A}) = m < n$. Let $\mathcal{B}$ be a subset of m elements from $\{1,\ldots,n\}$, and $\mathcal{C} = \{1,\ldots,n\}\backslash\mathcal{B}$. We can split $\boldsymbol{A}$ into $\boldsymbol{B} = \boldsymbol{A}_{\mathcal{B}} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{C} = \boldsymbol{A}_{\mathcal{C}} \in \mathbb{R}^{m\times n-m}$ obtained by picking the columns with index from $\mathcal{B}$ and $\mathcal{C}$, respectively, and we let $(\boldsymbol{c}_{\mathcal{B}}, \boldsymbol{c}_{\mathcal{C}})$ and $(\mathbf{x}_{\mathcal{B}}, \mathbf{x}_{\mathcal{C}})$ denote a similar splitting of $\boldsymbol{c}$ and $\mathbf{x}$. Then

$$(P) \qquad \min\left\{ \boldsymbol{c}^T\mathbf{x} \ : \ \boldsymbol{A}\mathbf{x} = \boldsymbol{b} \ , \ \mathbf{x} \geq \boldsymbol{0} \right\}$$

is equivalent with

$$(P') \qquad \min\left\{ \boldsymbol{c}_{\mathcal{B}}^T\mathbf{x}_{\mathcal{B}} + \boldsymbol{c}_{\mathcal{C}}^T\mathbf{x}_{\mathcal{C}} \ : \ \boldsymbol{B}\mathbf{x}_{\mathcal{B}} = \boldsymbol{b} - \boldsymbol{C}\mathbf{x}_{\mathcal{C}} \ , \ \mathbf{x}_{\mathcal{B}}, \mathbf{x}_{\mathcal{C}} \geq \boldsymbol{0} \right\} .$$

Now, assume that $\text{rank}(\boldsymbol{B}) = m$. Then

$$\mathbf{x}_{\mathcal{B}} = \boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{B}^{-1}\boldsymbol{C}\mathbf{x}_{\mathcal{C}} \equiv \widetilde{\boldsymbol{b}} - \widetilde{\boldsymbol{A}}\mathbf{x}_{\mathcal{C}} \tag{2.4a}$$

and

$$\boldsymbol{c}_{\mathcal{B}}^T\mathbf{x}_{\mathcal{B}} + \boldsymbol{c}_{\mathcal{C}}^T\mathbf{x}_{\mathcal{C}} = \boldsymbol{c}_{\mathcal{B}}^T\widetilde{\boldsymbol{b}} + \left(\boldsymbol{c}_{\mathcal{C}}^T - \boldsymbol{C}^T\boldsymbol{B}^{-T}\boldsymbol{c}_{\mathcal{C}}\right)^T\mathbf{x}_{\mathcal{C}} \equiv c_0 + \widetilde{\boldsymbol{c}}^T\mathbf{x}_{\mathcal{C}} . \tag{2.4b}$$

Therefore, (P) is equivalent with the following problem in canonical form with $n-m$ variables and m constraints,

$$(P_c) \qquad \min\left\{ \widetilde{\boldsymbol{c}}^T\boldsymbol{z} \ : \ -\widetilde{\boldsymbol{A}}\boldsymbol{z} \geq -\widetilde{\boldsymbol{b}}, \ \boldsymbol{z} \geq \boldsymbol{0} \right\} .$$

If the original problem has $p < m$ *free variables,* and their indices are part of $\mathcal{B}$, then the constraint $\mathbf{x}_{\mathcal{B}} \geq \boldsymbol{0}$ should be used only for the $n-p$ non free variables. This implies that in (P_c) the matrix $\widetilde{\boldsymbol{A}}$ is $(m-p)\times(n-m)$.

Example 2.5. The problem treated in Example 2.2 can be written

$$(P) \qquad \min\{ -13x_1 - 10x_2 + 0x_3 \ : \ \boldsymbol{A}\mathbf{x} = \boldsymbol{b}, \ x_1, x_3 \geq 0 \} ,$$

where x_3 is a slack variable, and

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 & 1 \\ -1 & 7 & 0 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 24 \\ 15 \end{bmatrix} .$$

We take $\mathcal{B} = \{3, 2\}$, including the index of the free variable x_2, and get

$$\boldsymbol{B} = \begin{bmatrix} 1 & 2 \\ 0 & 7 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 3 \\ -1 \end{bmatrix}, \quad \widetilde{\boldsymbol{A}} = \begin{bmatrix} 23/7 \\ -1/7 \end{bmatrix}, \quad \widetilde{\boldsymbol{b}} = \begin{bmatrix} 138/7 \\ 15/7 \end{bmatrix} ,$$

$$\widetilde{\boldsymbol{c}} = -13 - \begin{bmatrix} 3 \\ -1 \end{bmatrix}^T \begin{bmatrix} 1 & 0 \\ 2 & 7 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ -10 \end{bmatrix}^{-1} = -13 - \tfrac{10}{7} = -\tfrac{101}{7} .$$

Now, (2.4a) takes the form $\begin{bmatrix} x_3 \\ x_2 \end{bmatrix} = \begin{bmatrix} 138/7 \\ 15/7 \end{bmatrix} - \begin{bmatrix} 23/7 \\ -1/7 \end{bmatrix} x_1$,
and $x_3 \geq 0$, x_2 free, lead to the equivalent problem

$$(P') \qquad \min\left\{ -\frac{1380}{7} - \frac{101}{7}x_1 \;:\; \frac{138}{7} - \frac{23}{7}x_1 \geq 0 \;,\; x_1 \geq 0 \right\} ,$$

or, simpler

$$(P'') \qquad \min\{ -x_1 \;:\; -x_1 \geq -6 \;,\; x_1 \geq 0 \} \;.$$

Now it is easily seen that the optimal set is $x_1^* = 6$, $x_2^* = 3$ and $x_3^* = 0$ with the optimal value $\boldsymbol{c}^T\mathbf{x}^* = 108$.

Example 2.6. Many programs for solving LO problems demand that the problem is presented either in canonical form or in standard form. This, however, can give unnecessary trouble during the solution process caused by linear dependent rows in $\boldsymbol{A}$ (arising from transforming equality constraints to canonical form) or linear dependent columns (arising from the treatment of free variables).

Rather than expanding the number of constraints and/or variables, it may be better to reduce it, as described in Example 2.4. A deeper analysis is given in [39, Theorem I.15]: Let the LO problem (P) have m constraints and n variables, including m_0 equality constraints and n_0 free variables. Then there is an equivalent canonical problem (P'), for which $m'+n' \leq m+n-(m_0+n_0)$.

The better LO programs all include a *preprocessing* of the problem with the aim of identifying and removing free variables and redundancy; see [1], [15].

2.2. Duality

This is a fundamental concept in linear optimization – and plays an important role in the theory, in applications as well as in algorithms. Therefore it plays a major role in all textbooks on LO, but often the dual problem appears as Godsent. Our presentation is inspired by [5], and further leans on [20], [39], [38].

First, consider a problem in standard form

$$(P) \qquad \min\{ \boldsymbol{c}^T\mathbf{x} \;:\; \boldsymbol{A}\mathbf{x} = \mathbf{b},\; \mathbf{x} \geq \boldsymbol{0} \} \;, \tag{2.5}$$

with n variables and m constraints. We assume that the problems is feasible, i.e. the domain $\mathcal{P}$ is not empty. We relax (P) to

$$(R) \qquad \min\{ \boldsymbol{c}^T\mathbf{x} + \boldsymbol{y}^T(\mathbf{b} - \boldsymbol{A}\mathbf{x}) \;:\; \mathbf{x} \geq \boldsymbol{0} \} \;, \tag{2.6}$$

where $\boldsymbol{y}$ is an arbitrary vector in $\mathbb{R}^m$. The elements y_i are called *penalty factors*. The relaxed problem is also an LO problem, and for any $\boldsymbol{y} \in \mathbb{R}^m$ it has the simple feasible domain $\mathcal{R} = \mathbb{R}^n_+$.

We introduce the notation

$$\begin{aligned} f_R(\mathbf{x}, \boldsymbol{y}) &= \boldsymbol{c}^T\mathbf{x} + \boldsymbol{y}^T(\mathbf{b} - \boldsymbol{A}\mathbf{x}) \\ &= \mathbf{x}^T(\boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y}) + \mathbf{b}^T\boldsymbol{y} \;, \end{aligned} \tag{2.7}$$

and let $\mathbf{x}_R^*(\boldsymbol{y})$ denote an optimal solution to (R) for a given $\boldsymbol{y}$.

It is seen that $\mathcal{P} \subseteq \mathcal{R}$ and that $f_R(\mathbf{x}, \boldsymbol{y}) = f_P(\mathbf{x})$ for $\mathbf{x} \in \mathcal{P}$ (implying that $\mathbf{b} - \boldsymbol{A}\mathbf{x} = \boldsymbol{0}$). Therefore,

$$f_R(\mathbf{x}_R^*(\boldsymbol{y}), \boldsymbol{y}) \leq f_P(\mathbf{x}^*) \quad \text{for all } \boldsymbol{y} \in \mathbb{R}^m \;.$$

This means that for any m-vector $\boldsymbol{y}$ the solution to the relaxed problem gives a lower bound for the optimal value for (P). This is true also, if we choose $\boldsymbol{y} = \boldsymbol{y}^*$, the vector that maximizes this lower bound,

$$\max_{\boldsymbol{y} \in \mathcal{F}} \{ f_R(\mathbf{x}_R^*(\boldsymbol{y}), \boldsymbol{y}) \} \leq f_P(\mathbf{x}^*) \;, \tag{2.8}$$

where $\mathcal{F} = \mathbb{R}^m$ is the space from which we may choose $\boldsymbol{y}$.

The optimal value for the relaxed problem is easily computed: From the second expression for f_R in (2.7) and from the positivity of $\mathbf{x} \in \mathcal{R}$ it follows that if any $(\boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y})_i < 0$, then we let $x_i \to \infty$, and get the minimal value $-\infty$. To get the maximal lower bound we therefore must specify the condition

$$\boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y} \geq \boldsymbol{0} \;,$$

and see that then $\mathbf{x}_R^*(\boldsymbol{y}) = \boldsymbol{0}$ is optimal for (R) with $f_R(\mathbf{x}_R^*(\boldsymbol{y}), \boldsymbol{y}) = \mathbf{b}^T\boldsymbol{y}$, cf. (2.7). Thus, $\boldsymbol{y}^*$ is an optimal solution to the LO problem

$$(D) \qquad \max\{ \mathbf{b}^T\boldsymbol{y} \;:\; \boldsymbol{A}^T\boldsymbol{y} \leq \boldsymbol{c} \} \;. \tag{2.9}$$

This is a *dual problem* associated with the *primal problem* (P) given by (2.5). Note that in this LO problem the variables y_j are free.

Example 2.7. For the problem discussed in Example 1.1 we gave a standard formulation in Example 2.3. Using this we find the dual problem

$$(D) \qquad \max\left\{ y_1 + 2y_2 \;:\; \begin{array}{ll} \frac{1}{2}y_1 - \frac{2}{3}y_2 \le 2\,, & -y_1 \le 0 \\ y_1 + y_2 \le 3\,, & y_2 \le 0 \end{array} \right\} .$$

The feasible domain $\mathcal{D}$ is shown below. The solution can be found as discussed in connection with Figure 1.3, except that now we have a maximization problem. The solution is $\boldsymbol{y}^* = (3,0)$ with $f_D(\boldsymbol{y}^*) = 3$.

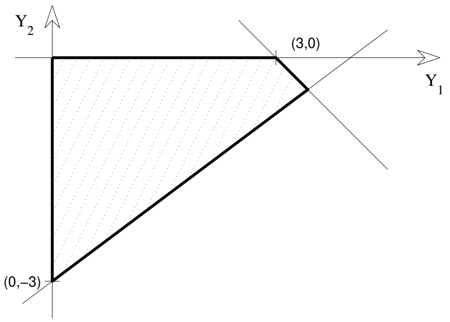


Figure 2.1. *$\mathcal{D}$*

Next, consider a problem in canonical form,

$$(P) \qquad \min\{\, \boldsymbol{c}^T\mathbf{x} \;:\; \boldsymbol{A}\mathbf{x} \ge \boldsymbol{b},\ \mathbf{x} \ge \boldsymbol{0} \,\} . \tag{2.10a}$$

Again, we want to find a lower bound on $f(\mathbf{x}^*)$ by introducing a relaxed problem (R) with objective function given by (2.7). Now, however, we must restrict the penalty factors to ensure that (2.8) holds. Since $\boldsymbol{b} - \boldsymbol{A}\mathbf{x} \le \boldsymbol{0}$, the restriction $\boldsymbol{y} \in \mathcal{F} = \mathbb{R}^m_+$ guarantees this, and we see that the dual problem to (2.10a) is

$$(D) \qquad \max\{\, \boldsymbol{b}^T\boldsymbol{y} \;:\; \boldsymbol{A}^T\boldsymbol{y} \le \boldsymbol{c},\ \boldsymbol{y} \ge \boldsymbol{0} \,\} . \tag{2.10b}$$

Example 2.8. The problem of Example 1.1 is stated in canonical form, and from (2.10) we see that its dual is

$$(D) \qquad \max\left\{ y_1 - 2y_2 \;:\; \begin{array}{l} \frac{1}{2}y_1 + \frac{2}{3}y_2 \le 2 \\ y_1 - y_2 \le 3 \end{array},\ \ y_1, y_2 \ge 0 \right\} .$$

This is seen to be equivalent with the result of Example 2.7 when we replace y_2 by $-y_2$.

Other types of constraints can be treated similarly, and to make a long story short,

- For each constraint in (P) there is a variable in (D).
- For each variable in (P) there is a constraint in (D).
- If (P) is a maximization problem, then the relaxed problem provides an **upper** bound on $f_P(\mathbf{x}^*)$.

Primal (P)		Dual (D)	
Objective	$\min\{\boldsymbol{c}^T\mathbf{x}\}$ $\max\{-\boldsymbol{c}^T\mathbf{x}\}$	$\max\{\boldsymbol{b}^T\boldsymbol{y}\}$ $\min\{-\boldsymbol{b}^T\boldsymbol{y}\}$	Objective
Constraint	$\mathbf{a}_i^T\mathbf{x} = b_i$ $\mathbf{a}_i^T\mathbf{x} \ge b_i$ $\mathbf{a}_i^T\mathbf{x} \le b_i$	y_i free $y_i \ge 0$ $y_i \le 0$	Variable
Variable	x_j free $x_j \ge 0$ $x_j \le 0$	$(\boldsymbol{A}^T)_{j,:}\boldsymbol{y} = c_j$ $(\boldsymbol{A}^T)_{j,:}\boldsymbol{y} \le c_j$ $(\boldsymbol{A}^T)_{j,:}\boldsymbol{y} \ge c_j$	Constraint

Table 2.1. *Transformation between primal and dual*

Exercise 2.9. Find the dual of problem (P) in Example 2.2, and show that it has the optimal value $f_D(\boldsymbol{y}^*) = 108$.

Example 2.10. The roles of the primal and dual problem may interchange: Consider the LO problem from Example 2.8. It can be written as

$$(Q) \qquad \min\left\{ -y_1 + 2y_2 \;:\; \begin{array}{l} \frac{1}{2}y_1 + \frac{2}{3}y_2 \le 2 \\ y_1 - y_2 \le 3 \end{array},\ \ y_1, y_2 \ge 0 \right\} ,$$

and by the rules of Table 2.1 we find its dual

$$(D) \qquad \max\left\{ 2x_1 + 3x_2 \;:\; \begin{array}{l} \frac{1}{2}x_1 + x_2 \le -1 \\ \frac{2}{3}x_1 - x_2 \le 3 \end{array},\ \ x_1, x_2 \le 0 \right\} .$$

If we replace $\mathbf{x}$ by $-\mathbf{x}$, we see that this is equivalent with the original problem from Example 1.1.

Now, let (P) be a minimization problem, and therefore (D) a maximization problem. For any pair of feasible vectors $\mathbf{x}\in\mathcal{P}$ and $\boldsymbol{y}\in\mathcal{D}$ we define the *duality gap*

$$\Delta(\mathbf{x},\boldsymbol{y}) \equiv f_P(\mathbf{x}) - f_D(\boldsymbol{y}) = \boldsymbol{c}^T\mathbf{x} - \boldsymbol{b}^T\boldsymbol{y}\ . \tag{2.11}$$

From the derivation – i.e. the definition of (D) – follows

The weak duality property

$$\boldsymbol{c}^T\mathbf{x} - \boldsymbol{b}^T\boldsymbol{y} \geq 0 \quad \text{for all } \mathbf{x}\in\mathcal{P},\ \boldsymbol{y}\in\mathcal{D}\ . \tag{2.12}$$

Further, it follows that if $\Delta(\mathbf{x},\boldsymbol{y})=0$ (the duality gap is closed), then $f_P(\mathbf{x})$ and $f_D(\boldsymbol{y})$ are respectively as small and as large as possible, which means that both $\mathbf{x}$ and $\boldsymbol{y}$ are optimal:

$$\Delta(\mathbf{x},\boldsymbol{y}) = 0 \quad\Rightarrow\quad \mathbf{x}\in\mathcal{P}^* \text{ and } \boldsymbol{y}\in\mathcal{D}^*\ . \tag{2.13}$$

In (Examples 1.1, 2.7) and (Example 2.5, Exercise 2.9) we saw that the converse may be true: $\mathbf{x}\in\mathcal{P}^*,\ \boldsymbol{y}\in\mathcal{D}^* \Rightarrow \Delta(\mathbf{x},\boldsymbol{y})=0$. In Section 2.4 we shall see, under what circumstances this is the case, but first we need some extra theory.

2.3. Complementarity

Consider the primal problem in canonical form,

$$(P) \qquad \min\{\ \boldsymbol{c}^T\mathbf{x}\ :\ \boldsymbol{A}\mathbf{x}\geq\boldsymbol{b},\ \mathbf{x}\geq\boldsymbol{0}\ \}\ , \tag{2.14a}$$

with the corresponding dual

$$(D) \qquad \max\{\ \boldsymbol{b}^T\boldsymbol{y}\ :\ \boldsymbol{A}^T\boldsymbol{y}\leq\boldsymbol{c},\ \boldsymbol{y}\geq\boldsymbol{0}\ \}\ . \tag{2.14b}$$

For any pair of feasible vectors $\mathbf{x}\in\mathcal{P}$ and $\boldsymbol{y}\in\mathcal{D}$ we define the *surplus vectors* (cf. (1.4))

$$\boldsymbol{s}(\mathbf{x}) = \boldsymbol{A}\mathbf{x} - \boldsymbol{b} \quad\text{and}\quad \boldsymbol{s}(\boldsymbol{y}) = \boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y}\ . \tag{2.15}$$

Note, that $\mathbf{x}, \boldsymbol{s}(\boldsymbol{y})\in\mathbb{R}^n_+$ and $\boldsymbol{y}, \boldsymbol{s}(\mathbf{x})\in\mathbb{R}^m_+$.

The definition (2.15) is equivalent with

$$\boldsymbol{b} = \boldsymbol{A}\mathbf{x} - \boldsymbol{s}(\mathbf{x}) \qquad \boldsymbol{c} = \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{s}(\boldsymbol{y})\ . \tag{2.16}$$

We insert these in the expression (2.11) for the duality gap,

$$\begin{aligned}\Delta(\mathbf{x},\boldsymbol{y}) &= \boldsymbol{c}^T\mathbf{x} - \boldsymbol{b}^T\boldsymbol{y}\\ &= \mathbf{x}^T\boldsymbol{A}^T\boldsymbol{y} + \mathbf{x}^T\boldsymbol{s}(\boldsymbol{y}) - \boldsymbol{y}^T\boldsymbol{A}\mathbf{x} + \boldsymbol{y}^T\boldsymbol{s}(\mathbf{x})\\ &= \mathbf{x}^T\boldsymbol{s}(\boldsymbol{y}) + \boldsymbol{y}^T\boldsymbol{s}(\mathbf{x})\ .\end{aligned} \tag{2.17}$$

All the vectors are nonnegative, so the duality gap vanishes only if each $x_j^* s_j(\boldsymbol{y}^*) = 0$ and $y_i^* s_i(\mathbf{x}^*) = 0$, or[1)]

$$\mathbf{x}^*\odot\boldsymbol{s}(\boldsymbol{y}^*) = \boldsymbol{0}\ ,\quad \boldsymbol{y}^*\odot\boldsymbol{s}(\mathbf{x}^*) = \boldsymbol{0}\ , \tag{2.18a}$$

This is the socalled *complementarity property* of optimal solution pairs $(\mathbf{x}^*\in\mathcal{P}^*,\ \boldsymbol{y}^*\in\mathcal{D}^*)$. If the optimal solutions to (P) and (D) further have the additional property

$$\mathbf{x}^* + \boldsymbol{s}(\boldsymbol{y}^*) > \boldsymbol{0}\ ,\quad \boldsymbol{y}^* + \boldsymbol{s}(\mathbf{x}^*) > \boldsymbol{0}\ , \tag{2.18b}$$

then we say that (P) and (D) have the *strict complementarity property.*

Example 2.11. In Examples 1.1 and 2.7–2.8 we treated a problem of the form (2.14) with

$$\boldsymbol{A} = \begin{bmatrix} 1/2 & 1\\ 2/3 & -1\end{bmatrix},\quad \boldsymbol{b} = \begin{bmatrix} 1\\ -2\end{bmatrix},\quad \boldsymbol{c} = \begin{bmatrix} 2\\ 3\end{bmatrix}.$$

The optimal solutions and corresponding surplus vectors are

$$\mathbf{x}^* = \begin{bmatrix} 0\\ 1\end{bmatrix},\quad \boldsymbol{y}^* = \begin{bmatrix} 3\\ 0\end{bmatrix},\quad \boldsymbol{s}(\mathbf{x}^*) = \begin{bmatrix} 0\\ 1\end{bmatrix},\quad \boldsymbol{s}(\boldsymbol{y}^*) = \begin{bmatrix} 1/2\\ 0\end{bmatrix},$$

and we find

$$\mathbf{x}^*\odot\boldsymbol{s}(\boldsymbol{y}^*) = \begin{bmatrix} 0\\ 0\end{bmatrix},\qquad \mathbf{x}^* + \boldsymbol{s}(\boldsymbol{y}^*) = \begin{bmatrix} 1/2\\ 1\end{bmatrix} > \boldsymbol{0}\ ,$$

$$\boldsymbol{y}^*\odot\boldsymbol{s}(\mathbf{x}^*) = \begin{bmatrix} 0\\ 0\end{bmatrix},\qquad \boldsymbol{y}^* + \boldsymbol{s}(\mathbf{x}^*) = \begin{bmatrix} 3\\ 1\end{bmatrix} > \boldsymbol{0}\ .$$

Thus, both (2.18a) and (2.18b) are satisfied in this case.

[1)] See "NOTATION" for the definition of $\boldsymbol{u}\odot\boldsymbol{v}$.

For a problem in standard form

$$(P) \qquad \min\{\, \boldsymbol{c}^T\mathbf{x} \;:\; \boldsymbol{A}\mathbf{x} = \boldsymbol{b},\; \mathbf{x} \geq \boldsymbol{0} \,\} \;, \tag{2.19a}$$

the dual problem is given by (2.9), which can be reformulated to

$$(D) \qquad \max\{\, \boldsymbol{b}^T\boldsymbol{y} \;:\; \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\; \boldsymbol{s} \geq \boldsymbol{0} \,\} \;. \tag{2.19b}$$

Thus, $\boldsymbol{y}$ is free and $\boldsymbol{s}(\mathbf{x}) = \boldsymbol{0}$ for a feasible $\mathbf{x}$. The problem (2.19) is said to have the strict complementarity property if it has an optimal solution $\mathbf{x}^*$, $\boldsymbol{y}^*$, $\boldsymbol{s}^*$ satisfying

$$\mathbf{x}^* \odot \boldsymbol{s}^* = \boldsymbol{0} \quad \text{and} \quad \mathbf{x}^* + \boldsymbol{s}^* > \boldsymbol{0} \;. \tag{2.20}$$

This result is often refered to as the *Goldman–Tucker Theorem.*

Note, that in this formulation we have the surplus vectors $\boldsymbol{s}(\mathbf{x}) = \boldsymbol{0}$ and $\boldsymbol{s}(\boldsymbol{y}) = \boldsymbol{s}$. For $\mathbf{x} \in \mathcal{P}$ and $\boldsymbol{y}, \boldsymbol{s} \in \mathcal{D}$ the expression (2.17) reduces to

$$\Delta(\mathbf{x}, \boldsymbol{y}) \equiv \boldsymbol{c}^T\mathbf{x} - \boldsymbol{b}^T\boldsymbol{y} = \mathbf{x}^T\boldsymbol{s} \;. \tag{2.21}$$

When (P) has the strict complementarity property, we can split the index set $\{1, 2, \ldots, n\}$ into the sets

$$\mathcal{B} \equiv \{\, i \mid x_i^* > 0\} \quad \text{and} \quad \mathcal{C} \equiv \{\, i \mid s_i^* > 0\} \;. \tag{2.22}$$

It follows from (2.20) that $\mathcal{B}$ and $\mathcal{C}$ are disjoint and $\mathcal{B} \cup \mathcal{C} = \{1, \ldots, n\}$. This splitting is called an *optimal partition* of (P).

Finally, consider the special LO problem

$$(S) \qquad \min\{\, \boldsymbol{q}^T\mathbf{x} \;:\; \boldsymbol{M}\mathbf{x} \geq -\boldsymbol{q},\; \mathbf{x} \geq \boldsymbol{0} \,\} \;, \tag{2.23a}$$

where $\boldsymbol{M}$ is *skew-symmetric* (i.e. $\boldsymbol{M}^T = -\boldsymbol{M}$), and $\boldsymbol{q} \geq \boldsymbol{0}$. By the rules of Table 2.1 the dual of this problem is

$$(DS') \qquad \max\{\, -\boldsymbol{q}^T\boldsymbol{y} \;:\; \boldsymbol{M}^T\boldsymbol{y} \leq \boldsymbol{q},\; \boldsymbol{y} \geq \boldsymbol{0} \,\} \;.$$

Since $\boldsymbol{M}^T = -\boldsymbol{M}$, this is seen to be equivalent with

$$(DS) \qquad \min\{\, \boldsymbol{q}^T\boldsymbol{y} \;:\; \boldsymbol{M}\boldsymbol{y} \geq -\boldsymbol{q},\; \boldsymbol{y} \geq \boldsymbol{0} \,\} \;. \tag{2.23b}$$

Problem (S) is said to be *self-dual.*

Lemma 2.1. The self-dual problem (S) with $\boldsymbol{q} \geq \boldsymbol{0}$ satisfies

1° Problem (S) is feasible and bounded.

2° The optimal set $\mathcal{S}^*$ contains $\widetilde{\mathbf{x}}$ that satisfies the complementarity properties $\widetilde{\mathbf{x}} \odot \boldsymbol{s}(\widetilde{\mathbf{x}}) = \boldsymbol{0}$ and $\widetilde{\mathbf{x}} + \boldsymbol{s}(\widetilde{\mathbf{x}}) > \boldsymbol{0}$.

3° S has an optimal partition $\mathcal{B}$, $\mathcal{C}$.

Proof: See [39, Chapter 2]. □

Example 2.12. Consider

$$(S) \qquad \min\left\{\, 2x_2 \;:\; \begin{array}{c} 0x_1 + x_2 \geq 0 \\ -x_1 + 0x_2 \geq -2 \end{array},\; x_1, x_2 \geq 0 \,\right\} \;.$$

The feasible set is the strip $0 \leq x_1 \leq 2$, $x_2 \geq 0$, and the optimal set is $\mathcal{S}^* = \{(u, 0) \mid u \in [0, 2]\}$. The corresponding surplus vectors are $\boldsymbol{s}(\mathbf{x}^*) = (0,\; 2-u)$. The points $\widetilde{\mathbf{x}}$, corresponding to u in the open interval $u \in (0, 2)$ are seen to satisfy (2.20), and the optimal partition of (S) is given by $\mathcal{B} = \{1\}$, $\mathcal{C} = \{2\}$.

2.4. Fundamental Theorems

Theorem 2.2. For the given pair (P) and (D) there are three alternatives

1° Both (P) and (D) are feasible and bounded, and there exists a strictly complementary optimal pair $(\widetilde{\mathbf{x}} \in \mathcal{P}^*,\; \widetilde{\boldsymbol{y}} \in \mathcal{D}^*)$ with $\boldsymbol{c}^T\widetilde{\mathbf{x}} = \boldsymbol{b}^T\widetilde{\boldsymbol{y}}$.

2° Either (P) or (D) is unbounded and the other is infeasible.

3° Both (P) and (D) are infeasible.

Proof: See Appendix A.1. □

Example 2.13. We already saw examples of point 1° in the theorem. Now, consider the infeasible problem from Example 1.2

$$(P') \qquad \min\left\{ 2x_1+3x_2 \ : \ \begin{array}{l} \frac{1}{2}x_1+x_2 \le 1 \\ \frac{2}{3}x_1-x_2 \le -2 \end{array}, \ x_1,x_2\ge 0 \right\} .$$

By the rules of Table 2.1 its dual is

$$(D') \qquad \max\left\{ y_1-2y_2 \ : \ \begin{array}{l} \frac{1}{2}y_1+\frac{2}{3}y_2 \le 2 \\ y_1-y_2 \le 3 \end{array}, \ y_1,y_2\le 0 \right\} .$$

This is a feasible problem, e.g. satisfied by $y_1=y_2=-t$ for all $t\ge 0$. The corresponding objective value is t, and we see that (D') is unbounded.

The unbounded primal problem

$$(U) \qquad \min\left\{ -2x_1-3x_2 \ : \ \begin{array}{l} \frac{1}{2}x_1+x_2 \ge 1 \\ \frac{2}{3}x_1-x_2 \ge -2 \end{array}, \ x_1,x_2\ge 0 \right\}$$

has the dual

$$(DU) \qquad \max\left\{ y_1-2y_2 \ : \ \begin{array}{l} \frac{1}{2}y_1+\frac{2}{3}y_2 \le -2 \\ y_1-y_2 \le -3 \end{array}, \ y_1,y_2\ge 0 \right\} .$$

This is seen to be an infeasible problem.

Consider the problem in standard form,

$$(P) \qquad \min\{\ \boldsymbol{c}^T\mathbf{x} \ : \ \boldsymbol{A}\mathbf{x}=\boldsymbol{b},\ \mathbf{x}\ge \boldsymbol{0}\ \} \quad ,$$

with the corresponding dual

$$(D) \qquad \max\{\ \boldsymbol{b}^T\boldsymbol{y} \ : \ \boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{s}=\boldsymbol{c},\ \boldsymbol{s}\ge \boldsymbol{0}\ \} \quad .$$

If (P) is feasible and bounded, then Theorem 2.2 implies that there is an optimal partition $\mathcal{B}, \mathcal{C}$. We can split the matrices and vectors as in Example 2.4, and see that the constraints imply

$$\boldsymbol{B}\mathbf{x}_{\mathcal{B}}=\boldsymbol{b}-\boldsymbol{C}\mathbf{x}_{\mathcal{C}}\ , \quad \boldsymbol{B}^T\boldsymbol{y}=\boldsymbol{c}_{\mathcal{B}}-\boldsymbol{s}_{\mathcal{B}}\ , \quad \boldsymbol{C}^T\boldsymbol{y}=\boldsymbol{c}_{\mathcal{C}}-\boldsymbol{s}_{\mathcal{C}}\ . \tag{2.24a}$$

Now, from (2.20)–(2.22) it follows that $\widetilde{\mathbf{x}}_{\mathcal{C}}=\boldsymbol{0}$ and $\widetilde{\boldsymbol{s}}_{\mathcal{B}}=\boldsymbol{0}$, so that $(\widetilde{\mathbf{x}}_{\mathcal{B}},\ \widetilde{\boldsymbol{y}},\ \widetilde{\boldsymbol{s}}_{\mathcal{C}})$ can be found from

$$\boldsymbol{B}\widetilde{\mathbf{x}}_{\mathcal{B}}=\boldsymbol{b}\ , \quad \boldsymbol{B}^T\widetilde{\boldsymbol{y}}=\boldsymbol{c}_{\mathcal{B}}\ , \quad \widetilde{\boldsymbol{s}}_{\mathcal{C}}=\boldsymbol{c}_{\mathcal{C}}-\boldsymbol{C}^T\widetilde{\boldsymbol{y}}\ . \tag{2.24b}$$

Let $\#\mathcal{B}$ denote the number of elements in $\mathcal{B}$. There are the following cases to consider,

1° $\#\mathcal{B}=\operatorname{rank}(\boldsymbol{B})=m$: $(\widetilde{\mathbf{x}}_{\mathcal{B}},\ \widetilde{\boldsymbol{y}},\ \widetilde{\boldsymbol{s}}_{\mathcal{C}})$ are uniquely determined by (2.24b), and $(\widetilde{\mathbf{x}},\ \widetilde{\boldsymbol{y}})$ is the unique optimal solution pair.

2° $\operatorname{rank}(\boldsymbol{B})=\#\mathcal{B}<m$: $\widetilde{\mathbf{x}}$ is the only point in $\mathcal{P}^*$, while
$$\widetilde{\boldsymbol{y}}=\widetilde{\boldsymbol{y}}^{(0)}+\boldsymbol{N}\boldsymbol{z}$$
with $\boldsymbol{B}^T\widetilde{\boldsymbol{y}}^{(0)}=\boldsymbol{c}_{\mathcal{B}}$ and the columns of $\boldsymbol{N}$ form a basis of the null space $\mathcal{N}(\boldsymbol{B}^T)$. $\boldsymbol{z}$ is any vector in $\mathbb{R}^{m-\#\mathcal{B}}$ for which $\boldsymbol{c}_{\mathcal{C}}-\boldsymbol{C}^T(\widetilde{\boldsymbol{y}}^{(0)}+\boldsymbol{N}\boldsymbol{z})>\boldsymbol{0}$.

3° $\operatorname{rank}(\boldsymbol{B})=m<\#\mathcal{B}$: $\widetilde{\boldsymbol{y}}$ and $\widetilde{\boldsymbol{s}}$ are unique, while
$$\widetilde{\mathbf{x}}=\widetilde{\mathbf{x}}^{(0)}+\boldsymbol{G}\boldsymbol{z}$$
with $\boldsymbol{B}\widetilde{\mathbf{x}}^{(0)}=\boldsymbol{b}$ and $\operatorname{span}(\boldsymbol{G})=\mathcal{N}(\boldsymbol{B})$. $\boldsymbol{z}$ is any vector in $\mathbb{R}^{\#\mathcal{B}-m}$ for which $\widetilde{\mathbf{x}}^{(0)}+\boldsymbol{G}\boldsymbol{z}>\boldsymbol{0}$.

4° $\operatorname{rank}(\boldsymbol{B})<\min\{\#\mathcal{B},m\}$: Neither $\widetilde{\mathbf{x}}$ nor $(\widetilde{\boldsymbol{y}},\ \widetilde{\boldsymbol{s}})$ is unique.

In the cases 2°–4° problem (P) is said to be *degenerate.*

Example 2.14. Problem (Q) from Example 1.2 is degenerate. In standard form it corresponds to $n=4$, $m=2$,

$$\boldsymbol{A}=\begin{bmatrix} 1/2 & 1 & -1 & 0 \\ 2/3 & -1 & 0 & -1 \end{bmatrix}, \quad \boldsymbol{b}=\begin{bmatrix} 1 \\ -2 \end{bmatrix}, \quad \boldsymbol{c}=\begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \end{bmatrix} .$$

$\mathcal{B}=\{1,\ 2,\ 4\}$, $\mathcal{C}=\{3\}$ is an optimal partition of (Q), corresponding to

$$\boldsymbol{B}=\begin{bmatrix} 1/2 & 1 & 0 \\ 2/3 & -1 & -1 \end{bmatrix}, \quad \boldsymbol{C}=\begin{bmatrix} -1 \\ 0 \end{bmatrix} .$$

We are in case 3° and find

$$\widetilde{\mathbf{x}}_{\mathcal{B}}=\begin{bmatrix} 2 \\ 0 \\ 10/3 \end{bmatrix}+z\begin{bmatrix} -2 \\ 1 \\ -7/3 \end{bmatrix}, \quad \widetilde{\boldsymbol{y}}=\begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad \widetilde{\boldsymbol{s}}=\begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} .$$

The condition $\widetilde{\mathbf{x}}_{\mathcal{B}}>\boldsymbol{0}$ is satisfied for $0<z<1$. Note, that $\boldsymbol{c}^T\widetilde{\mathbf{x}}=\boldsymbol{b}^T\widetilde{\boldsymbol{y}}=2$ for all z in this interval.

For $z=0$ the 2nd coordinate of $\widetilde{\mathbf{x}}_{\mathcal{B}}$ is zero, corresponding to $\mathcal{B}=\{1,\ 4\}$. The dual slack vector $\widetilde{s}$ is unaffected by z, so $\mathcal{C} = \{3\}$. The vectors

$$\mathbf{x}^* = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 10/3 \end{bmatrix}, \quad \boldsymbol{y}^* = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}^* = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix}$$

are also optimal, but here the condition $\mathbf{x}^*{+}\mathbf{s}^* > \boldsymbol{0}$ is violated by the 2nd coordinate. Similarly, for $z=1$ we also get an optimal solution satisfying the complementarity condition $\mathbf{x}^* \odot \boldsymbol{s}^* = \boldsymbol{0}$, but $\mathbf{x}^* + \boldsymbol{s}^* > \boldsymbol{0}$ is violated by the 1st coordinate.

Example 2.15. Let $\mathcal{B}$ and $\mathcal{C}$ with $\mathcal{B} \cup \mathcal{C} = \{1, \ldots, n\}$ be given, and split the vectors as in Example 2.4. Let $\widetilde{\mathbf{x}}_{\mathcal{C}} = \boldsymbol{0}$, $\widetilde{\mathbf{s}}_{\mathcal{B}} = \boldsymbol{0}$, and compute $\widetilde{\mathbf{x}}_{\mathcal{B}}$, $\widetilde{\boldsymbol{y}}$, $\widetilde{\mathbf{s}}_{\mathcal{C}}$ by means of (2.24). If $\widetilde{\mathbf{x}}_{\mathcal{B}} \geq \boldsymbol{0}$ and $\widetilde{\mathbf{s}}_{\mathcal{C}} \geq \boldsymbol{0}$, then $(\widetilde{\mathbf{x}}, \widetilde{\boldsymbol{y}}, \widetilde{\mathbf{s}})$ is optimal. This follows from the fact that both $\widetilde{\mathbf{x}}$ and $(\widetilde{\boldsymbol{y}}, \widetilde{\mathbf{s}})$ are feasible, and by using (2.21) we see that

$$\Delta(\widetilde{\mathbf{x}}, \widetilde{\boldsymbol{y}}) = \widetilde{\mathbf{x}}^T\widetilde{\mathbf{s}} = \widetilde{\mathbf{x}}_{\mathcal{B}}^T\widetilde{\mathbf{s}}_{\mathcal{B}} + \widetilde{\mathbf{x}}_{\mathcal{C}}^T\widetilde{\mathbf{s}}_{\mathcal{C}} = \widetilde{\mathbf{x}}_{\mathcal{B}}^T\boldsymbol{0} + \boldsymbol{0}^T\widetilde{\mathbf{s}}_{\mathcal{C}} = 0 \ .$$

We can use this to generate LO problems with known solution: Choose m, n, $\#\mathcal{B}$ and γ_x, γ_y, γ_s, where

n is the number of variables,

m is the number of equality constraints, $m < n$,

$\#\mathcal{B}$ is the number of elements in $\mathcal{B}$, $\#\mathcal{B} < n$,

γ_x, γ_y, γ_s are "descaling factors" for $\mathbf{x}$, $\boldsymbol{y}$ and $\boldsymbol{s}$.

The matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and the vector $\boldsymbol{y}^*$ are generated with uniform random elements in $[-1, 1]$ and $[-\gamma_y, \gamma_y]$, respectively. To get $\mathbf{x}^*$ and $\boldsymbol{s}^*$ with positive elements, we first generate the vectors $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{w}$ with

$$u_i = \frac{i}{n}, \quad v_i = \gamma_x^{u_i} \cdot u_i, \quad w_i = \gamma_s^{u_i} \cdot u_i, \quad i = 1, \ldots, n \ .$$

Next, the partition is found as follows: If $\#\mathcal{B} \leq \frac{n}{2}$ then $\mathcal{B} = \{2, 4, \ldots, 2\#\mathcal{B}\}$, otherwise we add a sufficient number of odd indices. $\mathcal{C}$ consists of the unused elements of $\{1, \ldots, n\}$, and we let

$$(x_i^*, \ s_i^*) = \begin{cases} (v_i, \ 0) & \text{for } i \in \mathcal{B} \\ (0, \ w_i) & \text{for } i \in \mathcal{C} \end{cases}$$

Finally, $\boldsymbol{b} := \boldsymbol{A}\mathbf{x}^*$ and $\boldsymbol{c} := \boldsymbol{A}^T\boldsymbol{y}^* + \boldsymbol{s}^*$. This generator is used to compare the algorithms in Sections 3 – 5.

The problem of feasability can also be investigated via the following, famous lemma, dating back to 1902.

Lemma 2.3. (Farkas' Lemma). Given $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{b} \in \mathbb{R}^m$, and consider the pairs of problems

	Problem A	*Problem B*
1	$\boldsymbol{A}\mathbf{x} \geq \boldsymbol{b}, \ \ \mathbf{x} \geq \boldsymbol{0}$	$\boldsymbol{A}^T\boldsymbol{y} \leq \boldsymbol{0}, \ \ \boldsymbol{b}^T\boldsymbol{y} > 0, \ \ \boldsymbol{y} \geq \boldsymbol{0}$
2	$\boldsymbol{A}^T\mathbf{x} \leq \boldsymbol{b}$	$\boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{0}, \ \ \boldsymbol{b}^T\boldsymbol{y} < 0, \ \ \boldsymbol{y} \geq \boldsymbol{0}$
3	$\boldsymbol{b}^T\mathbf{x} < 0, \ \ \boldsymbol{A}^T\mathbf{x} \geq \boldsymbol{0}$	$\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}, \ \ \boldsymbol{y} \geq \boldsymbol{0}$
4	$\boldsymbol{b}^T\mathbf{x} < 0, \ \ \boldsymbol{A}\mathbf{x} = \boldsymbol{0}$	$\boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{b}$

In each case either Problem A or Problem B (but not both) has a solution.

Proof: See Appendix A.2. □

Cases 3 and 4 in Farkas' Lemma have important applications in nonlinear optimization, where $\boldsymbol{b}$ is the *gradient* $\boldsymbol{g}$ of an objective function and we seek a *descent direction* $\boldsymbol{h}$ from a current iterate $\mathbf{x}$. As in the discussion in Chapter 1 we demand $\mathbf{g}^T\boldsymbol{h} < 0$, and the direction $\boldsymbol{h}$ is subject to constraints ensuring that the next iterate stays inside a certain domain.

Corresponding to Case 3 in Lemma 2.2 there are the two alternatives: If we can find $\boldsymbol{h}$ satisfying

$$\mathbf{g}^T\boldsymbol{h} < 0 \quad \text{and} \quad \boldsymbol{A}^T\boldsymbol{h} \geq \boldsymbol{0} \ ,$$

then $\mathbf{x}{+}t\boldsymbol{h}$ is a better approximation to the minimizer for t in a certain interval $(0, \delta)$. Otherwise, $\mathbf{x}$ an is optimal solution to the nonlinear problem, characterized by

$$\boldsymbol{g} = \boldsymbol{A}\boldsymbol{y} \quad \text{with} \quad \boldsymbol{y} \geq \boldsymbol{0} \ .$$

An alternative formulation and proof is given in the next example.

In Section 2.6 we apply Case 4 of Farkas' Lemma: Either we can get a better approximation $\mathbf{x}+t\boldsymbol{h}$ by using $\boldsymbol{h}$ satisfying

$$\mathbf{g}^T\boldsymbol{h} < 0 \quad \text{and} \quad \boldsymbol{A}\boldsymbol{h} = \boldsymbol{0} \ , \tag{2.25a}$$

or we have found the optimal solution, characterized by

$$\boldsymbol{g} = \boldsymbol{A}^T\boldsymbol{y} \tag{2.25b}$$

for some $\boldsymbol{y}$ in the relevant vector space. In Section 2.5 we give an alternative proof of this version of Farkas' Lemma.

Example 2.16. Another version of Farkas' Lemma is found in [11, Lemma 9.2.4]:

Given vectors $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ and $\mathbf{g}$ with $\mathbf{a}_j, \mathbf{g} \in \mathbb{R}^m$. Then the set

$$S = \left\{ \boldsymbol{h} \in \mathbb{R}^m \mid \mathbf{g}^T\boldsymbol{h} < 0, \ \mathbf{a}_i^T\boldsymbol{h} \geq 0, \ i=1,\ldots,n \right\}$$

is empty if and only if there exist multipliers $y_j \geq 0$ such that $\mathbf{g} = \sum_{j=1}^{n} y_j \mathbf{a}_j$.

Letting $\boldsymbol{A} = [\mathbf{a}_1 \ \ldots \ \mathbf{a}_n]$ and replacing $(\boldsymbol{b}, \mathbf{x})$ by $(\boldsymbol{g}, \boldsymbol{h})$, we see that this version is idential with Case 3 in Lemma 2.2.

This version of Farkas' Lemma has a more intuitive proof: In Figure 2.2 we show $\mathbf{a}_1$, $\mathbf{a}_2$ and $\mathbf{g}$ in the case $m = n = 2$, and dashed lines orthogonal to these, cf. Figure 1.2. The set S is indicated by the shaded area. If $\boldsymbol{g}$ is in the hatched angle between $\mathbf{a}_1$ and $\mathbf{a}_2$, then $\boldsymbol{g} = y_1\mathbf{a}_1 + y_2\mathbf{a}_2$ with $y_1, y_2 \geq 0$, and S is empty.

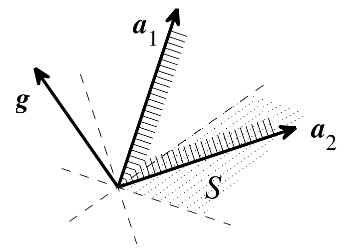


Figure 2.2. *Farkas' Lemma*

2.5. Projection onto Null Space

This is an important tool in connection with interior point methods.

Given a rectangular matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with $m < n$, and with linearly independent rows, i.e. $\mathrm{rank}(\boldsymbol{A}) = m$. The *row space* $\mathcal{R}(\boldsymbol{A})$ and *null space* $\mathcal{N}(\boldsymbol{A})$ of $\boldsymbol{A}$ are both subspaces of $\mathbb{R}^n$, defined by[2)]

$$\begin{aligned} \mathcal{R}(\boldsymbol{A}) &= \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} = \boldsymbol{A}^T\boldsymbol{y}, \ \boldsymbol{y} \in \mathbb{R}^m\} \ , \\ \mathcal{N}(\boldsymbol{A}) &= \{\mathbf{x} \in \mathbb{R}^n \mid \boldsymbol{A}\mathbf{x} = \boldsymbol{0}\} \ . \end{aligned} \tag{2.26}$$

Now, let $\mathbf{x}_{\mathcal{R}} \in \mathcal{R}(\boldsymbol{A})$ and $\mathbf{x}_{\mathcal{N}} \in \mathcal{N}(\boldsymbol{A})$. Then

$$\mathbf{x}_{\mathcal{R}}^T\mathbf{x}_{\mathcal{N}} = (\boldsymbol{A}^T\boldsymbol{y})^T\mathbf{x}_{\mathcal{N}} = \boldsymbol{y}^T\boldsymbol{A}\mathbf{x}_{\mathcal{N}} = \boldsymbol{y}^T\boldsymbol{0} = 0 \ ,$$

i.e. $\mathbf{x}_{\mathcal{R}}$ and $\mathbf{x}_{\mathcal{N}}$ are orthogonal. In words: *The null space of $\boldsymbol{A}$ is orthogonal to the row space of $\boldsymbol{A}$.*

Any $\mathbf{x} \in \mathbb{R}^n$ can be written as the direct sum of an $\mathbf{x}_{\mathcal{R}} \in \mathcal{R}(\boldsymbol{A})$ and an $\mathbf{x}_{\mathcal{N}} \in \mathcal{N}(\boldsymbol{A})$,

$$\mathbf{x} = \mathbf{x}_{\mathcal{R}} + \mathbf{x}_{\mathcal{N}} = \boldsymbol{A}^T\boldsymbol{y} + \mathbf{x}_{\mathcal{N}} \ .$$

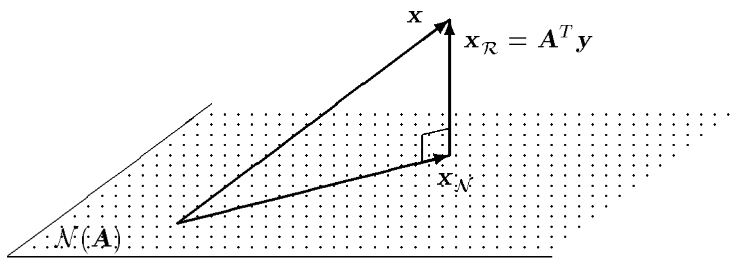


Figure 2.3. *Projection onto null space*

We can state the problem in matrix form

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\mathcal{N}} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \boldsymbol{0} \end{bmatrix} \ . \tag{2.27a}$$

2) Note, that $\mathbf{x} = \boldsymbol{A}^T\boldsymbol{y}$ is equivalent with $\mathbf{x} = y_1\boldsymbol{A}_{1,:}^T + \ldots y_m\boldsymbol{A}_{m,:}^T$, where $\boldsymbol{A}_{i,:}$ is the row vector with elements given by the ith row of $\boldsymbol{A}$.

By block elimination we see that $\boldsymbol{y}$ must satisfy

$$\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{A}\mathbf{x} \; . \tag{2.27b}$$

The assumption $\mathrm{rank}(\boldsymbol{A}) = m$ implies that the $m \times m$ matrix $\boldsymbol{A}\boldsymbol{A}^T$ is nonsingular, i.e. $(\boldsymbol{A}\boldsymbol{A}^T)^{-1}$ exists, and we find[3)]

$$\mathbf{x}_{\mathcal{N}} = \boldsymbol{P}_A\mathbf{x} \; , \quad \text{where} \quad \boldsymbol{P}_A = \boldsymbol{I} - \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{A} \tag{2.27c}$$

is the socalled *projection matrix* with respect to $\boldsymbol{A}$.

Among all vectors $\boldsymbol{z}$ in $\mathcal{N}(\boldsymbol{A})$, the vector $\mathbf{x}_{\mathcal{N}} = \boldsymbol{P}_A\mathbf{x}$ is the one that minimizes the distance $\|\mathbf{x}-\boldsymbol{z}\|_2$. This implies that the angle between $\mathbf{x}$ and $\mathbf{x}_{\mathcal{N}}$ is in the range $[0, \frac{\pi}{2}]$, so that $\mathbf{x}^T\mathbf{x}_{\mathcal{N}} \geq 0$.

In the next section we are given a gradient $\boldsymbol{g} \in \mathbb{R}^n$ and consider the problem of finding $\boldsymbol{h}$ such that

$$\boldsymbol{g}^T\boldsymbol{h} < 0 \quad \text{and} \quad \boldsymbol{A}\boldsymbol{h} = \boldsymbol{0} \; . \tag{2.28}$$

The vector in the null space of $\boldsymbol{A}$ which is closest to the steepest descent direction $-\boldsymbol{g}$ is found by projecting this vector onto $\mathcal{N}(\boldsymbol{A})$,

$$\boldsymbol{h} = -\boldsymbol{P}_A\boldsymbol{g} \; . \tag{2.29}$$

From the above observation we see that

$$\boldsymbol{g}^T\boldsymbol{h} = -\boldsymbol{g}^T\boldsymbol{P}_A\boldsymbol{g} \leq 0 \; .$$

If this is strictly negative, then (2.29) is a solution to (2.28). Otherwise, $\boldsymbol{g}$ is orthogonal to $\mathcal{N}(\boldsymbol{A})$, which means that it belongs to $\mathcal{R}(\boldsymbol{A})$ and can be expressed as $\boldsymbol{g} = \boldsymbol{A}^T\boldsymbol{y}$. Note, that this discussion can be seen as an alternative proof of Case 4 in Farkas' Lemma, p. 29.

3) Note, that $\boldsymbol{y}$ given by (2.27b) is the *least-squares solution* $\widehat{\mathbf{y}}$ to the overdetermined system of equations $\boldsymbol{A}^T\boldsymbol{y} \simeq \mathbf{x}$, and $\mathbf{x}_{\mathcal{N}}$ is the corresponding residual, $\mathbf{r}(\widehat{\mathbf{y}}) = \mathbf{x} - \boldsymbol{A}^T\widehat{\mathbf{y}}$. The matrix in (2.27a) is a socalled *augmented matrix,* and (2.27b) is the *normal equations*, see [14, Section 5.3.3] or [32, Section 6.2.1]. A more accurate solution can be found by *orthogonal transformation*, [14, Section 5.3.4], [32, Section 6.3]. Also the projection can be found via orthogonal transformation, but it is outside the scope of this introduction to go into this.

2.6. Logarithmic Barrier Function

We consider an LO problem in standard form with n variables

$$(P) \qquad \min\{\; \boldsymbol{c}^T\mathbf{x} \;:\; \boldsymbol{A}\mathbf{x} = \boldsymbol{b},\; \mathbf{x} \geq \boldsymbol{0} \;\}$$

and the associated problem

$$(P_\mu) \qquad \min\{\; \varphi_\mu(\mathbf{x}) \;:\; \boldsymbol{A}\mathbf{x} = \boldsymbol{b},\; \mathbf{x} > \boldsymbol{0} \;\} \; , \tag{2.30a}$$

where φ_μ is the *logarithmic barrier function*[4)]

$$\varphi_\mu(\mathbf{x}) = \boldsymbol{c}^T\mathbf{x} - \mu\sum_{j=1}^{n}\log x_j \; , \tag{2.30b}$$

with the *barrier parameter* $\mu > 0$. A large value of μ emphasizes the barrier term and keeps $\{x_j\}$ away from zero. A smaller value of μ emphasizes the objective value $f_P(\mathbf{x}) = \boldsymbol{c}^T\mathbf{x}$. Intuitively, a minimizer $\mathbf{x}(\mu)$ of (P_μ) should converge to an optimal solution of (P) if $\mu \searrow 0$.

The feasible domain of (P_μ), $\mathcal{P}_\mu = \{\mathbf{x} \in \mathbb{R}^n \mid \boldsymbol{A}\mathbf{x} = \boldsymbol{b},\; \mathbf{x} > \boldsymbol{0}\}$, is equal to $\mathcal{P}^+$, the *interior* of the feasible domain of (P). We assume that this set is not empty, and that we know a point $\mathbf{x} \in \mathcal{P}^+$. We seek a "new" point $\mathbf{x}+t\boldsymbol{h}$ so that

$$t > 0 \; , \quad \mathbf{x} + t\boldsymbol{h} \in \mathcal{P}^+ \quad \text{and} \quad \varphi_\mu(\mathbf{x} + t\mathbf{h}) < \varphi_\mu(\mathbf{x}) \; . \tag{2.31}$$

This means that $\boldsymbol{h}$ should be a *descent direction*, satisfying

$$\boldsymbol{h}^T\left(\nabla_x\varphi_\mu(\mathbf{x})\right) < 0 \; , \tag{2.32a}$$

where $\nabla_x\varphi_\mu$ is the gradient[5)]

$$\nabla_x\varphi_\mu(\mathbf{x}) = \boldsymbol{c} - \mu\mathbf{x}^{-1} \; . \tag{2.32b}$$

The Hessian of $\varphi_\mu(\mathbf{x})$ is

$$\Delta\varphi_\mu(\mathbf{x}) = \mu\mathbf{X}^{-2} \; , \tag{2.33}$$

4) This definition is equivalent with the definition (1.6) for (P) in canonical form. Now, $\boldsymbol{A}$ and $\mathbf{x}$ include slack variables (cf. Example 2.1), instead of the surplus vector used in (1.3).

5) $\mathbf{x}^{-1}$ is the vector with ith component x_i^{-1}. $\boldsymbol{X} = \mathrm{diag}(\mathbf{x})$, cf. NOTATION.

which is positive definite for any $\boldsymbol{x} \in \mathbb{R}^n_{++}$. This implies that if we find a stationary point of φ_μ, $\widetilde{\mathbf{x}} \in \mathcal{P}^+ \subseteq \mathbb{R}^n_{++}$, then this is the minimizer of (P_μ).

According to the discussion in the previous section, $\boldsymbol{x}$ is a stationary point if $\nabla g_\mu(\mathbf{x}) = \boldsymbol{A}^T\boldsymbol{y}$ for some $\boldsymbol{y} \in \mathbb{R}^m$. By putting $\boldsymbol{s} = \mu \boldsymbol{x}^{-1}$, which is equivalent with $\mathbf{x} \odot \boldsymbol{s} = \mu \boldsymbol{e}$, it follows that φ_μ has a minimizer if and only if there exist vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{s}$ such that[6]

$$\boldsymbol{A}\mathbf{x} = \boldsymbol{b} \;, \qquad \mathbf{x} > \boldsymbol{0} \;, \tag{2.34a}$$

$$\boldsymbol{A}^T\mathbf{y} + \boldsymbol{s} = \boldsymbol{c} \;, \qquad \boldsymbol{s} > \boldsymbol{0} \;, \tag{2.34b}$$

$$\mathbf{x} \odot \boldsymbol{s} = \mu \boldsymbol{e} \quad . \tag{2.34c}$$

The conditions $\boldsymbol{x} > \boldsymbol{0}$ and $\boldsymbol{s} > \boldsymbol{0}$ can be relaxed to $\boldsymbol{x} \geq \boldsymbol{0}$ and $\boldsymbol{s} \geq \boldsymbol{0}$, since (2.34c) ensures strict inequality. Therefore, (2.34a) is simply the feasibility constraints for (P). Similarly, (2.34b) is the feasibility constraints for the corresponding dual problem, cf. (2.19b).

Theorem 2.4. (Interior Point Condition). Let $\mu > 0$. Then the following statements are equivalent:

1° Both $\mathcal{P}$ and $\mathcal{D}$ contain a strictly positive vector.

2° There exists a unique minimizer $\mathbf{x}(\mu)$ of φ_μ on $\mathcal{P}^+$.

3° The KKT system (2.34) has a unique solution, $(\mathbf{x}(\mu), \boldsymbol{y}(\mu), \boldsymbol{s}(\mu))$.

Proof: See [39, Theorem II.4]. □

[6] (2.34) is known as the *Karush-Kuhn-Tucker conditions* for the nonlinear optimization problem (P_μ). We shall use "KKT system" to refer to this nonlinear system of equations.

2.7. The Central Path

The interior point condition Theorem 2.4 is independent of μ. This implies that if it is satisfied, then for each positive value of μ we have a unique $\mathbf{x}(\mu)$. The set of points

$$\{\, \mathbf{x}(\mu) \;:\; \mu > 0 \,\} \tag{2.35a}$$

is called the *central path* of (P). Similarly,

$$\{\, (\boldsymbol{y}(\mu), \boldsymbol{s}(\mu)) \;:\; \mu > 0 \,\} \tag{2.35b}$$

is the *central path* of (D), and (2.34c) is the *centering condition* with respect to μ.

In Example 1.3 we showed the central path for a simple problem, and saw that $\mathbf{x}(\mu)$ converged to an optimal solution as $\mu \searrow 0$. We now show that this must hold in general when (P) is feasible and bounded:

First, we remind of (2.21): For any $\boldsymbol{x} \in \mathcal{P}$ and $\boldsymbol{y} \in \mathcal{D}$ we can express the duality gap as $\Delta(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T\boldsymbol{s}$. For $(\boldsymbol{x}, \boldsymbol{y})$ on the central path this leads to

$$\begin{aligned} \Delta(\mathbf{x}(\mu), \boldsymbol{y}(\mu)) &= \mathbf{x}(\mu)^T \boldsymbol{s}(\mu) \\ &= \mathbf{e}^T(\mathbf{x}(\mu) \odot \boldsymbol{s}(\mu)) \\ &= \mathbf{e}^T(\mu \boldsymbol{e}) = n\mu \quad . \end{aligned} \tag{2.36}$$

Thus, if μ is sufficiently small, say $\mu \leq \frac{1}{n}\varepsilon$, then $f_P(\mathbf{x}(\mu)) - f_P(\mathbf{x}^*) \leq \varepsilon$, which means that we are close to the optimal value for (P).

3. Interior Point Methods

In this chapter we consider the primal problem (P) in standard form with n variables and m constraints

$$(P) \qquad \min\left\{\ \boldsymbol{c}^T\mathbf{x}\ :\ \boldsymbol{A}\mathbf{x}=\boldsymbol{b},\ \mathbf{x}\geq\boldsymbol{0}\ \right\}\ ,$$

and the corresponding dual problem

$$(D) \qquad \max\left\{\ \boldsymbol{b}^T\boldsymbol{y}\ :\ \boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{s}=\boldsymbol{c},\ \boldsymbol{s}\geq\boldsymbol{0}\ \right\}\ .$$

In Section 2.6 we introduced the logarithmic barrier function and modified (P) to (P_μ), whose minimizer is the solution to the *KKT system*

$$\begin{aligned} \boldsymbol{A}\mathbf{x} &= \boldsymbol{b}\ , \qquad \mathbf{x}>\boldsymbol{0}\ , && (3.1a)\\ \boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{s} &= \boldsymbol{c}\ , \qquad \boldsymbol{s}>\boldsymbol{0}\ , && (3.1b)\\ \mathbf{x}\odot\boldsymbol{s} &= \mu\boldsymbol{e}\ . && (3.1c) \end{aligned}$$

The solution $\mathbf{x}(\mu)$ and $(\boldsymbol{y}(\mu), \boldsymbol{s}(\mu))$ to this nonlinear system is on the *central path* of (P) and (D), respectively, and in Section 2.7 we saw that if $\mu \leq \frac{1}{n}\varepsilon$, then $\boldsymbol{c}^T(\mathbf{x}(\mu)-\mathbf{x}^*) \leq \varepsilon$, where $\mathbf{x}^*$ is an optimal solution for (P). Similarly, it can be shown that $\boldsymbol{b}^T(\boldsymbol{y}^*-\boldsymbol{y}(\mu)) \leq \varepsilon$, where $\boldsymbol{y}^*$ is an optimal solution for (D).

A point $(\mathbf{x}, \boldsymbol{y}, \boldsymbol{s})$ that satisfies the conditions (3.1a–b) is said to be *strictly feasible* and to lie in the *interior* of the feasible domain.

Roughly, Interior Point Methods (IPM) have the form sketched in Algorithm *IPM* below. There exist a number of different variants of IPM, all of which rely on Lemma 3.1. We concentrate on the socalled *Primal Dual Logarithmic Barrier Method*, which is currently considered to be the best. This method is described in Sections 3.1 – 3.5, and in Section 3.6 we give a brief review of Karmarkar's method.

Algorithm IPM
  Get an initial strictly feasible $\mathbf{x}$ and $\boldsymbol{s}$,
  and an initial value for $\boldsymbol{y}$ and μ.
  `repeat` {*outer loop*}
    `repeat` {*inner loop*}
      update $\mathbf{x}$, $\boldsymbol{y}$, $\boldsymbol{s}$
    `until` sufficiently close to $(\mathbf{x}(\mu), \boldsymbol{y}(\mu), \boldsymbol{s}(\mu))$
    reduce μ
  `until` $n\mu \leq \varepsilon$

Lemma 3.1. If the barrier parameter has the initial value μ_0 and each reduction has the form $\mu_k := (1-\theta)\mu_{k-1}$ with $0<\theta<1$, then after at most

$$\left\lceil \frac{1}{\theta}\ \log\frac{n\mu_0}{\varepsilon} \right\rceil$$

reductions we have $n\mu_k \leq \varepsilon$.

Proof: Evidently, $\mu_k = (1-\theta)^k\mu_0$, so that

$$n\mu_k \leq \varepsilon \quad\Leftrightarrow\quad n(1-\theta)^k\mu_0 \leq \varepsilon \quad\Leftrightarrow\quad k\ \log(1-\theta) \leq \log\frac{\varepsilon}{n\mu_0}\ ,$$

and using the relation $-\log(1-\theta) \geq \theta$ we get the result. □

3.1. Newton Iteration. I

In Section 2.6 we saw that the logarithmic barrier function is strictly convex on $\mathcal{P}^+$. This makes Newton's method an obvious choice for the inner loop iterations in Algorithm *IPM*.

Given a strictly feasible point $(\mathbf{x}, \boldsymbol{y}, \boldsymbol{s})$, i.e. (3.1a–b) is satisfied. We want to find $(\mathbf{x}+\boldsymbol{h}_x,\ \boldsymbol{y}+\boldsymbol{h}_y,\ \boldsymbol{s}+\boldsymbol{h}_s)$ so that also the centering condition (3.1c) is satisfied:

$$
\begin{aligned}
\boldsymbol{A}(\mathbf{x}+\boldsymbol{h}_x) &= \boldsymbol{b} \;, \\
\boldsymbol{A}^T(\boldsymbol{y}+\boldsymbol{h}_y)+\boldsymbol{s}+\boldsymbol{h}_s &= \boldsymbol{c} \;, \\
(\mathbf{x}+\boldsymbol{h}_x)\odot(\boldsymbol{s}+\boldsymbol{h}_s) &= \mu\boldsymbol{e} \;.
\end{aligned} \tag{3.2}
$$

We subtract $\boldsymbol{A}\mathbf{x}=\boldsymbol{b}$, $\boldsymbol{A}^T\boldsymbol{y}+\boldsymbol{s}=\boldsymbol{c}$ and neglect the quadratic term $\boldsymbol{h}_x\odot\boldsymbol{h}_s$ in the last equation. This defines the Newton step $\boldsymbol{h}=(\boldsymbol{h}_x,\ \boldsymbol{h}_y,\ \boldsymbol{h}_s)$:

$$
\begin{aligned}
\boldsymbol{A}\boldsymbol{h}_x &= \boldsymbol{0} \;, \\
\boldsymbol{A}^T\boldsymbol{h}_y+\boldsymbol{h}_s &= \boldsymbol{0} \;, \\
\boldsymbol{S}\boldsymbol{h}_x+\boldsymbol{X}\boldsymbol{h}_s &= \mu\boldsymbol{e}-\mathbf{x}\odot\boldsymbol{s} \;.
\end{aligned} \tag{3.3}
$$

This linear system, $2n+m$ equations with $2n+m$ unknowns, is discussed in Section 3.5. Here, we only mention that the system is nonsingular, so that $\boldsymbol{h}$ is unique.

The Newton step is said to be feasible[1] if $\mathbf{x}+\boldsymbol{h}_x>\boldsymbol{0}$ and $\boldsymbol{s}+\boldsymbol{h}_s>\boldsymbol{0}$. In order to clarify this problem and also to get a good measure of, how close $\mathbf{x}$ and $(\boldsymbol{y},\boldsymbol{s})$ are to the central paths of (P) and (D), we introduce a *scaling* of the variables,

$$
\begin{aligned}
&\mathbf{x}=\sqrt{\mu}\,\boldsymbol{D}\mathbf{u} \;, \qquad \boldsymbol{s}=\sqrt{\mu}\,\boldsymbol{D}^{-1}\mathbf{u} \;, \\
&\boldsymbol{h}_x=\sqrt{\mu}\,\boldsymbol{D}\boldsymbol{d}_x \;, \quad \boldsymbol{h}_s=\sqrt{\mu}\,\boldsymbol{D}^{-1}\boldsymbol{d}_s \;, \quad \boldsymbol{h}_y=\sqrt{\mu}\,\boldsymbol{d}_y \;,
\end{aligned} \tag{3.4a}
$$

where

$$
\mathbf{u}=\sqrt{\frac{1}{\mu}\,\mathbf{x}\odot\boldsymbol{s}} \;, \qquad \boldsymbol{D}=\operatorname{diag}(\sqrt{\mathbf{x}\odot\boldsymbol{s}^{-1}}) \;. \tag{3.4b}
$$

The system (3.2) changes to

$$
\begin{aligned}
\boldsymbol{A}\boldsymbol{D}\boldsymbol{d}_x &= \boldsymbol{0} \\
(\boldsymbol{A}\boldsymbol{D})^T\boldsymbol{d}_y+\boldsymbol{d}_s &= \boldsymbol{0} \\
\boldsymbol{d}_x+\boldsymbol{d}_s &= \mathbf{u}^{-1}-\mathbf{u} \;.
\end{aligned} \tag{3.5}
$$

If $(\mathbf{x},\boldsymbol{y},\boldsymbol{s})=(\mathbf{x}(\mu),\boldsymbol{y}(\mu),\boldsymbol{s}(\mu))$, then $\mathbf{x}\odot\boldsymbol{s}=\mu\boldsymbol{e}$, so that $\mathbf{u}=\boldsymbol{e}$ and $\mathbf{u}^{-1}-\mathbf{u}=\boldsymbol{0}$, implying $\boldsymbol{d}_x=\boldsymbol{d}_y=\boldsymbol{d}_s=\boldsymbol{0}$. Otherwise, $\boldsymbol{d}_x$ and $\boldsymbol{d}_s$ are nonzero, and the first two equations in (3.5) show that $\boldsymbol{d}_x\in\mathcal{N}(\boldsymbol{A}\boldsymbol{D})$ and $\boldsymbol{d}_s\in\mathcal{R}(\boldsymbol{A}\boldsymbol{D})$. From the discussion in Section 2.5 we see that $\boldsymbol{d}_x$ and $\boldsymbol{d}_s$ are orthogonal, and from the last equation in (3.5) it now follows that

[1] "Strictly feasible" would be more appropriate.

$$
\|\boldsymbol{d}_x\|_2^2+\|\boldsymbol{d}_s\|_2^2=\|\mathbf{u}^{-1}-\mathbf{u}\|_2^2 \;. \tag{3.6}
$$

This motivates the following measure for the proximity of $(\mathbf{x},\ \boldsymbol{y},\ \boldsymbol{s})$ to $(\mathbf{x}(\mu),\ \boldsymbol{y}(\mu),\ \boldsymbol{s}(\mu))$,[2]

$$
\delta(\mathbf{x},\boldsymbol{s};\,\mu)\equiv\frac{1}{2}\|\mathbf{u}^{-1}-\mathbf{u}\|_2=\frac{1}{2}\Big\|\sqrt{\frac{\mu}{\mathbf{x}\odot\boldsymbol{s}}}-\sqrt{\frac{\mathbf{x}\odot\boldsymbol{s}}{\mu}}\Big\|_2 \;. \tag{3.7}
$$

Without further ado we now summarize a number of properties of the Newton step.

Theorem 3.2. The Newton step satisfies

1° The vectors $\boldsymbol{h}_x$ and $\boldsymbol{h}_s$ are orthogonal, $\boldsymbol{h}_x^T\boldsymbol{h}_s=0$, and also $\boldsymbol{d}_x^T\boldsymbol{d}_s=0$.

2° The step is feasible if and only if $\mu\boldsymbol{e}+\boldsymbol{h}_x\odot\boldsymbol{h}_s>\boldsymbol{0}$.

3° $(\mathbf{x}+\boldsymbol{h}_x)^T(\boldsymbol{s}+\boldsymbol{h}_s)=n\mu$.

4° Let $\delta\equiv\delta(\mathbf{x},\boldsymbol{s};\,\mu)$. Then

$$
\|\boldsymbol{h}_x\odot\boldsymbol{h}_s\|_\infty\le\mu\delta^2 \quad\text{and}\quad \|\boldsymbol{h}_x\odot\boldsymbol{h}_s\|_2\le\sqrt{2}\mu\delta^2 \;.
$$

5° If $\delta\equiv\delta(\mathbf{x},\boldsymbol{s};\,\mu)<1$, then the step is feasible, and

$$
\delta(\mathbf{x}+\boldsymbol{h}_x,\boldsymbol{s}+\boldsymbol{h}_s;\,\mu)\le\frac{\delta^2}{\sqrt{2(1-\delta^4)}} \;.
$$

Proof: See Appendix A.3. □

By comparing (2.36) and point 3° in the theorem we see that after a feasible step the duality gap $\Delta(\mathbf{x}+\boldsymbol{h}_x,\boldsymbol{y}+\boldsymbol{h}_y)$ is equal to $\Delta(\mathbf{x}(\mu),\boldsymbol{y}(\mu))$. It is also important to note that Newton's method is quadratically convergent in the domain

$$
\Big\{(\mathbf{x},\boldsymbol{y})\in\mathcal{P}\times\mathcal{D}\;\;|\;\;\delta(\mathbf{x},\boldsymbol{s};\,\mu)\le\sqrt[4]{0.5}=0.8409\Big\} \;. \tag{3.8}
$$

Here, point 5° shows that $\delta(\mathbf{x}+\boldsymbol{h}_x,\boldsymbol{s}+\boldsymbol{h}_s;\,\mu)\le\delta^2$.

When such a good approximation has been found, it is time to reduce the barrier parameter in Algorithm *IPM*, p. 36. To see, how much μ can be reduced, we need

[2] The factor $\frac{1}{2}$ is introduced for convenience.

Lemma 3.3. Let $(\boldsymbol{x}, \boldsymbol{s})$ be a positive primal-dual pair with $\boldsymbol{x}^T\boldsymbol{s} = n\mu$. Moreover, let $\delta \equiv \delta(\mathbf{x}, \boldsymbol{s};\, \mu)$ and $\overline{\mu} = (1-\theta)\mu$ with $0 < \theta < 1$. Then
$$\delta(\mathbf{x}, \boldsymbol{s};\, \overline{\mu}) = \sqrt{(1-\theta)\delta^2 + \frac{\theta^2 n}{4(1-\theta)}}\ .$$

Proof: See [39, Lemma II.53]. □

In the following sections we shall discuss different strategies for performing the Newton iteration and for reducing the barrier parameter.

3.2. Full Newton Step

Assume that we are given starting values $\mathbf{x}_0 \in \mathcal{P}^+$, $\boldsymbol{y}_0 \in I\!R^m$ so that $\boldsymbol{s}_0 \equiv \boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y}_0 > \boldsymbol{0}$. Further, we are given a proximity parameter τ and a barrier reduction parameter θ, $0 < \tau, \theta < 1$.

The initial barrier parameter is found by $n\mu_0 = \boldsymbol{x}_0^T\boldsymbol{s}_0$, and we assume that $\delta(\boldsymbol{x}_0, \boldsymbol{s}_0; \mu_0) \le \tau$. We then proceed by the following simple version of the basic algorithm.

Algorithm IPM1

$\boldsymbol{x} := \boldsymbol{x}_0;\quad \boldsymbol{y} := \boldsymbol{y}_0;\quad \boldsymbol{s} := \boldsymbol{s}_0;\quad \mu := \mu_0;$
`repeat`
  compute $\boldsymbol{h}$ by (3.3)
  $\boldsymbol{x} := \boldsymbol{x} + \boldsymbol{h}_x;\quad \boldsymbol{y} := \boldsymbol{y} + \boldsymbol{h}_y;\quad \boldsymbol{s} := \boldsymbol{s} + \boldsymbol{h}_s;$
  $\mu := (1-\theta)\mu;$
`until` $n\mu \le \varepsilon$

Getting the starting value that satisfies $\delta(\boldsymbol{x}_0, \boldsymbol{s}_0; \mu_0) \le \tau$ may need several Newton iterations, maybe involving the use of dampeded Newton steps (see Section 3.3) to ensure that the next point is strictly feasible. Provided that we choose τ and θ sufficently small, however, the inner loop in the general algorithm (p. 36) is reduced to one iteration step:

Theorem 3.4. If $\tau \le \sqrt[4]{0.5}$ and $\theta = 1/(2\sqrt{n})$, then Algorithm *IPM1* requires at most
$$\left\lceil 2\sqrt{n}\ \log\frac{n\mu_0}{\varepsilon}\right\rceil$$
iterations to get $n\mu_k \le \varepsilon$.

Proof: The choice $\tau \le \sqrt[4]{0.5}$ guarantees that the starting guess is in the region of quadratic convergence, see (3.8). After the first Newton step with $\mu = \mu_0$ we therefore have $\delta(\boldsymbol{x}_1, \boldsymbol{s}_1; \mu_0) \le \tau^2 \le \sqrt{0.5}$, and Lemma 3.3 with $\theta = 1/(2\sqrt{n})$ shows that
$$\delta(\boldsymbol{x}_1, \boldsymbol{s}_1; \mu_1) \le \sqrt{\frac{1-\theta}{2} + \frac{1}{16(1-\theta)}} \le \sqrt{\frac{1}{2} + \frac{1}{8}} = 0.7906 < \sqrt[4]{0.5}\ .$$
Thus, we are in the region of quadratic convergence for the next Newton step, etc. The upper bound on the number of iterations follows from Lemma 3.1 with $\theta = 1/(2\sqrt{n})$. □

Example 3.1. The dominant work in Algorithm *IPM1* is the solution of the system (3.3) in each iteration. For a full matrix the cost is $O(n^3)$ flops per iteration. Since $\log n$ grows very slowly with n, the result of Theorem 3.4 is that we can find an "ε-approximation" to the solution with $O(n^{3.5})$ flops. This is referred to as *"polynomial cost"*.

If we replace ε by ε^2, then this cost doubles – because $log(1/\varepsilon^2) = 2\log(1/\varepsilon)$.

Example 3.2. In a number of examples (e.g. 1.1, 1.3, 2.3 and 2.7) we consider the problem given by $n = 4$, $m = 2$,
$$\boldsymbol{A} = \begin{bmatrix} 1/2 & 1 & -1 & 0 \\ -2/3 & 1 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \end{bmatrix}.$$
The vectors
$$\mathbf{x}_0 = \begin{bmatrix} 6 \\ 3 \\ 5 \\ 3 \end{bmatrix}, \quad \mathbf{y}_0 = \begin{bmatrix} 1.0 \\ -1.2 \end{bmatrix}, \quad \mathbf{s}_0 = \begin{bmatrix} 0.7 \\ 3.2 \\ 1.0 \\ 1.2 \end{bmatrix},$$

are easily seen to be stricly feasible, and we let $\mu_0 = (\mathbf{x}_0^T \mathbf{s}_0)/n = 5.6$. The corresponding measure of proximity is $\delta(\mathbf{x}_0, \mathbf{s}_0; \mu_0) = 0.3848$, so this is a good starting point for Algorithm *IPM1* with $\theta = 1/(2\sqrt{n}) = 0.25$.

The number K of iterations needed to get $n\mu$ below $\varepsilon = 10^{-4}$ can be estimated by Theorem 3.4: $K = 50$. We can make this a bit sharper, though: From the proof of Lemma 3.1 with $1-\theta = 0.75$ we get

$$K = \left\lceil \log \frac{\varepsilon}{n\mu_0} / \log(1-\theta) \right\rceil = \lceil 42.82 \rceil = 43 \ .$$

First, we show the behaviour of the first two components[3)] of $\mathbf{x}$, cf. Examples 1.1 and 1.3, and in Figure 3.1b we show the behaviour of $\boldsymbol{y}$, cf. Example 2.7.

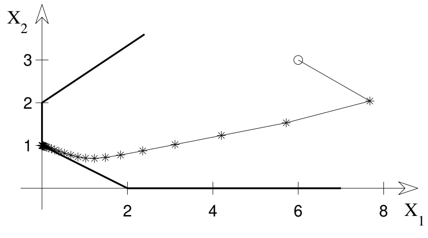


Figure 3.1a. *Algorithm IPM1. Results for* (x_1, x_2)

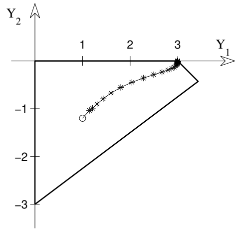


Figure 3.1b. *Algorithm IPM1. Results for* (y_1, y_2)

Finally, in Figure 3.1c we give proximity measures as functions of iteration number. We discuss these results in the next section.

3) Components $x_{3:4}$ are slack variables, and of no primary interest.

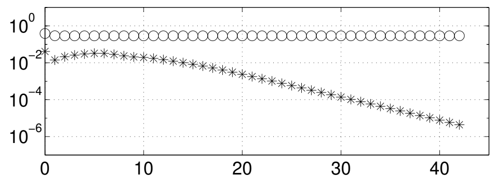


Figure 3.1c. *Algorithm IPM1. Proximity values* o : $\delta(\mathbf{x}_k, \mathbf{s}_k; \mu_k)$, $*$: $\delta(\mathbf{x}_k + \boldsymbol{h}_x^{(k)}, \mathbf{s}_k + \boldsymbol{h}_s^{(k)}; \mu_k)$

3.3. Newton Iteration. II

The behaviour shown in Figure 3.1c is typical for the simple algorithm *IPM1*: As the iterations proceed, the accuracy after the Newton step is almost "too good". This is caused by the fixed barrier reduction parameter θ, chosen to ensure that $(\boldsymbol{x}, \boldsymbol{s})$ is in the region of quadratic convergence, as indicated by $\delta(\boldsymbol{x}, \boldsymbol{s};\, \mu) \le \tau;\ \tau \le \sqrt[4]{0.5}$.

Another approach is to use a *"large update method"*, where the barrier reduction parameter θ is chosen e.g. as $\theta = 0.99$. Then the interior loop in Algorithm *IPM* p. 36 is necessary, and there is a risk that in the first steps of this iteration the Newton step is infeasible, i.e. $\boldsymbol{x} + \boldsymbol{h}_x > 0$ and $\boldsymbol{s} + \boldsymbol{h}_s > 0$ are not satisfied. This problem can be cured by choosing *damped Newton steps,* where we use the step $\alpha\boldsymbol{h}$ with $\alpha \le 1$ computed to keep the new iterate strictly positive:

$$\alpha = \min\{\Gamma(\boldsymbol{x}, \boldsymbol{h}_x, \beta), \Gamma(\boldsymbol{s}, \boldsymbol{h}_s, \beta)\} \ , \tag{3.9a}$$

with

$$\Gamma(\boldsymbol{z}, \mathbf{v}, \beta) = \min\{1,\ \beta \cdot \max\{\alpha \mid \boldsymbol{z} + \alpha\mathbf{v} \ge \boldsymbol{0}\}\} \ . \tag{3.9b}$$

Here, β is a chosen constant in the range $0 < \beta < 1$, e.g. $\beta = 0.99995$.

The large update method is summarized in algorithm *IPM2* below, where the accuracy parameter can be chosen as $\tau = \sqrt[4]{0.5}$.

Algorithm IPM2

$\mathbf{x} := \mathbf{x}_0$; $\boldsymbol{s} := \boldsymbol{s}_0$; $\mu := \mu_0$;
repeat
  $\mu := (1-\theta)\mu$;
  repeat
    compute $\boldsymbol{h}$ by (3.3)
    compute damping factor α by (3.9)
    $\mathbf{x} := \mathbf{x} + \alpha\boldsymbol{h}_x$; $\boldsymbol{s} := \boldsymbol{s} + \alpha\boldsymbol{h}_s$;
  until $\delta(\mathbf{x}, \boldsymbol{s};\, \mu) \le \tau$
until $n\mu \le \varepsilon$

The efficiency of the method depends on the choice of θ and β, and [39, Theorem II.74] shows that the required number of iterations is $O(n \log n)$. Also see [16] and the next example.

Example 3.3. We have used the large update method on the problem of Example 3.2 and with the same starting values, corresponding to $\mu_0 = 5.6$. For $\theta > 0.9999\,9554$ one outer iteration is sufficient to satisfy the stopping criterion with $\varepsilon = 10^{-4}$. Therefore we have strengthened the desired accuracy to $\varepsilon = 10^{-8}$, and give the number of iterations needed for different choices of θ and β.

θ \ β	0.95	0.995	0.99995
0.25	75	75	75
0.9	12	12	13
0.99	11	7	8
0.999	14	11	8
0.9999	12	9	7

In all cases the computed solution had an error

$$\max\{\|\mathbf{x}-\mathbf{x}^*\|_\infty,\ \|\boldsymbol{s}-\boldsymbol{s}^*\|_\infty,\ \|\boldsymbol{y}-\boldsymbol{y}^*\|_\infty\} < 10^{-8} .$$

Here, superscript * denotes exact solution, cf. Examples 1.1 and 2.7,

$$\mathbf{x}^* = \begin{bmatrix} 0\\1\\0\\1 \end{bmatrix}, \quad \boldsymbol{s}^* = \begin{bmatrix} 0.5\\0\\3\\0 \end{bmatrix}, \quad \boldsymbol{y}^* = \begin{bmatrix} 3\\0 \end{bmatrix} .$$

The choice $\theta = 0.25$ is equal to the value used in Example 3.1, and the large update method behaves like the simplest full Newton step algorithm.

The other θ values correspond to effective damping. The best combination of (θ, β) needs less than 10% of the iterations needed with algorithm *IPM1*. Also note that – even for this small problem – the algorithm seems to be surprisingly sensitive to the choice of the parameters θ and β.

In order to get the background for some of the more efficient versions of IPMs, we take a closer look at the Newton step defined by (3.3), which we write in matrix form,

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_x \\ \boldsymbol{h}_y \\ \boldsymbol{h}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \mu\boldsymbol{e} - \boldsymbol{X}\boldsymbol{s} \end{bmatrix} . \tag{3.10}$$

Note, that $\boldsymbol{X}\boldsymbol{s} = \boldsymbol{x}\odot\boldsymbol{s}$. The solution splits naturally into two parts corresponding to $-\boldsymbol{X}\boldsymbol{s}$ and $\mu\boldsymbol{e}$, respectively,

$$\boldsymbol{h}^N = \boldsymbol{h}^a + \boldsymbol{h}^c \quad \text{with} \quad \boldsymbol{B}\boldsymbol{h}^a = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ -\boldsymbol{X}\boldsymbol{s} \end{bmatrix}, \quad \boldsymbol{B}\boldsymbol{h}^c = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \mu\boldsymbol{e} \end{bmatrix}, \tag{3.11}$$

where $\boldsymbol{B}$ is the matrix in (3.10). The two contributions are called the *affine-scaling direction* and the *centering direction*, respectively. In Section 3.4 we show, how to make use of this splitting and the properties stated in the next theorem about the duality gap and the proximity measure defined by (3.7).

Theorem 3.5. Let $(\mathbf{x}, \boldsymbol{s})$ be a positive primal-dual pair with $\mathbf{x}^T\boldsymbol{s} = n\mu$, and let $\boldsymbol{h}^N \equiv \boldsymbol{h}$, $\boldsymbol{h}^a$ and $\boldsymbol{h}^c$ be defined by (3.11). Then:

For $0 < \alpha \le \alpha_0$, small enough to ensure feasibility,

1° $(\mathbf{x} + \alpha\boldsymbol{h}_x^a)^T(\boldsymbol{s} + \alpha\boldsymbol{h}_s^a) = (1-\alpha)n\mu = (1-\alpha)\mathbf{x}^T\boldsymbol{s}$,

2° $(\mathbf{x} + \alpha\boldsymbol{h}_x^c)^T(\boldsymbol{s} + \alpha\boldsymbol{h}_s^c) = (1+\alpha)n\mu = (1+\alpha)\mathbf{x}^T\boldsymbol{s}$,

3° $(\mathbf{x} + \alpha\boldsymbol{h}_x^N)^T(\boldsymbol{s} + \alpha\boldsymbol{h}_s^N) = n\mu = \mathbf{x}^T\boldsymbol{s}$.

Further, for sufficiently small values of α ,

4° $\delta(\mathbf{x} + \alpha\boldsymbol{h}_x^a, \boldsymbol{s} + \alpha\boldsymbol{h}_s^a; \mu) \simeq \sqrt{1+\alpha}\,\delta(\mathbf{x}, \boldsymbol{s}; \mu)$,

5° $\delta(\mathbf{x} + \alpha\boldsymbol{h}_x^c, \boldsymbol{s} + \alpha\boldsymbol{h}_s^c; \mu) < \sqrt{1-2\alpha}\,\delta(\mathbf{x}, \boldsymbol{s}; \mu)$,

6° $\delta(\mathbf{x} + \alpha\boldsymbol{h}_x^N, \boldsymbol{s} + \alpha\boldsymbol{h}_s^N; \mu) < \sqrt{1-\alpha}\,\delta(\mathbf{x}, \boldsymbol{s}; \mu)$.

Proof: The theorem is new and the proof is rather lengthy. It can be found in Appendix A.4. □

Point 3° in the theorem is identical with 3° in Theorem 3.2, and 1°–2° show that the duality gap decreases in the affine scaling direction, but increases in the centering direction. As regards proximity, however, points 4°–5° show that the opposite is true. Thus, the name "centering".

Example 3.4. Consider the problem from Example 3.2 with the feasible starting point

$$\mathbf{x}_0 = \begin{bmatrix} 6 \\ 3 \\ 5 \\ 3 \end{bmatrix}, \quad \boldsymbol{y}_0 = \begin{bmatrix} 1.0 \\ -1.2 \end{bmatrix}, \quad \boldsymbol{s}_0 = \begin{bmatrix} 0.7 \\ 3.2 \\ 1.0 \\ 1.2 \end{bmatrix},$$

and $\mu = (\mathbf{x}_0^T \boldsymbol{s}_0)/n = 5.6$. In figure 3.2a we show the first two coordinates of $\mathbf{x}_0$, $\mathbf{x}(5.6)$ on the central path, and the vectors $\boldsymbol{h}^a$, $\boldsymbol{h}^c$ and the full Newton step $\boldsymbol{h}^N = \boldsymbol{h}^a + \boldsymbol{h}^c$.

Note, that $\boldsymbol{h}^a$ is not feasible: $\mathbf{x}+\boldsymbol{h}^a$ is outside the feasible domain. Taking a damped step in this direction, however, would bring us close to the solution, whose firste two coordinates are (0,1), cf. Figure 3.1a.

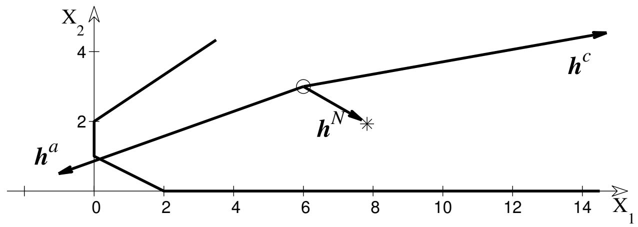


Figure 3.2a. *Affine-scaling, centering and Newton steps*

Figure 3.2b below shows the proximity measure δ as function of α and the direction. It confirms Theorem 3.5, although it is difficult to see that for $\alpha \ll 1$ we get better results by using the direction $\boldsymbol{h}^c$ than by using $\boldsymbol{h}^N$.

The Newton approximation to the KKT equations is derived under the assumption that $\|\boldsymbol{h}\|$ is so small that we can ignore the second order term $\boldsymbol{h}_x \odot \boldsymbol{h}_s$. This is **not** satisfied by the steps shown in Figure 3.2a, and the difference in their lengths further confuses the picture in Figure 3.2b. Also, we only show two out of the 10 coordinates in $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$.

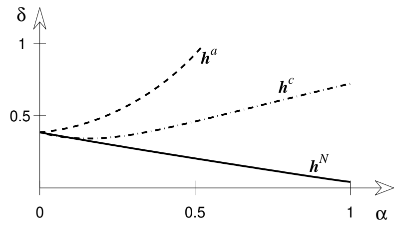


Figure 3.2b. *Proximity measure*

Example 3.5. In Section 3.4 we describe a socalled *predictor-corrector* algorithm, which can be outlined as follows:

Given $\boldsymbol{x}$,$\boldsymbol{y}$,$\boldsymbol{s}$. An appropriate value of the barrier parameter μ is found by going in the direction of $\boldsymbol{h}^a$. Next, the Newton step is computed by $\boldsymbol{h}^N = \boldsymbol{h}^a + \boldsymbol{h}^c$, where $\boldsymbol{h}^c$ is computed with this μ–value.

The computation can be done by accumulating (3.11) into a matrix equation

$$\boldsymbol{BH} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{Xs} & \boldsymbol{e} \end{bmatrix}.$$

Then, $\boldsymbol{h}^a$ comes out as the first column of $\boldsymbol{H}$, and $\boldsymbol{h}^c = \mu \boldsymbol{H}_{:,2}$. We discuss this aspect further in Section 3.5.

For the problem from the previous example we see from Figure 3.2a that the step $\frac{3}{4}\boldsymbol{h}^a$ is feasible, and from 1° in Theorem 3.5 we see that the corresponding μ-value is $\mu = (1 - \frac{3}{4})\mu_0 = 1.4$. This is shown below together with the first two coordinates of $\mathbf{x}(1.4)$. The full Newton step is seen to land close to this point. For comparison we also indicate the $\boldsymbol{h}^c$ from Example 3.4.

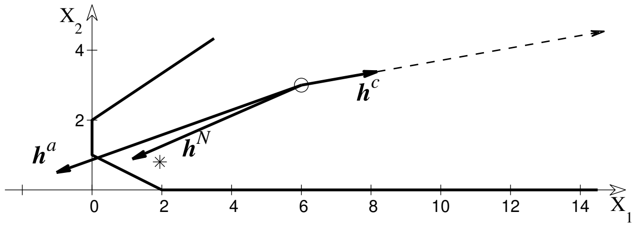


Figure 3.3. *Predictor–corrector method*

3.4. Infeasible Predictor–Corrector Method

This is currently considered to be the most efficient IPM, and among its advantages is that it does not need a feasible starting vector. This demand is not essential for applying Newton's method to (3.1).

Let $\boldsymbol{x},\boldsymbol{y},\boldsymbol{s}$ be given with $\boldsymbol{x},\boldsymbol{s} > \boldsymbol{0}$, and let[4)]

$$\boldsymbol{f}_P = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}, \quad \boldsymbol{f}_D = \boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{s}, \quad \boldsymbol{f}_C = \mu\boldsymbol{e} - \boldsymbol{X}\boldsymbol{s} \;. \tag{3.12a}$$

Then, the Newton step $\boldsymbol{h} = (\boldsymbol{h}_x, \boldsymbol{h}_y \boldsymbol{h}_s)$ is the solution to the system

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_x \\ \boldsymbol{h}_y \\ \boldsymbol{h}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_P \\ \boldsymbol{f}_D \\ \boldsymbol{f}_C \end{bmatrix} \;. \tag{3.12b}$$

We discuss this system i Section 3.5. Notice that if $(\boldsymbol{x}, \boldsymbol{y}\; \boldsymbol{s})$ are feasible, then $\boldsymbol{f}_P = \boldsymbol{0}$, $\boldsymbol{f}_D = \boldsymbol{0}$, and (3.12b) is identical with (3.10).

Exercise 3.6. Show that if $\boldsymbol{x}+\boldsymbol{h}_x \geq \boldsymbol{0}$ and $\boldsymbol{s}+\boldsymbol{h}_s \geq \boldsymbol{0}$, then the point $(\boldsymbol{x}+\boldsymbol{h}_x,\ \boldsymbol{y}+\boldsymbol{h}_y,\ \boldsymbol{s}+\boldsymbol{h}_s)$ is feasible.

We are interested in large reductions of the barrier parameter μ, and shall use the predictor-corrector method outlined in the previous section. The affine-scaling direction is found by solving

$$\boldsymbol{B}\boldsymbol{h}^a = \begin{bmatrix} \boldsymbol{f}_P \\ \boldsymbol{f}_D \\ -\boldsymbol{X}\boldsymbol{s} \end{bmatrix} \;, \tag{3.13}$$

where $\boldsymbol{B}$ is the matrix in (3.12b). A *predictor* is computed by taking a damped step in this direction[5)]

$$\widehat{\boldsymbol{x}} = \boldsymbol{x} + \alpha_P \boldsymbol{h}_x^a \;, \quad \widehat{\boldsymbol{y}} = \boldsymbol{y} + \alpha_D \boldsymbol{h}_y^a \;, \quad \widehat{\boldsymbol{s}} = \boldsymbol{s} + \alpha_D \boldsymbol{h}_s^a \;, \tag{3.14a}$$

with the damping factors determined by

4) Index P, D and C for Primal, Dual and Central path, respectively.

5) Experience shows that it pays to use different damping factors for the primal and dual variables. The function Γ is defined in (3.9b), p. 42.

$$\alpha_P = \Gamma(\boldsymbol{x}, \boldsymbol{h}_x^a, 1) \;, \quad \alpha_D = \Gamma(\boldsymbol{s}, \boldsymbol{h}_s^a, 1) \;. \tag{3.14b}$$

Next, the new value for the barrier parameter μ is computed. The following heuristic is recommended by [2] and [6]:

$$\mu := \eta^3\mu, \quad \text{where} \quad \eta = \min\left\{1, \frac{\widehat{\boldsymbol{x}}^T\widehat{\boldsymbol{s}}}{n\mu}\right\} \;. \tag{3.15}$$

Now, we compute the centering step by solving[6)]

$$\boldsymbol{B}\boldsymbol{h}^c = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \mu\boldsymbol{e} \end{bmatrix} \;, \tag{3.16}$$

and we get the next iterate (the *"corrector"*) by using a damped step in the Newton direction given by $\boldsymbol{h} = \boldsymbol{h}^a + \boldsymbol{h}^c$.

As regards stopping criteria, the central path is only defined for strictly feasible points, therefore we cannot use the proximity measure and barrier value as in algorithms *IPM1* and *IPM2*. Instead we check feasibility,

$$\|\boldsymbol{f}_P\|_\infty \leq \varepsilon_1(1 + \|\boldsymbol{b}\|_\infty) \;, \qquad \|\boldsymbol{f}_D\|_\infty \leq \varepsilon_1(1 + \|\boldsymbol{c}\|_\infty) \;, \tag{3.17a}$$

and the duality gap

$$|\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{y}| \leq \varepsilon_2(1 + \max\{|\boldsymbol{c}^T\boldsymbol{x}|,\ |\boldsymbol{b}^T\boldsymbol{y}|\}) \;. \tag{3.17b}$$

Further, we need criteria to stop the iteration if the problem is infeasible or unbounded. In that case the absolute value of the primal or dual objective function $|\boldsymbol{c}^T\boldsymbol{x}|$ or $|\boldsymbol{b}^T\boldsymbol{y}|$ will grow during iteration, and a simple check is given by[7)]

$$\max\left\{|\boldsymbol{c}^T\boldsymbol{x}|,\ |\boldsymbol{b}^T\boldsymbol{y}|\right\} \geq \frac{1}{\varepsilon_3}\left(1 + \max\left\{|\boldsymbol{c}^T\boldsymbol{x}_0|,\ |\boldsymbol{b}^T\boldsymbol{y}_0|\right\}\right) \;. \tag{3.17c}$$

Finally, as in all iterative processes, we must have a "safety valve"

$$k \geq k_{\max} \;. \tag{3.17d}$$

6) The solution of (3.13) and (3.16) can be combined as described in Example 3.5.

7) This check is not robust for large problems. A better approach is to embed the problem in a self-dual problem as in the proof of Theorem 2.2, see e.g. [2, Section 6], [39, Section 20.5].

Now, we can summarize the algorithm as follows,

Algorithm IPM3

$\mathbf{x} := \mathbf{x}_0;\quad \boldsymbol{y} := \boldsymbol{y}_0;\quad \boldsymbol{s} := \boldsymbol{s}_0;\quad \mu := \mathbf{x}^T\boldsymbol{s}/n;\quad k := 0$
repeat
  if (3.17c) **or** (3.17d) is satisfied **then** STOP
  if (3.17a) **and** (3.17b) are satisfied **then** STOP
  $k := k + 1$
  Compute $\boldsymbol{h}^a$ by (3.13)
  $\widehat{\mathbf{x}} := \mathbf{x} + \Gamma(\mathbf{x}, \boldsymbol{h}_x^a, 1)\boldsymbol{h}_x^a;\quad \widehat{\boldsymbol{s}} := \boldsymbol{s} + \Gamma(\boldsymbol{s}, \boldsymbol{h}_s^a, 1)\boldsymbol{h}_s^a;$
  Compute μ by (3.15)
  Compute $\boldsymbol{h}^c$ by (3.16)
  $\boldsymbol{h} := \boldsymbol{h}^a + \boldsymbol{h}^c;$
  $\alpha_P := \Gamma(\mathbf{x}, \boldsymbol{h}_x, \beta);\quad \alpha_D = \Gamma(\boldsymbol{s}, \boldsymbol{h}_s, \beta)$
  $\mathbf{x} := \mathbf{x} + \alpha_P \boldsymbol{h}_x;\quad \boldsymbol{y} := \boldsymbol{y} + \alpha_D \boldsymbol{h}_y;\quad \boldsymbol{s} := \boldsymbol{s} + \alpha_D \boldsymbol{h}_s$
end

In the examples we use the iteration parameters $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = 10^{-8}$, $k_{\max} = \max\{20,\ n\}$ and $\beta = 0.99995$.

Example 3.7. We have used Algorithm *IPM3* on the problem from Example 3.2. With the infeasible starting vector

$$\mathbf{x}_0 = \boldsymbol{s}_0 = \boldsymbol{e}\ ,\quad \boldsymbol{y}_0 = \boldsymbol{0} \tag{3.18}$$

we get the results shown below.

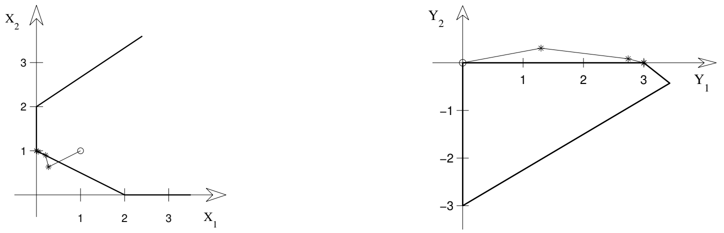


Figure 3.4a. *Algorithm IPM3. Results for* (x_1, x_2) *and* (y_1, y_2)

The stopping criteria (3.17) are satisfied after 5 iterations, and in Figure 3.4b we show the results for μ, the infinity norm of $\boldsymbol{f}_P$ and $\boldsymbol{f}_D$, and the duality gap. Note, that (3.17a) is satisfied before (3.17b). This is generally true with this algorithm.

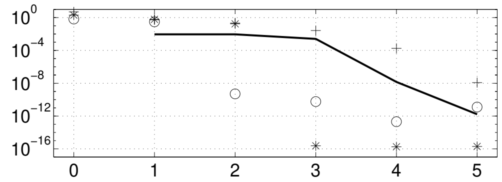


Figure 3.4b. *Algorithm IPM3. Barrier value and error measures*
`--` : μ, `o` : $\|\boldsymbol{f}_P\|_\infty$, `*` : $\|\boldsymbol{f}_D\|_\infty$, `+` : $|\boldsymbol{c}^T\mathbf{x} - \boldsymbol{b}^T\boldsymbol{y}|$

Naturally, the algorithm also works with a feasible starting point. If we use the starting point from Examples 3.2 and 3.4, the algorithm also needs 5 iterations, and the points stay feasible.

Example 3.8. For the infeasible problem (P') from Examples 1.2 and 2.13 the algorithm was stopped by (3.17c) after 5 iterations. The performance is illustrated below

k	$\|\boldsymbol{f}_P\|_\infty$	$\|\boldsymbol{f}_D\|_\infty$	$\|\boldsymbol{c}^T\mathbf{x}\|$	$\|\boldsymbol{b}^T\boldsymbol{y}\|$
0	6.67e-01	4.00e+00	-5.00e+00	0
1	1.60e-14	2.44e+00	-1.18e+01	-6.03e-01
2	3.15e-10	2.35e+00	-7.77e+01	-7.48e-01
3	3.43e-08	2.34e+00	-3.30e+04	-7.33e-01
4	2.51e-02	2.34e+00	-1.39e+08	-7.32e-01
5	5.53e+03	2.34e+00	-4.30e+15	-7.32e-01

After the first iteration we have primal feasibility, whereas we cannot get dual feasibility, cf. Example 2.13. The primal objective $\boldsymbol{c}^T\mathbf{x}$ seems to "converge" to $-\infty$, cf. Example 1.2, and the algorithm is stopped by (3.17c) after 5 iterations.

Example 3.9. To get an idea of the efficiency of algorithm *IPM3* we have generated larger problems as described in Example 2.15. In all cases we used the default parameter values $\varepsilon_1 = \varepsilon_2 = 10^{-8}$, $\beta = 0.99995$, and the starting point given by (3.18). The largest error in the solution, as measured by the duality gap was $7.1\cdot 10^{-9}$.

First, we consider "well scaled" problems: In the generator we let $n = 2m$ and $\gamma_x = \gamma_y = \gamma_s = 1$. The number of iterations is shown in Figure 3.5a, and the behaviour is illustrated in Figure 3.5b for the case needing most iterations. The problem is dual feasible after 3 iterations, and as in Example 3.7 we see that (3.17b) is the critical stopping criterion.

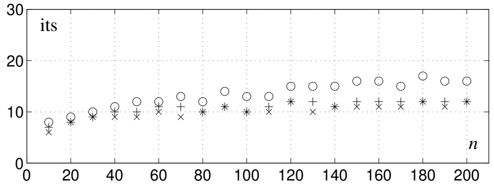


Figure 3.5a. *Algorithm IPM3. Well scaled problems.*
$+: \#\mathcal{B} = 0.8m, \quad \circ: \#\mathcal{B} = m, \quad \times: \#\mathcal{B} = 1.2m$

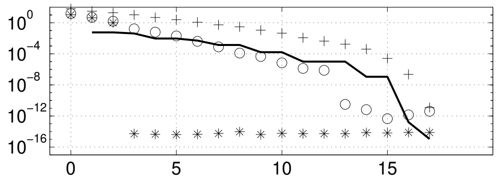


Figure 3.5b. *Algorithm IPM3. Well scaled problem.* $n = 180$.
$-- : \mu, \quad \circ: \|\boldsymbol{f}_P\|_\infty, \quad *: \|\boldsymbol{f}_D\|_\infty, \quad +: |\boldsymbol{c}^T\mathbf{x} - \boldsymbol{b}^T\boldsymbol{y}|$

Next, to get "descaled" problems we have used $\gamma_x = 10^3$, $\gamma_y = 1$, $\gamma_s = 10^{-3}$. Results are given in Figure 3.5c.

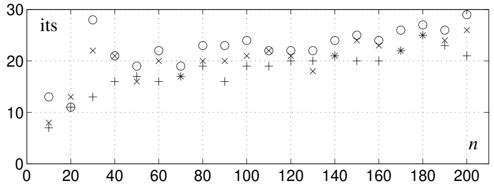


Figure 3.5c *Algorithm IPM3. Descaled problems.*
$+: \#\mathcal{B} = 0.8m, \quad \circ: \#\mathcal{B} = m, \quad \times: \#\mathcal{B} = 1.2m$

In both cases degeneracy (i.e. $\#\mathcal{B} \neq m$, cf. the discussion p. 26) seems to speed up convergence rather than impairing it.

From Figure 3.5a we see that the required number of iterations grows very slowly with n. This is in accordance with the literature, where $O(n^{1/4})$ or $O(\log n)$ is frequently mentioned.

A similar growth is seen in Figure 3.5c, and it "costs" about 10 more iterations for a problem of the same size. The reason is that for this set of problems the choice (3.18) is a very poor approximation to a point on the central path. It illustrates that IPMs are sensitive the the choice of starting point and motivates special algorithms for this. We shall not discuss this, but refer to [6].

Below we show the behaviour for the outlier at $n = 30$,

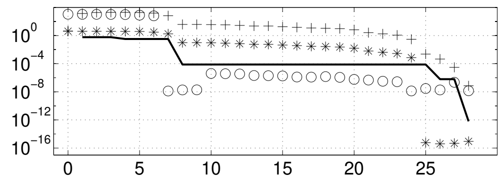


Figure 3.5d. *Algorithm IPM3. Descaled problem.* $n = 30$.
$-- : \mu, \quad \circ: \|\boldsymbol{f}_P\|_\infty, \quad *: \|\boldsymbol{f}_D\|_\infty, \quad +: |\boldsymbol{c}^T\mathbf{x} - \boldsymbol{b}^T\boldsymbol{y}|$

With the large value of γ_x relative to γ_y and γ_s, the first iterations seem to focus on getting primal feasibility. This is obtained after 7 iterations, and is followed by a large reduction in the barrier parameter. This stays fixed the next 18 iterations, while the error in dual feasibility and the duality gap decrease slowly. Once dual feasibility is obtained, we get fast final convergence.

It should be mentioned, that also for a feasible starting point we may find a similar behavior.

3.5. Solving the Newton Equations

In all the versions of the Primal Dual Method discussed in Sections 3.1–4 the Newton step is the solution to a linear system of equations of the form

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^T & \boldsymbol{I} \\ \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_x \\ \boldsymbol{h}_y \\ \boldsymbol{h}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_P \\ \boldsymbol{f}_D \\ \boldsymbol{f}_C \end{bmatrix}, \tag{3.19}$$

where $\boldsymbol{X} = \mathrm{diag}(\boldsymbol{x})$, $\boldsymbol{S} = \mathrm{diag}(\boldsymbol{s})$ and the right-hand side depend on the current iterate (and the expressions for the right-hand side depend on the version of the method). We can simplify the system: From the last block of equations we see that

$$\boldsymbol{h}_s = \boldsymbol{X}^{-1}\boldsymbol{f}_C - \boldsymbol{D}^{-2}\boldsymbol{h}_x \ , \tag{3.20a}$$

where we have used (3.4b),

$$\boldsymbol{D}^2 = \boldsymbol{S}^{-1}\boldsymbol{X} \ . \tag{3.20b}$$

With this expression for $\boldsymbol{h}_s$ we can compress (3.19) to

$$\begin{bmatrix} -\boldsymbol{D}^{-2} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_x \\ \boldsymbol{h}_y \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_D - \boldsymbol{X}^{-1}\boldsymbol{f}_C \\ \boldsymbol{f}_P \end{bmatrix} \equiv \begin{bmatrix} \boldsymbol{f}_x \\ \boldsymbol{f}_y \end{bmatrix} \ . \tag{3.21}$$

Exercise 3.10. Let $\boldsymbol{f}_P = \boldsymbol{0}$ (i.e. $\boldsymbol{x}$ is feasible). Show that $\boldsymbol{h}_x = -\boldsymbol{D}\boldsymbol{P}_{AD}(\boldsymbol{D}\boldsymbol{f}_x)$, i.e. $-\boldsymbol{D}$ times the projection of the vector $\boldsymbol{D}\boldsymbol{f}_x$ onto the null space of the scaled matrix $\boldsymbol{A}\boldsymbol{D}$.

The system can be compressed even further: Multiplying the first block of equations by $\boldsymbol{A}\boldsymbol{D}^2$ and adding to the second block we eliminate $\boldsymbol{h}_x$ and see that $\boldsymbol{h}_y$ is determined by

$$(\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T)\boldsymbol{h}_y = \boldsymbol{f}_y + \boldsymbol{A}\boldsymbol{D}^2\boldsymbol{f}_x \equiv \widetilde{\boldsymbol{f}} \ . \tag{3.22}$$

Exercise 3.11. The matrix $\boldsymbol{H} = \boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$ is obviously symmetric. Show, that if $\boldsymbol{A}$ has full rank, then $\boldsymbol{H}$ is *positive definite*, and therefore nonsingular.

This shows that (3.22) has a unique solution, and after computing this, the other components of $\boldsymbol{h}$ can be found from (3.19):

$$\boldsymbol{h}_s = \boldsymbol{f}_D - \boldsymbol{A}^T\boldsymbol{h}_y \ , \quad \boldsymbol{h}_x = \boldsymbol{S}^{-1}(\boldsymbol{f}_C - \boldsymbol{X}\boldsymbol{h}_s) \ . \tag{3.23}$$

As the iterates converge to the solution we see that

$$d_i^2 = \frac{x_i}{s_i} \to \begin{cases} \infty & \text{if } i \in \mathcal{B} \\ 0 & \text{if } i \in \mathcal{C} \end{cases} \ ,$$

where $\mathcal{B}$, $\mathcal{C}$ are defined in (2.22). This means the the diagonal of $\boldsymbol{D}$ will have both very small and very large elements, and it has the effect that the matrix in (3.22) gets increasingly ill conditioned. Fortunately, a close analysis [41] shows, that this ill conditioning has no damaging effects, provided that the implementation is done carefully. This is the subject of the remainder of this section.

In practical applications the number of constraints and variables (m and n) may be very large, and the solution of (3.21) in each step of the iteration will dominate the execution time. Fortunately, the matrix $\boldsymbol{A}$ is normally *sparse* – i.e. most of the $a_{ij} = 0$ – and this can be exploited in the solution, see [9] or [34].

The sparsity can be fully exploited by using an *iterative method* to solve the linear system, see e.g. [3]. To get an efficient method, a good *preconditioner* is needed, and this is used successfully in special applications, [30]. In spite of much effort, however, there still is no algorithm for getting a good preconditioner in the general case, and a direct method is recommended, [2], [15].

We can use a *direct method* on the *augmented system* (3.21) or on the *normal equations* formulation (3.22). Both choices have their pros and cons. The names come from the literature on least squares problems, cf. the relationship shown in Exercise 3.10 and in Section 2.5.

First, consider the normal equations approach. As discussed in Exercise 3.11, the matrix $\boldsymbol{H} = \boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$ is symmetric and positive definite. This implies that we can use *Cholesky factorization*[8)]

$$\boldsymbol{H} = \boldsymbol{C}\boldsymbol{C}^T \ , \tag{3.24}$$

where $\boldsymbol{C}$ is a lower triangular matrix. When $\boldsymbol{A}$ is sparse, then hopefully also $\boldsymbol{H}$ is sparse. Normally, however, we get *fill-ins,* i.e. $\boldsymbol{C}$ has more nonzeros than the lower triangle of $\boldsymbol{H}$.

---

8) See e.g. [32, Section 3.2.4] or [14, Section 4.2].

The number of fill-ins can be reduced by *pivoting.* If we take the pivots from the diagonal, then the symmetry is preserved, and we can write[9)]

$$\boldsymbol{P}\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T\boldsymbol{P}^T = \boldsymbol{C}\boldsymbol{C}^T \ . \tag{3.25a}$$

Now, the solution to (3.22) is found by[10)]

$$\boldsymbol{w} = \boldsymbol{C}^{-1}(\boldsymbol{P}\widetilde{\boldsymbol{f}}) \ , \qquad \boldsymbol{h}_y = \boldsymbol{P}(\boldsymbol{C}^{-T}\boldsymbol{w}) \ . \tag{3.25b}$$

In words, $\boldsymbol{w}$ is computed by forward substitution on a permuted version of $\widetilde{\boldsymbol{f}}$, and then $\boldsymbol{h}_y$ is found by permuting the vector obtained by back substitution on $\boldsymbol{w}$.

There exist a number of algorithms for finding $\boldsymbol{P}$, so that the number of fill-ins in $\boldsymbol{C}$ is kept low. See [34, Chapter 4], [9, Chapter 10] or [15].

Example 3.12. Consider the matrix $\boldsymbol{A}$ whose *sparsity pattern* is shown in Figure 3.6a. We have $m = 200$, $n = 300$ (and the trailing 100 columns could respresent slack variables). The number of nonzeros is denoted nz.

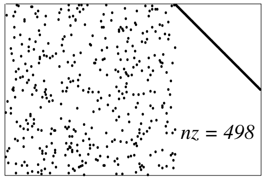
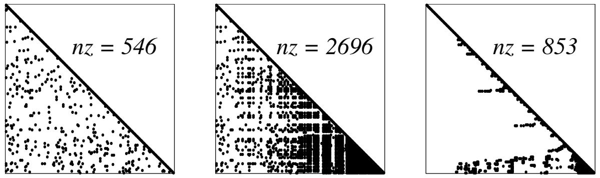


Figure 3.6a. *Constraint matrix* $\boldsymbol{A}$

Below we give the patterns of the lower triangle of $\boldsymbol{H}$ and the Cholesky factors given by (3.24) and by (3.25a) with $\boldsymbol{P}$ determined by the MATLAB implementation of the socalled *minimum degree algorithm.*

9) $\boldsymbol{P}$ is a *permutation matrix* representing the row and column interchanges. If $\boldsymbol{P} \neq \boldsymbol{I}$, then the $\boldsymbol{C}$ in (3.25) will be different from the $\boldsymbol{C}$ in (3.24).

10) As always in numerical linear algebra, an expression like "$\boldsymbol{C}^{-1}\boldsymbol{z}$" is short for "solve the linear system $\boldsymbol{C}\mathbf{x} = \boldsymbol{z}$". We (almost) never invert a matrix. The inverse of a sparse matrix is normally dense.



Lower triangle of $\boldsymbol{H}$ $\quad$ $\boldsymbol{C}_I$ *from (3.24)* $\quad$ $\boldsymbol{C}_P$ *from (3.25)*

Figure 3.6b. *Lower triangle of* $\boldsymbol{H}$ *and Cholesky factors*

The relative density in $\boldsymbol{A}$ is $498/(200 \cdot 300) = 0.0083$, and in the lower triangle of $\boldsymbol{H}$ it is $(2 \cdot 546)/(200 \cdot 201) = 0.0272$, i.e. more than three times larger. The relative number if fill-ins in the Cholesky factor is reduced from 3.94 to 0.56 when we use (3.25) instead of (3.24).

An advantage of the normal equations approach is that the choice of a good pivoting is independent of the actual values in $\boldsymbol{D}$. Therefore, this socalled ANALYSE is performed once, only. A serious disadvantage is that the inherent ill conditioning is enhanced by this approach. To see this, consider the following reformulation of the matrix,

$$\boldsymbol{H} = \boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T = \sum_{i=1}^{n} d_i^2 \boldsymbol{A}_{:,i}\boldsymbol{A}_{:,i}^T \ , \tag{3.26}$$

where $\boldsymbol{A}_{:,i}$ is the ith column of $\boldsymbol{A}$. Assume, that $\boldsymbol{A}_{:,j}$ and $\boldsymbol{A}_{:,k}$ have their nonzeros in the same rows, and that the nonzero values satisfy[11)] $|a_{ij}| \sim 1$. Further, assume that $d_k^2 < \varepsilon_M d_j^2$. Then

$$\mathsf{fl}\left(d_j^2 \boldsymbol{A}_{:,j}\boldsymbol{A}_{:,j}^T + d_k^2 \boldsymbol{A}_{:,k}\boldsymbol{A}_{:,k}^T\right) = d_j^2 \boldsymbol{A}_{:,j}\boldsymbol{A}_{:,j}^T \ ,$$

i.e. we loose the information from the kth column.

Example 3.13. From (3.26) it also follows, that if one of the columns is (almost) *dense,* then $\boldsymbol{H}$ will be (almost) dense.

This problem can be solved as follows: Assume that the columns of $\boldsymbol{A}$ are ordered so that the (almost) dense columns come last, and split the matrix

11) See NOTATION for the mening of "$\sim$". $\mathsf{fl}(\cdot)$ denotes "computed value of $(\cdot)$".

$$AD = [\, \widetilde{A}_S \quad \widetilde{A}_F \,] \,,$$

where $\widetilde{A}_F \in \mathbb{R}^{m\times q}$ is (almost) dense and $\widetilde{A}_S \in \mathbb{R}^{m\times(n-q)}$ is sparse. Then

$$H = (AD)(AD)^T = \widetilde{A}_S\widetilde{A}_S^T + \widetilde{A}_F\widetilde{A}_F^T \,.$$

Further, assume that the sparse part has full rank, i.e. in the Cholesky factorization

$$\widetilde{A}_S\widetilde{A}_S^T = C_S C_S^T \,,$$

the lower triangular C_S is nonsingular. Now, let V and z be the solutions to the sparse systems

$$C_S V = \widetilde{A}_F \,, \qquad C_S z = \widetilde{f} \,. \tag{3.27a}$$

Then the solution to (3.22) can be computed by the *Sherman–Morrison–Woodbury* formula

$$h_y = C_S^{-T}\left(z - V(I + V^T V)^{-1} V^T z\right) \,. \tag{3.27b}$$

The validity of this can be shown by insertion. The computation involves the solution of a $q\times q$ system with the symmetric, positive definite matrix $I + V^T V$, which is (almost) dense, but normally q is small. For further details see [2, Section 4].

Also the augmented system (3.21) is symmetric, but it is not positive definite, and numerical stability must be taken into account in the choice of pivots. It is possible to preserve symmetry, stability and sparsity by a factorization of the form

$$P \begin{bmatrix} -D^{-2} & A^T \\ A & 0 \end{bmatrix} P^T = LGL^T \,, \tag{3.28a}$$

where L is unit lower triangular and G has 1×1 and 2×2 symmetric blocks along the diagonal and zeros elsewhere, see [14, Section 4.4]. Having computed the factorization, the solution to (3.21) is found by

$$\mathbf{w} = L^{-1}\left(P\begin{bmatrix} f_x \\ f_y \end{bmatrix}\right), \quad \mathbf{v} = G^{-1}\mathbf{w}, \quad \begin{bmatrix} h_x \\ h_y \end{bmatrix} = P\left(L^{-T}\mathbf{v}\right) \,. \tag{3.28b}$$

This approach has better accuracy properties, and (almost) dense columns in A do not give excessive fill. However, the pivoting depends on the current values in D, so the ANALYSE must be repeated in each iteration step. Also, the demands on storage and computing time are normally more than twice the demands from the normal equations approach.

Example 3.14. In a predictor-corrector method we have to solve (3.19) with two different right-hand sides. This is performed with two "SOLVES", i.e. two applications of (3.25b) or (3.28b). Typically, each SOLVE costs less than 10% of a "FACTORIZE", and this is the background for more advanced predictor-corrector schemes, where the corrrector is recomputed several times, taking into account the term $h_x \odot h_s$, which was dropped in the derivation of the Newton equations (3.3); see e.g. [2, Section 6.62].

Example 3.15. The simplified system (3.21) was derived from the primal-dual method, (3.19). However, basically this is the system to be solved with any IPM, as we illustrate below and in the next example. Therefore, if two different IPMs use K_1 and K_2 iterations, then K_1/K_2 is a good estimate of the ratio between their execution times, since the solution of (3.21) dominates.

Consider a primal-dual method applied to an LO problem with both lower and upper constraints on the variables. As we saw in Example 2.1, we can write it in the form

$$(P') \qquad \min\left\{\, c^T\mathbf{x} \;:\; A\mathbf{x} = b,\; 0 \le \mathbf{x} \le u \,\right\}$$

or

$$(P) \qquad \min\left\{\, c^T\mathbf{x} \;:\; A\mathbf{x} = b,\; \mathbf{x} + \mathbf{v} = u,\; \mathbf{x}, \mathbf{v} \ge 0 \,\right\} \,.$$

Proceeding as in Section 2.6 we minimize

$$\varphi_\mu(\mathbf{x}, \mathbf{v}) = c^T\mathbf{x} - \mu\sum_{j=1}^{n} \log x_j v_j \,,$$

and get the *KKT* system

$$\begin{aligned} A\mathbf{x} &= b \,, & \mathbf{x} &> 0 \,, \\ \mathbf{x} + \mathbf{v} &= u \,, & \mathbf{x} &> 0 \,, \\ A^T y + \mathbf{s} - \mathbf{z} &= c \,, & \mathbf{s}, \mathbf{z} &> 0 \,, \quad y \text{ free} \,, \\ \mathbf{x} \odot \mathbf{s} &= \mu e \,. \end{aligned}$$

Given $\mathbf{x}, \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{z} > \boldsymbol{0}$ and $\boldsymbol{y}$. The Newton step is defined by

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}^T & \boldsymbol{0} & \boldsymbol{I} & -\boldsymbol{I} \\ \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{0} & \mathbf{X} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{Z} & \boldsymbol{0} & \boldsymbol{V} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_x \\ \boldsymbol{h}_y \\ \boldsymbol{h}_v \\ \boldsymbol{h}_s \\ \boldsymbol{h}_z \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} - \boldsymbol{A}\mathbf{x} \\ \boldsymbol{u} - \mathbf{x} - \mathbf{v} \\ \boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{s} + \boldsymbol{z} \\ \mu\boldsymbol{e} - \mathbf{x}\odot\boldsymbol{s} \\ \mu\boldsymbol{e} - \mathbf{v}\odot\boldsymbol{z} \end{bmatrix} \equiv \begin{bmatrix} \boldsymbol{f}_P \\ \boldsymbol{f}_U \\ \boldsymbol{f}_D \\ \boldsymbol{f}_{Cx} \\ \boldsymbol{f}_{Cv} \end{bmatrix} .$$

This can be simplified by the substitutions

$$\begin{aligned} \boldsymbol{h}_v &= \boldsymbol{f}_U - \boldsymbol{h}_x , \\ \boldsymbol{h}_s &= \boldsymbol{X}^{-1}(\boldsymbol{f}_{Cx} - \boldsymbol{S}\boldsymbol{h}_x) , \\ \boldsymbol{h}_z &= \boldsymbol{V}^{-1}(\boldsymbol{f}_{Cv} - \boldsymbol{Z}\boldsymbol{f}_U + \boldsymbol{Z}\boldsymbol{h}_x) , \end{aligned}$$

and the system reduces to the form (3.21) with

$$\begin{aligned} \boldsymbol{D}^{-2} &= \boldsymbol{X}^{-1}\boldsymbol{S} + \boldsymbol{V}^{-1}\boldsymbol{Z} , \\ \boldsymbol{f}_x &= \boldsymbol{f}_D - \boldsymbol{X}^{-1}\boldsymbol{f}_{Cx} + \boldsymbol{V}^{-1}(\boldsymbol{f}_{Cv} - \boldsymbol{Z}\boldsymbol{f}_U) , \\ \boldsymbol{f}_y &= \boldsymbol{f}_P . \end{aligned}$$

Example 3.16. We briefly outline two other IPMs. They differ from the primal-dual method in the definition of the Newton step $\boldsymbol{h}$, but use the framework of Algorithm *IPM* p. 36 as regards monitoring the barrier parameter μ, and the interior loop may involve damped Newton steps.

Primal Logarithmic Barrier Method. Let

$$(P) \qquad \min\left\{ \boldsymbol{c}^T\mathbf{x} \ : \ \boldsymbol{A}\mathbf{x} = \boldsymbol{b},\ \mathbf{x} \geq \boldsymbol{0} \right\} .$$

We use the same barrier function as in Section 2.6,

$$\begin{aligned} &\varphi_\mu(\mathbf{x}) = \boldsymbol{c}^T\mathbf{x} - \mu\textstyle\sum_{j=1}^n \log x_j , \\ &\nabla_x\varphi_\mu(\mathbf{x}) = \boldsymbol{c} - \mu\mathbf{x}^{-1} , \\ &\nabla_x\varphi_\mu(\mathbf{x}+\boldsymbol{h}) \simeq \boldsymbol{c} - \mu\mathbf{x}^{-1} + \mu\boldsymbol{X}^{-2}\boldsymbol{h} . \end{aligned}$$

Given a feasible $\mathbf{x}$, we seek $\boldsymbol{h}$ in the null space of $\boldsymbol{A}$, and so that $\nabla_x\varphi_\mu(\mathbf{x}+\boldsymbol{h})$ satisfies point 4 in Farkas' Lemma p. 29,

$$\boldsymbol{A}\boldsymbol{h} = \boldsymbol{0} , \quad \boldsymbol{c} - \mu\mathbf{x}^{-1} + \mu\boldsymbol{X}^{-2}\boldsymbol{h} = \boldsymbol{A}^T\boldsymbol{y} ,$$

or the following system, which is of the form (3.21),

$$\begin{bmatrix} -\mu\boldsymbol{X}^{-2} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{h} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c} - \mu\mathbf{x}^{-1} \\ \boldsymbol{0} \end{bmatrix} .$$

Dual Logarithmic Barrier Method. We consider the dual in the form

$$(D) \qquad \min\left\{ -\boldsymbol{c}^T\boldsymbol{y} \ : \ \boldsymbol{A}^T\boldsymbol{y} \leq \boldsymbol{c},\ \boldsymbol{y} \text{ free} \right\} ,$$

and use the barrier function

$$\begin{aligned} &\varphi_\mu(\boldsymbol{y}) = -\boldsymbol{b}^T\boldsymbol{y} - \mu\textstyle\sum_{j=1}^n \log s_j , \quad \text{with } \boldsymbol{s} = \boldsymbol{s}(\boldsymbol{y}) = \boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y} , \\ &\nabla_y\varphi_\mu(\boldsymbol{y}) = -\boldsymbol{b} + \mu\boldsymbol{A}\boldsymbol{s}^{-1} , \\ &\nabla_y\varphi_\mu(\boldsymbol{y}+\boldsymbol{h}_y) \simeq -\boldsymbol{b} + \mu\boldsymbol{A}\boldsymbol{s}^{-1} - \mu\boldsymbol{A}\boldsymbol{S}^{-2}\boldsymbol{h}_s , \quad \text{with } \boldsymbol{h}_s = -\boldsymbol{A}^T\boldsymbol{h}_y . \end{aligned}$$

The Newton step is therefore defined by

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}^T \\ \mu\boldsymbol{A}\boldsymbol{S}^{-2} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_s \\ \boldsymbol{h}_y \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{b} + \mu\boldsymbol{A}\boldsymbol{s}^{-1} \end{bmatrix} ,$$

or

$$\begin{bmatrix} -\frac{1}{\mu}\boldsymbol{S}^2 & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} -\mu\boldsymbol{S}^{-2}\boldsymbol{h}_s \\ \boldsymbol{h}_y \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{b} + \mu\boldsymbol{A}\boldsymbol{s}^{-1} \end{bmatrix} ,$$

which again is of the form (3.21).

3.6. Karmarkars Algorithm

Narendra Karmarkar published his method in 1984, and this started an explosive research in the field of IPMs. His method cannot compete with today's efficient versions, as presented in the previous sections, but for the sake of completeness we give a short presentation of it.

Consider

$$(P) \qquad \min\left\{ \boldsymbol{c}^T\mathbf{x} \ : \ \boldsymbol{A}\mathbf{x} = \boldsymbol{b},\ \mathbf{x} \geq \boldsymbol{0} \right\} ,$$

and let $\boldsymbol{z}$ be the current approximation to an optimal solution $\mathbf{x}^*$. The point $\boldsymbol{z}$ is assumed to be interior, and centered at this point we define two ellipsoids in $\mathbb{R}^n$, E' and E'', with their surfaces respectively inside and outside the feasible domain.

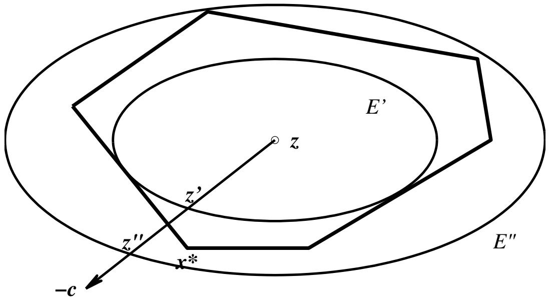


Figure 3.7. *Interior and exterior ellipsoid*

The points $\mathbf{z}'$ and $\mathbf{z}''$ are the intersections betweeen E' and E'' and the vector from $\mathbf{z}$ in the direction of the negative gradient. The ratio γ defined by

$$\mathbf{z}'' - \mathbf{z} = \gamma(\mathbf{z}' - \mathbf{z})$$

obviously satisfies $\gamma > 1$. It is also obvious that

$$\mathbf{c}^T\mathbf{z} \geq \mathbf{c}^T\mathbf{z}' \geq \mathbf{c}^T\mathbf{x}^* \geq \mathbf{c}^T\mathbf{z}'' ,$$

and from these relations we see that

$$\mathbf{c}^T(\mathbf{x}^* - \mathbf{z}') \geq \mathbf{c}^T(\mathbf{z}'' - \mathbf{z}') = (\gamma-1)\mathbf{c}^T(\mathbf{z}' - \mathbf{z}) ,$$

or

$$\mathbf{c}^T(\mathbf{z}' - \mathbf{x}^*) \leq (\gamma-1)\mathbf{c}^T(\mathbf{z} - \mathbf{z}') = (\gamma-1)\mathbf{c}^T\left(\mathbf{z} - \mathbf{x}^* - (\mathbf{z}' - \mathbf{x}^*)\right) ,$$

so that

$$\mathbf{c}^T(\mathbf{z}' - \mathbf{x}^*) \leq \left(1 - \frac{1}{\gamma}\right)\mathbf{c}^T(\mathbf{z} - \mathbf{x}^*) . \tag{3.29}$$

Since $1-\frac{1}{\gamma} < 1$, the point $\mathbf{z}'$ is a better approximation to $\mathbf{x}^*$, and we can repeat the process with ellipsoids centered at $\mathbf{z}'$; now, however the ratio γ is much larger, so we get almost no improvement.

Karmarkar's method is based on

1° A special formulation of the problem,

$$(P_K) \qquad \min\left\{ \hat{\mathbf{c}}^T\hat{\mathbf{x}} \;:\; \hat{\mathbf{A}}\hat{\mathbf{x}} = \mathbf{0},\ \mathbf{e}^T\hat{\mathbf{x}} = 1,\ \hat{\mathbf{x}} \geq \mathbf{0} \right\} , \tag{3.30}$$

constructed so that the vector $\hat{n}^{-1}\mathbf{e}$ is feasible, and the optimal value is $\hat{\mathbf{c}}^T\hat{\mathbf{x}}^* = 0$. In Example 3.17 we show, how a given problem can be written in this "Karmarkar standard form". For notational convenience we omit the "^" in the following.

2° The use of a transformation, whereby the current iterate is mapped into the centre of the transformed feasible domain.

In Figure 3.8 we show (for $n=3$) the feasible region, which is the part of the hyperplane $\mathbf{e}^T\mathbf{x} = 1$ in the positive orthant. The centre is $n^{-1}\mathbf{e}$, and

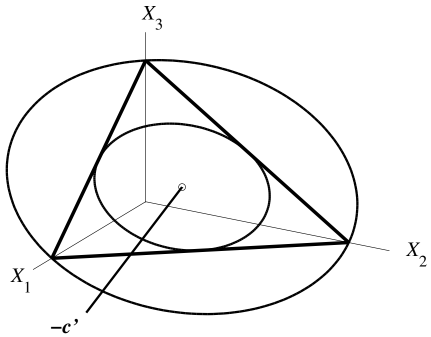


Figure 3.8. *Interior and exterior ellipsoid*

the largest interior and smallest exterior ellipsoids are balls with radius

$$r' = \sqrt{\frac{1}{n(n-1)}} \quad \text{and} \quad r'' = \sqrt{\frac{n-1}{n}} ,$$

respectively. Therefore, $\gamma = r''/r' = n-1$, and after k iterations we have[12)]

$$\mathbf{c}^T(\mathbf{x}_k - \mathbf{x}^*) \leq \left(1 - \frac{1}{n-1}\right)^k \mathbf{c}^T(\mathbf{x}_0 - \mathbf{x}^*) .$$

[12)] This is a simplified analysis. It assumes that in every iteration the point corresponding to $\mathbf{z}'$ in Figure 3.7 is in the strict interior of the feasible domain. Even worse, we do not use $-\mathbf{c}$ but a projection of it (marked $-\mathbf{c}'$ in the figure). However, a stringent analysis involving the *"Karmarkar potential"* $\varphi_K(\mathbf{x}) = n\log\mathbf{c}^T\mathbf{x} - \sum\log x_j$, yields a result similar to our simplified derivation, see [22] or [39, Theorem IV.5], and in practice considerably faster convergence is normally achieved.

Similar to Lemma 3.1 we see that after at most

$$\left\lceil (n-1)\log\frac{1}{\varepsilon}\right\rceil$$

iteration we have $\mathbf{c}^T(\mathbf{x}_k - \mathbf{x}^*) \le \varepsilon \cdot \mathbf{c}^T(\mathbf{x}_0 - \mathbf{x}^*)$.

Karmarkar uses the transformation

$$\widetilde{\mathbf{x}} = \frac{\mathbf{Z}^{-1}\mathbf{x}}{\mathbf{e}^T\mathbf{Z}^{-1}\mathbf{x}} \quad \Leftrightarrow \quad \mathbf{x} = \frac{\mathbf{Z}\widetilde{\mathbf{x}}}{\mathbf{e}^T\mathbf{Z}\widetilde{\mathbf{x}}} , \tag{3.31}$$

where $\boldsymbol{Z} = \operatorname{diag}(\boldsymbol{z})$, with $\boldsymbol{z}$ denoting the current iterate. This is mapped into $\widetilde{\boldsymbol{z}} = n^{-1}\mathbf{e}$, and problem (P_K) is transformed into

$$(P'_K) \qquad \min\left\{ \frac{\mathbf{c}^T\boldsymbol{Z}\widetilde{\mathbf{x}}}{\mathbf{e}^T\boldsymbol{Z}\widetilde{\mathbf{x}}} \;:\; \boldsymbol{A}\boldsymbol{Z}\widetilde{\mathbf{x}} = \boldsymbol{0},\; \mathbf{e}^T\widetilde{\mathbf{x}} = 1,\; \widetilde{\mathbf{x}} \ge \boldsymbol{0} \right\} .$$

The objective of this problem is nonlinear, but we know that the optimal value is zero, and this happens only if the nominator is zero. Therefore, we linearize (P'_K) to

$$(\widetilde{P}_K) \qquad \min\left\{ (\boldsymbol{Z}\boldsymbol{c})^T\widetilde{\mathbf{x}} \;:\; \boldsymbol{A}\boldsymbol{Z}\widetilde{\mathbf{x}} = \boldsymbol{0},\; \mathbf{e}^T\widetilde{\mathbf{x}} = 1,\; \widetilde{\mathbf{x}} \ge \boldsymbol{0} \right\} . \tag{3.32}$$

The next transformed iterate is found as $\widetilde{\mathbf{x}} = \widetilde{\boldsymbol{z}} + \alpha\widetilde{\boldsymbol{h}}$, which should be feasible, i.e. $\boldsymbol{A}\boldsymbol{Z}\widetilde{\boldsymbol{h}} = \boldsymbol{0}$ and $\mathbf{e}^T\widetilde{\boldsymbol{h}} = 0$. We get $\widetilde{\boldsymbol{h}}$ by projecting the negative gradient $-\boldsymbol{Z}\boldsymbol{c}$ onto the null space of the matrix

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{A}\boldsymbol{Z} \\ \mathbf{e}^T \end{bmatrix} .$$

According to (2.27a) we can find $\widetilde{\boldsymbol{h}}$ by solving[13)]

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{h}} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{Z}\boldsymbol{c} \\ \boldsymbol{0} \end{bmatrix} . \tag{3.33}$$

The value of α in $\widetilde{\mathbf{x}} = \widetilde{\boldsymbol{z}} + \alpha\widetilde{\boldsymbol{h}}$can be found as described in Section 3.3 for damped Newton steps, and to get back to $\boldsymbol{x}$ we use the back transformation in (3.31).

13) $\widetilde{\boldsymbol{h}} = -\boldsymbol{c}'$ in Figure 3.8. Note, that also in Karmarkar's method the dominating part of the computational work is the solution of a system of the form (3.21).

Example 3.17. Consider an LO problem in canonical form

$$(P) \qquad \min\{\, \mathbf{c}^T\mathbf{x} \;:\; \boldsymbol{A}\mathbf{x} \ge \boldsymbol{b},\; \mathbf{x} \ge \boldsymbol{0} \,\} ,$$
$$(D) \qquad \max\{\, \boldsymbol{b}^T\boldsymbol{y} \;:\; \boldsymbol{A}^T\boldsymbol{y} \le \boldsymbol{c},\; \boldsymbol{y} \ge \boldsymbol{0} \,\} .$$

Introducing slack variables we can write the problem in standard form,

$$(P') \qquad \min\{\, \mathbf{c}^T\mathbf{x} \;:\; \boldsymbol{A}\mathbf{x} - \boldsymbol{z} = \boldsymbol{b},\; \mathbf{x}, \boldsymbol{z} \ge \boldsymbol{0} \,\} ,$$
$$(D') \qquad \max\{\, \boldsymbol{b}^T\boldsymbol{y} \;:\; \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c},\; \boldsymbol{y}, \boldsymbol{s} \ge \boldsymbol{0} \,\} .$$

Now, let

$$\hat{n} = 2(n+m+1) , \quad \mathbf{x}_0 = \boldsymbol{s}_0 = \mathbf{e}_{(n)} , \quad \boldsymbol{y}_0 = \boldsymbol{z}_0 = \mathbf{e}_{(m)} ,$$

where $\mathbf{e}_{(n)}$ ($\mathbf{e}_{(m)}$) is the vector with n (m) ones, and let

$$\boldsymbol{q}_0 = \hat{n}\boldsymbol{b} - \boldsymbol{A}\mathbf{x}_0 + \boldsymbol{z}_0, \quad \mathbf{r}_0 = \hat{n}\boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y}_0 - \boldsymbol{s}_0, \quad \Delta_0 = \boldsymbol{b}^T\boldsymbol{y}_0 - \boldsymbol{c}^T\mathbf{x}_0 .$$

Then

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{A} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & -\hat{n}\boldsymbol{b} & \boldsymbol{q}_0 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}^T & \boldsymbol{I} & -\hat{n}\boldsymbol{c} & \mathbf{r}_0 \\ \boldsymbol{c}^T & \boldsymbol{0} & -\boldsymbol{b}^T & \boldsymbol{0} & 0 & \Delta_0 \end{bmatrix} , \quad \widehat{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \boldsymbol{z} \\ \boldsymbol{y} \\ \boldsymbol{s} \\ \nu \\ \lambda \end{bmatrix}$$

can be used in the Karmarkar standard formulation: It is seen that $\mathbf{x} = \frac{1}{\hat{n}}\boldsymbol{e}$ is feasible: $\widehat{\boldsymbol{A}}\mathbf{x} = \boldsymbol{0}$, $\mathbf{e}^T\mathbf{x} = 1$.

The problem

$$(P_K) \qquad \min\left\{ \widehat{\boldsymbol{c}}^T\widehat{\mathbf{x}} \;:\; \widehat{\boldsymbol{A}}\widehat{\mathbf{x}} = \boldsymbol{0},\; \mathbf{e}^T\widehat{\mathbf{x}} = 1,\; \widehat{\mathbf{x}} \ge \boldsymbol{0} \right\}$$

with $\widehat{\boldsymbol{c}} \in \mathbb{R}^{\hat{n}}$ having $\hat{c}_{\hat{n}} = 1$ as the only nonzero element, has the objective λ, which is nonnegative. Thus, (P_K) is both feasible and bounded, and according to Theorem 2.2 is has a solution $\widehat{\mathbf{x}}^+$ with $\lambda^+ = 0$.

Now, as in the proof of Theorem 2.2, we see that If $\nu^+ = 0$, then (P) is either infeasible or unbounded. Otherwise, also the vector $\beta\,\widehat{\mathbf{x}}^+$ with $\beta = 1/(\hat{n}\nu^+)$ satisfies the homogeneous constraints $\widehat{\boldsymbol{A}}(\beta\,\widehat{\mathbf{x}}^+) = \boldsymbol{0}$. This means, that the $\mathbf{x}$, $\boldsymbol{z}$, $\boldsymbol{y}$ and $\boldsymbol{s}$ parts of $\beta\,\widehat{\mathbf{x}}^+$ are feasible for the original problem, and their duality gap is zero. Therefore, we have found an optimal solution.

4. The Simplex Method

Consider an LO problem in canonical form

$$(P_c) \qquad \min\{\, \mathbf{c}^T\mathbf{x} \;:\; \mathbf{A}\mathbf{x} \geq \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \,\} \quad ,$$

with n variables and m inequality constraints. Each of the equations $\mathbf{A}_{i,:}\mathbf{x} = b_i$ and $x_j = 0$ defines a hyperplane in $\mathbb{R}^n$ (a line in $\mathbb{R}^2$), which splits the space into two parts, the feasible part $\mathbf{A}_{i,:}\mathbf{x} \geq b_i$ ($x_j \geq 0$) and the infeasible part $\mathbf{A}_{i,:}\mathbf{x} < b_i$ ($x_j < 0$). The hyperplanes define a polytope (also called a *simplex*) with *vertices* where n (or more) of these planes meet.

In Examples 1.1 and 1.2 we saw, that at least one member of the optimal set is at a vertex. In general there are up to $\vartheta(m,n) = \frac{(n+m)!}{m!n!}$ vertices, and even for small problems, this is a prohibitively[1] large number. The Simplex method often finds the solution much faster.

The fundamental idea was given in Figure 1.3 and is further illustrated in Figure 4.1: move from one vertex to an adjacent vertex as long as the objective $\mathbf{c}^T\mathbf{x}$ decreases.

We shall present the Simplex algorithm as applied to an LO problem in standard form[2]

$$(P) \qquad \min\{\, \mathbf{c}^T\mathbf{x} \;:\; \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \,\} \quad ,$$

with n variables and $m < n$ constraints. The discussion after Theorem 2.2 can be seen as another introduction to the Simplex idea: The index set $\{1,\ldots,n\}$ can be split into two disjoint sets $\mathcal{B}$ (with m elements) and $\mathcal{C}$

[1] $\vartheta(5,5) = 252$, $\vartheta(20,20) =$ `1.38e+11`, $\vartheta(80,80) =$ `9.20e+46`, ...

[2] As shown in Section 2.1, this can be obtained from (P_c) by extending $\mathbf{x}$ with m slack variables, implying that $n := n+m$. Conversely, for a problem given in standard form there is an equivalent canonical form, see Example 2.4. We use this relationship implicitly in the discussion when we illustrate some concepts by a two-dimensional figure.

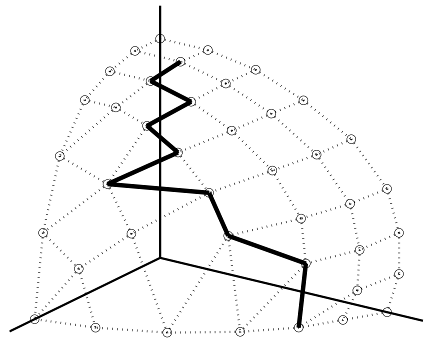


Figure 4.1. *Vertices and Simplex path*

(with $n{-}m$ elements), and we split $\boldsymbol{A}$, $\mathbf{c}$ and $\mathbf{x}$ accordingly, so that the constraints take the form

$$\boldsymbol{B}\mathbf{x}_{\mathcal{B}} = \boldsymbol{b} - \boldsymbol{C}\mathbf{x}_{\mathcal{C}} \quad , \tag{4.1}$$

where $\boldsymbol{B} = \boldsymbol{A}_{\mathcal{B}}$ and $\boldsymbol{C} = \boldsymbol{A}_{\mathcal{C}}$. We say that $\mathbf{x}_{\mathcal{B}}$ and $\mathbf{x}_{\mathcal{C}}$ are *basic* and *nonbasic* variables, respectively.

If $\boldsymbol{B}$ is nonsingular and $\boldsymbol{B}^{-1}\boldsymbol{b} \geq \boldsymbol{0}$, then the $\mathbf{x}$ given by $\mathbf{x}_{\mathcal{B}} = \boldsymbol{B}^{-1}\boldsymbol{b}$ and $\mathbf{x}_{\mathcal{C}} = \boldsymbol{0}$ is a vertex of the polytope, and an adjacent vertex is obtained by swapping an element in $\mathcal{B}$ with an element in $\mathcal{C}$. This is elaborated in Sections 4.1 – 4.4, and Sections 4.5 – 4.6 give implementation aspects and some variants of the method.

4.1. The Algorithm

Assume, that we are given a vertex $\mathbf{x}_k$, corresponding to a splitting $\mathcal{B} = \mathcal{B}_k$, $\mathcal{C} = \mathcal{C}_k$ so that

$$\operatorname{rank}(\boldsymbol{B}) = m \quad , \tag{4.2a}$$

$$\mathbf{x}_{\mathcal{C}}^{(k)} = \boldsymbol{0} \quad , \tag{4.2b}$$

$$\mathbf{x}_{\mathcal{B}}^{(k)} = \boldsymbol{B}^{-1}\boldsymbol{b} \geq \boldsymbol{0} \quad . \tag{4.2c}$$

To see, whether we can improve the objective, we look at the complete solution to the equality constraints, as given by (4.1),

$$\mathbf{x}_{\mathcal{B}} = \boldsymbol{B}^{-1}(\boldsymbol{b} - \boldsymbol{C}\mathbf{x}_{\mathcal{C}}) = \mathbf{x}_{\mathcal{B}}^{(k)} - \boldsymbol{B}^{-1}\boldsymbol{C}\mathbf{x}_{\mathcal{C}} \ , \tag{4.3a}$$

and the corresponding objective is[3)]

$$\begin{aligned} \mathbf{c}^T\mathbf{x} &= \mathbf{c}_{\mathcal{B}}^T\mathbf{x}_{\mathcal{B}} + \mathbf{c}_{\mathcal{C}}^T\mathbf{x}_{\mathcal{C}} \\ &= \mathbf{c}_{\mathcal{B}}^T\mathbf{x}_{\mathcal{B}}^{(k)} + \mathbf{d}^T\mathbf{x}_{\mathcal{C}} \\ &= \mathbf{c}^T\mathbf{x}_k + \mathbf{d}^T\mathbf{x}_{\mathcal{C}} \ , \end{aligned} \tag{4.3b}$$

where

$$\boldsymbol{d} = \boldsymbol{c}_{\mathcal{C}} - \boldsymbol{C}^T\boldsymbol{g} \quad \text{with} \quad \boldsymbol{g} = \boldsymbol{B}^{-T}\boldsymbol{c}_{\mathcal{B}} \ . \tag{4.3c}$$

The constraints demand that $\mathbf{x}_{\mathcal{C}} \geq \boldsymbol{0}$. Therefore, if $\boldsymbol{d} \geq \boldsymbol{0}$, then we cannot decrease the objective, and $\mathbf{x}_k$ is optimal.

Otherwise, let d_s be a negative component of $\boldsymbol{d}$, and change $(\mathbf{x}_{\mathcal{C}})_s$ to $\nu > 0$, while the other components of $\mathbf{x}_{\mathcal{C}}$ are kept at zero. Then the objective decreases, and the basic variables are changed to

$$\mathbf{x}_{\mathcal{B}} = \mathbf{x}_{\mathcal{B}}^{(k)} - \nu\boldsymbol{h} \ , \quad \text{where} \quad \boldsymbol{h} = \boldsymbol{B}^{-1}(\boldsymbol{C})_{:,s} \ . \tag{4.4}$$

Let

$$\sigma_j = \begin{cases} (\mathbf{x}_{\mathcal{B}}^{(k)})_j / h_j & \text{if } h_j > 0 \\ +\infty & \text{if } h_j \leq 0 \end{cases} \ , \tag{4.5a}$$

$$\alpha = \min\{\sigma_1, \ \ldots, \ \sigma_m\} \ . \tag{4.5b}$$

If $\alpha = +\infty$, then we can make ν arbitrarily large, and thus decrease the objective to $-\infty$: the problem is unbounded. Otherwise, the vector

$$\boldsymbol{z} = \mathbf{x}_{\mathcal{B}}^{(k)} - \alpha\boldsymbol{h} \tag{4.6}$$

is nonnegative and at least one of its elements – the qth – is zero[4)]. Now, the qth element in $\mathcal{B}$ is swapped with the sth element in $\mathcal{C}$, giving us $\mathcal{B}_{k+1}$,

[3)] $\boldsymbol{d} = \boldsymbol{d}_k$ is the socalled *reduced-cost vector.* For the sake of convenience we omit the index. Similarly, $\boldsymbol{h} = \boldsymbol{h}_k$ in (4.4).

[4)] The element x_q is called a *blocking variable.*

$\mathcal{C}_{k+1}$, and the next vertex has $\mathbf{x}_{\mathcal{B}}^{(k+1)} = \boldsymbol{z}$ except for $(\mathbf{x}_{\mathcal{B}}^{(k+1)})_q = \alpha$. The objective is decreased by $\alpha(-d_s)$.

The algorithm is summarized below. It assumes that $\mathcal{B}$ and $\mathcal{C} = \{1, \ldots, n\} \backslash \mathcal{B}$ are initialized so that the conditions in (4.2) are satisfied, and we use $\mathcal{B}(i)$ $(\mathcal{C}(i))$ to indicate the ith element in $\mathcal{B}$ $(\mathcal{C})$.

Algorithm SIMPLEX
  $\mathbf{x}_{\mathcal{B}} := \boldsymbol{B}^{-1}\boldsymbol{b}$
  `repeat`
    Compute $\boldsymbol{d}$ by (4.3c)
    `if` $\boldsymbol{d} \geq \boldsymbol{0}$ `then` STOP $\{\mathbf{x}$ is optimal$\}$
    Choose s so that $d_s < 0$
    $\boldsymbol{h} := \boldsymbol{B}^{-1}\boldsymbol{A}_{:,\mathcal{C}(s)}$
    `if` $\boldsymbol{h} \leq \boldsymbol{0}$ `then` STOP $\{(P)$ is unbounded$\}$
    Compute α by (4.5)
    Choose q so that $\sigma_q = \alpha$
    Swap $\mathcal{B}(q)$ and $\mathcal{C}(s)$
    $\mathbf{x}_{\mathcal{B}} := \mathbf{x}_{\mathcal{B}} - \alpha\boldsymbol{h}$; $(\mathbf{x}_{\mathcal{B}})_q := \alpha$
  `end`

Example 4.1. Consider the problem from Example 1.1. In standard form it is given by $n = 4$, $m = 2$,

$$\boldsymbol{A} = \begin{bmatrix} 1/2 & 1 & -1 & 0 \\ -2/3 & 1 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \end{bmatrix} .$$

The feasible region is given below, cf. Figure 1.1.

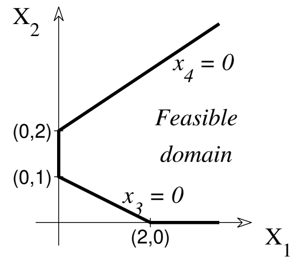


Figure 4.2

The choice $\mathcal{B}_0 = \{2,3\}$, $\mathcal{C}_0 = \{1,4\}$ satisfies the conditions of (4.2): $\boldsymbol{B}$ is nonsingular and

$$\mathbf{x}_{\mathcal{B}}^{(0)} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \geq \boldsymbol{0} \ .$$

This corresponds to $\mathbf{x}_0 = \begin{bmatrix} 0 & 2 & 1 & 0 \end{bmatrix}^T$. The last two components are slacks, so $\mathbf{x}_0$ corresponds to the point $(0,\ 2)$ in Figure 4.2.

Now, we start the iteration:

$$\boldsymbol{d} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 1/2 & -2/3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} - \begin{bmatrix} -2 \\ 3 \end{bmatrix} = \begin{bmatrix} 4 \\ -3 \end{bmatrix} .$$

$s = 2$, and

$$\boldsymbol{h} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \sigma_1 = \frac{2}{1} = 2\,, \ \sigma_2 = \frac{1}{1} = 1\,, \quad \alpha = 1\ .$$

We see that $q = 2$, $\mathcal{B}_1 = \{2,4\}$, $\mathcal{C}_1 = \{1,3\}$ and

$$\boldsymbol{z} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} - 1 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \mathbf{x}_{\mathcal{B}}^{(1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} .$$

Now for the next iteration:

$$\boldsymbol{d} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 1/2 & -2/3 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 3 \end{bmatrix} .$$

Both components of $\boldsymbol{d}$ are positive, so that $\mathbf{x}_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^T$ is optimal. This corresponds to the point $(0,\ 1)$ in in Figure 4.2.

Example 4.2. For the problem of the previous example we change the objective vector to $\boldsymbol{c} = \begin{bmatrix} 1 & 2 & 0 & 0 \end{bmatrix}^T$. Then the last expression for $\boldsymbol{d}$ is changed to

$$\boldsymbol{d} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1/2 & -2/3 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix} .$$

We still have $\boldsymbol{d} \geq \boldsymbol{0}$ so that $\mathbf{x}_1$ is optimal. However, the fact that $d_1 = 0$ shows that if we move in the direction given by $s = 1$,

$$\boldsymbol{h} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1/2 \\ -2/3 \end{bmatrix} = \begin{bmatrix} 1/2 \\ -7/6 \end{bmatrix} ,$$

then the objective value is unchanged. This is in agreement with Example 1.2: It corresponds to $\mathbf{x} = \begin{bmatrix} \nu & 1-\frac{\nu}{2} & 0 & 1+\frac{7\nu}{6} \end{bmatrix}^T$, which is feasible for $0 \leq \nu \leq 2$, and the first two elements give the line segment between the points $(0,\ 1)$ and $(2,\ 0)$ in Figure 4.2.

Problem (U) from Example 1.2 is obtained by changing $\boldsymbol{c}$ to $-\boldsymbol{c}$ so that the first $\boldsymbol{d}$ in Example 4.1 is changed to $\boldsymbol{d} = \begin{bmatrix} -4 & 3 \end{bmatrix}^T$. Now, $s = 1$, and

$$\boldsymbol{h} = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 1/2 \\ -2/3 \end{bmatrix} = \begin{bmatrix} -2/3 \\ -7/6 \end{bmatrix} .$$

Thus, $\boldsymbol{h} < \boldsymbol{0}$, indicating (as it should) that the problem is unbounded.

Assume, that $\mathbf{x}_{\mathcal{B}}^{(k)} > \boldsymbol{0}$ in every iteration step. Then $\alpha > 0$ in each step, and the objective decreases. This implies that we cannot return to a vertex already tried, and since the number of vertices is finite, the Simplex algorithm has finite termination.

Exercise 4.3. The matrix $\boldsymbol{B} = \boldsymbol{A}_{\mathcal{B}_k}$ has full rank. Show, that also $\boldsymbol{A}_{\mathcal{B}_{k+1}}$ obtained by swapping $\mathcal{B}(q)$ and $\mathcal{C}(s)$ has full rank.

Hint: Exploit, that q is chosen so that the qth component in $\boldsymbol{h} = \boldsymbol{B}^{-1}\boldsymbol{C}_{:,s}$ is nonzero.

The computational work is dominated by the solution of the two systems

$$\boldsymbol{B}^T\boldsymbol{g} = \boldsymbol{c}_{\mathcal{B}} \qquad \text{and} \qquad \boldsymbol{B}\boldsymbol{h} = \boldsymbol{C}_{:,s}$$

in each iteration step. This is discussed in Section 4.5. Before that, however, we look at the selection of incoming and outgoing basis variables and the choice of starting point.

4.2. Degeneracy

A vertex is said to be *degenerate* if one or more components in $\mathbf{x}_{\mathcal{B}}^{(k)}$ is zero. This happens if more than $n-m$ hyperplanes meet at the vertex, as illustrated below.

Example 4.4. For $n-m = 2$ we show a nondegenerate vertex $\mathbf{x}_{k-1}$ and an adjacent, degenerate vertex $\mathbf{x}_k$. The infeasible parts of $\mathbb{R}^2$ are indicated by shading.

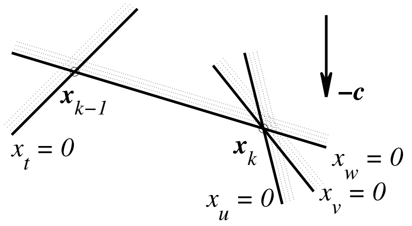


Figure 4.3. Degeneracy

At $\mathbf{x}_{k-1}$ the variables x_u and x_v are positive, so that $u, v \in \mathcal{B}_{k-1}$, while $t, w \in \mathcal{C}_{k-1}$. Let $\boldsymbol{h}$ correspond to the edge $x_w = 0$. Then both x_u and x_v are blocking variables[5]: $\sigma_{ju} = \sigma_{jv} = \alpha$. At $\mathbf{x}_k$ the nonbasic variable x_t is swapped with either x_u or x_v.

Now, consider the Simplex step from $\mathbf{x}_k$ (without knowledge of, how we got there). All three edges correspond to descent directions, but one of them is in $\mathcal{B}_k$ and the others are in $\mathcal{C}_k$. There are the following possibilities

In $\mathcal{C}_k$	"Chosen as s"	Blocking variable
v, w	v	x_u
v, w	w	x_u
u, w	u	–
u, w	w	x_v
u, v	u	–
u, v	v	–

In the cases marked "–" there is no problems in leaving $\mathbf{x}_k$. In the other cases we get $\alpha = 0$, i.e. no progress: $\mathbf{x}_{k+1} = \mathbf{x}_k$, but $\mathcal{B}$ and $\mathcal{C}$ have changed. Further, there is a risk of *"cycling"*: The iteration may go on for ever, alternating between $u \in \mathcal{B}$, $v, w \in \mathcal{C}$ and $v \in \mathcal{B}$, $u, w \in \mathcal{C}$.

This example may seem pathological, but for large, "real-life" problems it seems to be the rule rather than an exception that some of the vertices are degenerate.

[5] ju and jv are the indices in $\mathcal{B}_{k-1}$ of u and v, respectively.

Degeneracy of a vertex must not be confused with degeneracy of the problem, as discussed in Section 2.4. Problem (P) my be nondegenerate and still have degenerate vertices. If the solution $\mathbf{x}^*$ has less than m nonzero elements (case 2° page 26), then the final vertex is degenerate, while a nonunique solution to (P) (case 3° page 26) may give a nondegenerate final vertex.

The cycling can be avoided by flagging the variables so that we cannot return to a splitting $\mathcal{B}$, $\mathcal{C}$ already tried. A special version of this is *Bland's rule,* where s is always chosen as the first index in the current $\mathcal{C}$, for which $d_i < 0$ and q is the first index in the current $\mathcal{B}$, for which $\sigma_j = \alpha$, [13, Section 8.3.3]. Such a strategy guarantees that we leave a degenerate vertex after a finite number of iteration steps, but does not prevent *"stalling"* – there is no progress during these steps.

Another approach is based on *perturbing* the problem so that an intersection between $n-m+p$ hyperplanes is split into p nondegenerate (but close) vertices. Such a method is described in [12]. Basically, if α computed by (4.5) is smaller than a prescribed positive value $\alpha_{\min}$, it is replaced by $\alpha_{\min}$, implying that some components of $\mathbf{z}$, (4.6), are negative. Let $\mathcal{J} \subseteq \mathcal{B}$ denote the set of indices for which $z_j < 0$, then $\mathbf{z}'$ obtained by replacing these components by zero is feasible for the perturbed constraints

$$\boldsymbol{A}\mathbf{x} = \boldsymbol{b}' \ , \quad \text{where} \quad \boldsymbol{b}' = \mathbf{b} - \boldsymbol{A}_{:,\mathcal{B}(\mathcal{J})}\mathbf{z}_{\mathcal{J}} \ . \tag{4.7}$$

Alternatively, the code can be written so that slight infeasibilities in the variables and constraints are accepted. After all, the computed results are affected by rounding errors, and occational "purification" is used during the iteration, see Section 4.5. In this connection the perturbed iterate can also be purified, e.g. by the technique described in (4.9) – (4.10).

4.3. Pricing and Pivoting

These names are often used[6] in connection with the choice of s i.e. the incoming basis variable. Once this is chosen, the number q of the outgoing variable is well defined, except when the vertex is degenerate.

[6] Citing from [36, Section 10.8]: *"... the subject of linear programming is surrounded by notational and terminological thickets. Both of these thorny defenses are lovingly cultivated by a coterie of stern acolytes who have devoted themselves to the field. Actually, the basic ideas of linear programming are quite simple."*

Normally, there is more than one negative element in $\boldsymbol{d}$, and as illustrated in Figure 4.4, the choice between them may have a large effect on the number of Simplex steps. Starting at the vertex marked by a double circle, we can go to the left and reach the solution in two steps. If we go to the right, we have to use 5 steps.

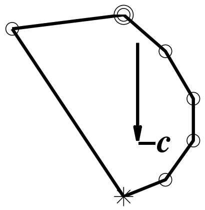


Figure 4.4

Ideally, we want to choose s so that the total computational work is minimized. This, however, is prohibitive. Instead we aim at optimizing the current Simplex step. Let $\mathcal{S}$ denote the set of candidates for incoming variable:

$$d_j < 0 \quad \Leftrightarrow \quad j \in \mathcal{S} \ .$$

Some simple choices of s are

1° Take s as the first member in $\mathcal{S}$.

2° $s = \underset{j \in \mathcal{S}}{\operatorname{argmax}} \{ |d_j| \}$.

The second choice implies, that if α were independent of s, then we maximize the gain $-\alpha d_s$. However, α depends on $\boldsymbol{h} = \boldsymbol{B}^{-1}\boldsymbol{C}_{:,s}$, and a more advanced strategy is to compute $\boldsymbol{h}$ for all $j \in \mathcal{S}$, and pick the one that maximizes the gain. This is also prohibitive,[7] but since the advent of efficient interior point methods a large effort has been made to get better Simplex methods, and especially this part has shown great improvement. See [29] for an overview and a list of references.

4.4. Initialization

It is common usage to distinguish between

Phase 1: Compute a starting point satisfying the conditions in (4.2).

Phase 2: The algorithm, as presented in the previous sections.

[7] and does not guarantee the best overall performance: In Figure 4.4 the first step to the right gives a larger decrease in the objective than the step to the left.

There exists a number of strategies for Phase 1. The simplest idea is based on solving the expanded problem

$$(\widehat{P}) \qquad \min\left\{ \ \widehat{\boldsymbol{c}}^T\widehat{\mathbf{x}} \ : \ \widehat{\mathbf{A}}\widehat{\mathbf{x}} = \boldsymbol{b}, \ \widehat{\mathbf{x}} \geq \boldsymbol{0} \ \right\} \ , \tag{4.8a}$$

where $\widehat{\mathbf{x}}, \widehat{\boldsymbol{c}} \in \mathbb{R}^{n+m}$ and $\widehat{\mathbf{A}} \in \mathbb{R}^{m\times(n+m)}$ are given by

$$\widehat{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ \boldsymbol{z} \end{bmatrix}, \quad \widehat{\boldsymbol{c}} = \begin{bmatrix} \boldsymbol{0} \\ \mathbf{e} \end{bmatrix}, \quad \widehat{\mathbf{A}} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{I} \end{bmatrix} \ . \tag{4.8b}$$

Since $\boldsymbol{z} \geq \boldsymbol{0}$, the minimal value of the objective $\widehat{\boldsymbol{c}}^T\widehat{\mathbf{x}} = \mathbf{e}^T\boldsymbol{z}$ occurs for $\boldsymbol{z} = \boldsymbol{0}$. Assume that all[8] $b_i \geq 0$, then $\widehat{\mathbf{x}} = \begin{bmatrix} \boldsymbol{0}^T & \boldsymbol{b}^T \end{bmatrix}^T$ satisfies the constraints. Thus, $(\widehat{P})$ is both bounded and feasible, and according to Theorem 2.2 it has a solution.

We can use SIMPLEX p. 68 to find this and note, that the choice $\widehat{\mathcal{B}}_0 = \{n{+}1, \ldots, n{+}m\}$ (corresponding to $\boldsymbol{B} = \boldsymbol{I}$, $\widehat{\mathbf{x}}_{\mathsf{B}}^{(0)} = \boldsymbol{b}$) satisfies the conditions in (4.2). Let $\widehat{\mathbf{x}}^+$ denote the result. If $\mathbf{e}^T\boldsymbol{z}^+ > 0$, then the original problem (P) is infeasible. Otherwise, $\boldsymbol{z}^+ = \boldsymbol{0}$, and $\mathbf{x}^+$ is feasible for (P). Even better, $\mathbf{x}^+$ is a vertex for (P). To see this, let p be the number of $\mathsf{b}_j^+ \leq n$ in the final set $\widehat{\mathcal{B}}^+$. If $\widehat{\mathbf{x}}^+$ i nondegenerate, then $p = m$ and we can start Phase 2 with $\mathcal{B}_0 = \widehat{\mathcal{B}}^+$. Also for a degenerate $\widehat{\mathbf{x}}^+$ it may happen that $p = m$, otherwise we have to supplement with $m{-}p$ other columns from $\mathbf{A}$ to get a full rank basis, see Section 4.5.

Example 4.5. For the problem of Example 4.1, $\boldsymbol{b} \geq \boldsymbol{0}$ and the expanded problem has the constraint matrix

$$\widehat{\mathbf{A}} = \begin{bmatrix} 1/2 & 1 & -1 & 0 & 1 & 0 \\ -2/3 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \ .$$

The choice $\widehat{\mathcal{B}}_0 = \{5, 6\}$, $\widehat{\mathcal{C}}_0 = \{1, 2, 3, 4\}$ gives

$$\widehat{\mathbf{x}}_{\mathcal{B}}^{(0)} = \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \ , \quad \boldsymbol{d} = \boldsymbol{0} - \boldsymbol{A}^T\mathbf{e} = \begin{bmatrix} \frac{1}{6} & -2 & 1 & -1 \end{bmatrix}^T \ .$$

We choose $s = 2$ and get

[8] If $b_k < 0$, we can replace $\mathbf{A}_{k,:}\mathbf{x} = b_k$ by $-\mathbf{A}_{k,:}\mathbf{x} = -b_k$.

$$\boldsymbol{h} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \alpha = \min\{1,2\} = 1 , \quad \boldsymbol{z} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} - 1 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} .$$

Thus, $q=1$, $\widehat{\mathcal{B}}_1 = \{2,6\}$, $\widehat{\mathcal{C}}_1 = \{1,5,3,4\}$, $\widehat{\mathbf{x}}_{\mathcal{B}}^{(1)} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$.

In the next iteration we find $\boldsymbol{d} = \begin{bmatrix} \frac{7}{6} & 2 & -1 & -1 \end{bmatrix}^T$.
We choose $s=3$ and get $\boldsymbol{h} = \begin{bmatrix} -1 & 1 \end{bmatrix}^T$, $\alpha=1$, $\boldsymbol{z} = \begin{bmatrix} 3 & 0 \end{bmatrix}^T$, $q=2$.
This leads to $\widehat{\mathcal{B}}_2 = \{2,3\}$, $\widehat{\mathcal{C}}_2 = \{1,5,6,4\}$, $\widehat{\mathbf{x}}_{\mathcal{B}}^{(2)} = \begin{bmatrix} 3 & 1 \end{bmatrix}^T$. Now, $\widehat{\boldsymbol{c}}_{\mathcal{B}} = \boldsymbol{0}$, so that $\boldsymbol{d} = \widehat{\boldsymbol{c}}_{\mathcal{C}} = \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix}^T \geq \boldsymbol{0}$, and Phase 1 is finished.

$\widehat{\mathcal{B}}^{+} = \widehat{\mathcal{B}}_2$ has both elements in the range $[1,\ n]$. Therefore, $\mathcal{B}_0 = \{2,3\}$ can be used for starting SIMPLEX on the original problem – as it was done in Example 4.1.

The Simplex method is well suited for *warm start.* This is of interest e.g., when a nonlinear problem is solved by iteration involving a series of LO problems, where the elements in $\boldsymbol{A}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ may have changed from the previous problem, but as the outer iteration proceeds, the optimal partion will be (almost) the same.

In such a case initialization via the expanded problem $(\widehat{P})$ is wasteful. Instead we can use the $\mathcal{B}$ from the previous problem as starting point for a special SIMPLEX iteration. We assume, that condition (4.2a) is satisfied[9] also with the new elements in $\boldsymbol{A}$, and modify the algorithm focusing on feasibility rather than the objective.

If $\mathbf{x}_{\mathcal{B}}^{(0)} = \boldsymbol{B}^{-1}\boldsymbol{b} \geq \boldsymbol{0}$, then all conditions in (4.2) are satisfied, and we can go directly to Phase 2. Otherwise, let $\mathcal{V} = \mathcal{V}(\mathbf{x}) \subseteq \mathcal{B}$ denote the set of violated constraints,

$$x_j < 0 \quad \Leftrightarrow \quad j \in \mathcal{V} , \tag{4.9a}$$

and let

$$\varphi(\mathbf{x}) = \sum_{j \in \mathcal{V}(\mathbf{x})} (-x_j) = \widetilde{\boldsymbol{c}}^T\mathbf{x} , \tag{4.9b}$$

with

[9] If not, then we have to adjust to get a full rank $\boldsymbol{B}$, see Section 4.5.

$$\widetilde{c}_j = \widetilde{c}_j(\mathbf{x}) = \begin{cases} -1 & \text{if } j \in \mathcal{V}(x) \\ 0 & \text{otherwise} \end{cases} . \tag{4.9c}$$

The function $\varphi(\mathbf{x})$ is nonnegative and *piecewise linear* since $\widetilde{\mathbf{c}}(\mathbf{x})$ is constant in regions, where no x_j changes sign.

The minimum of $\varphi(\mathbf{x})$ is zero – when all variables are feasible. We can use the Simplex ideas to find this minimum: Since $\widetilde{\boldsymbol{c}}_{\mathcal{C}} = \boldsymbol{0}$, Equation (4.3) reduces to

$$\widetilde{\mathbf{c}}^T\mathbf{x} = \widetilde{\mathbf{c}}^T\mathbf{x}_k + \mathbf{d}^T\mathbf{x}_{\mathcal{C}} \quad \text{with} \quad \boldsymbol{d} = -\boldsymbol{C}^T\boldsymbol{B}^{-T}\widetilde{\boldsymbol{c}}_{\mathcal{B}} .$$

If $\boldsymbol{d} \geq \boldsymbol{0}$, then we cannot decrease φ, and (P) is infeasible. Otherwise, choose s so that $d_s < 0$; compute $\boldsymbol{h}$ as in (4.4); and

$$\sigma_j = \begin{cases} \left.\begin{matrix} (\mathbf{x}_{\mathcal{B}}^{(k)})_j / h_j & \text{if } h_j < 0 \\ -\infty & \text{otherwise} \end{matrix}\right\} & \text{if } j \in \mathcal{V} \\ \left.\begin{matrix} (\mathbf{x}_{\mathcal{B}}^{(k)})_j / h_j & \text{if } h_j > 0 \\ +\infty & \text{otherwise} \end{matrix}\right\} & \text{if } j \in \mathcal{B} \ominus \mathcal{V} \end{cases} \tag{4.10a}$$

$$\alpha = \min\left\{ \min_{j \in \mathcal{B} \ominus \mathcal{V}} \{\sigma_j\},\ \max_{j \in \mathcal{V}} \{\sigma_j\} \right\} . \tag{4.10b}$$

The step taken to get the next iterate is $-\alpha\boldsymbol{h}$. The construction ensures that feasible variables stay feasible, and aims at getting as many variables feasible as possible. See [13, Section 8.6.4] for more details and a proof that this algorithm stops after a finite number of steps.

Example 4.6. For the problem of Example 4.1 take $\mathcal{B}_0 = \{3,4\}$. The matrix $\boldsymbol{B} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$ has full rank, and $\mathbf{x}_{\mathcal{B}}^{(0)} = \boldsymbol{B}^{-1}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$, so that $\mathcal{V}(\mathbf{x}_0) = \{1\}$.

We find $\boldsymbol{d} = \begin{bmatrix} -1/2 & -1 \end{bmatrix}^T$, and the choice $s=2$ results in $\boldsymbol{h} = \begin{bmatrix} -1 & 1 \end{bmatrix}^T$, $\alpha=1$ and $\boldsymbol{z} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$, $q=1$.

Thus, after one iteration we have $\mathcal{B}_1 = \{2,4\}$, $\mathbf{x}_{\mathcal{B}}^{(1)} = \begin{bmatrix} 1 & 1 \end{bmatrix}^T$, which is feasible, and we are ready for Phase 2. Actually, this starting point is the optimal solution.

4.5. Basis Factorization

Given the splitting of the index set $\{1,\ldots,n\}$ into $\mathcal{B}$ and $\mathcal{C}$, and the corresponding splitting of $\boldsymbol{A}$ into $\boldsymbol{B}$ and $\boldsymbol{C}$, we have to solve

$$\boldsymbol{B}^T\boldsymbol{g} = \boldsymbol{c}_\mathcal{B} \quad \text{and} \quad \boldsymbol{B}\boldsymbol{h} = \boldsymbol{C}_{:,s} , \tag{4.11}$$

where the choice of s was discussed in Section 4.3. If m is large, then the solution of these systems in each Simplex step dominates the execution time. Therefore it is necessary to have an efficient (and accurate) implementation.

As mentioned in Section 3.5, the constraint matrix $\boldsymbol{A}$ – and therefore the *basis matrix* $\boldsymbol{B}$ – is normally sparse, and this should be exploited. In contrast to the coefficient matrix in IPMs, $\boldsymbol{B}$ is neither symmetric nor positive definite (but normally considerably sparser than $\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$). This implies that we have to take into account both sparsity and numerical stability in the factorization of $\boldsymbol{B}$, see [34] or [9]. The result can be expressed as

$$\boldsymbol{P}\boldsymbol{B}\boldsymbol{Q} = \boldsymbol{L}\boldsymbol{U} , \tag{4.12}$$

where $\boldsymbol{P}$ and $\boldsymbol{Q}$ are permutation matrices representing the row and column interchanges, respectively; $\boldsymbol{L}$ is a unit lower triangular matrix; and $\boldsymbol{U}$ is an upper triangular matrix with nonzero diagonal elements (since $\boldsymbol{B}$ has full rank). If the problem is sparse, then so are $\boldsymbol{L}$ and $\boldsymbol{U}$.

Once the factorization is known, the vector $\boldsymbol{h}$ can be computed by[10)]

$$\boldsymbol{h} = \boldsymbol{Q}\boldsymbol{U}^{-1}\boldsymbol{L}^{-1}\boldsymbol{P}\boldsymbol{C}_{:,s} . \tag{4.13a}$$

This involves the solution of two triangular systems and two sets of interchanges of elements in a vector, and in sparse matrix terminology it is one SOLVE.

From (4.12) we see that $\boldsymbol{B}^T = \boldsymbol{Q}\boldsymbol{U}^T\boldsymbol{L}^T\boldsymbol{P}$, and the vector $\boldsymbol{g}$ can therefore be found as

$$\boldsymbol{g} = \boldsymbol{P}^T\boldsymbol{L}^{-T}\boldsymbol{U}^{-T}\boldsymbol{Q}^T\boldsymbol{c}_\mathcal{B} . \tag{4.13b}$$

10) See the footnote on page 55. For a permutation matrix $\boldsymbol{P}^{-1} = \boldsymbol{P}^T$.

This, again is a SOLVE, where the roles of $(\boldsymbol{P}, \boldsymbol{Q})$ are replaced by $(\boldsymbol{Q}^T, \boldsymbol{P}^T)$ and the lower (upper) triangular matrix $\boldsymbol{L}$ $(\boldsymbol{U})$ is replaced by the lower (upper) $\boldsymbol{U}^T$ $(\boldsymbol{L}^T)$.

The conclusion is, that each Simplex step involves one factorization and two SOLVEs. For a dense matrix the factorization involves about $\frac{2}{3}m^3$ flops, and each SOLVE "costs" $2m^2$ flops. For a sparse matrix a SOLVE costs about 1% of a facorization.

The next basis matrix $\boldsymbol{B}^{(k+1)}$ differs from $\boldsymbol{B}^{(k)}$ only in the qth column. This can be exploiting by computing the factorization for $\boldsymbol{B}^{(k+1)}$ by *updating* the previous factorization. We shall present the *Bartels-Golub algorithm,* where the $\boldsymbol{L}$ matrix is represented in product form of the inverse

$$\boldsymbol{M}_r \cdots \boldsymbol{M}_2\boldsymbol{M}_1\boldsymbol{B} = \boldsymbol{P}\boldsymbol{U}\boldsymbol{Q} . \tag{4.14}$$

Each $\boldsymbol{M}_i$ is a matrix which differs from $\boldsymbol{I}$ in just one of the off-diagonal elements. This factorization is derived in Appendix A.5.

Example 4.7. Let $\boldsymbol{M}_i$ have the off-diagonal element $-\ell$ in row k, column j. Then $\boldsymbol{v} = \boldsymbol{M}_i\boldsymbol{u}$ is identical with $\boldsymbol{u}$ except for $v_k = u_k - \ell u_j$. Thus, $\boldsymbol{M}_i$ represents an *elementary row operation.* It can be stored as a triple $(-\ell, k, j)$.

With this factorization of $\boldsymbol{B}$, the solutions to (4.11) are found as

$$\begin{aligned} \boldsymbol{g} &= \boldsymbol{M}_1^T\boldsymbol{M}_2^T \cdots \boldsymbol{M}_r^T\boldsymbol{P}\boldsymbol{U}^{-T}\boldsymbol{Q}\boldsymbol{c}_\mathcal{B} , \\ \boldsymbol{h} &= \boldsymbol{Q}^T\boldsymbol{U}^{-1}\boldsymbol{P}^T\boldsymbol{M}_r \cdots \boldsymbol{M}_2\boldsymbol{M}_1\boldsymbol{C}_{:,s} . \end{aligned} \tag{4.15}$$

Let (4.14) correspond to $\boldsymbol{B} = \boldsymbol{B}^{(k)}$. The next basis matrix $\boldsymbol{B}^{(k+1)}$ differs from $\boldsymbol{B}^{(k)}$ only in the qth column, which is replaced by $\boldsymbol{C}_{:,s}$. From (4.14) we see that the upper triangular $\boldsymbol{U}$ can be expressed as

$$\boldsymbol{U} = \boldsymbol{P}^T\boldsymbol{M}_r \cdots \boldsymbol{M}_2\boldsymbol{M}_1\boldsymbol{B}^{(k)}\boldsymbol{Q}^T .$$

This implies that

$$\boldsymbol{S} = \boldsymbol{P}^T\boldsymbol{M}_r \cdots \boldsymbol{M}_2\boldsymbol{M}_1\boldsymbol{B}^{(k+1)}\boldsymbol{Q}^T$$

is identical with $\boldsymbol{U}$ except for the $\overline{q}$th column, where $\overline{q}$ is the column number in $\boldsymbol{B}\boldsymbol{Q}^T$ of $\boldsymbol{B}_{:,q}$. This column is a *"spike"* with elements given by

$$\boldsymbol{S}_{:,\overline{q}} = \boldsymbol{P}^T \boldsymbol{M}_r \cdots \boldsymbol{M}_2 \boldsymbol{M}_1 \boldsymbol{C}_{:,s} \ .$$

The next $\boldsymbol{U}$ is obtained by transforming $\boldsymbol{S}$ to upper triangular form. The work is minimized by first moving the $\overline{q}$th column to the last position and shifting each of the columns $\overline{q}+1, \ldots, m$ one place to the left. The resulting matrix $\boldsymbol{S}\widetilde{\boldsymbol{Q}}$ is said to have *upper Hessenberg form.*[11] Finally, $\overline{\boldsymbol{U}}$ is obtained by applying at most $m-\overline{q}$ elemenatry row operations. The three matrices are indicated in Figure 4.5.

$$
\begin{array}{ccccccc}
 & & \overline{q} & & & & \\
\times & \times & \times & \times & \times & \times & \times \\
 & \times & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times \\
 & & \times & & \times & \times & \times \\
 & & \times & & & \times & \times \\
 & & \times & & & & \times \\
\end{array}
\qquad
\begin{array}{ccccccc}
 & & \overline{q} & & & & \\
\times & \times & \times & \times & \times & \times & \times \\
 & \times & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times \\
 & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times \\
 & & & & & \times & \times \\
\end{array}
\qquad
\begin{array}{ccccccc}
 & & & & & & \\
\times & \times & \times & \times & \times & \times & \times \\
 & \times & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times \\
 & & & \times & \times & \times & \times \\
 & & & & \times & \times & \times \\
 & & & & & \times & \times \\
 & & & & & & \times \\
\end{array}
$$

$\boldsymbol{S}$ $\qquad$ $\boldsymbol{S}\widetilde{\boldsymbol{Q}}$ $\qquad$ $\overline{\boldsymbol{U}}$

Figure 4.5. *Updating* $\boldsymbol{U}$

The accumulated result of these transformations is that

$$\boldsymbol{M}_{\overline{r}} \cdots \boldsymbol{M}_r \cdots \boldsymbol{M}_1 \boldsymbol{B}^{(k+1)} = \overline{\boldsymbol{P}}\,\overline{\boldsymbol{U}}\,\overline{\boldsymbol{Q}} \ ,$$

i.e. a factorization of the form (4.14), with more $\{\boldsymbol{M}_i\}$, however, and we can use (4.15) with $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{U}$, r updated to $\overline{\boldsymbol{P}}$, $\overline{\boldsymbol{Q}}$, $\overline{\boldsymbol{U}}$, $\overline{r}$.

In the transformation from $\boldsymbol{S}\widetilde{\boldsymbol{Q}}$ to $\overline{\boldsymbol{U}}$ row pivoting is used for stability reasons. Reid [37] showed that in the sparse case row and column permutations can often be used to decrease the number $(\overline{r}-r)$ of row operations. In this connection row interchanges is also used to reduce fill-in.

In the dense case the transformation from $\boldsymbol{S}$ to $\overline{\boldsymbol{U}}$ costs at most m^2-1 flops (when $\overline{q}=1$). Even for medium-sized problems this is considerably smaller than the $\frac{2}{3}m^3$ flops needed to factorize $\boldsymbol{B}^{(k+1)}$ "from scratch".

[11] This means that in each column it has at most one nonzero element below the diagonal. The step $\boldsymbol{S} \to \boldsymbol{S}\widetilde{\boldsymbol{Q}}$ is shared by another updating algorithm known by the names Forest-Tomlin. It differs from the Bartels-Golub algorithm (B-G) in some details, and experience indicate [29], that especially for sparse problems the B-G algorithm is superior.

As the Simplex iteration proceeds, the number r grows, implying that storage demands for the $\{\boldsymbol{M}_i\}$; time for performing the r simple row operations; and the effect of rounding errors all grow. In the sparse case $\boldsymbol{U}$ will also suffer from an increasing amount of fill-in. Therefore, an occational *refactorization* is used, e.g. every mth Simplex step or monitored by the number of nonzeros in $\boldsymbol{U}$ or by the *growth factor.* This is the largest element in $|\boldsymbol{U}|$ relative to the largest element in $|\boldsymbol{B}^{(k)}|$. If this is large, it indicates a risk of excessive effect of rounding errors.

Example 4.8. We have implemented the Simplex method as described above, with cold start Phase 1 defined by (4.8) and the choice of incoming varible given by 2° page 73. In case of a degenerate vertex we modify the problem by (4.7), and recompute (once) with warm start on the original problem. We refactorize if r exceeds $m(m-1)$, i.e. twice the number of nonzeros in $\boldsymbol{L}$ in an LU-factorization.

The costly part of each iteration is

Task	Dense matrix (flops)
Compute $\boldsymbol{g}$ and $\boldsymbol{h}$	$2\cdot(2m^2+2r)$
Compute $\boldsymbol{d}$	$2(n-m)m$
Update factorization	$2r+(m-\overline{q})^2$

On the average we can expect $r=\frac{3}{4}m^2$ and $m-\overline{q}=\frac{1}{2}m$, and this leads to the following estimate of the cost when $n=2m$,

$$R\cdot\frac{2}{3}m^3 + K_{\text{SPX}}\cdot 10m^2 \ ,$$

where R and K_{SPX} are the numbers of refactorization and iterations, respectively.

For comparison, suppose that we use the normal equations approach (see Section 3.5) in connection with algorithm *IPM3*, p. 49. The cost per iteration with this is dominated by

Task	Dense matrix (flops)
Compute $\boldsymbol{f}_P$ and $\boldsymbol{f}_D$	$2\cdot 2mn$
Compute $\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T$	$n\cdot m^2$
Compute $\boldsymbol{C}$	$\frac{1}{3}m^3+m^2$
Compute $\boldsymbol{h}_y^a$ and $\boldsymbol{h}_y^c$	$2\cdot 2m^2$

Thus, if $n=2m$, then the cost of K_{IPM} iterations with this method is about

$$K_{\text{IPM}}\left(\frac{7}{3}m^3+13m^2\right) \ .$$

We use these estimates in the next example.

From the two estimates we see that, if $K_{\text{SPX}} \lesssim 0.23m \cdot K_{\text{IPM}}$, then we can expect Simplex to be faster than IPM. Normally, however, K_{SPX} grows a little faster than m, while K_{IPM} is almost constant, so IPM wins for large problems. If $n \gg m$, then this change-over occurs for a smaller m-value.

As mentioned several times, "real-life problems" are often sparse, and this is exploited in both a Simplex and an IPM code. A comparison of computational effort is complicated by the fact, that normally the Cholesky factor in the IPM method is more dense[12] than the $\boldsymbol{L}$ (as represented by the $\boldsymbol{M}_r$) and the $\boldsymbol{U}$ in the factorization of $\boldsymbol{B}$, leading to the change-over for a larger m-value.

Example 4.9. We have used the algorithm from the previous example on the same problems as treated by algorithm *IPM3* in Example 3.9. In all cases shown, the largest relative error in the objective was `5.3e-13`.

First, Figure 4.6a shows results for well scaled problems. The notation $\#\mathcal{B} = 0.8m$ means that the solution is a degenerate vertex with $0.2m$ zeros in $\widetilde{\mathbf{x}}_{\mathcal{B}}$, while $\#\mathcal{B} = 1.2m$ means that the solution is not unique. The number of

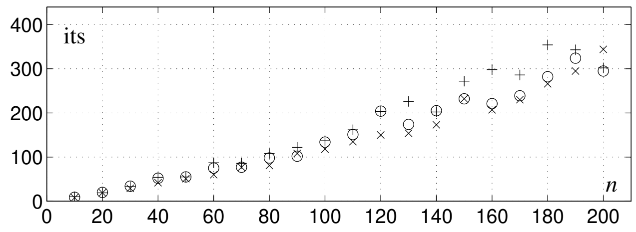


Figure 4.6a. *Simplex. Well scaled problems.*
+ : $\#\mathcal{B} = 0.8m$, o : $\#\mathcal{B} = m$, x : $\#\mathcal{B} = 1.2m$

iterations seems to grow slightly faster than linearly[13] with n, and the degenerate problems need slightly more iterations than the others. The worst

[12] See Figures 3.6a–b.

[13] A least sqares fit to the results shows, that $K_{\text{SPX}} \simeq 0.5 \cdot n^{1.2}$.

case is $(n, \#\mathcal{B}) = (180,\ 72)$, which needs 354 iterations (216 in Phase 1, 138 in Phase 2) and one refactorization. For comparison, this problem needed 12 iterations with *IPM3*, and the estimates in the previous example give

$$\text{work}_{\text{SPX}} = \texttt{2.92e+07}\ \text{flops}, \qquad \text{work}_{\text{IPM}} = \texttt{2.17e+07}\ \text{flops} \ .$$

In Figure 4.6b we show results for "descaled" problems, $\gamma_x = 10^3$, $\gamma_y = 1$, $\gamma_s = 10^{-3}$. The number of iterations is larger than for the well scaled problems, but the Simplex method seems to be less sensitive than the IPM method used in Example 3.9.

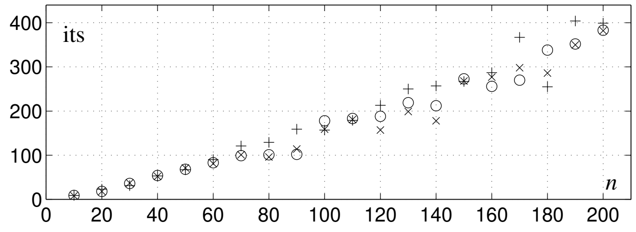


Figure 4.6b. *Simplex. Descaled problems.*
+ : $\#\mathcal{B} = 0.8m$, o : $\#\mathcal{B} = m$, x : $\#\mathcal{B} = 1.2m$

4.6. Dual Simplex

For the primal problem in standard form

$$(P) \qquad \min\{\ \mathbf{c}^T\mathbf{x}\ :\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\ \} \ , \tag{4.16a}$$

the dual is

$$(D) \qquad \max\{\ \mathbf{b}^T\mathbf{y}\ :\ \mathbf{A}^T\mathbf{y} + \mathbf{s} = \mathbf{c},\ \mathbf{s} \geq \mathbf{0}\ \} \ . \tag{4.16b}$$

Assume, that we have solved (P) by the Simplex method with the final splitting $\mathcal{B}$, $\mathcal{C}$. Then it follows from (2.24) that an optimal solution to (D) is given by

$$\widetilde{\mathbf{y}} = \mathbf{B}^{-T}\mathbf{c}_{\mathcal{B}}\ , \quad \widetilde{\mathbf{s}}_{\mathcal{B}} = \mathbf{0}\ , \quad \widetilde{\mathbf{s}}_{\mathcal{C}} = \mathbf{c}_{\mathcal{C}} - \mathbf{C}^T\widetilde{\mathbf{y}}\ . \tag{4.17}$$

By comparison with Section 4.1 we see that $(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{s}}_{\mathcal{C}})$ are the final version of the "Simplex vectors" $(\boldsymbol{g}, \boldsymbol{d})$.

Example 4.10. Assume that (P) is nondegenerate, i.e. $\widetilde{\mathbf{x}}_{\mathcal{B}} > \boldsymbol{0}$, and let us perturb the right-hand side to $\boldsymbol{b}+\mathbf{r}$. Then $\widetilde{\mathbf{x}}_{\mathcal{B}}$ is changed to $\widetilde{\mathbf{x}}_{\mathcal{B}} + \boldsymbol{B}^{-1}\mathbf{r}$, and for $\|\mathbf{r}\|$ small enough, this stays positive, so that we have the same optimal splitting. The objective is changed by

$$\boldsymbol{c}_{\mathcal{B}}^T(\boldsymbol{B}^{-1}\mathbf{r}) = \widetilde{\boldsymbol{y}}^T\mathbf{r} \ .$$

Thus, $\widetilde{\boldsymbol{y}}$ expresses the sensitivity of the optimal value with respect to small changes in the right-hand side.

Economists played an important role in the early days of linear programming and the development of the Simplex method. Problem (P) was often formulated as the minimization of expenditure, so the relation between $\widetilde{\boldsymbol{y}}$ and the vector $\boldsymbol{d}$ justifies the name "pricing vector", cf. Section 4.3. In the same vein, $\widetilde{y}_i$ is called the *marginal cost* of constraint number i.

Example 4.11. The problem of Example 4.1 has the optimal splitting $\mathcal{B} = \{2, 4\}$, $\mathcal{C} = \{1, 3\}$, and

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} , \quad \boldsymbol{C} = \begin{bmatrix} 1/2 & -1 \\ -2/3 & 0 \end{bmatrix} .$$

The solution to the dual problem was given in Example 3.3, $\widetilde{\boldsymbol{y}} = \begin{bmatrix} 3 & 0 \end{bmatrix}^T$, $\widetilde{\mathbf{s}} = \begin{bmatrix} 0.5 & 0 & 3 & 0 \end{bmatrix}^T$, and it is easily seen that they satisfy (4.17).

From $\widetilde{y}_2 = 0$ we se that a small change in b_2 should not change the objective:

$$\mathbf{x}_{\mathcal{B}} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 2+r \end{bmatrix} = \begin{bmatrix} 1 \\ 1+r \end{bmatrix} ,$$

which stays positive for $r \geq -1$, and thus, is optimal in this range. The objective is

$$\boldsymbol{c}^T\mathbf{x} = \boldsymbol{c}_{\mathcal{B}}^T\mathbf{x}_{\mathcal{B}} = 3 \cdot 1 + 0 \cdot (1+r) = 3 ,$$

which is independent of r.

The *Dual Simplex Method* is based on this relationship between the dual solution and the vectors used in the primal Simplex method described in the previous sections. Instead of the warm start algorithm of Section 4.4 it may be advantageous to use Dual Simplex.

Suppose that we are given a splitting $\mathcal{B}$, $\mathcal{C}$, so that

$$\boldsymbol{y} = \boldsymbol{B}^{-T}\boldsymbol{c}_{\mathcal{B}} \tag{4.18a}$$

is feasible for (D), i.e.

$$\boldsymbol{s}_{\mathcal{B}} \equiv \boldsymbol{c}_{\mathcal{B}} - \boldsymbol{B}^T\boldsymbol{y} = \boldsymbol{0} \quad \text{and} \quad \boldsymbol{s}_{\mathcal{C}} \equiv \boldsymbol{c}_{\mathcal{C}} - \boldsymbol{C}^T\boldsymbol{y} \geq \boldsymbol{0} \ . \tag{4.18b}$$

Then the basic solution given by $\mathbf{x}_{\mathcal{B}} = \boldsymbol{B}^{-1}\boldsymbol{b}$ is said to be *dual feasible.* If it is also primal feasible (i.e. $\mathbf{x}_{\mathcal{B}} \geq \boldsymbol{0}$), then it is optimal. Otherwise, let q be the index of a negative component of $\mathbf{x}_{\mathcal{B}}$. We change $\boldsymbol{y} = \boldsymbol{y}^{(k)}$ to

$$\boldsymbol{y} - \nu\mathbf{u} \quad \text{with} \quad \mathbf{u} = \boldsymbol{B}^{-T}\mathbf{e}_q \ , \quad \nu > 0 \ . \tag{4.19}$$

Then the dual objective increases:

$$\begin{aligned} \boldsymbol{b}^T(\boldsymbol{y} - \nu\mathbf{u}) &= \boldsymbol{b}^T\boldsymbol{y} - \nu\mathbf{e}_q^T\boldsymbol{B}^{-1}\boldsymbol{b} \\ &= \boldsymbol{b}^T\boldsymbol{y} - \nu(\mathbf{x}_{\mathcal{B}})_q > \boldsymbol{b}^T\boldsymbol{y} \end{aligned}$$

since $(\mathbf{x}_{\mathcal{B}})_q < 0$. As regards dual feasibility, we see that

$$\begin{aligned} &\boldsymbol{s}_{\mathcal{B}} = \boldsymbol{c}_{\mathcal{B}} - \boldsymbol{B}^T(\boldsymbol{y} - \nu\mathbf{u}) = \nu\mathbf{e}_q \ , \\ &\boldsymbol{s}_{\mathcal{C}} = \boldsymbol{s}_{\mathcal{C}}^{(k)} + \nu\boldsymbol{h} \quad \text{with} \quad \boldsymbol{h} = \boldsymbol{C}^T\mathbf{u} \ . \end{aligned}$$

If $\boldsymbol{h} \geq \boldsymbol{0}$, then we can increase ν infinitely: (D) is unbounded. Otherwise, we proceed similar to (4.5),

$$\sigma_j = \begin{cases} (-\boldsymbol{s}_{\mathcal{C}}^{(k)})_j / h_j & \text{if } h_j < 0 \\ +\infty & \text{otherwise} \end{cases} , \quad j = 1, \ldots, n-m \ , \tag{4.20a}$$

$$p = \operatorname{argmin}\{\sigma_j\} \ , \qquad \alpha = \sigma_p \ . \tag{4.20b}$$

Now, the $\mathcal{B}(q)$ is swapped with $\mathcal{C}(p)$; $\boldsymbol{y}^{(k+1)} = \boldsymbol{y}^{(k)} - \alpha\boldsymbol{h}$; $\boldsymbol{B}$ is updated as described in Section 4.5; and $\mathbf{x}_{\mathcal{B}}^{(k+1)} = \boldsymbol{B}^{-1}\boldsymbol{b}$.

Example 4.12. We shall apply the Dual Simplex Method to the problem of Examples 4.1 and 4.11:

$$A = \begin{bmatrix} 1/2 & 1 & -1 & 0 \\ -2/3 & 1 & 0 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 2 & 3 & 0 & 0 \end{bmatrix}^T .$$

For $b = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$ the optimal splitting is $\mathcal{B} = \{2,4\}$, $\mathcal{C} = \{1,3\}$, and $\widetilde{y} = \begin{bmatrix} 3 & 0 \end{bmatrix}^T$, $\widetilde{s} = \begin{bmatrix} 0.5 & 0 & 3 & 0 \end{bmatrix}^T$.

Now, we change the right-hand side to $b = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$, and from Example 4.11 (with $r = -2$) we see that $\mathbf{x}_{\mathcal{B}} = \begin{bmatrix} 1 & -1 \end{bmatrix}^T$, which is not primal feasible. The change in b does not, however, affect the dual constraints, so with the choice $y^{(0)} = \widetilde{y}$, the vector $\mathbf{x}_{\mathcal{B}}$ is dual feasible.

With $q = 2$ we find

$$u = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-T} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad h = \begin{bmatrix} 1/2 & -1 \\ -2/3 & 0 \end{bmatrix}^T \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -7/6 \\ 1 \end{bmatrix},$$

$$\sigma_1 = \frac{-0.5}{-7/6} = \frac{3}{7}, \quad \sigma_2 = +\infty, p = 1, \ \alpha = \frac{3}{7} .$$

The next splitting is $\mathcal{B} = \{2,1\}$, $\mathcal{C} = \{4,3\}$, and we find

$$\mathbf{x}_{\mathcal{B}} = \begin{bmatrix} 1 & 1/2 \\ 1 & -2/3 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 4/7 \\ 6/7 \end{bmatrix},$$

which is positive, so we are finished. The optimal vector is $\mathbf{x}^* = \begin{bmatrix} \frac{6}{7} & \frac{4}{7} & 0 & 0 \end{bmatrix}^T$.

5. CONTINUATION METHODS

This class of methods can be said to have the flavour of both IPMs and the Simplex method. With the former they share the advantage of continuous differentiability and not being restricted to the vertices of a polytope. With the latter they share the possibility of cheap updates of the factorization.

We shall restrict ourselves to discussing a special version, based on the work of Madsen et al. [24] – [26] and on [23]. The method is formulated in Section 5.1 and further developed in the rest of the chapter.

5.1. Formulation

Again, we consider an LO problem in standard form

$$(P) \qquad \min\{\ c^T x \ : \ Ax = b, \ x \geq 0\ \}$$

with the dual

$$(D) \qquad \max\{\ b^T y \ : \ A^T y \leq c, \quad y \text{ free}\ \} .$$

We solve the dual problem via a series of smooth approximations depending on a positive parameter γ, which we let go to zero. These problems have the form

$$(D_\gamma) \qquad \max\left\{\ \varphi_\gamma(y) \equiv b^T y - \sum_{i=1}^{n} \rho_\gamma(s_i(y)) \ : \ y \text{ free} \right\}, \tag{5.1a}$$

where $s_i(y)$ is the ith component of the surplus vector

$$s = s(y) = c - A^T y , \tag{5.1b}$$

and ρ_γ is defined by

$$\rho_\gamma(s) = \begin{cases} \frac{1}{2\gamma}(\gamma - s)^2 & \text{if } s \leq \gamma \\ 0 & \text{if } s > \gamma \end{cases} . \tag{5.1c}$$

The feasibility condition $\boldsymbol{A}^T\boldsymbol{y} \le \boldsymbol{c}$ is equivalent with $\boldsymbol{s}(\boldsymbol{y}) \ge \boldsymbol{0}$, and the function ρ_γ is chosen to smooth the separation between feasible and infeasible parts of the domain and to penalize infeasible points.

Example 5.1. In Figure 5.1 we show $\rho_\gamma(s)$ for two values of γ. The function is a piecewise second order polynomial with continuous first derivative. As $\gamma \searrow 0$, the shape of the graph approaches a corner.

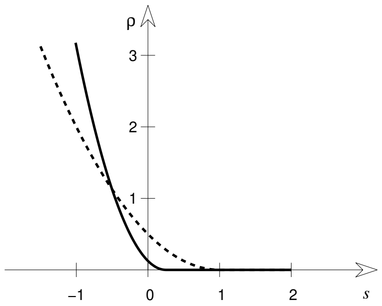


Figure 5.1. *Smoothing function for $\gamma = 1$ (dotted line) and $\gamma = \frac{1}{4}$ (full line)*

We introduce an "activity matrix" $\boldsymbol{W}_\gamma(\boldsymbol{y}) = \mathrm{diag}(w_{(\gamma)i}(\boldsymbol{y}))$ as follows[1)]

$$w_{(\gamma)i}(\boldsymbol{y}) = \begin{cases} 1 & \text{if} \;\; s_i(\boldsymbol{y}) \le \gamma \\ 0 & \text{if} \;\; s_i(\boldsymbol{y}) > \gamma \end{cases} \quad . \tag{5.2}$$

The *active set* $\mathcal{A}(\gamma, \boldsymbol{y})$ consists of the indices for which $w_{(\gamma)i}(\boldsymbol{y}) = 1$.

Now, we can write the objective function φ_γ in the form

$$\varphi_\gamma(\boldsymbol{y}) = \boldsymbol{b}^T\boldsymbol{y} - \frac{1}{2\gamma}(\gamma\boldsymbol{e} - \boldsymbol{s})^T \boldsymbol{W}(\gamma\boldsymbol{e} - \boldsymbol{s}) \ , \tag{5.3a}$$

where we have omitted the index γ and argument $\boldsymbol{y}$ for notational convenience.

1) Note, that $\boldsymbol{W}^2 = \boldsymbol{W}$. A matrix with this property is said to be *idempotent*.

The hyperplanes $s_i(\boldsymbol{y}) = \gamma$ divide $\mathbb{R}^m$ into subregions, inside each of which $\mathcal{A}(\gamma, \boldsymbol{y})$ and therefore $\boldsymbol{W}_\gamma(\boldsymbol{y})$ are constant. This means that $\varphi_\gamma(\boldsymbol{y})$ is a piecewise second order polynomial. It is continuous, and also the gradient

$$\nabla_y \varphi_\gamma(\boldsymbol{y}) = \boldsymbol{b} - \tfrac{1}{\gamma}\boldsymbol{A}\boldsymbol{W}(\gamma\boldsymbol{e} - \boldsymbol{s}) \tag{5.3b}$$

varies continuously across the hyperplanes.

Example 5.2. For the problem of Example 1.1 we have

$$\boldsymbol{A} = \begin{bmatrix} 1/2 & 1 & -1 & 0 \\ -2/3 & 1 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} 2 & 3 & 0 & 0 \end{bmatrix}^T .$$

and the hyperplanes $s_i(\boldsymbol{y}) = \gamma$ are the lines shown in Figure 5.2 for $\gamma = 1$ and $\gamma = \frac{1}{4}$. In the shaded areas the active set is $\mathcal{A}(\gamma, \boldsymbol{y}) = \{1, 2, 4\}$. We have also marked the solution $\boldsymbol{y}^* = (3, 0)$, cf. Example 2.7.

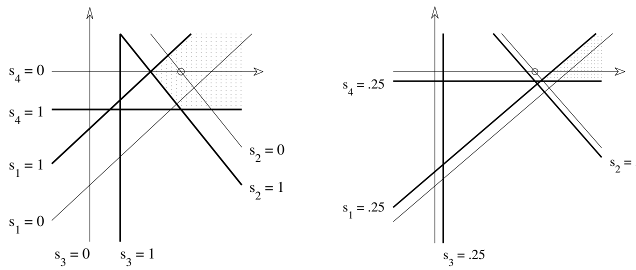


Figure 5.2. *Dividing hyperplanes for $\gamma = 1$ (left) and $\gamma = \frac{1}{4}$ (right)*

A maximizer $\boldsymbol{y}_\gamma$ of (D_γ) satisfies $\nabla_y \varphi_\gamma(\boldsymbol{y}_\gamma) = \boldsymbol{0}$, or

$$\boldsymbol{b} - \boldsymbol{A}\boldsymbol{W}(\boldsymbol{e} - \tfrac{1}{\gamma}\boldsymbol{s}) = \boldsymbol{0} \ , \tag{5.4a}$$

where $\boldsymbol{W} = \boldsymbol{W}_\gamma(\boldsymbol{y}_\gamma)$, $\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{y}_\gamma)$. We can reformulate this to

$$\boldsymbol{A}\boldsymbol{W}\boldsymbol{A}^T\boldsymbol{y}_\gamma = \boldsymbol{A}\boldsymbol{W}\boldsymbol{c} + \gamma(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{W}\boldsymbol{e}) \ . \tag{5.4b}$$

If we knew $\boldsymbol{W} = \boldsymbol{W}_\gamma(\boldsymbol{y}_\gamma)$, then this were a linear problem. Without this knowledge, however, it is a nonlinear problem, which we solve by Newton iteration, see Section 5.2. Having found the solution, we reduce γ. Equation (5.4b) shows that if $\boldsymbol{W}$ stays equal to $\boldsymbol{W}_\gamma(\boldsymbol{y}_\gamma)$, then $\boldsymbol{y}_\gamma$ changes linearly with γ:

$$\boldsymbol{y}_\gamma = \boldsymbol{y}_{(0)} + \gamma \boldsymbol{u} \ , \tag{5.5a}$$

where

$$\boldsymbol{AWA}^T\boldsymbol{y}_{(0)} = \boldsymbol{AWc}, \quad \boldsymbol{AWA}^T\boldsymbol{u} = \boldsymbol{b} - \boldsymbol{AWe} \ . \tag{5.5b}$$

A simpler expression for $\boldsymbol{u}$ is

$$\boldsymbol{u} = \tfrac{1}{\gamma}\left(\boldsymbol{y}_\gamma - \boldsymbol{y}_{(0)}\right) \ . \tag{5.5c}$$

Now, for $0 \le \delta \le \gamma$ let

$$\begin{aligned} \boldsymbol{y}_{(\gamma-\delta)} &= \boldsymbol{y}_{(0)} + (\gamma-\delta)\boldsymbol{u} \\ &= \boldsymbol{y}_\gamma - \delta\boldsymbol{u} \end{aligned} \tag{5.6a}$$

with the surplus vector

$$\begin{aligned} \boldsymbol{s}(\boldsymbol{y}_{(\gamma-\delta)}) &= \boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y}_{(\gamma-\delta)} \\ &= \boldsymbol{s}(\boldsymbol{y}_\gamma) + \delta\boldsymbol{v} \ , \quad \text{where} \quad \boldsymbol{v} = \boldsymbol{A}^T\boldsymbol{u} \ . \end{aligned} \tag{5.6b}$$

If $\mathcal{A}(\gamma-\delta, \boldsymbol{y}_{(\gamma-\delta)}) = \mathcal{A}(\gamma, \boldsymbol{y}_\gamma)$ for all $\delta \in [0, \gamma]$, then this set is equal to the set $\mathcal{B}$ containing the indices of basic variables of the solution – cf. the introduction to Chapter 4. A sufficient condition is that

$$\left.\begin{array}{ll} k \in \mathcal{A}(\gamma, \boldsymbol{y}_\gamma): & s_k(\boldsymbol{y}_\gamma) + \delta v_k \le \gamma-\delta \\ k \notin \mathcal{A}(\gamma, \boldsymbol{y}_\gamma): & s_k(\boldsymbol{y}_\gamma) + \delta v_k > \gamma-\delta \end{array}\right\} \text{ for } \ 0 \le \delta \le \gamma \ .$$

This is satisfied only if

$$\begin{array}{ll} k \in \mathcal{A}(\gamma, \boldsymbol{y}_\gamma): & s_k(\boldsymbol{y}_{(0)}) = 0, \quad v_k = -s_k(\boldsymbol{y}_\gamma) \ , \\ k \notin \mathcal{A}(\gamma, \boldsymbol{y}_\gamma): & v_k > -\frac{1}{\gamma}s_k(\boldsymbol{y}_\gamma) \quad . \end{array} \tag{5.7}$$

If one or more components of $\boldsymbol{s}(\boldsymbol{y}_{(\gamma-\delta)})$ pass the value $\gamma-\delta$ for some δ in the range $]0, \gamma[$, then the active set changes, and we compute $\boldsymbol{y}_\gamma$ for a reduced value of γ. In [26] we have shown, that for γ smaller than a finite, positive value we can identify the active set corresponding to $\gamma = 0$.

As regards the primal solution, let

$$\widetilde{\mathbf{x}} = \boldsymbol{W}_\gamma(\boldsymbol{y}_\gamma)\left(\boldsymbol{e} - \tfrac{1}{\gamma}\boldsymbol{s}(\boldsymbol{y}_\gamma)\right) \ . \tag{5.8}$$

From (5.2) it follows that $\widetilde{\mathbf{x}} \ge \boldsymbol{0}$, and (5.4a) shows that this vector satisfies the constraints $\boldsymbol{A}\widetilde{\mathbf{x}} = \boldsymbol{b}$. This means that $\widetilde{\mathbf{x}}$ is feasible for (P), and the duality gap is

$$\boldsymbol{c}^T\widetilde{\mathbf{x}} - \boldsymbol{b}^T\boldsymbol{y}_{(0)} = \widetilde{\mathbf{x}}^T\left(\boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y}_{(0)}\right) = \widetilde{\mathbf{x}}^T\boldsymbol{W}_\gamma(\boldsymbol{y}_\gamma)\boldsymbol{s}(\boldsymbol{y}_{(0)}) \ .$$

Thus, if (5.7) is satisfied, then the duality gap is closed, and $\widetilde{\mathbf{x}}$ is optimal for (P).

The algorithm[2)] is summarized in

Algorithm PP
```
Choose starting value γ
repeat
   Compute y_γ                              {Section 5.2}
   if  (final set)  then                    {Section 5.3}
      Compute x by (5.8);   STOP
   else
      Reduce γ                              {Section 5.3}
   end
end
```

Example 5.3. For the problem of the previous example the threshold value $\gamma = 1$ gives the active set $\mathcal{A}(\gamma, \boldsymbol{y}_\gamma) = \{1, 2, 4\}$ and

$$\boldsymbol{y}_\gamma = \begin{bmatrix} 2.681 \\ 0.223 \end{bmatrix}, \quad \boldsymbol{s}(\boldsymbol{y}_\gamma) = \begin{bmatrix} 0.809 \\ 0.096 \\ 2.681 \\ -0.223 \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} -0.638 \\ 0.447 \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} -0.617 \\ -0.191 \\ 0.638 \\ 0.447 \end{bmatrix} .$$

For $\delta = 0.5$ the index $k = 1$ leaves the active set.

With $\gamma = 0.25$ we get $\mathcal{A}(\gamma, \boldsymbol{y}_\gamma) = \{2, 4\}$ and

$$\boldsymbol{y}_\gamma = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \quad \boldsymbol{s}(\boldsymbol{y}_\gamma) = \begin{bmatrix} 0.5 & 0 & 3 & 0 \end{bmatrix}^T, \quad \boldsymbol{u} = \boldsymbol{0}, \quad \boldsymbol{v} = \boldsymbol{0} \ .$$

[2)] The name *"PP"* is chosen because the method is based on piecewise polynomials.

This means that the conditions in (5.7) are satisfied: $\mathcal{B} = \{2, 4\}$, and (5.8) gives the primal solution (in agreement with e.g. Example 3.3),

$$\widetilde{\mathbf{x}} = \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 0 & \\ & & & 1 \end{bmatrix} \left(\boldsymbol{e} - \begin{bmatrix} 2 \\ 0 \\ 12 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} .$$

The result $\boldsymbol{u} = \boldsymbol{0}$ is special for this problem, where $\boldsymbol{b} = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$, and

$$\boldsymbol{b} - \boldsymbol{A}\boldsymbol{W}\boldsymbol{e} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \boldsymbol{0} .$$

Therefore $\boldsymbol{u} = \boldsymbol{0}$ according to (5.4b), and for all $\gamma < 0.5$ we find $\boldsymbol{y}_\gamma = \boldsymbol{y}_{(0)}$.

5.2. Compute Maximizer

We use Newton iteration with line search to find the maximizer $\boldsymbol{y}_\gamma$: Let $\boldsymbol{y}$ denote the current iterate with corresponding $\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{y})$ and $\boldsymbol{W} = \boldsymbol{W}_\gamma(\boldsymbol{y})$. The Newton direction $\boldsymbol{h}$ is found by linearizing[3)] the expression (5.3b) for $\nabla_y \varphi_\gamma(\boldsymbol{y}+\boldsymbol{h}) = \boldsymbol{0}$,

$$\nabla_y \varphi_\gamma(\boldsymbol{y} + \boldsymbol{h}) \simeq \boldsymbol{b} - \frac{1}{\gamma}\boldsymbol{A}\boldsymbol{W}(\gamma \boldsymbol{e} - \boldsymbol{s} + \boldsymbol{A}^T\boldsymbol{h}) = \boldsymbol{0} ,$$

or

$$\boldsymbol{A}\boldsymbol{W}\boldsymbol{A}^T\boldsymbol{h} = \gamma\boldsymbol{b} - \boldsymbol{A}\boldsymbol{W}(\gamma\boldsymbol{e} - \boldsymbol{s}) \equiv \boldsymbol{g} . \qquad (5.9)$$

If $\boldsymbol{g} = \boldsymbol{0}$, then $\boldsymbol{y}$ is optimal, $\boldsymbol{y} = \boldsymbol{y}_\gamma$. Otherwise we use line search to find the maximizer of

$$\begin{aligned} \psi(\alpha) &\equiv \gamma\varphi_\gamma(\boldsymbol{y} + \alpha\boldsymbol{h}) \\ &= \gamma(\boldsymbol{y} + \alpha\boldsymbol{h})^T\boldsymbol{b} - \tfrac{1}{2}(\gamma\boldsymbol{e} - \boldsymbol{s} + \alpha\boldsymbol{d})^T\boldsymbol{W}(\gamma\boldsymbol{e} - \boldsymbol{s} + \alpha\boldsymbol{d}) \\ &= c_0(\alpha) + c_1(\alpha)\alpha - \tfrac{1}{2}c_2(\alpha)\alpha^2 . \qquad (5.10a) \end{aligned}$$

Here,

$$\boldsymbol{d} = \boldsymbol{A}^T\boldsymbol{h} , \qquad (5.10b)$$

3) If $\boldsymbol{W}$ does not change, then there is no approximation involved.

and the coefficients

$$\begin{aligned} c_0(\alpha) &= \gamma\boldsymbol{y}^T\boldsymbol{b} - \tfrac{1}{2}(\gamma\boldsymbol{e} - \boldsymbol{s})^T\boldsymbol{W}(\gamma\boldsymbol{e} - \boldsymbol{s}) \\ c_1(\alpha) &= \gamma\boldsymbol{h}^T\boldsymbol{b} - \boldsymbol{d}^T\boldsymbol{W}(\gamma\boldsymbol{e} - \boldsymbol{s}) \qquad (5.10c) \\ c_2(\alpha) &= \boldsymbol{d}^T\boldsymbol{W}\boldsymbol{d} \end{aligned}$$

are piecewise constant. They change at the *kink values* $\{\alpha_k\}$, where a component of $\boldsymbol{s}(\boldsymbol{y}+\alpha\boldsymbol{h}) = \boldsymbol{s} - \alpha\boldsymbol{d}$ passes the value γ. Let $s_i(\boldsymbol{y}+\alpha\boldsymbol{h}) = \gamma$ for $\alpha = \alpha_k > 0$. This implies

$$\alpha_k = \frac{s_i - \gamma}{d_i} , \qquad (5.11)$$

and only pairs[4)] $(s_i > \gamma,\ d_i > 0)$ or $(s_i < \gamma,\ d_i < 0)$ give rise to kink values. We shall assume that the kink values are sorted in increasing order,

$$0 < \alpha_1 \le \alpha_2 \le \cdots \alpha_K .$$

It is easily seen that $K \le n$. Further, $\psi(\alpha)$ is concave, which implies that either it has a unique maximizer α^* in the range $[0, \alpha_K]$, or (D_γ) – and therefore (D) – is unbounded. If the problem is bounded, then the maximizer is the zero of the piecewise first order polynomial

$$\psi'(\alpha) = c_1(\alpha) - c_2(\alpha)\alpha , \qquad (5.12)$$

and it can be computed by the algorithm

$c_1 := c_1(0);\quad c_2 := c_2(0);\quad k := 1$
while $(k < K)$ **and** $(c_1 - c_2\alpha_k > 0)$
  Update c_1 and c_2
  $k := k+1$ (5.13)
end
if $k \le K$ **then** $\alpha^* := c_1/c_2$
  else Unbounded

From (5.9) – (5.10) we see that

$$c_1(0) = \boldsymbol{h}^T\boldsymbol{g} = \boldsymbol{h}^T\boldsymbol{H}\boldsymbol{h} = c_2(0) , \qquad (5.14)$$

4) The case $(s_i = \gamma,\ d_i < 0)$ $(i \in \mathcal{A}(\gamma, \boldsymbol{y})$ but leaves immediately) must be treated specially, see [35].

where we have introduced

$$H = AWA^T \ . \tag{5.15}$$

This matrix is symmetric and positive semidefinite. In this section[5)] we assume that $\boldsymbol{AW}$ has full rank, in which case $\boldsymbol{H}$ is positive definite, and $\boldsymbol{g} \neq \boldsymbol{0} \ \Rightarrow \ \boldsymbol{h} = \boldsymbol{H}^{-1}\boldsymbol{g} \neq \boldsymbol{0}$, so that $c_1(0) = c_2(0) > 0$.

If the smallest kink value $\alpha_1 \geq 1$, then $\alpha^* = 1$, and the vector

$$\widetilde{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{h}$$

has the same active set as $\boldsymbol{y}$. This implies that there is no approximation involved in the derivation of (5.9), and $\nabla_y \varphi_\gamma(\widetilde{\boldsymbol{y}}) = \boldsymbol{0}$, so that $\widetilde{\boldsymbol{y}} = \boldsymbol{y}_\gamma$. Otherwise, we need to *update* the coefficients: Assume that index i changes activity as α passes the kink value α_k. Then

$$\begin{aligned} &\texttt{if} \quad i \text{ enters } \mathcal{A} \quad \texttt{then} \quad c_1 := c_1 + \alpha_k d_i^2; \quad c_2 := c_2 + d_i^2; \\ &\qquad\qquad\qquad\quad\ \ \texttt{else} \quad c_1 := c_1 - \alpha_k d_i^2; \quad c_2 := c_2 - d_i^2; \end{aligned} \tag{5.16}$$

The updating of c_2 follows directly from (5.10c), and the expression for c_1 is derived from the continuity of $\psi'(\alpha)$.

The algorithm for computing the maximizer is summarized below. In [24] we have shown that this algorithm finds the maximizer in a finite number of steps.

Algorithm PPNewton
Get initial $\boldsymbol{y}$; $\quad \boldsymbol{s} := \boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y}$
`repeat`
  Compute $\boldsymbol{h}$ by (5.9); $\quad \boldsymbol{d} := \boldsymbol{A}^T\boldsymbol{h}$
  Compute kinks $\{\alpha_k\}$ by (5.11)
  Compute α^* by (5.13)
  $\boldsymbol{y} := \boldsymbol{y} + \alpha^*\boldsymbol{h}; \quad \boldsymbol{s} := \boldsymbol{s} - \alpha^*\boldsymbol{d}$
`until` $(\alpha_1 \geq 1)$ `or` $(\|\alpha^*\boldsymbol{h}\| \leq \varepsilon\|\boldsymbol{y}\|)$

In the examples we use $\varepsilon = 10^{-12}$. After a threshold reduction from γ to $\gamma{-}\delta$ we use (5.6a) to get the initial $\boldsymbol{y}$.

---

5) In the discussion of implementation aspects in Section 5.4 we also deal with the rank deficient case.

5.3. Update Threshold

Given γ and $\boldsymbol{y}_\gamma$, we can use (5.6) to see whether the active set changes as γ is reduced to $\gamma{-}\delta$.

Example 5.4. Below we illustrate five different possibilities for the behaviour of the surplus vector as $\gamma{-}\delta$ goes to zero.

The dotted line shows the critical value. The two components of $\boldsymbol{s}(\boldsymbol{y}_{(\gamma-\delta)})$ indicated by full line do not change activity, while the two dashed lines correspond to indices entering or leaving $\mathcal{A}$.

Finally, the line marked by circles illustrates that $\mathcal{B}$ may contain indices that are not in $\mathcal{A}(\gamma, \boldsymbol{y}_\gamma)$ for any strictly positive γ. This can occur for degenerate LO problems.

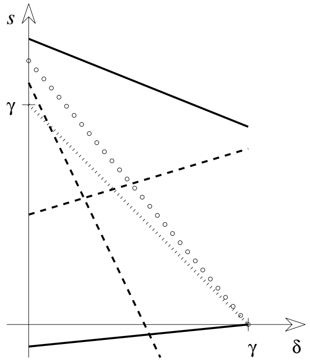


Figure 5.3

As in the line search in the previous section we define *kink values* $0 < \delta_1 \leq \delta_2 \leq \cdots \leq \delta_K < \gamma$ for the threshold reduction: Suppose that the ith component of $\boldsymbol{s}$ passes the critical value for $\delta = \delta_k$. From (5.6b) we see that this implies

$$s_i + \delta_k v_i = \gamma - \delta_k \ ,$$

or

$$\delta_k = \frac{\gamma - s_i}{1 + v_i} \ , \tag{5.17}$$

where $s_i = s_i(\boldsymbol{y}_\gamma)$. Only values $\delta_k \in]0, \gamma[$ are of interest. They correspond to pairs $(s_i > \gamma, \ v_i < -s_i/\gamma)$ or $(s_i < \gamma, \ v_i > -s_i/\gamma)$.

If this set is empty $(K = 0)$, then $\mathcal{A}(\gamma, \boldsymbol{y}_\gamma) = \mathcal{A}(0, \boldsymbol{y}_{(0)})$, we compute the primal solution by (5.8) and stop. Otherwise, the next threshold value $\widetilde{\gamma}$ is found by the following heuristics,

$$\widetilde{\gamma} = \begin{cases} \frac{1}{2}(\gamma - \delta_1) & \text{if } K = 1 \\ \min\{\frac{1}{2}\gamma, \ \gamma - \delta^*\} & \text{otherwise} \end{cases} \ . \tag{5.18a}$$

Here,

$$\delta^* = \tfrac{1}{2}(\delta_r + \delta_{r+1}) \tag{5.18b}$$

with

$$r = \max\{1,\ \min\{K-1,\ |\nu - m|\}\}\ , \tag{5.18c}$$

where $\nu = \#\mathcal{A}(\gamma, \boldsymbol{y}_\gamma)$. This heuristic aims at having m elements in the active set. Similarly, the *initial threshold value* is chosen as

$$\gamma = a_m\ , \tag{5.19}$$

where the vector $\boldsymbol{a}$ is found by sorting the elements of $|\boldsymbol{c}|$ in increasing order. This ensures that $\boldsymbol{s}(\boldsymbol{0}) = \boldsymbol{c}$ has at least m active elements.

Example 5.5. We have used the generator of Example 2.15 to get a well scaled problem $(\gamma_x = \gamma_y = \gamma_s = 1)$ of size $m = \#\mathcal{B} = 50,\ n = 100$.

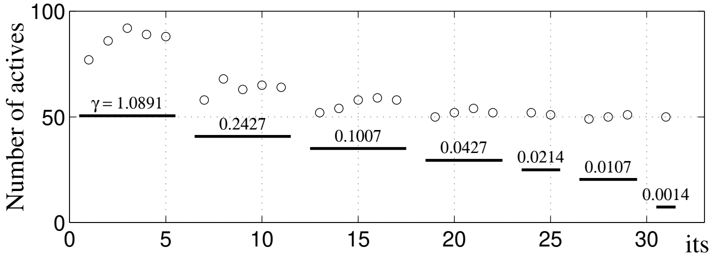


Figure 5.4. *Performance of Algorithm PP.* $m = \#\mathcal{B} = 50,\ n = 100$

The figure shows results from *Algorithm PP*, which found the solution to full accuracy after 32 iteration steps, where a step is counted as the solution of (5.20) below for determining either the Newton direction $\boldsymbol{h}$ or the vector $\boldsymbol{y}_{(0)}$. The iteration involved 6 threshold reductions. We give the number of elements in $\mathcal{A}$ as a function of iteration number and also show the γ-values (in logarithmic scale). The horizontal gap between different γ-values illustrates the computation of $\boldsymbol{y}_{(0)}$ by (5.5b); this is part of the threshold reduction scheme.

5.4. Implementation Aspects

The work involved with the piecewise polynomial algorithm is dominated by the solution of (5.9)[6)]

$$\boldsymbol{H}\boldsymbol{h} = \boldsymbol{g}\ , \quad \text{where} \quad \boldsymbol{H} = \boldsymbol{A}\boldsymbol{W}\boldsymbol{A}^T\ . \tag{5.20}$$

This looks like the basic problem in IPMs: solving systems of the form

$$(\boldsymbol{A}\boldsymbol{D}^2\boldsymbol{A}^T)\boldsymbol{h} = \boldsymbol{f}\ ,$$

cf. Section 3.5. In both cases the matrix is symmetric, and the "middle matrix" $\boldsymbol{W}$ or $\boldsymbol{D}^2$ changes from one iteration step to the next.

As discussed in Section 3.5, the "IPM matrix" is positive definite, and the solution can be found via Cholesky factorization. However, all elements in the diagonal of $\boldsymbol{D}$ change, so a refactorization is needed in each step.

In contrast, the matrix

$$\boldsymbol{H} = \boldsymbol{A}\boldsymbol{W}\boldsymbol{A}^T = \sum_{j \in \mathcal{A}} \boldsymbol{A}_{:,j}\boldsymbol{A}_{:,j}^T = \boldsymbol{A}_{\mathcal{A}}\boldsymbol{A}_{\mathcal{A}}^T \tag{5.21}$$

(with $\mathcal{A} = \mathcal{A}(\gamma, \boldsymbol{y})$) is only guaranteed to be positive semidefinite – it may be singular. This is the bad news. The good news is, that normally the change in the active set is relatively small. This means that relatively few of the elements in $\boldsymbol{W}$ change, and we can save work by *updating* the factorization of $\boldsymbol{H}$ instead of doing a refactorization. Also, the number $\nu = \#\mathcal{A}$ is typically about m, so that if a refactorization is performed, then it involves ν columns of $\boldsymbol{A}$, which can be much smaller than taking all the n columns as needed in an IPM.

Example 5.6. For the problem of the previous example we show below the number of elements that enter and leave the active set at each iteration step. The active set $\mathcal{A}(\gamma, \boldsymbol{y}_\gamma)$ for the first γ-value has 88 elements, and the first $\mathcal{A}(\gamma, \boldsymbol{y})$ after the reduction has 58 elements – 34 elements left the set and 4 entered it. Here, the algorithm uses a refactorization. In most of the other steps the number of changes is very small, and no more refactorizations are used.

6) Note, that also $\boldsymbol{y}_{(0)}$ is found by solving (5.5b) of this form, cf. Example 5.5.

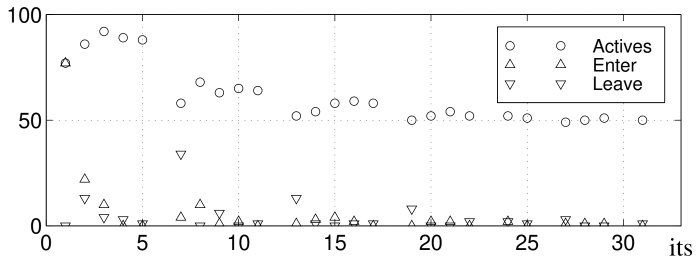


Figure 5.5. *Changes in active set.* $m = \#\mathcal{B} = 50$, $n = 100$

Factorization: We use a Marquardt-like modification of the matrix, and replace (5.20) by

$$\widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{h}} = \boldsymbol{g} , \quad \text{with} \quad \widetilde{\boldsymbol{H}} = \boldsymbol{AWA}^T + \mu^2 \boldsymbol{I} , \tag{5.22}$$

where μ is a small positive number. The matrix $\widetilde{\boldsymbol{H}}$ is symmetric and positive definite, and we can write it in the form

$$\widetilde{\boldsymbol{H}} = \begin{bmatrix} \mu \boldsymbol{I} & \boldsymbol{A}_{\mathcal{A}} \end{bmatrix} \begin{bmatrix} \mu \boldsymbol{I} \\ \boldsymbol{A}_{\mathcal{A}}^T \end{bmatrix} .$$

We introduce the QR factorization[7)]

$$\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix} = \boldsymbol{Q}^T \begin{bmatrix} \mu \boldsymbol{I} \\ \boldsymbol{A}_{\mathcal{A}}^T \end{bmatrix} , \tag{5.23a}$$

where $\boldsymbol{Q}$ is orthogonal and $\boldsymbol{R} \in \mathbb{R}^{m \times m}$ is upper triangular. It is seen, that this is equivalent with the triangular factorization

$$\widetilde{\boldsymbol{H}} = \boldsymbol{R}^T \boldsymbol{R} . \tag{5.23b}$$

Further, let $\mathcal{E}$ and $\mathcal{L}$ denote the set of indices that respectively enter and leave the active set. From the above it follows that the new $\widetilde{\widetilde{\boldsymbol{H}}}$ can be expressed as

$$\boldsymbol{R}^T \boldsymbol{R} := \boldsymbol{R}^T \boldsymbol{R} + \boldsymbol{A}_{\mathcal{E}} \boldsymbol{A}_{\mathcal{E}}^T - \boldsymbol{A}_{\mathcal{L}} \boldsymbol{A}_{\mathcal{L}}^T . \tag{5.24a}$$

[7)] cf. [32, Chapter 4] or [14, Section 5.2].

The change from $\boldsymbol{R}^T \boldsymbol{R}$ to $\overline{\boldsymbol{R}}^T \overline{\boldsymbol{R}} = \boldsymbol{R}^T \boldsymbol{R} + \boldsymbol{A}_{\mathcal{E}} \boldsymbol{A}_{\mathcal{E}}^T$ can be found from the *updating* formula

$$\begin{bmatrix} \overline{\boldsymbol{R}} \\ \boldsymbol{0} \end{bmatrix} = \overline{\boldsymbol{Q}}^T \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{A}_{\mathcal{E}}^T \end{bmatrix} , \tag{5.24b}$$

while $\widehat{\boldsymbol{R}}^T \widehat{\boldsymbol{R}} = \boldsymbol{R}^T \boldsymbol{R} - \boldsymbol{A}_{\mathcal{L}} \boldsymbol{A}_{\mathcal{L}}^T$ is equivalent with $\boldsymbol{R}^T \boldsymbol{R} = \widehat{\boldsymbol{R}}^T \widehat{\boldsymbol{R}} + \boldsymbol{A}_{\mathcal{L}} \boldsymbol{A}_{\mathcal{L}}^T$, or the *downdating* formula

$$\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{bmatrix} = \widetilde{\boldsymbol{Q}}^T \begin{bmatrix} \widehat{\boldsymbol{R}} \\ \boldsymbol{A}_{\mathcal{L}}^T \end{bmatrix} . \tag{5.24c}$$

A change involving both up- and downdates is performed by (5.24c) followed by (5.24b) with $\widehat{\boldsymbol{R}}$ playing the role of $\boldsymbol{R}$ in the right-hand side. This order is chosen because the downdating is inherently unstable, and a refactorization is performed if there is indication of dominating effect of rounding errors, or if the accumulated number of columns in $\{\boldsymbol{A}_{\mathcal{L}}\}$ exceeds m. The initial factorization and the refactorizations are performed by (5.24b) with $\boldsymbol{R} = \mu \boldsymbol{I}$, $\boldsymbol{A}_{\mathcal{E}} = \boldsymbol{A}_{\mathcal{A}}$. The orthogonal transformations are performed by Householder matrices, see [35] for details.

The parameter μ is chosen so that

1° If $\boldsymbol{H}$ has full rank, then the difference between $\boldsymbol{h}$ and $\widetilde{\boldsymbol{h}}$ is negligible.

2° If $\boldsymbol{H}$ is rank deficient, then the direction $\widetilde{\boldsymbol{h}}$ should ensure that the first change in active set during the line search (see Section 5.2) is that an index **enters** the set.

3° The accuracy of the factorization is not spoiled by a "downdate".

Partly based on experiments and supported by the analysis in Appendix A.6 we have found that the choice

$$\mu^2 = \varepsilon_M^{1.5} \cdot \|\boldsymbol{A}\|_1 \cdot \|\boldsymbol{A}\|_\infty \tag{5.25}$$

is a good compromise between these demands.

Newton iteration: With the modified matrix $\widetilde{\boldsymbol{H}}$ and Newton direction $\widetilde{\boldsymbol{h}}$ the relation (5.14) no longer holds, but must be replaced by

$$c_1(0) = \widetilde{\mathbf{h}}^T \boldsymbol{g}, \qquad c_2(0) = \boldsymbol{d}^T \boldsymbol{W}\boldsymbol{d}\ ,$$

where $\boldsymbol{d} = \boldsymbol{A}^T\widetilde{\mathbf{h}}$. If (5.20) is *consistent* (especially if $\boldsymbol{H}$ has full rank), then the relative difference[8] $(c_1(0) - c_2(0))\,/c_1(0) \sim \varepsilon_M$.

For an *inconsistent* problem – $\boldsymbol{H}$ is rank deficient and (5.20) has no solution – the analysis in Appendix A.6 shows that $\boldsymbol{W}\boldsymbol{d} \simeq \boldsymbol{0}$. This means that the actives stay active, and the kink points correspond to indices entering $\mathcal{A}$. Since $c_2(0)$ is very small, the function $\psi'(\alpha)$, (5.12), is positive for $0 \le \alpha \le \alpha_1$, and the next $\boldsymbol{H}$ corresponds to an augmented active set and will be closer to a full rank matrix.

Sparse matrix aspects: The discussion in Section 3.5 is valid also for the PP algorithm with two major differences,

1° We cannot use one initial ANALYSE since the sparsity pattern will change from one iteration step to the next.

2° The modified system (5.22) is suited for iterative solution with an occational refactorization acting as preconditioner.

The last claim is supported by [10], dealing with a sparse implementation of the PP algorithm outlined in Example 5.8 below.

Example 5.7. We have used *Algorithm PP* on the well scaled problems treated in Examples 3.9 and 4.9, and the resulting number of iterations are shown below.

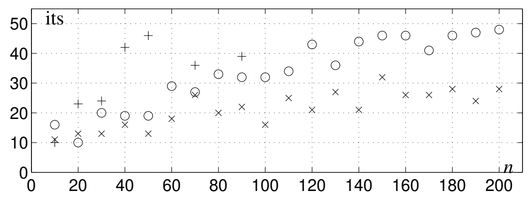


Figure 5.6. PP algorithm. Well scaled problems.
+ : $\#\mathcal{B} = 0.8m$, o : $\#\mathcal{B} = m$, x : $\#\mathcal{B} = 1.2m$

8) See NOTATION for the meaning of "$\sim$".

In all cases the solution was found with a relative error in the objective less than `8.9e-15`. Further, we can give the following statistics,

$\#\mathcal{B}$	max no. of refac.s	max no. of γ-reduc.s	no. of it.s
$0.8m$	33	12	$O(n)$
m	6	8	$O(n^{1/2})$
$1.2m$	3	4	$O(n^{1/3})$

Thus, this simple version of the algorithm seems to be unsuited for problems where there exist solutions with less than m nonzero components in $\mathbf{x}^*$.

We have also tried the algorithm on the descaled problems from Examples 3.9 and 4.9. Except for the smallest of these problems the algorithm needed more than n iteration steps (which is used as a "safety valve" in *Algorithm PP*). We are currently (December 1997) working on several possible improvements. One of them is presented in the next section.

Example 5.8. In [26] we present another piecewise polynomial algorithm, which is based on the formulation

$$(P) \qquad \max\left\{\ \boldsymbol{c}^T\mathbf{x}\ :\ \boldsymbol{A}\mathbf{x} = \boldsymbol{b},\ \ -\boldsymbol{e} \le \mathbf{x} \le \boldsymbol{e}\ \right\}$$

with the dual

$$(D) \qquad \min\left\{\ \boldsymbol{b}^T\boldsymbol{y} + \|\boldsymbol{s}(\boldsymbol{y})\|_1\ :\ \boldsymbol{y}\ \text{free}\ \right\}\ .$$

As in Section 5.1 we introduce a smooth approximation

$$(D_\gamma) \qquad \max\left\{\ \boldsymbol{b}^T\boldsymbol{y} + \textstyle\sum_{i=1}^n \sigma_\gamma(s_i(\boldsymbol{y}))\ :\ \boldsymbol{y}\ \text{free}\ \right\}\ ,$$

where[9]

$$\sigma_\gamma(s) = \begin{cases} \frac{1}{2\gamma}s^2 & \text{if } |s| \le \gamma \\ |s| - \frac{1}{2}\gamma & \text{if } |s| > \gamma \end{cases}\ .$$

In Figure 5.7 we show this function for two values of γ. As $\gamma \searrow 0$, the shape of the graph approaches the graph of $|s|$.

The active set contains the indices of the small components of the surplus vector $\boldsymbol{s}(\boldsymbol{y}) = \boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y}$: $i \in \mathcal{A}(\gamma, \boldsymbol{y}) \Leftrightarrow |s_i| \le \gamma$, and the Newton direction is found by a system of the form (5.20). Now, however, the right-hand side

9) This smoothing function was taken from robust data fitting, and we refer to it as the *"Huber smoothing function"*, [18].

depends also on the sign of the non active components of $s(\boldsymbol{y})$. We refer to [25], [26] and [31] for details, and to [27], [28] and [33] for a generalization of the ideas to solving *quadratic programming* problems.

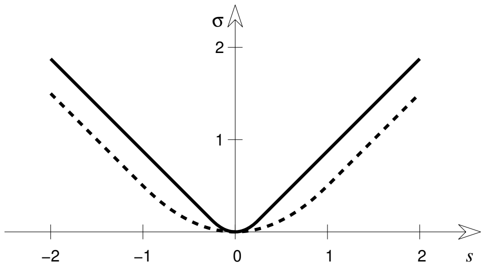


Figure 5.7. *Huber function for $\gamma = 1$ (dotted line) and $\gamma = \frac{1}{4}$ (full line)*

5.5. A Hybrid Method

Let $\mathcal{B}$ denote a subset of $\{1, \ldots, n\}$ and let $\mathcal{S}(\mathcal{B})$ denote the part of $\mathbb{R}^m$ containing the vectors $\boldsymbol{y}$, for which $\mathcal{A}(\gamma, \boldsymbol{y}) = \mathcal{B}$. Figure 5.2 indicates, that as $\gamma \searrow 0$, this subregion gets close to a vertex of the polytope from Section 4.1 plus part of the exterior of this polytope (or $\mathcal{S}(\mathcal{B})$ may be empty). Further, the definition (5.1c) of ρ_γ makes this exterior increasingly "expensive". This has the effect that $\|\boldsymbol{W s}\| \sim \boldsymbol{c}$, and (5.9) shows that the Newton direction is determined by a vector $\boldsymbol{h}$ with $\|\boldsymbol{h}\| \sim \gamma$.

Therefore, as $\gamma \searrow 0$, the behaviour of the PP algorithm will be similar to the Simplex algorithm, but less robust because the small $\boldsymbol{h}$ will need a large α^* to bring us to another vertex, and this implies, that a small error in the direction defined by $\boldsymbol{h}$ can have serious effects. This is (at least partly) the explanation of the poor results reported in Example 5.7.

One remedy is to use the PP method only for reasonably large γ-values, to get us close to the optimal partition of (P), and then shift to the Simplex algorithm.[10)] Such a hybrid method can have the form given in *Algorithm PP_Simplex* below.

10) In the notation of Section 4.4 we use the PP method for Phase 1. For well scaled, non degenerate problems the solution is often found before the start of Phase 2.

Algorithm PP_Simplex
Choose starting value γ; $r := 0$;
repeat
  Compute $\boldsymbol{y}_\gamma$ by *Algorithm PPNewton*, p 93
  if (final set) **then** Compute $\mathbf{x}$ by (5.8); STOP
    else Reduce γ; $r := r+1$
until $r = R$
Get $\mathcal{B}$, $\boldsymbol{B}$ and $\mathbf{x}_\mathcal{B}$
Find $\mathbf{x}^*$ by *Algorithm* SIMPLEX, page 68

Experiments indicate that a shift after $R = 2$ γ-reductions generally gives good performance, and this is the value used in the examples.

The transition to a starting point for Simplex from the last $\boldsymbol{y}_\gamma$ with corresponding $\mathcal{A} = \mathcal{A}(\gamma, \boldsymbol{y}_\gamma)$ and $\widetilde{\mathbf{x}}$ computed by (5.8) is made as follows,[11)]

if $\#\mathcal{A} < m$ **then**
  Augment $\mathcal{A}$ so that $\#\mathcal{A} = m$
elseif $\#\mathcal{A} > m$ **then**
  Reduce $\mathcal{A}$ until $\#\mathcal{A} = m$
  with simultaneous updating of $\widetilde{\mathbf{x}}$
end
$\mathcal{B} := \mathcal{A}$; $\boldsymbol{B} := \mathbf{A}_\mathcal{A}$; $\mathbf{x}_\mathcal{B} = \widetilde{\mathbf{x}}_\mathcal{A}$

The adjustments are performed with the goal

$$\min\{\boldsymbol{c}_\mathcal{B}^T \mathbf{x}_\mathcal{B} \ : \ \mathbf{A}_\mathcal{B} \mathbf{x}_\mathcal{B} = \boldsymbol{b}, \ \mathbf{x}_\mathcal{B} \geq \boldsymbol{0}\}$$

in mind.

In the case $\nu = \#\mathcal{A}(\gamma, \boldsymbol{y}_\gamma) < m$ we simply augment $\mathcal{A}$ with $\{j_1, \ldots j_{m-\nu}\}$, the indices of the $m-\nu$ smallest elements in the non-active part of $\boldsymbol{c}$. Note, that components $j_1, \ldots j_{m-\nu}$ in $\widetilde{\mathbf{x}}$ are all zero, and therefore the condition $\mathbf{A}_\mathcal{A} \widetilde{\mathbf{x}}_\mathcal{A} = \boldsymbol{b}$ is also satisfied with the augmented $\mathcal{A}$.

The case $\#\mathcal{A}(\gamma, \boldsymbol{y}_\gamma) > m$ is treated iteratively: If any component of $\widetilde{\mathbf{x}}_\mathcal{A}$ is zero, reduce $\mathcal{A}$ by removing this index. Otherwise, consider the problem

$$(T) \qquad \min\left\{ \ \boldsymbol{c}_\mathcal{A}^T(\widetilde{\mathbf{x}}_\mathcal{A} + t\mathbf{h}) \ : \ \mathbf{A}_\mathcal{A}(\widetilde{\mathbf{x}}_\mathcal{A} + t\mathbf{h}) = \boldsymbol{b}, \ \widetilde{\mathbf{x}}_\mathcal{A} + t\mathbf{h} \geq \boldsymbol{0} \ \right\}$$

11) In this presentation we assume that a matrix $\mathbf{A}_\mathcal{A}$ has full rank, $\mathrm{rank}(\mathbf{A}_\mathcal{A}) = \min\{m, \#\mathcal{A}\}$. See [35] for a discussion of what to do when this is not satisfied.

for $t > 0$. The current $\widehat{\mathbf{x}}$ satisfies the constraints $\boldsymbol{A}_{\mathcal{A}} \widetilde{\mathbf{x}}_{\mathcal{A}} = \boldsymbol{b}$, $\widetilde{\mathbf{x}}_{\mathcal{A}} > \boldsymbol{0}$, so $\boldsymbol{h}$ should satisfy

$$\boldsymbol{c}_{\mathcal{A}}^T \boldsymbol{h} \le 0, \qquad \widetilde{\mathbf{x}}_{\mathcal{A}} \boldsymbol{h} = \boldsymbol{0} \ .$$

These conditions are met if we compute $\boldsymbol{h}$ as the projection of $-\boldsymbol{c}_{\mathcal{A}}$ onto the null space of $\boldsymbol{A}_{\mathcal{A}}$, cf. Section 2.5. Applying (2.27c) to the present problem we see that

$$\boldsymbol{h} = \boldsymbol{A}_{\mathcal{A}}^T \boldsymbol{u} - \boldsymbol{c}_{\mathcal{A}}, \quad \text{where} \quad (\boldsymbol{A}_{\mathcal{A}} \boldsymbol{A}_{\mathcal{A}}^T)\boldsymbol{u} = \boldsymbol{A}_{\mathcal{A}} \boldsymbol{c}_{\mathcal{A}} \ .$$

This system has the form $(\boldsymbol{A}\boldsymbol{W}\boldsymbol{A}^T)\boldsymbol{u} = \boldsymbol{A}\boldsymbol{W}\boldsymbol{c}$, and is solved as described in Section 5.4. Then, we determine t^* as the smallest value for which a component of $\widetilde{\mathbf{x}}_{\mathcal{A}} + t\boldsymbol{h}$ is equal to zero, let $\widetilde{\mathbf{x}}_{\mathcal{A}} := \widetilde{\mathbf{x}}_{\mathcal{A}} + t^*\boldsymbol{h}$, and can remove at least one element from $\mathcal{A}$. Therefore, this iteration stops after at most $\#\mathcal{A}(\gamma, \boldsymbol{y}_\gamma) - m$ steps. Each of these is similar to a Simplex step, and in the examples we count them as Simplex steps.

Example 5.9. We have used *Algorithm PP_Simplex* on the problems in Examples 3.9 and 4.9. By comparison we see that the number of PP steps is about the same as the number of IPM steps – and much cheaper. For the well scaled problems the number of Simplex steps is about 1/10 of the steps needed by a pure Simplex method. For the descaled problems this ratio is about 1/3.

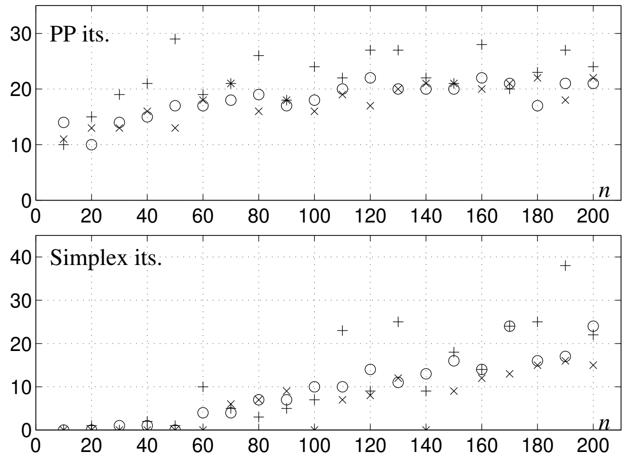


Figure 5.8a. *Algorithm PP_Simplex. Well scaled problems.*
+ : $\#\mathcal{B} = 0.8m$, o : $\#\mathcal{B} = m$, x : $\#\mathcal{B} = 1.2m$

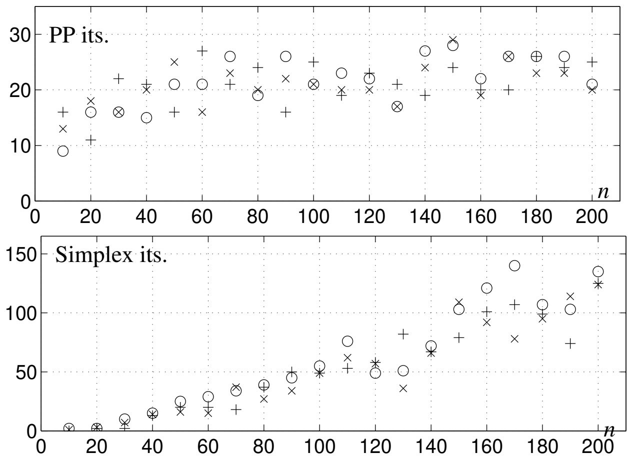


Figure 5.8b. *Algorithm PP_Simplex. Descaled problems.*
+ : $\#\mathcal{B} = 0.8m$, o : $\#\mathcal{B} = m$, x : $\#\mathcal{B} = 1.2m$

Example 5.10. The Simplex method is notorious for the possibility that the number of steps may grow exponentially with the size of the problem. In practice the "normal" behaviour seems to be that the number of steps grows slightly faster than n – cf. Example 4.9, but in pathological cases the exponential growth occurs. A famous example, due to Klee and Minty [21], is given by

$$(E) \qquad \min\left\{ \ \boldsymbol{c}^T\mathbf{x} \ : \ \boldsymbol{A}\mathbf{x} = \boldsymbol{b}, \ \mathbf{x} \ge \boldsymbol{0} \ \right\}$$

with $n = 2m$ and

$$a_{ij} = \begin{cases} 2\tau^{i-j} & , \ j = 1, \ldots, i-1 \\ 1 & , \ j = i \text{ and } j = m+i \\ 0 & , \ \text{otherwise} \end{cases} , \tag{5.26a}$$

$$b_i = \tau^{2(i-1)} , \quad i = 1, \ldots, m , \tag{5.26b}$$

$$c_j = \begin{cases} -\tau^{m-j} & , \ j = 1, \ldots, m \\ 0 & , \ j = m+1, \ldots, n \end{cases} , \tag{5.26c}$$

where $\tau > 2$ is a chosen value. The optimal partitioning is $\mathcal{B} = \{m, \ m+1, \ \ldots, \ n-1\}$, and

$$\widetilde{\mathbf{x}}_{\mathcal{B}} = \begin{bmatrix} \tau^{2(m-1)} \\ 1 \\ \vdots \\ \tau^{2(m-2)} \end{bmatrix}, \quad \widetilde{\boldsymbol{y}} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}, \quad \widetilde{\boldsymbol{s}}_{\mathcal{C}} = \begin{bmatrix} \tau^{m-1} \\ \vdots \\ \tau \\ 1 \end{bmatrix}.$$

With the Simplex starting point $\mathcal{B} = \{m+1, \ldots, 2m\}$, $\boldsymbol{B} = \boldsymbol{I}$, $\mathbf{x}_{\mathcal{B}} = \boldsymbol{b}$, and with the pivoting rule 2°, p. 73, the number of Simplex steps is 2^m-1. This is demonstrated in the following table, where we also give results from *Algorithm IPM3* (p. 49) and *Algorithm PP* (p. 90). In *IPM3* the maximum mumber of steps is set to 200 (and marked by ∞ if this limit is reached), and in PP we return the result after at most $R=3$ reductions of γ. The "error" is the relative duality gap, $|\boldsymbol{c}^T\mathbf{x} - \boldsymbol{b}^T\boldsymbol{y}|/|\boldsymbol{c}^T\mathbf{x}|$.

m	$\kappa(\boldsymbol{A})$	SIMPLEX		*IPM3*		*PP*		
		its	error	its	error	its	redu	error
2	8.12e+00	3	0	∞	7.99e-09	8	2	0
3	3.40e+01	7	0	14	2.88e-10	15	3	0
4	1.36e+02	15	0	21	7.23e-09	21	3	0
5	5.46e+02	31	0	36	1.61e-14	26	3	0
6	2.18e+03	63	0	∞	3.89e-06	10	0	9.31e-09
7	8.74e+03	127	0	∞	8.76e-10	12	0	1.46e-10
8	3.50e+04	255	0	54	1.63e-12	15	0	2.19e-12
9	1.40e+05	511	0	∞	3.86e-10	11	0	4.57e-13

Results for problems generated by (5.26) with $\tau = 4$.

The failing of *IPM3* to stop is primarily caused by the severe descaling of the solution, and show that our implementation needs some improvement.

As mentioned, this is a pathological problem, but it demonstrates that even small size LO problems can be hard to solve.

APPENDIX A. SELECTED PROOFS

A.1. Proof of Theorem 2.2

We consider the primal problem in canonical form,

$$\begin{aligned} &(P) \qquad \min\{\ \boldsymbol{c}^T\mathbf{x}\ :\ \boldsymbol{A}\mathbf{x} \geq \boldsymbol{b},\ \mathbf{x} \geq \boldsymbol{0}\ \}\ , \\ &(D) \qquad \max\{\ \boldsymbol{b}^T\boldsymbol{y}\ :\ \boldsymbol{A}^T\boldsymbol{y} \leq \boldsymbol{c},\ \boldsymbol{y} \geq \boldsymbol{0}\ \}\ , \end{aligned} \tag{A.1}$$

and introduce the homogeneous problem

$$(E) \qquad \min\{\ \boldsymbol{0}^T\boldsymbol{z}\ :\ \boldsymbol{M}\boldsymbol{z} \geq \boldsymbol{0},\ \boldsymbol{z} \geq \boldsymbol{0}\ \}\ , \tag{A.2a}$$

where the $(m+n+1)$-vector $\boldsymbol{z}$ and the $(m+n+1)^2$-matrix $\boldsymbol{M}$ are given by

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{y} \\ \mathbf{x} \\ \lambda \end{bmatrix}, \qquad \boldsymbol{M} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{A} & -\boldsymbol{b} \\ -\boldsymbol{A}^T & \boldsymbol{0} & \boldsymbol{c} \\ \boldsymbol{b}^T & \boldsymbol{c}^T & 0 \end{bmatrix}. \tag{A.2b}$$

The matrix is skew-symmetric, and problem (E) is of the form (2.23). This implies the existence of an optimal solution $\widetilde{\boldsymbol{z}}$, that satisfies the strict complementarity properties

$$\widetilde{\boldsymbol{z}} \odot \boldsymbol{s}(\widetilde{\boldsymbol{z}}) = \boldsymbol{0}\ , \qquad \widetilde{\boldsymbol{z}} + \boldsymbol{s}(\widetilde{\boldsymbol{z}}) > \boldsymbol{0}\ . \tag{A.3}$$

The surplus vector $\boldsymbol{s}(\boldsymbol{z})$ for (E) splits into

$$\begin{aligned} \boldsymbol{s}(\mathbf{x}) &= \boldsymbol{A}\mathbf{x} - \lambda\boldsymbol{b}\ , \\ \boldsymbol{s}(\boldsymbol{y}) &= \lambda\boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{y}\ , \\ s(\lambda) &= \boldsymbol{b}^T\boldsymbol{y} - \boldsymbol{c}^T\mathbf{x}\ . \end{aligned} \tag{A.4}$$

The feasible set of (E) is $\mathcal{E} = \{\boldsymbol{z} \in I\!R^{m+n+1}\,|\,\boldsymbol{z}, \boldsymbol{M}\boldsymbol{z} \geq \boldsymbol{0}\}$. This implies, that if $\widetilde{\boldsymbol{z}}$ is an optimal solution, then $\alpha\widetilde{\boldsymbol{z}}$ is also optimal for any choice of $\alpha \geq 0$, and this leads to two alternatives:

Case 1. $\widetilde{\lambda} \neq 0$: We can normalize $\widetilde{\mathbf{z}}$ so that $\widetilde{\lambda} = 1$, and the above splitting of $\boldsymbol{s}(\widetilde{\mathbf{z}})$ gives $\boldsymbol{s}(\widetilde{\mathbf{x}})$ and $\boldsymbol{s}(\widetilde{\boldsymbol{y}})$ identical with (2.15). (A.3) further implies

$$\begin{aligned} \widetilde{\boldsymbol{y}} \odot \boldsymbol{s}(\widetilde{\mathbf{x}}) &= \boldsymbol{0} , & \widetilde{\boldsymbol{y}} + \boldsymbol{s}(\widetilde{\mathbf{x}}) &> \boldsymbol{0} , \\ \widetilde{\mathbf{x}} \odot \boldsymbol{s}(\widetilde{\boldsymbol{y}}) &= \boldsymbol{0} , & \widetilde{\mathbf{x}} + \boldsymbol{s}(\widetilde{\boldsymbol{y}}) &> \boldsymbol{0} , \\ 1 \cdot (\boldsymbol{b}^T\widetilde{\boldsymbol{y}} - \boldsymbol{c}^T\widetilde{\mathbf{x}}) &= 0 , & 1 + (\boldsymbol{b}^T\widetilde{\boldsymbol{y}} - \boldsymbol{c}^T\widetilde{\mathbf{x}}) &> 0 . \end{aligned}$$

Thus, the case $\widetilde{\lambda} \neq 0$ corresponds to alternative 1° in the theorem.

Case 2. $\widetilde{\lambda} = 0$: Now, (A.3) – (A.4) imply that $\boldsymbol{A}\widetilde{\mathbf{x}} \geq \boldsymbol{0}$, $\boldsymbol{A}^T\widetilde{\boldsymbol{y}} \leq \boldsymbol{0}$ and $\boldsymbol{b}^T\widetilde{\boldsymbol{y}} - \boldsymbol{c}^T\widetilde{\mathbf{x}} > 0$. The last relation further implies that $\boldsymbol{b}^T\widetilde{\boldsymbol{y}} > 0$, $\boldsymbol{c}^T\widetilde{\mathbf{x}} < 0$, or both $\boldsymbol{b}^T\widetilde{\boldsymbol{y}} > 0$ and $\boldsymbol{c}^T\widetilde{\mathbf{x}} < 0$.

Case 2a. $\boldsymbol{b}^T\widetilde{\boldsymbol{y}} > 0$: The feasible set $\mathcal{P}$ is empty, since otherwise we would have the contradiction

$$0 \geq \mathbf{x}^T(\boldsymbol{A}^T\widetilde{\boldsymbol{y}}) = (\boldsymbol{A}\mathbf{x})^T\widetilde{\boldsymbol{y}} \geq \boldsymbol{b}^T\widetilde{\boldsymbol{y}} > 0 .$$

Moreover, if (D) is feasible, and $\boldsymbol{y} \in \mathcal{D}$, then the relation $\boldsymbol{A}^T\widetilde{\boldsymbol{y}} \leq \boldsymbol{0}$ implies that $\boldsymbol{y} + \alpha\widetilde{\boldsymbol{y}}$ is feasible for all positive values of α. The objective value

$$\boldsymbol{b}^T(\boldsymbol{y} + \alpha\widetilde{\boldsymbol{y}}) = \boldsymbol{b}^T\boldsymbol{y} + \alpha\boldsymbol{b}^T\widetilde{\boldsymbol{y}}$$

can be arbitrarily large, i.e. (D) is unbounded.

Case 2b. $\boldsymbol{c}^T\widetilde{\mathbf{x}} < 0$: Similar arguments show that (D) is infeasible and (P) is either infeasible or unbounded.

Thus, we have shown that $\widetilde{\lambda} = 0$ corresponds to alternatives 2° and 3° in the theorem. □

A.2. Proof of Lemma 2.3

A solution to Problem 1A is the feasible set of the LO problem

$$(P_1) \qquad \min\{\ \boldsymbol{0}^T\mathbf{x}\ :\ \boldsymbol{A}\mathbf{x} \geq \boldsymbol{b},\ \mathbf{x} \geq \boldsymbol{0}\ \}$$

with the dual

$$(D_1) \qquad \max\{\ \boldsymbol{b}^T\boldsymbol{y}\ :\ \boldsymbol{A}^T\boldsymbol{y} \leq \boldsymbol{0},\ \boldsymbol{y} \geq \boldsymbol{0}\ \} .$$

Using the notation and results from the proof of Theorem 2.2 we find

$\widetilde{\lambda} \neq 0$: $\mathcal{P}_1$ is not empty, i.e. Problem 1A has solutions. Also (D_1) is feasible, and $\boldsymbol{b}^T\boldsymbol{y} \leq \boldsymbol{0}^T\widetilde{\mathbf{x}} = 0$ for any $\boldsymbol{y} \in \mathcal{D}_1$. This means that the constraint $\boldsymbol{b}^T\boldsymbol{y} > 0$ in 1B cannot be satisfied.

$\widetilde{\lambda} = 0$: As in the proof of Theorem 2.2 we see that $\boldsymbol{b}^T\widetilde{\boldsymbol{y}} - \boldsymbol{0}^T\widetilde{\mathbf{x}} = \boldsymbol{b}^T\widetilde{\boldsymbol{y}} > 0$, and $\widetilde{\boldsymbol{y}}$ solves Problem 1B, while $\mathcal{P}_1$ is empty; i.e. 1A has no solution.

In *case 2* we consider

$$\begin{aligned} (P_2) \qquad & \min\{\ \boldsymbol{0}^T\mathbf{x}\ :\ \boldsymbol{A}\mathbf{x} \leq \boldsymbol{b}\ \} , \\ (D_2) \qquad & \max\{\ \boldsymbol{b}^T\boldsymbol{y}\ :\ \boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{0},\ \boldsymbol{y} \leq \boldsymbol{0}\ \} . \end{aligned}$$

As in Case 1 we see that either 2A has a solution, or 2B is solved by $\boldsymbol{y} = -\widetilde{\boldsymbol{y}}$, satisfying $\boldsymbol{b}^T\widetilde{\boldsymbol{y}} > 0$, $\boldsymbol{A}^T\widetilde{\boldsymbol{y}} = \boldsymbol{0}$, $\widetilde{\boldsymbol{y}} \leq \boldsymbol{0}$.

In *case 3* we consider

$$\begin{aligned} (P_3) \qquad & \min\{\ \boldsymbol{b}^T\mathbf{x}\ :\ \boldsymbol{A}\mathbf{x} \geq \boldsymbol{0}\ \} , \\ (D_3) \qquad & \max\{\ \boldsymbol{0}^T\boldsymbol{y}\ :\ \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b},\ \ \boldsymbol{y} \geq \boldsymbol{0}\ \} . \end{aligned}$$

If both problems are feasible, then $\boldsymbol{b}^T\mathbf{x} \geq \boldsymbol{0}^T\widetilde{\boldsymbol{y}} = 0$ for any $\mathbf{x} \in \mathcal{P}_3$. Problem 3B but not 3A has a solution. Otherwise, $\boldsymbol{b}^T\widetilde{\mathbf{x}} < 0$, $\mathcal{D}_3$ is empty, and $\widetilde{\mathbf{x}}$ solves 3A.

Finally, in *case 4* we consider

$$\begin{aligned} (P_4) \qquad & \min\{\ \boldsymbol{b}^T\mathbf{x}\ :\ \boldsymbol{A}\mathbf{x} = \boldsymbol{0}\ \} , \\ (D_4) \qquad & \max\{\ \boldsymbol{0}^T\boldsymbol{y}\ :\ \boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{b}\ \} , \end{aligned}$$

and the discussion is similar to case 3. □

A.3. Proof of Theorem 3.2

We already discussed the orthogonality of $\boldsymbol{d}_x$ and $\boldsymbol{d}_s$ (before (3.6)). Similarly, $\boldsymbol{h}_x^T\boldsymbol{h}_s = 0$ follows from $\boldsymbol{h}_x \in \mathcal{N}(\boldsymbol{A})$ and $\boldsymbol{h}_s \in \mathcal{R}(\boldsymbol{A})$.

To prove 2° we introduce

$$\mathbf{x}(\alpha) = \mathbf{x} + \alpha\boldsymbol{h}_x , \qquad \boldsymbol{s}(\alpha) = \boldsymbol{s} + \alpha\boldsymbol{h}_s \tag{A.5a}$$

and

$$\begin{aligned} \boldsymbol{z}(\alpha) &= \mathbf{x}(\alpha)\odot\boldsymbol{s}(\alpha) \\ &= \boldsymbol{x}\odot\boldsymbol{s} + \alpha(\boldsymbol{x}\odot\boldsymbol{h}_s + \boldsymbol{s}\odot\boldsymbol{h}_x) + \alpha^2(\boldsymbol{h}_x\odot\boldsymbol{h}_s) \\ &= \boldsymbol{x}\odot\boldsymbol{s} + \alpha(\mu\boldsymbol{e} - \mathbf{x}\odot\boldsymbol{s}) + \alpha^2(\boldsymbol{h}_x\odot\boldsymbol{h}_s)\ . \end{aligned} \tag{A.5b}$$

Here, we have used the last equation in (3.3). We consider this vector $\boldsymbol{z}(\alpha)$ in the range $0\le\alpha\le 1$, noting that $\boldsymbol{z}(0)=\boldsymbol{x}\odot\boldsymbol{s} > \boldsymbol{0}$ and that $\boldsymbol{z}(1) = (\boldsymbol{x}+\boldsymbol{h}_x)\odot(\boldsymbol{s}+\boldsymbol{h}_s)$ also is a positive vector if the step is feasible.

Now, if the step is feasible, then

$$\boldsymbol{0} < \boldsymbol{z}(1) = \mu\boldsymbol{e} + \boldsymbol{h}_x\odot\boldsymbol{h}_s\ ,$$

i.e. we have shown the 'only if' part. For the 'if' part we know that $\boldsymbol{z}(1) = \mu\boldsymbol{e}+\boldsymbol{h}_x\odot\boldsymbol{h}_s > \boldsymbol{0}$, and combining this with (A.5) we see that

$$\begin{aligned} \boldsymbol{z}(\alpha) &\ge \mathbf{x}\odot\boldsymbol{s} + \alpha(\mu\boldsymbol{e} - \mathbf{x}\odot\boldsymbol{s}) - \alpha^2(\mu\boldsymbol{e}) \\ &= (1-\alpha)(\mathbf{x}\odot\boldsymbol{s} + \alpha\mu\boldsymbol{e}) > \boldsymbol{0} \quad \text{for} \quad 0\le\alpha<1\ . \end{aligned} \tag{A.6}$$

Both $\mathbf{x}(\alpha)$ and $\boldsymbol{s}(\alpha)$ vary continuously with α, and (A.6) shows that all their components stay positive as α increases from 0 to 1. Also $\boldsymbol{z}(1)>\boldsymbol{0}$, implying that $\mathbf{x}(1), \boldsymbol{s}(1) > \boldsymbol{0}$, i.e. the step is feasible. This completes the proof of 2°.

Next, $(\boldsymbol{x}+\boldsymbol{h}_x)^T(\boldsymbol{s}+\boldsymbol{h}_s) = \mathbf{e}^T\boldsymbol{z}(1) = \mathbf{e}^T(\mu\mathbf{e}) + \boldsymbol{h}_x^T\boldsymbol{h}_s = n\mu$. Here, we have used the ortogonality of $\boldsymbol{h}_x$ and $\boldsymbol{h}_s$.

Finally, for 4° and 5°, see [39, Section 7.4]. □

A.4. Proof of Theorem 3.5

As in the proof of Theorem 3.2 we see that

$$(\boldsymbol{h}_x^a)^T\boldsymbol{h}_s^a = (\boldsymbol{h}_x^c)^T\boldsymbol{h}_s^c = 0\ . \tag{A.7}$$

Further, for $g = a, c, N$ we generalize (A.5) to

$$\mathbf{x}^g(\alpha) = \mathbf{x} + \alpha\boldsymbol{h}_x^g\ , \qquad \boldsymbol{s}^g(\alpha) = \boldsymbol{s} + \alpha\boldsymbol{h}_s^g\ , \tag{A.8a}$$

$$\begin{aligned} \mathbf{z}^g(\alpha) &= \mathbf{x}^g(\alpha)\odot\boldsymbol{s}^g(\alpha) \\ &= \mathbf{x}\odot\boldsymbol{s} + \alpha(\mathbf{x}\odot\boldsymbol{h}_s^g + \boldsymbol{s}\odot\boldsymbol{h}_x^g) + \alpha^2(\boldsymbol{h}_x^g\odot\boldsymbol{h}_s^g)\ . \end{aligned} \tag{A.8b}$$

For the affine-scaling and the centering directions we have

$$\mathbf{x}\odot\boldsymbol{h}_s^a + \boldsymbol{s}\odot\boldsymbol{h}_x^a = -\mathbf{x}\odot\boldsymbol{s}\ , \quad \mathbf{x}\odot\boldsymbol{h}_s^c + \boldsymbol{s}\odot\boldsymbol{h}_x^c = \mu\boldsymbol{e}\ ,$$

respectively, so that

$$\mathbf{z}^a(\alpha) = (1-\alpha)\mathbf{x}\odot\boldsymbol{s} + \alpha^2(\boldsymbol{h}_x^a\odot\boldsymbol{h}_s^a)\ , \tag{A.9a}$$

$$\mathbf{z}^c(\alpha) = \mathbf{x}\odot\boldsymbol{s} + \alpha\mu\boldsymbol{e} + \alpha^2(\boldsymbol{h}_x^c\odot\boldsymbol{h}_s^c)\ . \tag{A.9b}$$

Now, using the orthogonalities (A.7) we find

$$(\mathbf{x}^a(\alpha))^T\,\boldsymbol{s}^a(\alpha) = \mathbf{e}^T\mathbf{z}^a(\alpha) = (1-\alpha)\mathbf{x}^T\boldsymbol{s} = (1-\alpha)n\mu\ ,$$

$$(\mathbf{x}^c(\alpha))^T\,\boldsymbol{s}^c(\alpha) = \mathbf{e}^T\mathbf{z}^c(\alpha) = \mathbf{x}^T\boldsymbol{s} + \alpha\mu n = (1+\alpha)n\mu\ ,$$

and we have proven points 1° and 2°. Point 3° follows by simple addition.

For the proximity we consider the function obtained from (3.7),

$$\begin{aligned} \varphi(\boldsymbol{h},\alpha) &= 4\delta^2(\mathbf{x}+\alpha\boldsymbol{h}_x, \boldsymbol{s}+\alpha\boldsymbol{h}_s; \mu) \\ &= \|\mathbf{u} - \mathbf{u}^{-1}\|_2^2 = \mathbf{e}^T\left(\mathbf{u}^2 + \mathbf{u}^{-2} - 2\boldsymbol{e}\right)\ , \end{aligned} \tag{A.10a}$$

where $\mathbf{u}$ is given by (3.4),

$$\mathbf{u}^2 = \mu^{-1}(\mathbf{x}+\alpha\boldsymbol{h}_x)\odot(\boldsymbol{s}+\alpha\boldsymbol{h}_s)\ . \tag{A.10b}$$

We let

$$\mathbf{u}_0^2 \equiv \mu^{-1}\mathbf{x}\odot\boldsymbol{s}\ , \quad \boldsymbol{U}_0 = \mathrm{diag}(\mathbf{u}_0)\ , \tag{A.11a}$$

and see that

$$\mathbf{e}^T\mathbf{u}_0^2 = \mu^{-1}\mathbf{x}^T\boldsymbol{s} = n \tag{A.11b}$$

and

$$\begin{aligned} \varphi(\boldsymbol{h},0) &= 4\delta_0^2 \equiv 4\delta^2(\mathbf{x},\boldsymbol{s};\mu) \\ &= \mathbf{e}^T\left(\mathbf{u}_0^2 + \mathbf{u}_0^{-2}\right) - 2n = \mathbf{e}^T\mathbf{u}_0^{-2} - n\ . \end{aligned} \tag{A.11c}$$

For small values of α Taylor's theorem shows that

$$\varphi(\boldsymbol{h},\alpha) \simeq 4\delta_0^2 + \alpha\varphi'(\boldsymbol{h},0)\ , \tag{A.12a}$$

where

$$\begin{aligned}
\varphi'(\boldsymbol{h},0) &= \boldsymbol{h}_x^T(\nabla_x\varphi) + \boldsymbol{h}_s^T(\nabla_s\varphi)\\
&= \boldsymbol{h}_x^T(\boldsymbol{I}-\boldsymbol{U}_0^{-4})\nabla_x\mathbf{u}_0^2 + \boldsymbol{h}_s^T(\boldsymbol{I}-\boldsymbol{U}_0^{-4})\nabla_s\mathbf{u}_0^2\\
&= \mu^{-1}\left(\boldsymbol{h}_x^T(\boldsymbol{I}-\boldsymbol{U}_0^{-4})\boldsymbol{s} + \boldsymbol{h}_s^T(\boldsymbol{I}-\boldsymbol{U}_0^{-4})\mathbf{x}\right)\\
&= \mu^{-1}\boldsymbol{e}^T(\boldsymbol{I}-\boldsymbol{U}_0^{-4})(\boldsymbol{S}\boldsymbol{h}_x + \boldsymbol{X}\boldsymbol{h}_s)\ . \qquad \text{(A.12b)}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\varphi'(\boldsymbol{h}^a,0) &= \mu^{-1}\boldsymbol{e}^T(\boldsymbol{I}-\boldsymbol{U}_0^{-4})(-\mu\mathbf{u}_0^2)\\
&= \boldsymbol{e}^T(\mathbf{u}_0^{-2}-\mathbf{u}_0^2) = \boldsymbol{e}^T\mathbf{u}_0^{-2} - n = 4\delta_0^2.
\end{aligned}$$

Combining this with (A.12a) and (A.10a) we get point 4°.

For the centering direction we find

$$\begin{aligned}
\varphi'(\boldsymbol{h}^c,0) &= \mu^{-1}\boldsymbol{e}^T(\boldsymbol{I}-\boldsymbol{U}_0^{-4})(\mu\boldsymbol{e})\\
&= \boldsymbol{e}^T(\boldsymbol{e}-\mathbf{u}_0^{-4}) = \boldsymbol{e}^T\left(\boldsymbol{e}-(\mathbf{u}_0^{-2}-\boldsymbol{e}+\boldsymbol{e})^2\right)\\
&= \boldsymbol{e}^T\left(-2(\mathbf{u}_0^{-2}-\boldsymbol{e})-(\mathbf{u}_0^{-2}-\boldsymbol{e})^2\right)\\
&= -8\delta_0^2 - \boldsymbol{e}^T(\mathbf{u}_0^{-2}-\boldsymbol{e})^2 < -8\delta_0^2\ ,
\end{aligned}$$

since all elements in both $\boldsymbol{e}$ and $(\mathbf{u}_0^{-2}-\boldsymbol{e})^2$ are positive. Point 5° follows immediately. Finally,

$$\begin{aligned}
\varphi'(\boldsymbol{h}^N,0) &= \boldsymbol{e}^T(\mathbf{u}_0^{-2}-\mathbf{u}_0^2\ +\ \boldsymbol{e}-\mathbf{u}_0^{-4})\\
&= \boldsymbol{e}^T(\mathbf{u}_0^{-2}-\mathbf{u}_0^{-4}) - n + n\\
&= -\boldsymbol{e}^T\left(\mathbf{u}_0^{-2}-\boldsymbol{e}+(\mathbf{u}_0^{-2}-\boldsymbol{e})^2\right)\\
&= = -4\delta_0^2 - \boldsymbol{e}^T(\mathbf{u}_0^{-2}-\boldsymbol{e})^2 < -4\delta_0^2\ ,
\end{aligned}$$

and 6° follows. □

A.5. Proof of (4.14)

Consider the general step in transforming the full rank matrix $\boldsymbol{B}$ to an upper triangular matrix by means of row and column interchanges and elementary row operations

$$\begin{aligned}
\boldsymbol{P}_s\boldsymbol{B}_{s-1}\boldsymbol{Q}_s &= \begin{bmatrix} \times & \times & \times & \times & \times\\ & \times & \times & \times & \times\\ & & y & \times & \times\\ & & & \times & \times\\ & & z & \times & \times \end{bmatrix}\\
&= \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & \ell & & 1 \end{bmatrix}\begin{bmatrix} \times & \times & \times & \times & \times\\ & \times & \times & \times & \times\\ & & y & \times & \times\\ & & & \times & \times\\ & & 0 & * & * \end{bmatrix} = \widetilde{\boldsymbol{M}}_s^{-1}\boldsymbol{B}_s\ .
\end{aligned}$$

Blanks signify already created zeros; $\boldsymbol{P}_s$ and $\boldsymbol{Q}_s$ are elementary permutation matrices satisfying $\boldsymbol{P}_s^{-1} = \boldsymbol{P}_s^T = \boldsymbol{P}_s$, $\boldsymbol{Q}_s^{-1} = \boldsymbol{Q}_s^T = \boldsymbol{Q}_s$; the elimination factor $\ell = z/y$; and changed values are marked by "$*$". It is easily verified (by matrix multiplication), that $\widetilde{\boldsymbol{M}}$ is obtained from $\widetilde{\boldsymbol{M}}_s^{-1}$ simply by changing the sign of the off-diagonal element.

After r ($r \le \frac{1}{2}m(m-1)$) of these steps the matrix $\boldsymbol{U}=\boldsymbol{B}_r$ is upper triangular, and we see that

$$\begin{aligned}
\boldsymbol{U} &= \widetilde{\boldsymbol{M}}_r\boldsymbol{P}_r\boldsymbol{B}_{r-1}\boldsymbol{Q}_r\\
&= \widetilde{\boldsymbol{M}}_r\boldsymbol{P}_r\widetilde{\boldsymbol{M}}_{r-1}\boldsymbol{P}_{r-1}\boldsymbol{B}_{r-2}\boldsymbol{Q}_{r-1}\boldsymbol{Q}_r\\
&= \cdots\\
&= \widetilde{\boldsymbol{M}}_r\boldsymbol{P}_r\widetilde{\boldsymbol{M}}_{r-1}\boldsymbol{P}_{r-1}\cdots\widetilde{\boldsymbol{M}}_1\boldsymbol{P}_1\boldsymbol{B}\boldsymbol{Q}_1\cdots\boldsymbol{Q}_{r-1}\boldsymbol{Q}_r
\end{aligned}$$

$$\Updownarrow$$

$$\boldsymbol{P}\boldsymbol{U}\boldsymbol{Q} = \boldsymbol{P}\widetilde{\boldsymbol{M}}_r\boldsymbol{P}_r\widetilde{\boldsymbol{M}}_{r-1}\boldsymbol{P}_{r-1}\cdots\widetilde{\boldsymbol{M}}_1\boldsymbol{P}_1\boldsymbol{B}\ , \qquad \text{(A.13)}$$

where

$$\boldsymbol{P} = \boldsymbol{P}_1\cdots\boldsymbol{P}_{r-1}\boldsymbol{P}_r\ , \qquad \boldsymbol{Q} = \boldsymbol{Q}_r\boldsymbol{Q}_{r-1}\cdots\boldsymbol{Q}_1\ .$$

Now, exploiting that $\boldsymbol{P}_i^{-1} = \boldsymbol{P}_i^T = \boldsymbol{P}_i$, we get

$$\begin{aligned}
\boldsymbol{P}_1\cdots\boldsymbol{P}_s\widetilde{\boldsymbol{M}}_s\boldsymbol{P}_s &= \boldsymbol{P}_1\cdots\boldsymbol{P}_s\widetilde{\boldsymbol{M}}_s\boldsymbol{P}_s\cdots\boldsymbol{P}_1\boldsymbol{P}_1\cdots\boldsymbol{P}_{s-1}\\
&= \boldsymbol{M}_s\boldsymbol{P}_1\cdots\boldsymbol{P}_{s-1}\ , \qquad \text{(A.14)}
\end{aligned}$$

where $\boldsymbol{M}_s$ is found from $\widetilde{\boldsymbol{M}}_s$ by applying the first s row interchanges and the same column interchanges. This means, that also $\boldsymbol{M}_s$ has ones on the diagonal, and the $-\ell$ lands in an off-diagonal position.

Finally, (4.14) is obtained from (A.13) by consecutive use of (A.14). $\square$

A.6. Discussion of (5.22)

Let the symmetric, positive semidefinite matrix $\boldsymbol{H}$ have the eigensolutions $(\lambda_1, \mathbf{v}_1), \ldots, (\lambda_m, \mathbf{v}_m)$ with orthonormal eigenvectors $\{\mathbf{v}_j\}$ and ordered so that

$$\lambda_1 \geq \cdots \geq \lambda_p > 0 \;, \quad \lambda_{p+1} = \cdots = \lambda_m = 0 \;. \tag{A.15}$$

The number $p \leq m$ is the *rank* of $\boldsymbol{H}$, and the *row space* and *null space* (cf. (2.26)) are

$$\mathcal{R}(\boldsymbol{H}) = \mathrm{span}(\mathbf{v}_1, \ldots, \mathbf{v}_p), \qquad \mathcal{N}(\boldsymbol{H}) = \mathrm{span}(\mathbf{v}_{p+1}, \ldots, \mathbf{v}_m) \;.$$

The modified matrix $\widetilde{\boldsymbol{H}}$ has the same eigenvectors as $\boldsymbol{H}$, while the eigenvalues are changed to $\widetilde{\lambda}_j = \lambda_j + \mu^2,\ j = 1, \ldots, m$.

In (5.20) let the right-hand side have the expansion

$$\boldsymbol{g} = \boldsymbol{g}_{\mathcal{R}} + \boldsymbol{g}_{\mathcal{N}}; \quad \boldsymbol{g}_{\mathcal{R}} = \textstyle\sum_{j=1}^{p} \eta_j \mathbf{v}_j, \quad \boldsymbol{g}_{\mathcal{N}} = \textstyle\sum_{j=p+1}^{p} \eta_j \mathbf{v}_j \;.$$

Then

$$\widetilde{\boldsymbol{h}} = \widetilde{\boldsymbol{H}}^{-1}\boldsymbol{g} = \sum_{j=1}^{p} \frac{\eta_j}{\lambda_j + \mu^2}\mathbf{v}_j + \mu^{-2} \sum_{j=p+1}^{m} \eta_j \mathbf{v}_j \;. \tag{A.16}$$

First, consider the case where $\boldsymbol{H}$ has full rank, $p = m$. The solution to (5.20) is

$$\boldsymbol{h} = \textstyle\sum_{j=1}^{m} (\eta_j/\lambda_j)\mathbf{v}_j \;,$$

and we see that $\boldsymbol{h}$ and $\widetilde{\boldsymbol{h}}$ are close if μ^2 is negligible compared to all the eigeinvalues. This is the case if $\mu^2 \ll \lambda_m$, which we generalize to

$$\mu^2 \ll \lambda_p \;. \tag{A.17}$$

Next, if the system is rank deficient, it may be *consistent:* $\boldsymbol{g} \in \mathcal{R}(\boldsymbol{H})$, i.e. $\boldsymbol{g}_{\mathcal{N}} = \boldsymbol{0}$. Then (5.20) has the complete solution

$$\boldsymbol{h} = \sum_{j=1}^{p} \frac{\eta_j}{\lambda_j}\mathbf{v}_j + \sum_{j=p+1}^{p} \beta_j \mathbf{v}_j \;,$$

where the $\{\beta_j\}$ can take any value. The *minimum norm solution* is given by $\beta_j = 0,\ j = p+1, \ldots, m$. Provided that (A.17) holds, we see that in this case, $\widetilde{\boldsymbol{h}}$ is close to this vector.

If the system (5.20) is *inconsistent*, i.e. $\boldsymbol{g}_{\mathcal{N}} \neq \boldsymbol{0}$, then it has no solution, and provided that $\|\boldsymbol{g}_{\mathcal{N}}\|/\|\boldsymbol{g}_{\mathcal{R}}\|$ is not very small and that (A.17) holds, then it follows from (A.16) that

$$\widetilde{\boldsymbol{h}} \simeq \mu^{-2}\boldsymbol{g}_{\mathcal{N}} \;.$$

This $\widetilde{\boldsymbol{h}}$ is orthogonal to $\mathcal{R}(\boldsymbol{H})$, and an SVD analysis [35] shows that with this Newton direction we get $\boldsymbol{W}\boldsymbol{A}^T\widetilde{\boldsymbol{h}} \simeq \boldsymbol{0}$.

The μ-value given in (5.25) is based on analyzing the effects of rounding errors and using ideas from *regularization,* [17]. The depency on $\boldsymbol{A}$ is derived from the following estimate of the largest eigenvalue,

$$\begin{aligned}\lambda_1 \sim \|\boldsymbol{H}\|_\infty &= \|\boldsymbol{A}\boldsymbol{W}\boldsymbol{A}^T\|_\infty \sim \|\boldsymbol{A}\boldsymbol{A}^T\|_\infty \\ &\sim \|\boldsymbol{A}\|_\infty \cdot \|\boldsymbol{A}^T\|_\infty = \|\boldsymbol{A}\|_\infty \cdot \|\boldsymbol{A}\|_1 \;.\end{aligned}$$

We refer to [35] for details. $\square$

Appendix B. Answers to Exercises

Exercise 2.9. By the rules of Table 2.1 we find

$$(D) \qquad \min\left\{ -24y_1 - 15y_2 \ : \ \begin{matrix} 3y_1 - y_2 \le -13 \\ 2y_1 + 7y_2 = -10 \end{matrix}, \ y_1 \le 0 \right\} .$$

The free variable y_2 is determined from the equality constraint, $y_2 = -\frac{1}{7}(10+2y_1)$, and we see that (D) is equivalent with the scalar problem

$$(D') \qquad \min\left\{ -\frac{138}{7}y_1 + \frac{150}{7} \ : \ \frac{23}{7}y_1 + \frac{10}{7} \le -13, \ y_1 \le 0 \right\} .$$

Here, the negativity constraint $y_1 \le 0$ is redundant. The optimal value is attained for $y_1^* = \frac{7}{23}(-13 - \frac{10}{7}) = -\frac{101}{7}$ and gives the value

$$f_D(\boldsymbol{y}^*) = -\frac{138}{7}(-\frac{101}{7}) + \frac{150}{7} = 108 .$$

Exercise 3.6. A feasible point satisfies all constraints. The positivity constraints are satisfied by the assumption, and from the definitions we se that

$$\begin{aligned} &\boldsymbol{A}(\mathbf{x}+\boldsymbol{h}_x) = \boldsymbol{A}\mathbf{x} + \boldsymbol{f}_P = \boldsymbol{b} , \\ &\boldsymbol{A}^T(\boldsymbol{y}+\boldsymbol{h}_y) + (\boldsymbol{s}+\boldsymbol{h}_s) = \boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{s} + \boldsymbol{f}_D = \boldsymbol{c} . \end{aligned}$$

Thus, also the equality constraints are satisfied.

Exercise 3.10. The system (3.21) is equivalent with

$$\begin{bmatrix} -\boldsymbol{D}^{-1} & \boldsymbol{D}\boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}_x \\ \boldsymbol{h}_y \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}\boldsymbol{f}_x \\ \boldsymbol{f}_y \end{bmatrix}$$

$$\Updownarrow$$

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{D}\boldsymbol{A}^T \\ \boldsymbol{A}\boldsymbol{D} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} -\boldsymbol{D}^{-1}\boldsymbol{h}_x \\ \boldsymbol{h}_y \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}\boldsymbol{f}_x \\ -\boldsymbol{f}_y \end{bmatrix} .$$

Since $(\boldsymbol{A}\boldsymbol{D})^T = \boldsymbol{D}^T\boldsymbol{A}^T = \boldsymbol{D}\boldsymbol{A}^T$, and $\boldsymbol{f}_y = \boldsymbol{f}_P = \boldsymbol{0}$, we recognize this as the system (2.27a) defining the projection $-\boldsymbol{D}^{-1}\boldsymbol{h}_x$ of $\boldsymbol{D}\boldsymbol{f}_x$ onto the null space of $\boldsymbol{A}\boldsymbol{D}$.

Exercise 3.11. The full rank assumption implies that for all $\mathbf{v} \ne \boldsymbol{0}$ the vector $\boldsymbol{z} = \boldsymbol{A}^T\mathbf{v} \ne \boldsymbol{0}$. Further, $\boldsymbol{D}^2 = \mathrm{diag}(d_i^2)$ with $d_i^2 = x_i/s_i > 0$. Therefore,

$$\forall \mathbf{v} \in \mathbb{R}^m, \ \mathbf{v} \ne \boldsymbol{0} : \quad \mathbf{v}^T\boldsymbol{A}\,\boldsymbol{D}^2\boldsymbol{A}^T\mathbf{v} = \boldsymbol{z}^T\boldsymbol{D}^2\boldsymbol{z} = \sum d_i^2 z_i^2 > 0 .$$

This is the definition of $\boldsymbol{H}$ being positive definite.

Exercise 4.3. The columns in $\boldsymbol{B}$ are linearly independent. Therefore, $\boldsymbol{A}_{\mathcal{B}_{k+1}}$ is rank deficient if and only if $\boldsymbol{C}_{:,s}$ is a linear combination of the columns in $\boldsymbol{B}$ with index $j \ne q$. This, however, is **not** the case, since

$$\boldsymbol{B}\boldsymbol{h} = \boldsymbol{C}_{:,s} \quad \Leftrightarrow \quad \boldsymbol{C}_{:,s} = h_1\boldsymbol{B}_{:,1} + \ldots + h_q\boldsymbol{B}_{:,q} + \ldots + h_m\boldsymbol{B}_{:,m}$$

with $h_q \ne 0$.

References

[1] E.D. Andersen and K.D. Andersen: *Presolving in Linear Programming.* Mathematical Programming **71**, 221–245. 1995.

[2] E.D. Andersen, J. Gondzio, C. Mészáros and X. Xu: *Implementation of Interior-Point Methods for Large Scale Linear Programs.* Chapter 6, pp 189–245 in [40]. 1996.

[3] V.A. Barker: *Iterative Methods for Sparse Systems of Linear Equations.* Lecture Notes #62, Institute for Numerical Analysis (now part of IMM), Technical University of Denmark. 1992.

[4] R.E. Bellman and S.E. Dreyfus: *Applied Dynamic Programming.* Princeton University Press, Princeton, N.J. 1962.

[5] J. Clausen: *Teaching Duality in Linear Programming – the Multiplyer Approach.* Proceedings of the Fourth Nordic MPS Meeting, 15 pp. 1996.

[6] J. Czyzyk, S. Mehotra and S.J. Wright: *PCx User Guide.* Technical Report OTC 96/01, Optimization Technology Center. 1997.

[7] G.B. Dantzig: *Linear Programming and Extensions.* Princeton University Press, Princeton, N.J. 1963.

[8] J. Dongarra, C.B. Moler, J.R. Bunch and G.W. Stewart: *An Extended Set of Fortran Basic Linear Algebra Subprograms,* ACM Trans. Math. Soft. **14**, 1–17. 1988.

[9] I.S. Duff, A.M. Erisman and J.K. Reid: *Direct Methods for Sparse Matrices.* Clarendon Press, Oxford. 1986.

[10] O. Edlund, Department of Mathematics, Luleå University of Technology, S-971 87 Luleå: Private communication. 1996, 1997.

[11] R. Fletcher: *Practical Methods of Optimization.* Wiley, Chichester. 2nd ed. 1987.

[12] P.E. Gill, W. Murray, M.A. Saunders and M.H. Wright: *A Practical Anti-Cycling Procedure for Linearly Constrained Optimization.* Mathematical Programming **45**, 437–474. 1989.

[13] P.E. Gill, W. Murray and M.H. Wright: *Numerical Linear Algebra and Optimization. Volume 1.* Addison–Wesley, Redwood City. 1991.

[14] G.H. Golub and C.F. Van Loan: *Matrix Computations, 2nd edition.* John Hopkins University Press, Baltimore. 1989.

[15] J. Gondzio and T. Terlaky: *A Computational View of Interior Point Methods for Linear Programming.* Chapter 3, pp 103–144 in J. Beasley (ed): *Advances in Linear and Integer Programming.* Oxford University Press, Oxford. 1996.

[16] P. Gudmundsson: *Experiments with a Network Interior Point Method.* Master thesis. Report IMM-EKS–1996–25, IMM, DTU, DK-2800 Lyngby. 1996.

[17] P.C. Hansen: *Rank-Deficient and Discrete Ill-Posed Problems.* Doctoral Dissertation. UNI•C, DTU, DK-2800 Lyngby. 1996.

[18] P. Huber: *Robust Statistics.* John Wiley, New York. 1981.

[19] S.K. Jacobsen: *Elementær Operationsanalyse.* IMM, DTU, DK-2800 Lyngby. 1983.

[20] H. Karloff: *Linear Programming.* Birkhäuser, Boston. 1991.

[21] V. Klee and G.J. Minty: *How Good is the Simplex Algorithm?* Pp 159–175 in O. Shisha (ed): *Inequalities – III.* Academic Press, New York. 1972.

[22] N.K. Karmarkar: *A New Polynomial-Time Algorithm for Linear Programming.* COMBINATORICA **4**, 373–395. 1984.

[23] W. Li and J. Swetits: *A New Algorithm for Solving Strictly Convex Quadratic Programs,* SIAM J. on Optimization **7**, 595–619. 1997.

[24] K. Madsen and H.B. Nielsen: *Finite Algorithms for Robust Linear Regression,* BIT **30**, 682–699. 1990.

[25] K. Madsen, H.B. Nielsen: *A Finite Smoothing Algorithm for Linear L_1 Estimation,* SIAM J. on Optimization **3**, 223–235. 1993.

[26] K. Madsen, H.B. Nielsen and M.Ç. Pınar: *A New Finite Continuation Algorithm for Linear Programming,* SIAM J. on Optimization **6**, 600–616. 1996.

[27] K. Madsen, H.B. Nielsen and M.Ç. Pınar: *Bound Constrained Quadratic Programming via Piecewise Quadratic Functions.* Report IMM–1997-05,IMM, DTU, DK-2800 Lyngby. 1996. Available as http://www.imm.dtu.dk/~hbn/publ/TR9705.ps

[28] K. Madsen, H.B. Nielsen and M.Ç. Pınar: *A Finite Continuation Algorithm for Bound Constrained Quadratic Programming.* Accepted for publication in SIAM J. on Optimization. 1998.

[29] I. Maros and G. Mitra: *Simplex Algorithms.* Chapter 1, pp 1–46 in J. Beasley (ed): *Advances in Linear and Integer Programming.* Oxford University Press, Oxford. 1996.

[30] S. Mehotra and J.-S. Wang: *Conjugate Gradient Based Implementation of Interior Point Methods for Network Flow Problems,* Technical Report 95–70.1, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208-3119, USA. 1995.

[31] H.B. Nielsen: *Implementation of a Finite Algorithm for linear ℓ_1 Estimation,* Report NI-91-01, Institute for Numerical Analysis, DTU, DK-2800 Lyngby. 1991.

[32] H.B. Nielsen: *Numerisk lineær algebra.* (textbook in Danish). IMM, DTU, DK-2800 Lyngby. 1996.

[33] H.B. Nielsen: *Bound Constrained Quadratic Programming Solved via Piecewise Quadratic Functions: Implementation.* Report IMM–1996–21, IMM, DTU, DK-2800 Lyngby. 1996. Available as http://www.imm.dtu.dk/~hbn/publ/TR9621.ps

[34] H.B. Nielsen: *Direct Methods for Sparse Matrices (an Introduction).* IMM, DTU, DK-2800 Lyngby. 1997. Available as http://www.imm.dtu.dk/~hbn/publ/Nsparse.ps

[35] H.B. Nielsen: *Implementation of a Piecewice Polynomial Algorithm for Linear Optimization,* Report IMM–1998–??, IMM, DTU, DK-2800 Lyngby. To appear in 1998.

[36] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery: *Numerical Recipes in C, 2nd Edition.* Cambridge University Press, Cambridge. 1992.

[37] J.K. Reid: *A Sparsity-Exploiting Variant of the Bartels-Golub Decomposition for Linear Programming Bases.* Mathematical Programming **24**, 55–69. 1982.

[38] R.T. Rockafellar: *Convex Analysis.* Princeton University Press, Princeton, N.J. 1970.

[39] C. Roos, T. Terlaky, J.-Ph. Vial: *Theory and Algorithms for Linear Optimization – an Interior Point Approach.* Wiley, Chichester. 1997.

[40] T. Terlaky (ed): *Interior Point Methods of Mathematical Programming.* Kluwer, Dordrecht. 1996.

[41] S.J. Wright: *Stability of Augmented System Factorizations in Interior-Point Methods.* SIAM J. MATRIX ANAL. APPL. **18**, 1, 191–222. 1997.

Index