Augmented Primal-Dual Methods for Linear Programs and SOC Problems

Inaugural-Dissertation

zur

Erlangung des Doktorgrades der

Mathematisch-Naturwissenschaftlichen Fakultät

der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Katrin Schmallowsky, M. Sc.

aus

Köthen/Anhalt

Mai 2008

Aus dem Institut für Mathematik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Florian Jarre

Koreferent: Prof. Dr. Marlis Hochbruck

Tag der mündlichen Prüfung: 2008/07/02

DEDICATION

I dedicate this thesis in loving memory to my parents. I hope, that this achievement will complete the dream that you had for me all those years ago.

Abstract

This thesis deals with linear conic programs

$$(P) \qquad \text{minimize} \quad c^T x \quad \text{s.t.} \quad x \in \mathcal{K} \cap (\mathcal{L} + b),$$

where $\mathcal{L}$ is an affine set and $\mathcal{K}$ is a closed, convex cone. We consider two choices for the cone $\mathcal{K}$, first the case that $\mathcal{K}$ is given by the positive orthant, i.e. $\mathcal{K} = \mathbb{R}^n_+$ and secondly that $\mathcal{K}$ is given as the second order cone $\mathcal{K} = \mathcal{Q}_n$. These problems are special cases of convex optimization problems.

In the first part, the equivalence of solving a linear program and the minimization of a convex, differentiable function f, which is piecewise quadratic on the space $\mathbb{R}^{n+m}$, is discussed. In this approach the affine set and the cone $\mathbb{R}^n_+$ are modelled by the function f. For the minimization of this function a generalized Newton method is used. To bound the number of iterations for this method, the properties of the conjugate function of f are exploited.

This approach establishes a basis for the next part of this thesis. By linear transformations the function f can be converted to a piecewise quadratic function in the primal-dual space. A closely related version of this function is considered in later chapters. First, the solution of perturbed linear second order cone programs is investigated, when the data is subject to arbitrary, but small changes. We then show that primal and dual nondegeneracy of linear second order cone programs is equivalent to uniqueness and strict complementarity of the optimal solution. Furthermore, the augmented primal-dual method is extended to linear second order cone programs.

In the third part of this thesis the implementation of the augmented primal-dual method is discussed. For large scale problems we have to observe the limited memory capacity, hence, a limited memory BFGS method is used. Finally, in an application of the preceeding results, we consider the cone of completely positive matrices

$$\mathcal{C}_n^* = \left\{ X \in \mathbb{R}^{n \times n} | X = \sum_{k \in K} v^k (v^k)^T \text{ for some finite } \{v^k\}_{k \in K} \in \mathbb{R}^n_+ \right\}.$$

The question, whether a given matrix B is in $\mathcal{C}_n^*$ or not, is an $\mathcal{NP}$-complete problem. We propose an algorithm that either computes a certificate proving that $B \in C^*$ or converges to a matrix $\bar{S}$ in C^* which in some sense is "close" to B. We further introduce a regularization approach to improve the algorithm in cases, where convergence is not satisfactory. The thesis is completed with numerical results for the algorithms presented here.

Zusammenfassung

Diese Arbeit betrachtet lineare konische Probleme

$$(P) \qquad \text{minimiere } c^T x \text{ s.d. } x \in \mathcal{K} \cap (\mathcal{L} + b),$$

wobei $\mathcal{L}$ ein affiner Raum und $\mathcal{K}$ ein abgeschlossener, konvexer Kegel ist. Wir betrachten zwei Beispiele für den Kegel $\mathcal{K}$, zum Einen den positiven Orthanten, also $\mathcal{K} = \mathbb{R}^n_+$ und zum Anderen den second order cone, also $\mathcal{K} = \mathcal{Q}_n$. Diese Probleme sind Spezialfälle der konvexen Optimierungsprobleme.
Im ersten Teil wird erläutert, dass die Lösung linearer Programme äquivalent ist zur Minimierung einer konvexen, differenzierbaren, stückweise quadratischen Funktion f auf dem Raum $\mathbb{R}^{n+m}$. Die affine Menge $\mathcal{L}$ und der Kegel $\mathbb{R}^n_+$ sind dabei durch quadratische Terme in f beschrieben. Zur Minimierung der Funktion f wird ein verallgemeinertes Newton Verfahren verwendet. Die Anzahl der Iterationen für dieses Verfahren wird beschränkt, indem die Eigenschaften der zu f konjugierten Funktion ausgenutzt werden.
Dieser Ansatz bildet die Grundlage für den nächsten Teil der Arbeit. Durch lineare Transformationen kann die Funktion f zu einer stückweise quadratischen Funktion auf dem primal-dualen Raum umformuliert werden. Eine eng verwandte Funktion wird in späteren Kapiteln betrachtet. Zunächst wird die Lösung von gestörten linearen second order cone Programmen untersucht, bei denen die Daten beliebigen, kleinen Änderungen unterliegen. Außerdem wird die Äquivalenz von primaler und dualer Nicht-Entartung und der Eindeutigkeit und strikten Komplementarität der Optimallösung gezeigt. Schließlich wird das erweiterte primal-duale Verfahren zur Lösung linearer konischer Probleme auf second order cone Programme ausgeweitet.
Im dritten Teil der Arbeit wird die Implementierung des erweiterten primal-dualen Verfahrens diskutiert. Für Probleme mit großer Dimension muss die begrenzte Speicherkapazität berücksichtigt werden. Daher wird ein limited-memory BFGS Verfahren verwendet. Eine Anwendung der bisherigen Ergebnisse ergibt sich bei Betrachtung des Kegels der vollständig positiven Matrizen

$$\mathcal{C}_n^* = \left\{ X \in \mathbb{R}^{n\times n} | X = \sum_{k\in K} v^k (v^k)^T \text{ for some finite } \{v^k\}_{k\in K} \in \mathbb{R}^n_+ \right\}.$$

Die Frage, ob eine gegebene Matrix B in $\mathcal{C}_n^*$ enthalten ist oder nicht, ist ein $\mathcal{NP}$-vollständiges Problem. Es wird ein Algorithmus entwickelt, der entweder bestätigt, dass $B \in C^*$ oder gegen eine Matrix $\bar{S}$ aus C^* konvergiert, die in einem bestimmten Sinn in der Nähe von B liegt. Des Weiteren wird ein Regularisierungsschritt vorgestellt, der den Algorithmus verbessern soll, wenn die Konvergenz nicht zufriedenstellend ist. Schließlich werden numerische Ergebnisse zu den unterschiedlichen hier vorgestellten Algorithmen präsentiert.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A general *optimization problem* is given by

$$\text{minimize} \quad f_0(x) \quad \text{s.t.} \quad f_i(x) \leq b_i \text{ for } i = 1, \ldots, m. \tag{1.1}$$

Here, $x \in \mathbb{R}^n$ is a vector of unknowns, $f_0 : \mathbb{R}^n \mapsto \mathbb{R}$ is called *objective function* and $f_i : \mathbb{R}^n \mapsto \mathbb{R}$ for $i = 1, \ldots, m$ are called *(inequality) constraints* with limits $b_i \in \mathbb{R}$ for $i = 1, \ldots, m$. The goal is to find the best choice of x from a set of vectors that satisfy the constraints, i.e. the vector x with the smallest objective value. Problem 1.1 is called a *convex optimization program* if the objective and constraint functions are convex. A special case of convex optimization programs are *linear programs*, where the objective and constraint functions are linear.

Linear programs appear in many applications like engineering, transportation, telecommunications and other economic situations. In these applications, linear programs can be used in different procedures like planning, scheduling or routing. Often, linear programs arise as subproblems in other algorithms and this is also the case in later chapters of this thesis.

This thesis deals with linear conic programs, where a linear function f is minimized over the intersection of an affine set and a closed convex cone $\mathcal{K}$. Obviously, these problems belong to the class of convex optimization problems. If the cone $\mathcal{K}$ is given as the positive orthant $\mathbb{R}^n_+$, then the constraints $x \in \mathbb{R}^n_+$ are linear as well, so these problems are just the linear programs specified above.

Linear programs and the solution of the same will be discussed in chapter 3. The approach uses a reformulation of the linear program as a convex, differentiable, piecewise quadratic minimization problem as well as an augmented Lagrangian (see e.g. [23, 45]) technique. A survey on this technique will also be given in chapter 3. The complexity analysis of our approach is based on a generalized Newton method applied to the piecewise quadratic function f. The number of steps of the generalized Newton method is bounded by exploiting the properties of the conjugate function for f.
Augmented Lagrangian approaches have been successfully applied to nonlinear and non-convex programs, see e.g. [12, 13], and are the subject of ongoing research, see e.g. [18, 47]. The application to nonlinear programs is well understood. It simplifies considerably when applied to linear programs. Such an application is discussed in chapter 3.

The convex, differentiable, piecewise quadratic function f, which is to be minimized in chapter 3 represents the set of feasible points for the primal and dual linear program as well as the duality gap. The latter is given by the difference of the objective function values for the primal and dual problem. More precisely, this function f determines the sum of the distances of a given point to the set of primal-dual feasible points and the set of points that have a duality gap of zero. With this observation, this approach can be generalized to linear conic programs. These programs can be analogously reformulated as minimization problems with a certain function in the primal-dual space.

In chapter 4 we pursue this generalization and focus on the solution of *linear second order cone programs*. As already mentioned, in a linear second order cone program a linear function is minimized over the intersection of an affine set and the cartesian product of second order (quadratic) cones, see e.g.[1, 2]. The conic condition in these programs is not limited to just one cone, it may consist of a cartesian product of several cones.
First, the perturbation of unique and strictly complementary optimal solutions of linear second order cone programs when the data is subject to small arbitrary changes, is considered. This part is an extension of a result given in [17].

Using the notion of nondegeneracy given in [1], we then show that the standard notion of primal and dual nondegeneracy for second order cone programs is equivalent to uniqueness and strict complementarity. While this result is certainly not surprising, we believe that it has not been rigorously analyzed so far. Based on these results the augmented primal-dual method for solving conic programs given in [25] is extended to second order cone programs. In [25] the augmented primal-dual method is sucessfully applied to large scale semidefinite programs, and we anticipate that the generalization given here proves to be suitable for large scale second order cone programs as well.
A link between second order cone programs and semidefinite programs is established in [8].

Although we do not treat semidefinite programs, the results obtained in this thesis together with the results in [25] can be applied to optimization programs with a mixture of linear, second order cone and semidefinite constraints. A view on linear, second order cone and semidefinite programs and existing algorithms to solve these problems is given in chapter 2 below.

Semidefinite programs arise for example from the determination of a maximum stable set of a graph G. A subset S of the set of vertices in G is called a stable set, if the nodes in S are pairwise not adjacent to each other. The maximum stable set problem has a semidefinite relaxation, which was introduced by Lovasz in 1979 [33]. In this relaxation, the unknown matrix X is, among other constraints, asked to be positive semidefinite. In 2003, de Klerk and Pasechnik [30] replaced the positive semidefinite constraint by a completely positive constraint. With this substitution, they obtained a sharp relaxation for the maximum stable set problem. In this context, the question whether a given matrix is completely positive or not, arised.

The cone of completely positive matrices C^* is the convex hull of all symmetric rank-1-matrices xx^T with nonnegative entries. This cone is considered in chapter 5. The concept of completely positive matrices has been introduced more than 40 years ago, [21, 34].

More recently the interest in completely positive matrices has gained new momentum in the context of combinatorial optimization problems, [6, 9]. Important theoretical properties are summarized in [4, 15]. The question 'whether or not a given matrix B is completely positive' does not only belong to the class of $\mathcal{NP}$-complete problems (see e.g. [39]), but there is also no simple heuristics to date to approach this problem for matrices of moderate dimension. We introduce a simple algorithm which – for a given input B – either determines a certificate proving that $B \in C^*$ or converges to a matrix $\bar{S}$ in C^* which in some sense is "close" to B. A normalization of a matrix $B \succeq 0$ and the computation of a "central" starting point in C^* is discussed and a linearization technique to approach a given matrix B from within C^* is presented. The resulting minimization problem can be written as a linear conic program over the intersection of the positive orthant and a second order cone and thus, the apd-method presented in chapter 4 can be applied to this program. This approach may stagnate before converging to B. Therefore, a refactorization-heuristics to recover from such stagnation is introduced.

The approaches described so far are implemented with MATLAB. In chapter 6, numerical results for the algorithms introduced in chapter 3 and chapter 5 are presented and the implementation of the apd-algorithm introduced in chapter 4 is specified. As the complexity of the computation of the Hessian for the function Ψ in chapter 4 gets rather high for large scale problems, a quasi-Newton method for the determination of the search direction is used; more precisely, we use a BFGS method. For the solution of large scale second order cone problems the memory capacity must be handled. Therefore, a limited memory BFGS method is used, where only few of the recent iterate- and gradient-differences are kept and used for the computation of the next search direction.

Finally, chapter 7 summarizes the results of this thesis and provides an outlook on future research.

Chapter 2

Notation and Fundamentals of Linear Conic Optimization

In this chapter we introduce some notation used in this thesis and give a review on optimization problems. We focus on linear conic problems and present the main properties for this class of problems when applied to different cones.

2.1 Notation

In the remainder of this thesis the following notation is used. For a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ we denote the derivative of f at y by the row vector $Df(y)$ and the gradient by the column vector $\nabla f(y) := Df(y)^T$. Second derivatives are denoted by $D^2 f(y)$ or $\nabla^2 f(y)$, i.e. we do not distinguish between square matrices and bilinear forms. If ∇f is differentiable almost everywhere, the generalized Hessian of f at a point x is given by the convex hull of the limits of $\nabla^2 f(y)$ where $y \to x$ such that $\nabla^2 f(y)$ is well defined; see e.g. [11]. (This definition is not to be confused with other versions of generalized derivatives by Sobolev or Lanczos which are based on partial integration.)

When there is a given set of measure zero (e.g. the set where the second derivative of a given function is not defined) we say a point is in general position, if it does not lie within this set. A point that is generated by some random process with a continuous density function always lies in general position – with probability one.

By E we always denote the matrix of all ones, I the identity matrix, and e the vector of all ones. The dimensions will always be evident from the context. The pseudo inverse of a linear operator M is denoted by $M^\dagger$, see e.g. [20]. The columns of a matrix $A \in \mathbb{R}^{m\times n}$ are denoted by a_i for $1 \le i \le n$, the components of $c \in \mathbb{R}^n$ by c_i.

By $\mathbb{R}_+$ we denote the set of nonnegative numbers, $\mathbb{R}_+ = \{t \in \mathbb{R} \mid t \ge 0\}$. The inequality $X \ge 0$ is used to indicate that the matrix X only has nonnegative entries; such X is called nonnegative.
The space of real $n \times n$-symmetric matrices is denoted by $\mathcal{S}^n$. By $B \succeq 0$ we indicate that the symmetric matrix B is positive semidefinite, and write $\mathcal{S}^n_+ := \{S = S^T \in \mathbb{R}^{n\times n} \mid S \succeq 0\}$.

2.2 Linear Optimization Problems

A linear conic optimization program is given by

$$(P) \qquad \text{minimize } \langle c, x\rangle \text{ s.t. } x \in \mathcal{K} \cap (\mathcal{L} + b).$$

Here, we have denoted the primal variable by x. The linear set $\mathcal{L}$ is given by

$$\mathcal{L} = \{x \mid \mathcal{A}(x) = 0\}, \text{ resp. } \mathcal{L} + b = \{x \mid \mathcal{A}(x) = \mathcal{A}(b) = \bar{b}\},$$

where $\mathcal{A}$ is a matrix or some other representation of a linear operator. If the variable x is a vector, as it is the case in linear and linear second order cone programming, $\mathcal{A}$ can be represented by a matrix. In the above formulation $\mathcal{K}$ is a closed convex cone in a finite dimensional Euclidean space $\mathcal{E}$ and $b, c \in \mathcal{E}$ are given data. With these notations, the dual program is given by

$$(D) \qquad \text{maximize } \langle \bar{b}, y\rangle \text{ s.t. } c - \mathcal{A}^*(y) =: s \in \mathcal{K}^D.$$

The dual variable is denoted by (y, s). In the dual formulation, $\mathcal{K}^D$ is the dual cone of $\mathcal{K}$, i.e.

$$\mathcal{K}^D = \{s \in \mathcal{E} \mid \langle s, x\rangle \ge 0 \ \forall\ x \in \mathcal{K}\}.$$

With the definition of $\mathcal{L}$, the linear set $\{(y, s) | s = \mathcal{A}^*(y)\}$ is the orthogonal complement of $\mathcal{L}$. Since

$$\langle b, s\rangle = \langle b, c - \mathcal{A}^*(y)\rangle = \langle b, c\rangle - \langle \bar{b}, y\rangle,$$

the optimal solution of

$$(D') \qquad \text{minimize} \quad \langle b, s\rangle \quad \text{s.t.} \quad s \in \mathcal{K}^D \cap (\mathcal{L}^\perp + c),$$

corresponds with the optimal solution of (D). It is easily verified that weak duality holds, i.e.

$$\langle b, c\rangle \leq \langle c, x\rangle + \langle b, s\rangle \tag{2.1}$$

for all x, s that are feasible for (P) and (D).
Furthermore, if (P) (or (D)) satisfies Slaters condition, i.e. there exists a strictly feasible point $x > 0, \mathcal{A}x = \bar{b}$ (resp. $s > 0, \mathcal{A}(y) + s = c$), and (P) (resp. (D)) posseses a finite optimal value, then an optimal solution for (D) (resp. (P)) exists and strong duality holds, see e.g. [26]. In this case, a point x (resp. (y, s)) is optimal for (P) (resp. (D)) if, and only if, there exists a point s (resp. x) feasible for (D) (resp. (P)) with

$$\langle b, c\rangle = \langle c, x\rangle + \langle b, s\rangle.$$

The optimality conditions for a primal-dual pair of linear conic programs are thus given by

$$\begin{aligned} \mathcal{A}x &= b, \\ \mathcal{A}(y) + s &= c, \\ \langle x, s\rangle &= 0, \\ x \in \mathcal{K}, \quad s &\in \mathcal{K}^D. \end{aligned}$$

The first two equations ensure that x and (y, s) satisfy the linear constraints of the primal-dual pair (P) and (D), the third equation is called complementarity condition and implies that strong duality holds and the last condition guarantees the fulfilment of the cone constraints.

The cone $\mathcal{K}$ is often given as the positive orthant $\mathbb{R}^n_+$, the second order cone $\mathcal{Q}_n$ or the cone of semidefinite matrices $\mathcal{S}^n_+$. Observe, that all of these cones are self-dual, i.e. $\mathcal{K}^D = \mathcal{K}$. In the sequel we give a short replication of the main properties of these cones and the corresponding problems.

2.2.1 The Positive Orthant

If $\mathcal{K} = \mathbb{R}^n_+$, then (P) and (D) are given as the usual linear programs

$$(P^{LP}) \qquad \text{minimize} \;\; c^T x \;\; \text{s.t.} \;\; Ax = Ab = \bar{b}, \qquad x \geq 0$$

and

$$(D^{LP}) \qquad \text{maximize} \;\; \bar{b}^T y \;\; \text{s.t.} \; c - A^T y =: s \geq 0.$$

Here we see, that the dual of a dual linear program is the original primal linear program. Geometrically, the linear constraints define a convex polyhedron, which is called the feasible region. Since the objective function is also linear, hence a convex function, all local optima are automatically global optima.
A linear program can also be unbounded or infeasible. Weak duality (2.1) states that if the primal program is unbounded then the dual program is infeasible. Likewise, if the dual program is unbounded, then the primal program must be infeasible.

The first algorithm for solving linear programs, the *simplex algorithm*, was developed by George Dantzig [14]. Initially, the simplex algorithm constructs a feasible solution at a vertex of the polyhedron. It then walks along edges of the polyhedron to vertices with successively lower values of the objective function until the optimum is reached. This algorithm proved to be quite efficient in practice and can be guaranteed to find the global optimum if certain precautions against cycling are taken. However, in 1972, Klee and Minty [29] constructed a linear program, where the objective function is minimized over a deformed simplex. This problem is given by

$$\max \left\{ \sum_{j=1}^{n} \epsilon^{n-j} x_j \;\middle|\; x_i + 2 \sum_{j=1}^{i-1} \epsilon^{i-j} x_j \leq 1 \;\; \text{for} \;\; 1 \leq i \leq n, \;\; x \geq 0 \right\},$$

with $0 < \epsilon < \frac{1}{2}$. For this program, the simplex algorithm can be shown to take a number of steps exponential in the problem size. To illustrate the performance of the approach in chapter 3, this problem will appear as a numerical experiment in chapter 6.

In 1979 Leonid Khachiyan [28] analyzed the *ellipsoid method*, the first polynomial-time algorithm for solving linear programs. This method either finds a point in a polyhedron or observes that the polyhedron is empty. Khachiyan verified, that a linear program can be solved with this technique. Although the simplex algorithm performed better in almost all linear programs, the ellipsoid method initiated new lines of research in linear programming with the development of *interior point methods*, which can be implemented as a practical tool.

These algorithms have been inspired by Karmarkar's algorithm [27]. The approach reformulates a linear program as a nonlinear problem and solves the resulting problem with certain modified Newton methods. In contrast to the simplex algorithm, which proceeds along points on the boundary of a polyhedral set, interior point methods move through the interior of the feasible region. The class of primal-dual path-following interior point methods is considered the most successful. Most implementations of interior point methods are based on Mehrotra's predictor-corrector algorithm [36].

2.2.2 The Second Order Cone

As a next example we consider the second order cone $\mathcal{Q}_n$. Second order cone programs have received attention in recent studies of optimization because of their wide applicability and computational efficiency. Formulating optimization problems as second order cone programs provides computational advantages. First, they can be solved with interior point methods and hence in polynomial time. Secondly, in practice, the number of iterations required to find a solution is not much affected by a choice of initial points.

The second order cone (or Lorentz-cone or ice-cream cone) of dimension n is defined by

$$\mathcal{Q}_n := \{x := (x_0; \bar{x}) = (x_0, x_1, \ldots, x_{n-1})^T \in \mathbb{R}^n \mid x_0 \geq \|\bar{x}\|_2\}.$$

Geometrically it looks like the picture below, in two, respectively three dimensions:

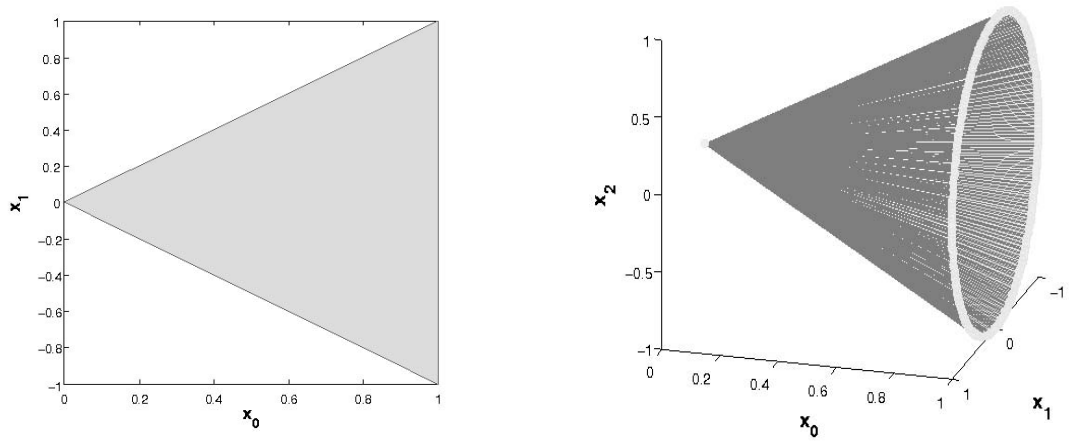

Figure 2.1: The second order cone in two and three dimensions

A second order cone constraint of dimension n specifies that the euclidean norm of $n-1$ variables must be less than or equal to the magnitude of the nth variable. A special case is the second order cone of dimension one, which is given by $\mathcal{Q}_1 = \mathbb{R}_+$.

For vectors u, $v \in \mathbb{R}^n$ we consider the following multiplication [1]:

$$u \circ v := \begin{pmatrix} u^T v \\ u_0 \bar{v} + v_0 \bar{u} \end{pmatrix},$$

and for vectors $u = (u_1, \ldots, u_n)^T, v = (v_1, \ldots, v_n)^T \in \mathbb{R}^{r_1} \times \cdots \times \mathbb{R}^{r_n}$, we set

$$u \circ v = (u_1 \circ v_1, \ldots, u_n \circ v_n)^T.$$

Associated to these vectors we now consider the cartesian product

$$\mathcal{Q} = \mathcal{Q}_{n_1} \times \cdots \times \mathcal{Q}_{n_r}$$

of r second order cones of dimensions $n_1, \ldots, n_r$. Let $n := n_1 + \cdots + n_r$. Then, the following canonical partition of some vectors c, x and $s \in \mathbb{R}^n$ and a matrix $A \in \mathbb{R}^{m \times n}$ is obviously associated with $\mathcal{Q}$:

$$\begin{aligned} c &= (c_1; \ldots; c_r), \text{ where } c_i \in \mathbb{R}^{n_i} \\ x &= (x_1; \ldots; x_r), \text{ where } x_i \in \mathcal{Q}_{n_i}, \\ s &= (s_1; \ldots; s_r), \text{ where } s_i \in \mathcal{Q}_{n_i}, \\ A &= (A_1, \ldots, A_r), \text{ where each } A_i \in \mathbb{R}^{m \times n_i}. \end{aligned}$$

In this thesis we study linear second order cone programs of the form

$$(P^{SOC}) \qquad \begin{array}{ll} \min & c_1^T x_1 + \cdots + c_r^T x_r \\ \text{s. t.} & A_1 x_1 + \cdots + A_r x_r = b, \\ & x_i \in \mathcal{Q}_{n_i}, \text{ for } i = 1, \ldots, r. \end{array}$$

Here, we assume that the matrix $A = (A_1, \ldots, A_r)$ has full row rank m. We use the dual program as introduced in [1],

$$(D^{SOC}) \qquad \begin{array}{ll} \max & b^T y \\ \text{s. t.} & A_i^T y + s_i = c_i, \text{ for } i = 1, \ldots, r, \\ & s_i \in \mathcal{Q}_{n_i}, \text{ for } i = 1, \ldots, r. \end{array}$$

The following statement for second order cone programs is shown e.g. in Theorem 4.2.1 in [40] and Theorem 16 in [1].
If (P^{SOC}) or (D^{SOC}) satisfies Slater's condition, then the optimal values of (P^{SOC}) and (D^{SOC}) coincide. If both problems satisfy Slater's condition, then the optimal solutions x^* and s^* of both problems exist and satisfy the complementarity condition

$$x^* \circ s^* = 0,$$

and, with the definition of s^*, this means that strong duality holds. Conversely, if x and $(y,\ s)$ are feasible points for (P^{SOC}) and (D^{SOC}), respectively, and if $x \circ s = 0$, then x is an optimal solution of (P^{SOC}) and $(y,\ s)$ is an optimal solution of (D^{SOC}).
Second order cone programming is a problem class that lies between linear programming and semidefinite programming, which will be considered in the next section. Like linear programs and semidefinite programs, second order cone programs can be solved very efficiently by primal-dual interior-point methods.

Below, we give two short examples how to convert different constraints to second order cone constraints.

- One example of a second order cone constraint that arises frequently in engineering is the least squares problem with further constraints: Find the vector $x \geq 0$ that minimizes the euclidean norm of $Ax - b$ (where A is a $m \times n$-matrix and x and b are vectors of appropriate dimensions).

If we denote $z_i := a_i^T x - b_i$, $i = 1, \ldots, n$, the original problem

$$\min\{\|Ax - b\|_2 | x \geq 0\}$$

can be written as the following second order cone program

$$\min\{z_0 | (z_0, z_1, \ldots, z_n) = z \in \mathcal{Q}_{n+1}, z_i = a_i^T x - b_i, i = 1, \ldots, n, x \geq 0\}.$$

Note, that this is a problem with mixed linear and second order cone constraints.

- A quadratic objective $x^T Q x$ can be handled by introducing a new variable t such that $x^T Q x \leq t$. Taking the Cholesky decomposition of $Q = LL^T$ and defining $z := L^T x$, this inequality is equivalent to $z^T z \leq t$ and thus a minimization problem with a quadratic objective can be written as

$$\min\{t | (t, z) \in \mathcal{Q}_{n+1}, z = L^T x\},$$

along with the constraints of the original problem.

In the application in chapter 5 will arise further examples of constraints that can be handled as second order cone constraints.

2.2.3 The Semidefinite Cone

As a last example we consider the cone of semidefinite matrices S_+^n. Semidefinite programming can be regarded as an extension of linear programming where the componentwise inequalities between vectors are replaced by matrix inequalities. It is therefore not surprising that the theory of semidefinite programming closely parallels the theory of linear programming. There are some important differences. For example there is no straightforward or practical simplex method for semidefinite programs.

The standard scalar product on the space of $n \times n$- matrices is given by

$$\langle C, X \rangle := C \bullet X := \text{trace}(C^T X).$$

For given matrices $A^{(i)} \in \mathcal{S}^n, i = 1, \ldots, m$, we define a linear map $\mathcal{A} : \mathcal{S}^n \mapsto \mathbb{R}^m$ by

$$\mathcal{A}(X) := \begin{bmatrix} A^{(1)} \bullet X \\ \vdots \\ A^{(m)} \bullet X \end{bmatrix}, \ X \in \mathcal{S}^n.$$

The adjoint operator $\mathcal{A}^* : \mathbb{R}^m \mapsto \mathcal{S}^n$ is given by

$$\mathcal{A}^*(y) = \sum_{i=1}^{m} y_i A^{(i)}, \ y \in \mathbb{R}^m.$$

With these notations, the standard pair of primal and dual linear semidefinite programs can be stated as follows:

$$(P^{SDP}) \qquad \text{minimize} \ \ C \bullet X \ \ \text{s.t.} \ \ \mathcal{A}(X) = \bar{b}, \qquad X \succeq 0$$

and

$$(D^{SDP}) \qquad \text{maximize} \ \ \bar{b}^T y \ \ \text{s.t.} \ \mathcal{A}^*(y) + S = C, \qquad S \succeq 0.$$

There are good reasons for studying semidefinite programs. First, positive semidefinite constraints arise directly in a number of important applications like differential equations, statistics or control theory. Secondly, many convex optimization problems, e.g., linear programming and (convex) quadratically constrained quadratic programming, can be cast as semidefinite programs, so semidefinite programming offers a unified way to study the properties of and derive algorithms for a wide variety of convex optimization problems.
Most importantly, however, semidefinite programs can be solved very efficiently, both in theory and in practice. Particularly, interior-point methods are applicable to semidefinite programs.

Chapter 3

Linear Programs and Implicit Functions

This chapter explores the solution of linear programs based on the minimization of convex, differentiable, piecewise quadratic functions. One of the approaches is based on an augmented Lagrangian method. The content of the present chapter is published in [22].
The method introduced in this chapter provides a basis for the approach which will be presented in the next chapter. While the present method is based on the solution of linear programs, the next chapter will concentrate on the solution of linear second order cone programs.

We recall the linear program

$$(P) \qquad \text{minimize} \quad c^T z \quad \text{s.t.} \quad Az = b, \qquad z \geq 0$$

and its dual

$$(D) \qquad \text{maximize} \quad b^T y \quad \text{s.t.} \quad A^T y \leq c.$$

For later convenience we have denoted the primal variable by z in this chapter, all other notation follows the standard conventions, i.e. the data is given by a matrix $A \in \mathbb{R}^{m \times n}$ with $n > m$ and the vectors b and c of appropriate dimensions. Throughout this chapter we assume that the matrix A defining the linear program (P) has full row rank m.

For the remainder of this chapter we make the following assumption:

Assumption 1. *We assume from now on that there is no direction y with $A^T y \leq 0$ and $b^T y > 0$.*

If there was a y violating Assumption 1 then (D) would not have a finite optimal solution and Algorithm 1 below would identify this case.

3.1 Newton's method for certain piecewise quadratic functions

For a real number α we set $\alpha^+ = \max\{0, \alpha\}$ and for a vector $z \in \mathbb{R}^n$ we denote by z^+ the vector with components $(z^+)_i = (z_i)^+$ for $1 \leq i \leq n$. Using the optimality conditions of (P) and (D), it is straightforward to see that a point $(\bar{z},\ \bar{y})$ solves (P) and (D), if, and only if, it minimizes the convex, differentiable, piecewise quadratic function

$$f^{(P),(D)}(z,y) := (c^T z - b^T y)^2 + \|Az - b\|_2^2 + \sum_{i=1}^{n}((a_i^T y - c_i)^+)^2 + ((-z_i)^+)^2 \quad (3.1)$$

and satisfies $f^{(P),(D)}(\bar{z},\bar{y}) = 0$. A function that is closely related to $f^{(P),(D)}$ will be considered in the next chapter in the context of second order cone programs. Next we consider the minimization of $f^{(P),(D)}$ by a generalized Newton approach with line search.

To analyze the generalized Newton path we consider certain convex, differentiable, piecewise quadratic functions f. For simplicity, the function f below is defined on $\mathbb{R}^m$, the transfer of the results for f to $f^{(P),(D)} : \mathbb{R}^{n+m} \to \mathbb{R}$ is straightforward.

We say f is piecewise quadratic on $\mathbb{R}^m$ if $\mathbb{R}^m$ is partitioned into a finite number of polyhedra and f is quadratic on each of these polyhedra. In this section we always consider functions $f : \mathbb{R}^m \to \mathbb{R}$ of the special form

$$f(y) := q(y) + \frac{1}{2}\sum_{i=1}^{n}((a_i^T y - \gamma_i)^+)^2, \quad (3.2)$$

where q is a convex quadratic function,

$$q(y) = b^T y + \frac{1}{2} y^T H y.$$

Other types of convex, differentiable, piecewise quadratic functions will be considered in Section 3.2.1.

The indices i of the "plus-squared" terms in (3.2) are divided into active, weakly active and inactive indices.

Definition 1. *An index i of (3.2) is called active at y if the i-th component satisfies $a_i^T y - \gamma_i > 0$. It is called weakly active if $a_i^T y - \gamma_i = 0$. Otherwise it is called inactive. Indices are called linearly independent if the associated vectors a_i are linearly independent.*

3.1.1 The generalized Newton path

Next, we consider two types of straightforward generalizations of the Newton step for minimizing f as in (3.2). In (3.3) below, we consider the case where the Hessian of f exists but may be singular, and in (3.5) below, we consider certain points where the Hessian is not defined. The generalized Newton step $\Delta\hat{y}$ for minimizing a convex function f starting at a point $\hat{y}$ is defined as follows: When ∇f is differentiable at $\hat{y}$ we set

$$\Delta\hat{y} := \begin{cases} \lim_{\epsilon\to 0,\ \epsilon>0} -(\nabla^2 f(\hat{y}) + \epsilon I)^{-1}\nabla f(\hat{y}) & \text{if this is finite} \\ \lim_{\epsilon\to 0,\ \epsilon>0} -\epsilon(\nabla^2 f(\hat{y}) + \epsilon I)^{-1}\nabla f(\hat{y}) & \text{else.} \end{cases} \tag{3.3}$$

(Here I denotes the identity matrix.) Hence, when $\nabla^2 f(\hat{y})$ is invertible $\Delta\hat{y}$ is defined by the first case in (3.3) and coincides with the Newton step. When $\nabla^2 f(\hat{y})$ is singular and the gradient of f is not contained in the null space of $\nabla^2 f(\hat{y})$, the generalized Newton step is defined by the second case in (3.3). Using the eigenvalue decomposition of $\nabla^2 f(\hat{y})$ it then follows that $\Delta\hat{y}$ is the orthogonal projection of the negative gradient onto the null space of $\nabla^2 f(\hat{y})$. Finally, if the gradient of f is contained in the null space of $\nabla^2 f(\hat{y})$, then the generalized Newton step is defined again by the first case in (3.3) and coincides with $-(\nabla^2 f(\hat{y}))^\dagger \nabla f(\hat{y})$, where $\dagger$ denotes the pseudo inverse.

If f is a quadratic function on all of $\mathbb{R}^m$ and the step $\Delta\hat{y}$ is defined by the first case in (3.3), the minimum of f is given by the step length $t_{max}(\hat{y}) := 1$; the point $\hat{y} + t_{max}(\hat{y})\Delta\hat{y}$ is a minimizer of f.

If f is quadratic on all of $\mathbb{R}^m$ and $\Delta\hat{y}$ is defined by the second case in (3.3), the function f does not have a minimum and $t_{max}(\hat{y}) := \infty$.

When $\nabla^2 f(\hat{y})$ is not defined, the generalized Hessian of f at $\hat{y}$ contains several elements. A general analysis of this case is complicated; we only consider the case when there is exactly one weakly active index $\hat{i}$ with $a_{\hat{i}}^T\hat{y} - \gamma_{\hat{i}} = 0$ and assume that $\nabla^2 f(y)$ is positive definite for y near $\hat{y}$ and $a_{\hat{i}}^T y - \gamma_{\hat{i}} \neq 0$, say $\nabla^2 f(y) =: \tilde{H} \succ 0$ for y near $\hat{y}$ and $a_{\hat{i}}^T y - \gamma_{\hat{i}} < 0$. For such y the Newton step Δy is a well defined function of y. (One could use the notation $\Delta y = \Delta(y)$ to indicate that Δy depends on y.) The rank-1-update formula for inverse matrices then implies that the sign of the scalar product of $a_{\hat{i}}$ with Δy is the same for all y near $\hat{y}$ with $a_{\hat{i}}^T y \neq \gamma_{\hat{i}}$, i.e. $sign(a_{\hat{i}}^T \Delta y) \equiv const.$ Namely, if the gradient of f is denoted by $g = g(y)$, and $a = a_{\hat{i}}$, then

$$\begin{aligned} sign(a^T(\tilde{H} + aa^T)^{-1}g) &= sign(a^T(\tilde{H}^{-1} - \frac{\tilde{H}^{-1}aa^T\tilde{H}^{-1}}{1 + a^T\tilde{H}^{-1}a})g) \\ = sign(a^T\tilde{H}^{-1}g(1 - \frac{a^T\tilde{H}^{-1}a}{1 + a^T\tilde{H}^{-1}a})) &= sign(a^T\tilde{H}^{-1}g). \end{aligned} \tag{3.4}$$

Note that g is a continuous function of y. Hence, if $a^T\tilde{H}^{-1}g \neq 0$, then either $a_{\hat{i}}^T\Delta y > 0$ for all y near $\hat{y}$ with $a_{\hat{i}}^T y \neq \gamma_{\hat{i}}$ or $a_{\hat{i}}^T\Delta y < 0$ for all such y.

This observation shall be used to generalize the Newton step also for such y near $\hat{y}$ that satisfy $a_{\hat{i}}^T y = \gamma_{\hat{i}}$. In the sequel we will minimize a function f by following the generalized Newton steps. If a Newton step Δy starts at a point y with $a_i^T y \neq \gamma_i$ for all i and crosses the first weakly active index $a_{\hat{i}}^T(y + t\Delta y) = \gamma_{\hat{i}}$ at some point $\hat{y} = y + \hat{t}\Delta y$ with $\hat{t} < t_{max}(y)$, then it is easy to see that $a_{\hat{i}}^T\tilde{H}^{-1}g(\hat{y}) \neq 0$. (If $\hat{t} = t_{max}(y)$, the minimum is found and the algorithm stops.) Hence we assume $a_{\hat{i}}^T\tilde{H}^{-1}g(\hat{y}) \neq 0$ from now on and based on (3.4) we may define the generalized Newton step $\Delta\hat{y}$ starting at $\hat{y}$ by

$$\Delta\hat{y} := \begin{cases} \lim_{y\to\hat{y},\ a_{\hat{i}}^T y > \gamma_{\hat{i}}} \Delta y & \text{if } sign(a_{\hat{i}}^T\Delta y) = 1, \\ \lim_{y\to\hat{y},\ a_{\hat{i}}^T y < \gamma_{\hat{i}}} \Delta y & \text{if } sign(a_{\hat{i}}^T\Delta y) = -1. \end{cases} \tag{3.5}$$

This generalization allows us to define a piecewise linear continuous path based on the relation

$$\dot{y}^+(t) = \frac{\Delta y(t)}{\|\Delta y(t)\|_2}, \tag{3.6}$$

where $\Delta y(t)$ is the generalized Newton step starting at $y(t)$ and

$$\dot{y}^+(t) := \lim_{\Delta t \to 0,\ \Delta t > 0} \frac{y(t + \Delta t) - y(t)}{\Delta t}.$$

Due to (3.5), the one sided derivative $\dot{y}^+(t)$ is defined also at points $y(t)$ with exactly one weakly active index $\hat{i}$, as long as the Hessian of f is nonsingular for y near $y(t)$ and $a_{\hat{i}}^T \tilde{H}^{-1} g(y(t)) \neq 0$.

The case when there is exactly one weakly active index at $\hat{y}$ but $\nabla^2 f(y)$ is not positive definite for y near $\hat{y}$ is illustrated in Case 2. in Section 3.1.2 below. The case when there is more than one weakly active index at $\hat{y}$ is illustrated in Case 1.

We now assume that $\nabla^2 f(y) \succ 0$ everywhere except for such points y that have weakly active constraints [1], i.e. for which $\nabla^2 f(y)$ is not defined. We consider the analogue of Newton's method where the generalized Newton direction is updated repeatedly as we encounter weakly active constraints. We assume that

$$\text{exactly one weakly active constraint exists at each iterate}^2. \tag{3.7}$$

In this case we may define the generalized Newton path $y(t)$ starting from y^0 by (3.6). The points on this path form a piecewise linear curve leading from its initial point $y(0) = y^0$ to the minimizer y^* of f if it exists. Tracing the path is simple: Given an initial point in general position the path crosses just one weakly active index at a time, and the new direction can be computed by a rank-one-update formula in order m^2 operations. The possibility of rank-one-updates for a Newton path has been observed earlier in [16], for example.

The complexity of following the generalized Newton path depends on the number of points with weakly active indices that are crossed by the path. Note that the straight line $[y^0, y^*]$ intersects at most n points with weakly active indices.

---

[1] This assumption may not be satisfied for all (z, y) when f is of the form (3.1). Modifications to account for singular Hessians are tedious and are therefore omitted here.

[2] Assumption (3.7) is generically satisfied: Let $\tilde{S}$ be the set of points that have two or more weakly active constraints. Then, $\tilde{S}$ has dimension $n - 2$. The set of points leading – via the generalized Newton path – to $\tilde{S}$ therefore has dimension $n - 1$. A point in general position will lie outside this set.

Unfortunately, as we will see next, the generalized Newton path may pass the same weakly active index multiple times. We indicate an example where the path contains $n^2/4$ or more points with weakly active indices.

3.1.2 A small example

We return to the function f of (3.2). For illustration we consider the following function $f : \mathbb{R}^2 \to \mathbb{R}$:

$$f(y) = -y_1 + 2y_2 + ((y_1)^+)^2 + ((y_1 - y_2)^+)^2.$$

This function has weakly active indices at all points with $y_1 = 0$ or with $y_1 = y_2$. The generalized Newton path starting at $y^0 := (-1, 2)^T$ leads along $y^0 + t(1, -2)^T$ for $0 \leq t \leq 1$ and then continues along the line $(0, 0)^T + t(-1, -1)^T$ for $t \geq 0$.

Case 1. The derivative of the path is not defined at $(0, 0)^T$; by distinguishing the four cases $y_1 \geq 0$ and/or $y_1 - y_2 \geq 0$, one easily finds that the continuation of the path in $(0, 0)^T$ is uniquely defined as stated above. Hence, the path does not pass through the line $y_1 = 0$ but is "reflected" at this line. As indicated in (3.4), such a reflection cannot occur, when there is just one weakly active index!

Case 2. When the initial point is changed to $y^0 = (-1, 3)^T$, the path will lead from y^0 to $(0, 1)^T$, then to $(0, 0)^T$, and then along the line $(0, 0)^T + t(-1, -1)^T$ for $t \geq 0$.

Case 3. If, in addition, a "prox-term" is added, $f(y) \longrightarrow f(y) + \epsilon y^T y$, the path will pass through the line $y_1 = 0$ near $(0, 1)^T$, then through the line $y_1 = y_2$, and will then pass the line $y_1 = 0$ a second time for some $y_2 < 0$. Hence, we cannot guarantee that the generalized Newton path will cross the same weakly active index (here $y_1 = 0$) only once.

The above cases are pictured in Figures 3.1 – 3.3. In fact, the negative result of the previous example can be strengthened: By adding $\hat{n} - 1$ further $((\ .\ )^+)^2$-terms, the example can be extended to cross the line $y_1 = 0$ exactly $\hat{n} + 1$ times along a zigzag-line.

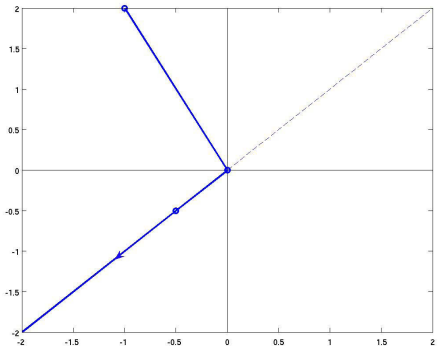

Figure 3.1: Case 1

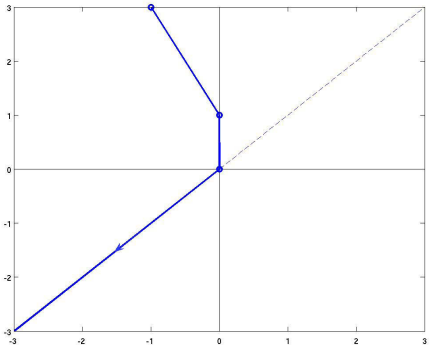

Figure 3.2: Case 2

Then, the term $((y_1)^+)^2$ in the definition of f can be replaced by $\sum_{i=1}^{\hat{n}} \frac{1}{\hat{n}}((y_1 + \epsilon_i)^+)^2$, so that each of these new $((\ .\ )^+)^2$-terms is crossed $\hat{n}+1$ times. We thus obtain a function f of the form (3.2) defined with $n = 2\hat{n}$ "$((\ .\ )^+)^2$-terms" and a piecewise linear generalized Newton path that consists of $\hat{n}(\hat{n}+2) = n + n^2/4$ linear segments.

To estimate the worst-case-complexity for following the generalized Newton path, we like to bound the number of linear segments on the generalized Newton path.

Note that in the situation discussed in Case 1 above, the definition of the generalized Newton path may be difficult. We therefore consider the case of a strictly convex function q in (3.2), i.e. $H \succ 0$.

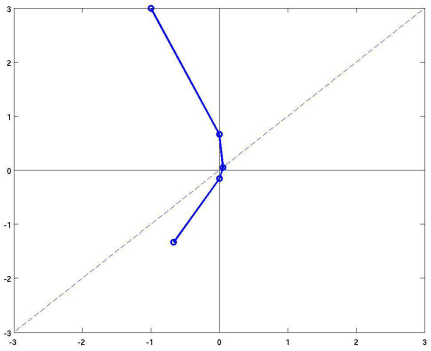

Figure 3.3: Case 3

When applying the generalized Newton method for minimizing f we obtain the following algorithm:

Algorithm 1 (Minimizing a strictly convex piecewise quadratic f)**.**

1. *Let a vector $y^0 \in \mathbb{R}^m$ in general position be given. Set $k = 0$.*
2. *Compute the generalized Newton step Δy^k at y^k.*
3. *Determine the smallest number $\bar{\lambda}_k \in (0, \infty]$ such that $y^k + \bar{\lambda}_k \Delta y^k$ contains a weakly active index. (Then f is quadratic on the line segment $[y^k, y^k + \bar{\lambda}_k \Delta y^k]$.)*

   Determine λ_k minimizing $f(y^k + \lambda \Delta y^k)$ for $\lambda \in (0, \bar{\lambda}_k]$.
   If $\lambda_k = \infty$ then Stop, f does not have a minimum.
4. *Set $y^{k+1} := y^k + \lambda_k \Delta y^k$.*
5. *If $\nabla f(y^{k+1}) = 0$ Stop, else set $k := k + 1$ and go to Step 2.*

Note: The case $\lambda_k = \infty$ in *Step 3.* cannot occur when $H \succ 0$.

3.1.3 The conjugate of a differentiable, piecewise quadratic, strictly convex function

As in the previous example we will minimize a strictly convex, differentiable, piecewise quadratic function f by tracing the generalized Newton path.

In the gradient space the generalized Newton path is a straight line. The link of the primal space and the gradient space is established via the conjugate function f^*. While f is strictly convex and quadratic on each cell of the primal arrangement, f^* is strictly convex and quadratic on each cell of a corresponding dual arrangement. Since the generalized Newton path is a line segment in the gradient space, the number of Newton steps needed to minimize f is the number of cells intersected by the line segment in the dual space. Subsection 3.1.3 studies in more detail the cell structure.

To simplify the analysis, we assume in this subsection that the function q in (3.2) is strictly convex [3], i.e. $H \succ 0$.

Since f is a strictly convex differentiable function, the gradient $v = \nabla f(y)$ is a one to one mapping from $\mathbb{R}^m$ to $\mathbb{R}^m$, and the conjugate function f^* is a strictly convex differentiable function which is given by

$$f^*(v) := \max_{y \in \mathbb{R}^m} \{v^T y - f(y)\}.$$

The function f^* is an implicit function that is closely related to the generalized Newton path. As shown in Theorem 26.6 in [48] it can also be written as

$$f^*(v) = [(\nabla f)^{-1}(v)]^T v - f((\nabla f)^{-1}(v)).$$

Strict monotonicity of ∇f, i.e.

$$[\nabla f(y) - \nabla f(x)]^T (y - x) > 0, \text{ (if } y \neq x)$$

also holds due to strict convexity and differentiability of f (Theorem IV.4.1.4 in [24]). In the sequel, the space $\{v \mid v = \nabla f(y), y \in \mathbb{R}^m\}$ is referred to as the dual space.

For $J \subset \{1, \ldots, n\}$ let $\mathcal{P}_J$ be the polyhedron

$$\mathcal{P}_J := \{y \mid a_i^T y \geq \gamma_i \text{ for } i \in J, \quad a_i^T y \leq \gamma_i \text{ for } i \notin J\}.$$

[3] If not, a regularization term $\epsilon y^T y$ may be added to f to obtain a regularized function for which the generalized Newton path is uniquely defined. This path may then be used as a reference path to define the generalized Newton path for f; however, this approach is somewhat tedious and does not seem to be of practical or theoretical importance. It is therefore omitted.

By definition f is a quadratic function on each $\mathcal{P}_J$. For $y \in \mathcal{P}_J$, $\nabla f(y)$ is written as follows:

$$\nabla f(y) = (H + \sum_{j \in J} a_j a_j^T)y + b - \sum_{j \in J} \gamma_j a_j.$$

We analyze the gradient of f on each $\mathcal{P}_J$ and define $\tilde{\mathcal{P}}_J$ as the corresponding polyhedron of $\mathcal{P}_J$, i.e.

$$\tilde{\mathcal{P}}_J := \nabla f(\mathcal{P}_J) = (H + \sum_{j \in J} a_j a_j^T)\mathcal{P}_J + b - \sum_{j \in J} \gamma_j a_j.$$

Therefore, $\tilde{\mathcal{P}}_J$ is a polyhedron, and since ∇f is one to one from $\mathbb{R}^m$ to $\mathbb{R}^m$, the union of the polyhedra $\tilde{\mathcal{P}}_J, J \subset \{1, \ldots, n\}$ satisfies

$$\bigcup_{J \subset \{1,\ldots,n\}} \tilde{\mathcal{P}}_J = \mathbb{R}^m.$$

Obviously, for two sets $J, \bar{J} \subset \{1, \ldots, n\}$, $\mathcal{P}_J$ and $\mathcal{P}_{\bar{J}}$ are neighbors, if and only if $\tilde{\mathcal{P}}_J$ and $\tilde{\mathcal{P}}_{\bar{J}}$ are neighbors. It is easily seen that f^* is a continuous strictly convex piecewise quadratic function. On each of the $\tilde{\mathcal{P}}_J$ it is a quadratic function.

In the dual space, the path generated by Algorithm 1 is written as

$$\nabla f(y(t)) = tv^0, \ (t \in [0, 1]),$$

where $v^0 = \nabla f(y^0)$. The number of $\tilde{\mathcal{P}}_J$ intersected by the path is exactly the number of steps needed in Algorithm 1. Since $\tilde{\mathcal{P}}_J := \nabla f(\mathcal{P}_J)$, the number of polyhedra $\tilde{\mathcal{P}}_J$ is the same as the number of $\mathcal{P}_J$ dividing $\mathbb{R}^m$. Since this number is bounded by 2^n, the number of iterations of Algorithm 1 is bounded by 2^n. We summarize the discussion in the following lemma:

Lemma 1. *In Algorithm 1, the Hessian of a strictly convex function f can be updated with order n^2 operations at each step if there is only one weakly active constraint at each iteration. In this case the number of generalized Newton steps is bounded by at most 2^n.*

Remark 1. *By the footnote to Assumption* (3.7) *the existence of exactly one weakly active constraint at each iteration is guaranteed if the starting point y^0 is given in general position.*

We note that the computation of a generalized Newton step for weakly convex f is somewhat more complicated than the computation of a simplex step. We believe that the upper bound of 2^n generalized Newton steps is overly pessimistic, the worst example we found is given in Section 3.1.2 which obtains an upper bound of $n + n^2/4$ for even numbers n.

3.2 An implicit function derived from the augmented Lagrangian

The function $f^{(P),(D)}$ of Section 3.1 is closely related to the augmented Lagrangian function. It does not need any penalty parameter but it depends on $n + m$ unknowns while the augmented Lagrangian can be written as a function of only m variables. In this section we derive further theoretical results based on the augmented Lagrangian.

3.2.1 The augmented Lagrangian for linear programs: Basic results

In mathematical optimization problems, the method of Lagrange multipliers is a method for finding the extrema of a function of several variables subject to one or more constraints; it is the basic tool in nonlinear constrained optimization. It reduces finding stationary points of a constrained function in n variables with m constraints to finding stationary points of an unconstrained function in m variables.
The method introduces a new unknown scalar variable (called the Lagrange multiplier) for each constraint, and defines a new function (called the Lagrangian) in terms of the original function, the constraints, and the Lagrange multipliers.
The augmented Lagrangian is given as the usual Lagrangian with an additional penalty term, that penalizes the violation of the equality constraints, i.e.

$$\Lambda(y, z, r) := c^T z + \frac{r}{2}((Az - b + \frac{y}{r})^+)^T(Az - b + \frac{y}{r})^+ - \frac{y^T y}{2r}$$

for a given penalty parameter $r > 0$. The augmented Lagrangian for the dual problem (D) is given by

$$\Lambda(y, z, r) := -b^T y + \frac{r}{2}((A^T y - c + \frac{z}{r})^+)^T (A^T y - c + \frac{z}{r})^+ - \frac{z^T z}{2r}.$$

Note: A derivation of the augmented Lagrangian can be found, for example, in [5], p. 395. There are several variants of augmented Lagrangian functions. Other (partially) augmented Lagrangian functions use quadratic penalty terms only for equality constraints and leave simple bounds unmodified. In this case, inequalities are treated via slack variables. Our approach is based on the (fully) augmented Lagrangian as given above, where also inequalities are penalized. We have chosen the dual problem (D) to define Λ so that there is only one type of constraint.

The gradient of Λ with respect to y and z is given by

$$\nabla_y \Lambda(y, z, r) = -b + rA(A^T y - c + \frac{z}{r})^+$$

and

$$\nabla_z \Lambda(y, z, r) = (A^T y - c + \frac{z}{r})^+ - \frac{z}{r}.$$

The next proposition is well known in a more general context; in the case of linear programs it can be stated in a slightly stronger and particularly simple fashion:

Proposition 1. *For fixed $z \in \mathbb{R}^n$ the function $y \mapsto \Lambda(y, z, r)$ is convex, and for fixed $y \in \mathbb{R}^m$ the function $z \mapsto \Lambda(y, z, r)$ is concave. A point $(\bar{y}, \bar{z})$ satisfies*

$$\nabla_y \Lambda(\bar{y}, \bar{z}, r) = 0 \qquad \textit{and} \qquad \nabla_z \Lambda(\bar{y}, \bar{z}, r) = 0, \tag{3.8}$$

if, and only if, it is an optimal solution of (D) and (P).

Proof. The convexity with respect to y is evident; concavity with respect to z follows from a standard argument. Let (3.8) be satisfied. Relation $(A^T \bar{y} - c + \frac{\bar{z}}{r})^+ - \frac{\bar{z}}{r} = 0$ implies $A^T \bar{y} \leq c$ (dual feasibility), $\bar{z} \geq 0$, and $\bar{z}_i = 0$ if $(A^T \bar{y})_i < c_i$ (complementarity). The relation

$$0 = \nabla_y \Lambda(\bar{y}, \bar{z}, r) = -b + A(r(A^T \bar{y} - c) + \bar{z})^+$$

implies that $(r(A^T \bar{y} - c) + \bar{z})^+$ is feasible for (P), and by complementarity it follows furthermore that $(r(A^T \bar{y} - c) + \bar{z})^+ = \bar{z}$.

Hence, $(\bar{y}, \bar{z})$ is an optimal solution of (D) and (P). Likewise, when $(\bar{y}, \bar{z})$ is an optimal solution of (D) and (P), relation (3.8) follows. □

For given (y, z, r) let $\sigma \in \mathbb{R}^n$ be defined by

$$\sigma_i := \sigma_i(y, z, r) := \begin{cases} 1 & \text{if } (A^T y - c + \frac{z}{r})_i \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{3.9}$$

and let

$$\Sigma := \mathrm{Diag}(\sigma)$$

be the $n \times n$ diagonal matrix with diagonal entries $\Sigma_{ii} = \sigma_i$. Let (y, z, r) be given such that $(A^T y - c + \frac{z}{r})_i \neq 0$ for all i. Then the function $\Lambda(\ .\ ,\ .\ , r)$ is twice differentiable at (y, z) and the second derivatives of Λ with respect to y and z are given by

$$\nabla_y^2 \Lambda(y, z, r) = rA\Sigma A^T \succeq 0$$

and

$$\nabla_z^2 \Lambda(y, z, r) = -\frac{1}{r}(I - \Sigma) \preceq 0.$$

(The latter, along with the differentiability of Λ, also implies concavity of Λ with respect to z.)

Let z be fixed arbitrarily. By Assumption 1 the function $y \mapsto \Lambda(y, z, r)$ is bounded below and due to its piecewise quadratic structure, the solution set $Y(z)$ of the problem

$$\text{minimize}_{y \in \mathbb{R}^m} \quad \Lambda(y, z, r) \tag{3.10}$$

is nonempty. (On each of the finitely many polyhedra on which Λ is quadratic there exists at least one minimizer y; the ones with the smallest value of Λ solve (3.10).) Hence, by Assumption 1 there always exists

$$y \in Y(z) := \mathrm{argmin}_y\{\Lambda(\ .\ , z, r)\}. \tag{3.11}$$

Conversely, if problem (3.10) has a solution for some $z \in \mathbb{R}^n$, then Assumption 1 must hold. This is readily verified: If Assumption 1 does not hold then there is a vector Δy with $b^T \Delta y > 0$ and $A^T \Delta y \leq 0$. Then for any y and $\lambda > 0$ we have

$$\Lambda(y + \lambda \Delta y, z, r) \leq \Lambda(y, z, r) - \lambda b^T \Delta y \overset{\lambda \to \infty}{\longrightarrow} -\infty,$$

so that $\Lambda(\ .\ , z, r)$ does not have a minimum. #

Since for fixed y, the function Λ is concave with respect to z, also

$$\varphi(z) := \Lambda(Y(z), z, r) = \min_{y \in \mathbb{R}^m} \{\Lambda(y, z, r)\}$$

is a concave function of z (the minimum of concave functions is concave).

To avoid set-valued functions we define the point

$$y(z) := \operatorname{argmin}\{\|y\|_2^2 \mid y \in Y(z)\}. \tag{3.12}$$

The constraint $y \in Y(z)$ is equivalent to the equation $\nabla_y \Lambda(y, z, r) = 0$. Note that for fixed z, the set $Y(z) = \{y \mid \nabla_y \Lambda(y, z, r) = 0\}$ of minimizers of the convex function $\Lambda(\ .\ , z, r)$ is a convex set. On the other hand, by definition of $\nabla_y \Lambda$,

$$Y(z) = \{y \mid -b + rA(A^T y - c + \frac{z}{r})^+ = 0\}. \tag{3.13}$$

While it is not evident from representation (3.13) that $Y(z)$ is convex for fixed z, this representation is certainly piecewise linear. Convexity and piecewise linearity imply that $Y(z)$ is a convex polyhedron. Hence, it can be written as

$$Y(z) = \{y \mid B_z y \leq \tilde{b}_z\},$$

where the matrix B_z and the vector $\tilde{b}_z$ depend on z. From (3.13) it also follows that the constraints of $Y(z)$ are piecewise linear also with respect to z implying that B_z and $\tilde{b}_z$ can be written as piecewise linear functions of z. The KKT-conditions of

$$y(z) := \operatorname{argmin}\{\|y\|_2^2 \mid B_z y \leq \tilde{b}_z\}$$

imply that $y(z)$ is a piecewise linear function of B_z and $\tilde{b}_z$ and hence a piecewise linear function of z. Thus $\varphi(z) = \Lambda(y(z), z, r)$ is a piecewise quadratic function of z. Note that continuity of φ follows from the concavity of φ.

Moreover, for any $z \in \mathbb{R}^n$, $d \in \mathbb{R}^n$ the function $y(z)$ posseses a directional derivative $y'(z, d)$. It follows that the derivative of φ is given by

$$D_z\varphi(z) = \underbrace{D_y\Lambda(y(z), z, r)}_{=0} y'(z,\ .\ ) + D_z\Lambda(y(z), z, r) = D_z\Lambda(y(z), z, r). \tag{3.14}$$

Hence, the following observation holds:

Proposition 2. *The function φ is differentiable everywhere. To solve the linear programs (P) and (D) it suffices to find a point z such that $D_z\varphi(z) = 0$.*

The proposition is evident as $D_z\varphi(z) = 0$ implies $D_z\Lambda(y(z), z, r) = 0$ and by definition of $y(z)$, also $D_y\Lambda(y(z), z, r) = 0$. #

3.2.2 The structure of the implicit function φ

We consider the case where $\nabla_y^2\Lambda(y(z),z,r) \succ 0$. In this case, $Y(z) = \{y(z)\}$ contains exactly one element, and by the implicit function theorem, its total derivative $D_z y(z) =: \dot{y}(z)$ exists. Taking the derivative with respect to z of the equation $\nabla_y\Lambda(y(z),z,r) \equiv 0$ yields

$$D_y^2\Lambda(y(z),z,r)\dot{y}(z) + D_z(\nabla_y\Lambda(y(z),z,r)) = 0.$$

The second term on the left hand side is given by $D_z(\nabla_y\Lambda(y(z),z,r)) = A\Sigma$. We obtain

$$\dot{y}(z) = -(D_y^2\Lambda(y(z),z,r))^{-1}A\Sigma = -\frac{1}{r}(A\Sigma A^T)^{-1}A\Sigma.$$

From this and (3.14) we derive

$$\begin{aligned} D^2\varphi(z) &= D_y(\nabla_z\Lambda(y(z),z,r))\dot{y}(z) + D_z^2\Lambda(y(z),z,r) \\ &= -\frac{1}{r}\Sigma A^T(A\Sigma A^T)^{-1}A\Sigma - \frac{1}{r}(I-\Sigma) \preceq 0. \end{aligned} \tag{3.15}$$

The piecewise linear function $\nabla\varphi$ is differentiable almost everywhere. Whenever it is differentiable its derivative satisfies relation (3.15). This confirms the earlier observation that φ is concave for all $z \in \mathbb{R}^n$ and all $r > 0$.
Due to the piecewise linear-quadratic structure of φ it follows that φ is unbounded above when the primal linear program (P) does not have an optimal solution. (Indeed, if φ is bounded above, due to the piecewise quadratic structure it must have a maximum z^{opt}. Since $\nabla\varphi(z^{opt}) = 0$ it follows that z^{opt} solves (P) which is a contradiction.)

Observe that $\Sigma A^T(A\Sigma A^T)^{-1}A\Sigma = \Sigma$ when there are exactly $e^T\sigma = m$ linearly independent columns a_i of A with $\sigma_i = 1$. In this case we obtain

$$D^2\varphi(z) = -\frac{1}{r}I. \tag{3.16}$$

For such points, the Powell-update rule (see [45]) for z

$$z^{k+1} = z^k + r\nabla\varphi(z^k) = z^k + r\nabla_z\Lambda(y(z^k),z^k,r) = (r(A^Ty(z^k) - c) + z^k)^+$$

coincides with the Newton step for maximizing φ. When $e^T\sigma > m$ the matrix $D^2\varphi(z)$ is not invertible.

In this case $\Sigma A^T(A\Sigma A^T)^{-1}A\Sigma$ is a projection matrix and $D^2\varphi(z)$ has the eigenvalue zero of multiplicity $e^T\sigma - m$, and the eigenvalue $-\frac{1}{r}$ of multiplicity $n+m-e^T\sigma$. This in turn implies that the Powell-update Δz is too short, a line search minimizing the unknown distance $\|z + \alpha\Delta z - z^{opt}\|_2$ would return a step $\alpha\Delta z$ with $\alpha \geq 1$.

Remark 2. *If (3.16) was true for all $z \in \mathbb{R}^n$, the Powell-update would return an optimal solution z^{opt} of (P) in one step. Of course, this is generally not the case. However, when (P) and (D) have unique optimal solutions z^{opt} and y^{opt}, z is fixed, and r is sufficiently large, say $r \geq \bar{r}$, then $y(z)$ is close to y^{opt}. Then, each inactive constraint $\bar{i}$ of (D) with $a_{\bar{i}}^T y^{opt} < c_{\bar{i}}$ induces an inactive index $\bar{i}$ with $a_{\bar{i}}^T y(z) - c_{\bar{i}} + \frac{z}{r} < 0$. The remaining m indices must be active, so that (3.16) holds at z. In fact, (3.16) holds on the entire line segment $[z, z^{opt}]$ and the Powell-update does return the optimal solution z^{opt} of (P) in one step.*

The closeness of $y(z)$ to y^{opt} follows in a straightforward fashion from Pietrzykowskis theorem (see [44] or Thm.11.1.5 in [26]) which states that for a constrained problem with a strict (local) minimizer, the minimizers of the penalty problem converge to the minimizer of the constrained problem. Here, the perturbation $\frac{z}{r}$ of the constraints tends to zero for large r, and uniqueness of y, z allows the use of the implicit function theorem.

We summarize the results of this section in Proposition 3.

Proposition 3. *The function φ is concave, piecewise linear-quadratic, and differentiable for all $z \in \mathbb{R}^n$ and all $r > 0$; its second derivative multiplied by "$-r$" is an orthogonal projection whenever it is defined. (P) has an optimal solution if, and only if, φ has a maximum. The latter is the case if, and only if, φ is bounded above. In this case each maximizer of φ is an optimal solution of the linear program (P).*

3.2.3 Conjugate functions of the implicit function φ

The Powell update for z is closely related to the Newton step for maximizing φ. As φ is piecewise quadratic, the complexity of Newton's method for maximizing φ can again be related to the conjugate function of $-\varphi$.

We recall the definition of the implicit function φ,

$$\varphi(z) = \min_{y\in\mathbb{R}^m}\{\Lambda(y,z,r)\}.$$

We assume for the moment that the set of optimal solutions of (D) is bounded. To simplify the notation we also assume $r = 1$ from now on; (this can be done without loss of generality). We obtain

$$\varphi(z) = \min_{y\in\mathbb{R}^m}\ \{-b^Ty + \frac{1}{2}\sum_{i=1}^{n}((a_i^Ty - c_i + z_i)^+)^2 - z_i^2\}.$$

Since φ is concave the convex conjugate function of $-\varphi$ is given by

$$(-\varphi)^*(\tilde{z}) = \max_{z\in\mathbb{R}^n}\{\tilde{z}^Tz + \min_{y\in\mathbb{R}^m}\ \{-b^Ty + \frac{1}{2}\sum_{i=1}^{n}((a_i^Ty - c_i + z_i)^+)^2 - z_i^2\}\}. \quad (3.17)$$

For a given $\tilde{z}\in\mathbb{R}^n$ we define the function $l = l_{\tilde{z}}$ of the variables y and z by

$$l(y,z) = \tilde{z}^Tz - b^Ty + \frac{1}{2}\sum_{i=1}^{n}(((a_i^Ty - c_i + z_i)^+)^2 - z_i^2).$$

As noted before, l is convex with respect to y and concave with respect to z. Since the set of optimal solutions of (D) is bounded, there does not exist a $y\neq 0$ with $b^Ty\geq 0$ and $A^Ty\leq 0$. This implies that $\lim_{\|y\|\to\infty} l(y,z) = \infty$. Now assume that $\tilde{z}$ is given such that there exists a y^0 with $A^Ty^0 < c-\tilde{z}$. Assume $A^Ty^0 \leq c-\tilde{z}-\epsilon e$ for some $\epsilon > 0$. Then, when $z_i\to+\infty$, the i-th component in l can be bounded above by

$$\begin{aligned} & \tilde{z}_iz_i + \frac{1}{2}(((a_i^Ty - c_i + z_i)^+)^2 - z_i^2) \\ = \ & \tilde{z}_iz_i + \frac{1}{2}((a_i^Ty - c_i + z_i)^2 - z_i^2) \\ \leq \ & \frac{1}{2}(a_i^Ty - c_i)^2 - \epsilon z_i \to -\infty. \end{aligned}$$

For $z_i\to-\infty$, the i-th component in l tends to $-\infty$ as well.
Hence, $\lim_{\|z\|\to\infty} l(y,z) = -\infty$. Hence, assumptions $(H1)$ to $(H4)$ of Theorem VII,4.3.1 in [24] are satisfied, and there exists a saddle point of $l = l_{\tilde{z}}$ so that the order of the minimization and the maximization may be interchanged. We then obtain from (3.17)

$$(-\varphi)^*(\tilde{z}) = \min_{y\in\mathbb{R}^m}\ \{-b^Ty + \max_{z\in\mathbb{R}^n}\{\tilde{z}^Tz + \frac{1}{2}\sum_{i=1}^{n}((a_i^Ty - c_i + z_i)^+)^2 - z_i^2\}\}. \quad (3.18)$$

Let $\hat{c} = \hat{c}(y) := A^T y - c$. The inner maximization in (3.18) with respect to z then implies

$$\tilde{z} = z - (\hat{c} + z)^+,$$

or, equivalently,

$$z_i = \begin{cases} \tilde{z}_i & \text{if } \tilde{z}_i < -\hat{c}(y)_i, \\ \geq \tilde{z}_i & \text{if } \tilde{z}_i = -\hat{c}(y)_i, \\ \text{undefined} & \text{if } \tilde{z}_i > -\hat{c}(y)_i. \end{cases}$$

Hence, the maximum is finite if, and only if, $\hat{c}(y) \leq -\tilde{z}$. Note that in case of $\hat{c}(y)_i = -\tilde{z}_i$ we have

$$\tilde{z}_i z_i + \frac{1}{2}\left(((a_i^T y - c_i + z_i)^+)^2 - z_i^2\right) = \frac{1}{2}\tilde{z}_i^2$$

for all $z_i \geq \tilde{z}_i$. Hence, we may replace $z_i = \tilde{z}_i$ for all i, and the function $(-\varphi)^*$ reduces to

$$\begin{aligned} (-\varphi)^*(\tilde{z}) &= \min_{y:\ A^T y - c \leq -\tilde{z}} \{-b^T y + \tilde{z}^T \tilde{z} + \frac{1}{2}\sum_{i=1}^{n}(((a_i^T y - c_i + \tilde{z}_i)^+)^2 - \tilde{z}_i^2)\} \\ &= \min_{y:\ A^T y - c \leq -\tilde{z}} \{-b^T y + \frac{1}{2}\tilde{z}^T \tilde{z}\} \qquad (3.19) \\ &= \frac{1}{2}\tilde{z}^T \tilde{z} - \max_{y:\ A^T y \leq c - \tilde{z}} \{b^T y\}. \qquad (3.20) \end{aligned}$$

This function is piecewise quadratic, but not differentiable everywhere since $-\varphi$ is not strictly convex (see again Theorem 26.3 in [48]). Note that the optimal value (not the optimal solution) of the maximization problem in (3.20) is a continuous function of the data $(A, b, c, \tilde{z})$ whenever it is finite.

The conjugate function of $(-\varphi)^*$ in (3.20) is given by

$$\begin{aligned} (-\varphi)^{**}(z) &= \max_{\tilde{z}} \{z^T \tilde{z} - \frac{1}{2}\tilde{z}^T \tilde{z} + \max_y \{b^T y \mid A^T y \leq c - \tilde{z}\}\} \\ &= -\min_{\tilde{z}, y} \{-b^T y - z^T \tilde{z} + \frac{1}{2}\tilde{z}^T \tilde{z} \mid A^T y \leq c - \tilde{z}\}. \end{aligned}$$

We thus obtain another representation of $(-\varphi)^{**}(z) = -\varphi(z)$ as a solution of a convex quadratic program with linear constraints.

Since φ is not convex but concave, the segment $[0, \tilde{z}^0]$ on which the gradient of φ^* corresponds to the generalized Newton path of φ is given by

$$-\tilde{z}^0 = \nabla\varphi(z^0) = (A^T y(z^0) - c + z^0)^+ - z^0 \geq A^T y(z^0) - c.$$

We write this as $A^T y(z^0) \leq c - \tilde{z}^0$. By our assumption, $A^T y \leq c$ has a feasible solution and by convexity, $A^T y \leq c - t\tilde{z}^0$ has a feasible solution for $t \in [0,1]$, so that formula (3.20) is applicable along the line $t\tilde{z}^0$ for $t \in [0,1]$.

We consider the polyhedra (in the $\tilde{z}$-space) in which φ^* is quadratic. These polyhedra are bounded by the manifolds at which the active indices of strictly complementary solutions y of $\max_{y:\ A^T y \leq c - \tilde{z}} \{b^T y\}$ in (3.20) are changing. Unfortunately, there may be exponentially many points along the line $c - \tilde{z}$ where φ^* changes the quadratic representation.

Let $z \in \mathbb{R}^n$ be given in *general position* such that

$$(P_1) \qquad \text{minimize} \quad (c+z)^T x \quad \text{s.t.} \quad x \in \mathcal{P}$$

has a finite optimal solution, i.e. such that

$$(D_1) \qquad \text{maximize} \quad b^T y \quad \text{s.t.} \quad y \in \mathcal{D}_1 := \{y \ \mid \ A^T y \leq c + z\}$$

is feasible. In this case, (P_1) and (D_1) also have a unique optimal primal dual solution.

Consider the function $\phi : [0,1] \to \mathbb{R}$ defined by

$$\phi(t) := \underbrace{\min\{(c+tz)^T x \mid \ Ax = b,\ x \geq 0\}}_{(P_t)} = \underbrace{\max\{b^T y \mid \ A^T y \leq c + tz\}}_{(D_t)}. \quad (3.21)$$

As indicated, we refer to the parameterized problems by (P_t) and (D_t). The function ϕ is concave and piecewise linear.

Concavity follows directly from the definition of (D_t); if $y(t)$ is an optimal solution for (D_t), then $\lambda y(t_1) + (1-\lambda) y(t_2)$ is feasible for $(D_{\lambda t_1 + (1-\lambda) t_2})$, and hence the optimal value $\phi(\lambda t_1 + (1-\lambda) t_2)$ is at least $\lambda \phi(t_1) + (1-\lambda)\phi(t_2)$. #

Following the generalized Newton path for φ is identical to following the path of ϕ, and as shown in [3], this path may have an exponential number of linear segments.

From the approach introduced in the present chapter the conception arose to equate solving a primal-dual pair of linear conic programs with minimizing a certain function. The function $\hat{f}$ in the present chapter consists of terms that describe the duality gap and terms that describe primal and dual feasibility.

The function value of $\hat{f}$ measures the distance of a given point from the set of feasible points and the set of points that have a duality gap of zero.
In the next chapter this approach is generalized to linear second order cone programs. The primal-dual feasible set together with the equation for a duality gap of zero will be partitioned into an affine space K_1 and a closed convex cone K_2. Thus, solving the primal-dual pair of linear conic programs is equivalent to minimizing the sum of the distance functions of points (x, s) to K_1 and K_2.

Chapter 4

On the Regularity of Second Order Cone Programs and an Application to Solving Large Scale Problems

We now consider optimization problems over the cartesian product of second order cones. Of central importance is the nonsingularity of the standard primal-dual system for second order cone programs. Assuming Slater's condition and uniqueness and strict complementarity of the optimal solution we establish nonsingularity. This result is applied to the analysis of the augmented primal-dual method for solving linear programs over second order cones. The content of this chapter is published in [49].

4.1 Known results

We recall the definition of the second order cone of dimension n

$$\mathcal{Q}_n := \{x := (x_0; \bar{x}) = (x_0, x_1, \ldots, x_{n-1})^T \in \mathbb{R}^n \mid x_0 \geq \|\bar{x}\|_2\}$$

and the primal-dual pair of linear second order cone programs

$$(P^{SOC}) \qquad \begin{array}{ll} \min & c_1^T x_1 + \cdots + c_r^T x_r \\ \text{s. t.} & A_1 x_1 + \cdots + A_r x_r = b, \\ & x_i \in \mathcal{Q}_{n_i}, \text{ for } i = 1, \ldots, r \end{array}$$

and

$$(D^{SOC}) \qquad \begin{array}{ll} \max & b^T y \\ \text{s. t.} & A_i^T y + s_i = c_i, \text{ for } i = 1, \ldots, r, \\ & s_i \in \mathcal{Q}_{n_i}, \text{ for } i = 1, \ldots, r. \end{array}$$

For the second order cone, let

$$\text{bd } \mathcal{Q}_n := \{x \in \mathcal{Q}_n \mid x_0 = \|\bar{x}\|_2 \text{ and } x \neq 0\}$$

denote the boundary of $\mathcal{Q}_n$ *without* the origin 0 and let

$$\text{int } \mathcal{Q}_n := \{x \in \mathcal{Q}_n \mid x_0 > \|\bar{x}\|_2\}$$

denote the interior of $\mathcal{Q}_n$.
The second order cone is selfdual, i.e.

$$\mathcal{Q}_n^D = \{s \in \mathbb{R}^n \mid \langle x, s\rangle \geq 0 \, \forall \, x \in \mathcal{Q}_n\} = \mathcal{Q}_n,$$

where $\langle \cdot, \cdot \rangle$ denotes the standard scalar product on $\mathbb{R}^n$, given by

$$\langle c, x\rangle = c^T x.$$

We remind of the multiplication " $\circ$ " [1]:

$$u \circ v := \begin{pmatrix} u^T v \\ u_0 \bar{v} + v_0 \bar{u} \end{pmatrix}.$$

Lemma 2. *For vectors u and $v \in \mathcal{Q}_n$ the following statements hold:*
i) $u \circ v = 0 \Leftrightarrow u^T v = 0$
ii) $u \circ v = 0 \Leftrightarrow u = 0$ *or there exists* $\alpha \geq 0$ *such that*
$v_0 = \alpha u_0$ *and* $\bar{v} = -\alpha \bar{u}$ *for* $u \in$ *bd* $\mathcal{Q}_n$.

For a proof of this Lemma see Lemma 15 in [1].

For a vector $x \in \mathbb{R}^n$ the *arrow-shaped* matrix $Arw(x)$ is given by

$$Arw(x) := \begin{pmatrix} x_0 & \bar{x}^T \\ \bar{x} & x_0 I \end{pmatrix},$$

see e.g.[1]. It is easily verified that

$$x \in \mathcal{Q}_n (x \in int(\mathcal{Q}_n)) \text{ iff } Arw(x) \succeq \ 0 (Arw(x) \succ 0).$$

This holds due to the fact that the eigenvalues of $Arw(x)$ are given by x_0 with multiplicity $n-2$ and $x_0 - \|\bar{x}\|_2$ and $x_0 + \|\bar{x}\|_2$, each with multiplicity one.
Observe the following identity for $x \in \mathbb{R}^n, \bar{x} \neq 0$:

$$x = \frac{1}{2}(x_0 + \|\bar{x}\|_2) \begin{pmatrix} 1 \\ \frac{\bar{x}}{\|\bar{x}\|_2} \end{pmatrix} + \frac{1}{2}(x_0 - \|\bar{x}\|_2) \begin{pmatrix} 1 \\ -\frac{\bar{x}}{\|\bar{x}\|_2} \end{pmatrix}. \tag{4.1}$$

Thus, defining $w := \frac{1}{2} \begin{pmatrix} 1 \\ \frac{\bar{x}}{\|\bar{x}\|_2} \end{pmatrix}$ and $w' := \frac{1}{2} \begin{pmatrix} 1 \\ -\frac{\bar{x}}{\|\bar{x}\|_2} \end{pmatrix}$, relation (4.1) simplifies to

$$x = (x_0 + \|\bar{x}\|_2)w + (x_0 - \|\bar{x}\|_2)w'.$$

Here, $w = R_n w'$ with

$$R_n := \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

Note that w and w' lie on the boundary of $\mathcal{Q}_n$. If $x \in \text{bd } \mathcal{Q}_n$, equation (4.1) reduces to

$$x = \frac{1}{2}(x_0 + \|\bar{x}\|_2)w.$$

Furthermore, note that $w \circ w' = 0$ and therefore also $w^T w' = 0$. Thus, (4.1) is an orthogonal decomposition of x.
In the following we will use the definition of nondegeneracy as given in [1].

Definition 1. *Let $\mathcal{T}_x$ be the tangent space at x to $\mathcal{Q}_n$. Then a primal-feasible point x is primal nondegenerate if*

$$\mathcal{T}_x + Ker(A) = \mathbb{R}^n;$$

otherwise x is primal degenerate.

Here, $Ker(A)$ denotes the kernel of the matrix A, i.e. $Ker(A) = \{x : Ax = 0\}$. The definition of nondegeneracy states that $\mathcal{Q}$ and the affine set $\mathcal{A} := \{x : Ax = b\}$ intersect transversally at x, i.e. the tangent spaces at x to $\mathcal{A}$ and $\mathcal{Q}_n$ span $\mathbb{R}^n$. Let x be given in a cartesian product of second order cones.

We assume without loss of generality that all blocks x_i with $x_i \in \text{bd } \mathcal{Q}_{n_i}$ are grouped together in x_B; all blocks x_i with $x_i = 0$ are grouped together in x_N; and all blocks x_i with $x_i \in \text{int } \mathcal{Q}_{n_i}$ are grouped together in x_I, i.e. $x = (x_B; x_I; x_N)$. We partition the matrix A in the same manner, that is $A = (A_B, A_I, A_N)$. We assume further that x_B and A_B have p blocks, i.e. $x_B = (x_1; \ldots; x_p)$ and $A_B = (A_1, \ldots, A_p)$ and that the dimensions of x_I and x_N are given by n_I and n_N. When $i \in \{1, \ldots, p\}$, we call x_i a *boundary block*.

For $x_i \in \text{int } \mathcal{Q}_{n_i}$, the tangent space to $\mathcal{Q}_{n_i}$ is all of $\mathbb{R}^{n_i}$. For $x_i = 0$, the tangent space is $\{0\}$. Now, for $x \in \text{bd } \mathcal{Q}_{n_i}$ we can write $x_i = \alpha_i w_i$ with $\alpha_i = x_{i0} + \|\bar{x}_i\|_2 = 2x_{i0}$. Here, the tangent space is given by the $n_i - 1$ dimensional space $\{z : w_i'^T z = 0\}$, where $w_i' = R_{n_i} w_i$. Primal nondegeneracy then means that

$$(\mathcal{T}_{x_B} \times \mathcal{T}_{x_I} \times \mathcal{T}_{x_N}) + Ker((A_B, A_I, A_N)) = \mathbb{R}^n.$$

As shown in [1], primal nondegeneracy is equivalent to

$$(((\alpha_1 w_1^{'}) \times \cdots \times (\alpha_p w_p^{'})) \times \{0\} \times \mathbb{R}^{n_N}) \cap \text{Span}((A_B, A_I, A_N)^T) = \{0\}. \quad (4.2)$$

For any $x = \alpha_i w_i + \tilde{\alpha}_i w_i'$ (with $\tilde{\alpha}_i = x_{i0} - \|\bar{x}_i\|_2$), let $\hat{Q}_i \in \mathbb{R}^{n_i \times (n_i - 2)}$ be a matrix whose columns are orthonormal and orthogonal to w_i and w_i'. Then, the columns of the orthogonal matrix

$$Q_i = (\sqrt{2} w_i', \hat{Q}_i, \sqrt{2} w_i)$$

are the eigenvectors of $Arw(x_i)$. For a boundary block x_i and when $s_i \in \text{bd}(\mathcal{Q}_{n_i})$ and $x_i \circ s_i = 0$ then $x_i = \alpha_i w_i$ and $s_i = \beta_i w_i'$, where $w_i' = R_{n_i} w_i$, $\alpha_i = x_{i0} + \|\bar{x}_i\|_2 = 2x_{i0} > 0$ and $\beta_i = s_{i0} + \|\bar{s}_i\|_2 = 2s_{i0} > 0$. In this case (see also Theorem 6 in [1]), the matrices $Arw(x_i)$ and $Arw(s_i)$ commute and share a system of eigenvectors, namely setting $Q_i = (\sqrt{2} w_i', \hat{Q}_i, \sqrt{2} w_i)$, we may write

$$Q_i^T Arw(x_i) Q_i = \begin{pmatrix} 2x_{i0} & 0 & 0 \\ 0 & x_{i0} I & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

and

$$Q_i^T Arw(s_i) Q_i = \begin{pmatrix} 0 & 0 & 0 \\ 0 & s_{i0} I & 0 \\ 0 & 0 & 2s_{i0} \end{pmatrix}.$$

Recalling the decompositions $A_B = (A_1, \ldots, A_p)$ and $x_B = (x_1; \ldots; x_p)$, we repeat Theorem 20 from [1] without proof.

Theorem 2. *For each boundary block $x_i = \alpha_i w_i$ let $Q_i = (\sqrt{2}w_i', \hat{Q}_i, \sqrt{2}w_i) =: (\sqrt{2}w_i', \bar{Q}_i)$ be the matrix of eigenvectors of $Arw(x_i)$. Then $x = (x_1; \ldots; x_p; x_I; x_N)$ is primal nondegenerate if and only if the matrix*

$$(A_1\bar{Q}_1, \ldots, A_p\bar{Q}_p, A_I)$$

has linearly independent rows.

Likewise, we define dual nondegeneracy:

Definition 3. *A dual feasible point (y, s) is dual nondegenerate if*

$$\mathcal{T}_s + Span(A^T) = \mathbb{R}^n;$$

otherwise (y, s) is dual degenerate.

We partition the dual variable s into three parts $s = (s_{\tilde{B}}; s_{\tilde{N}}; s_{\tilde{I}})$ and the matrix A accordingly ($A = (A_{\tilde{B}}, A_{\tilde{N}}, A_{\tilde{I}})$), where the dimensions of the blocks $s_{\tilde{B}}, s_{\tilde{N}}$ and $s_{\tilde{I}}$ are given by $n_{\tilde{B}}, n_{\tilde{N}}$ and $n_{\tilde{I}}$ and $\tilde{B}$ contains the indices of the boundary blocks of s, while $s_i = 0$ for i in $\tilde{N}$ and $s_i \in \text{int}(\mathcal{Q}_{n_i})$ for i in $\tilde{I}$. Then, this definition is equivalent to

$$(\mathcal{T}_{s_{\tilde{B}}} \times \mathcal{T}_{s_{\tilde{N}}} \times \mathcal{T}_{s_{\tilde{I}}}) + \text{Span}((A_{\tilde{B}}, A_{\tilde{N}}, A_{\tilde{I}})^T) = \mathbb{R}^n. \tag{4.3}$$

Note that $B = \tilde{B}$, $N = \tilde{I}$ and $I = \tilde{N}$ holds for strictly complementary solutions $(x; s)$. Assume that $s_{\tilde{B}}$ and $A_{\tilde{B}}$ consist of q blocks, i.e. $s_{\tilde{B}} = (s_1; \ldots; s_q)$ and $A_{\tilde{B}} = (A_1, \ldots, A_q)$. For each boundary block s_i we may write $s_i = \beta_i w_i'$, $i = 1, \ldots, q$ with $\beta_i > 0$. Taking as in [1] the orthogonal complement of (4.3), a solution is dual nondegenerate iff

$$((\beta_1 w_1) \times \cdots \times (\beta_q w_q) \times \mathbb{R}^{n_{\tilde{N}}} \times \{0\}) \cap \text{Ker}((A_1, \ldots, A_q, A_{\tilde{N}}, A_{\tilde{I}})) = \{0\}. \tag{4.4}$$

Let $n_1, \ldots, n_q$ be the dimensions of the blocks $s_1, \ldots, s_q$. We repeat Theorem 21 from [1] without proof.

Theorem 4. *The dual feasible solution (y, s) with $s = (s_1; \ldots; s_q; s_N; s_I)$ is dual nondegenerate if and only if the matrix*

$$(A_1 R_{n_1} s_1, \ldots, A_q R_{n_q} s_q, A_{\tilde{N}})$$

has linearly independent columns.

4.2 A perturbation theorem

In this section we give a proof of the analogue of Theorem 1 in [17] for the case of second order cone programs. In the following Lemma we first show, that uniqueness of the optimal solution $(x^*; s^*)$ implies already primal and dual nondegeneracy. The reverse is shown in Theorem 22 in [1].

Lemma 3. *For the pair of second order cone programs (P^{SOC}) and (D^{SOC}) the following inclusions hold:*

1. *If a primal optimal solution $x^* = (x_B^*; x_I^*; x_N^*)$ of the linear second order cone program (P^{SOC}) is unique, then the dual optimal solution $s^* = (s_{\tilde{B}}^*; s_{\tilde{N}}^*; s_{\tilde{I}}^*)$ is nondegenerate.*

2. *If a dual optimal solution y^*, s^* of the linear second order cone program (D^{SOC}) is unique, then the primal optimal solution x^* is nondegenerate.*

Note that strict complementarity is not needed to establish nondegeneracy.

Proof.
We prove the contrapositive of both statements.

1. Assume that s^* is dual degenerate. Then, according to (4.4), there exists

$$\tilde{z} \in ((\beta_1 w_1) \times \cdots \times (\beta_q w_q)) \times \mathbb{R}^{n_{\tilde{N}}} \times \{0\} \cap \mathrm{Ker}((A_1, \ldots, A_q, A_{\tilde{N}}, A_{\tilde{I}}))$$

with $\tilde{z} \neq 0$. Therefore $\tilde{z} \in \mathrm{Ker}(A)$ and hence, $A(x^* + \tilde{z}) = b$. Furthermore, since $w_i^T w_i' = 0$, s^* and $\tilde{z}$ obviously satisfy $(s_i^*)^T \tilde{z}_i = 0$ for all $i = 1, \ldots, r$. Therefore we have $(s^*)^T \tilde{z} = 0$. Now, observe that

$$c^T(x^* + \tilde{z}) = c^T x^* + c^T \tilde{z} = c^T x^* + (A^T y^* + s^*)^T \tilde{z}$$

$$= c^T x^* + (y^*)^T A\tilde{z} + (s^*)^T \tilde{z} = c^T x^*.$$

Since $x^* + \epsilon\tilde{z} \in \mathcal{Q}_n$ for $\epsilon > 0$ small enough, $(x^* + \epsilon\tilde{z})$ and s^* satisfy the complementarity condition $(x^* + \epsilon\tilde{z}) \circ s^* = 0$, which is equivalent to $(x_i^* + \epsilon\tilde{z}_i) \circ s_i^* = 0$ for all $i = 1, \ldots, r$. Therefore $x^* + \epsilon\tilde{z}$ is another optimal solution for the primal linear second order cone program (P^{SOC}) in contradiction to uniqueness of the primal optimal solution.

2. Now assume that x^* is primal degenerate. Then, according to (4.2), there exists

$$z \in ((\alpha_1 w_1^{'}) \times \cdots \times (\alpha_p w_p^{'})) \times \{0\} \times \mathbb{R}^{n_N}) \cap \mathrm{Span}((A_1, \ldots, A_p, A_I, A_N)^T)$$

with $z \neq 0$. Therefore $z \in \mathrm{Span}(A^T)$ and thus there exists a vector $\tilde{y}$ such that $A^T\tilde{y} = z$, more precisely $A_i^T\tilde{y} = z_i$ for $i = 1, \ldots, r$. Then, $(y - \tilde{y}, s^* + z)$ satisfies $A_i^T(y^* - \tilde{y}) + s_i^* + z_i = c_i$ for all $i = 1, \ldots, r$. Furthermore, since $w_i^T w_i^{'} = 0$, x^* and z obviously satisfy $(x_i^*)^T z_i = 0$ for all $i = 1, \ldots, r$ and therefore we have $(x^*)^T z = 0$. Now, observe that

$$b^T(y^* - \tilde{y}) = b^T y^* - (Ax^*)^T\tilde{y} = b^T y^* - (x^*)^T A^T y^*$$

$$= b^T y^* - (x^*)^T z = b^T y^*.$$

Since $s^* + \epsilon z \in \mathcal{Q}$ for $\epsilon > 0$ small enough, x^* and $s^* + \epsilon z$ satisfy the complementarity condition $x^* \circ (s^* + \epsilon z) = 0$, which is equivalent to $x_i^* \circ (s_i^* + \epsilon z_i) = 0$ for all $i = 1, \ldots, r$. Therefore $(y^* - \epsilon\tilde{y}, s^* + \epsilon z)$ is another optimal solution for the dual linear second order cone program (D^{SOC}) in contradiction to uniqueness of the dual optimal solution.

□

We make the following assumption for the remainder of this chapter.

Assumption 2. *We assume that (P^{SOC}) and (D^{SOC}) are strictly feasible and that there is a unique and strictly complementary solution $z^* = (x^*; s^*)$ of (P^{SOC}) and (D^{SOC}) satisfying $x^* + s^* \in int(\mathcal{Q})$.*

The condition

$$x^* + s^* \in int(\mathcal{Q})$$

is called *strict complementarity condition.* This condition is equivalent to $x_i^* + s_i^* \in$ int $\mathcal{Q}_{n_i}$ for all $i = 1, \ldots, r$. As shown in Corollary 24 in [1], strict complementarity for problems (P^{SOC}) and (D^{SOC}) holds iff for all blocks either both x_i^* and s_i^* are in bd $\mathcal{Q}_{n_i}$, or if one is zero and the other is in the interior of $\mathcal{Q}_{n_i}$.

Now we present our result on the perturbation of strictly complementary solutions of pairs of linear second order cone programs of the form (P) and (D).

Theorem 5. *Let matrices $A_i \in \mathbb{R}^{m\times n_i}, i = 1,\dots,r$ with $n_1 + \cdots + n_r =: n, A := (A_1,\dots,A_r)$ and vectors $b \in \mathbb{R}^m, c_i \in \mathbb{R}^{n_i}, i = 1,\dots,r$ with $c := (c_1;\dots;c_r)$ be the data of a pair (P^{SOC}) and (D^{SOC}) of primal and dual linear second order cone programs. Under Assumption 2, i.e. for x^*, y^*, s^* with*

$$Ax^* = b,\ A^Ty^* + s^* = c,\ x^* \circ s^* = 0,\ x^* \in \mathcal{Q},\ s^* \in \mathcal{Q},\ x^* + s^* \in int\ \mathcal{Q}, \tag{4.5}$$

the following statements hold.
If the data of (P^{SOC}) and (D^{SOC}) is changed by sufficiently small perturbations ΔA, Δb and Δc, then the optimal solutions of the perturbed second order cone programs are differentiable functions of the perturbations. Furthermore, the derivatives

$$\dot{x} := D_{A,b,c}x^*[\Delta A, \Delta b, \Delta c],\ \dot{y} := D_{A,b,c}y^*[\Delta A, \Delta b, \Delta c]$$

$$\text{and } \dot{s} := D_{A,b,c}s^*[\Delta A, \Delta b, \Delta c],$$

of the solution x, y, s at x^*, y^*, s^* satisfy

$$\begin{aligned} A\dot{x} &= \Delta b - \Delta A x^*, \\ A^T\dot{y} + \dot{s} &= \Delta c - \Delta A^T y^*, \\ \dot{x} \circ s^* + x^* \circ \dot{s} &= 0, \end{aligned} \tag{4.6}$$

and system (4.6) is nonsingular.

First of all, we repeat Definition 26 and Lemma 27 from [1].

Definition 6. *Let*

$$J = \begin{pmatrix} 0 & 0 & B_1^T & I & 0 \\ 0 & 0 & B_2^T & 0 & I \\ B_1 & B_2 & 0 & 0 & 0 \\ V_1 & 0 & 0 & U_1 & 0 \\ 0 & V_2 & 0 & 0 & U_2 \end{pmatrix},$$

where the first, second, third, fourth and fifth block rows and columns have dimensions m, $n-m$, m, m and $n-m$, respectively. We say J is a primal-dual block canonical matrix (PDBC matrix for short) if

1. *$B_1 \in \mathbb{R}^{m\times m}$ is a nonsingular matrix,*

2. $V_2 \in \mathbb{R}^{(n-m)\times(n-m)}$ *and* $U_1 \in \mathbb{R}^{m\times m}$ *are symmetric positive definite,*

3. $V_1 \in \mathbb{R}^{m\times m}$ *and* $U_2 \in \mathbb{R}^{(n-m)\times(n-m)}$ *are symmetric positive semidefinite,*

4. V_1 *and* U_1 *commute and likewise* V_2 *and* U_2 *commute.*

Lemma 4. *Every primal-dual block canonical matrix is nonsingular.*

For a proof of Lemma 4 see [1].

Proof *of Theorem 5.*
We first observe that uniqueness of x^* and y^* implies that $\text{rank}(A) = m$. Indeed, assume that $\text{rank}(A^T) = \text{rank}(A) < m$. Then, there exists a vector $\Delta y \in \mathbb{R}^m, \Delta y \neq 0$, such that $A^T\Delta y = 0$. If $b^T\Delta y = 0$, then $y^* + \Delta y$ is also an optimal solution of (D^{SOC}) in contradiction to the uniqueness of y^*. If $b^T\Delta y \neq 0$, then (D^{SOC}) does not have a finite optimal solution, which is again a contradiction.
Slater's condition states that

$$\exists\ y,\ x_0 > \|\bar{x}\|_2,\ s_0 > \|\bar{s}\|_2\ :\ Ax = b,\ A^Ty + s = c.$$

By continuity and the observation that rank $A = m$, Slater's condition is also satisfied for all sufficiently small perturbations of the problem data. Hence, the perturbed problem possesses optimal solutions $x^* + \Delta x$, $y^* + \Delta y$ and $s^* + \Delta s$. The optimality conditions of the perturbed problem are given by

$$(x^* + \Delta x)_0 \geq \|\bar{x}^* + \Delta\bar{x}\|_2,\ (s^* + \Delta s)_0 \geq \|\bar{s}^* + \Delta\bar{s}\|_2,$$

and

$$\begin{aligned} (A + \Delta A)(x^* + \Delta x) &= b + \Delta b, \\ (A^T + \Delta A^T)(y^* + \Delta y) + s^* + \Delta s &= c + \Delta c, \\ (x^* + \Delta x) \circ (s^* + \Delta s) &= 0. \end{aligned} \tag{4.7}$$

Substracting from these equations the first three equations of (4.5) yields

$$\begin{aligned} (A + \Delta A)\Delta x &= \Delta b - \Delta A x^*, \\ (A^T + \Delta A^T)\Delta y + \Delta s &= \Delta c - \Delta A^T y^*, \\ \Delta x \circ s^* + x^* \circ \Delta s &= -\Delta x \circ \Delta s. \end{aligned} \tag{4.8}$$

The left-hand side of (4.8) represents a system of $N := m+2n+2$ linear equations for the N unknowns Δx, Δy and Δs. Since the existence of Δx, Δy and Δs follows from Slater's condition, it suffices to verify uniqueness and differentiability. Neglecting the second-order terms in (4.8), we then obtain the result claimed in (4.6).

We show that (4.6) has a unique solution. We replicate the proof of Theorem 28 in [1], as there are some confusing typos in the proof given there.
The homogeneous version of equation (4.6) is given by

$$\begin{aligned} A\Delta x &= 0 \\ A^T\Delta y + \Delta s &= 0 \\ \Delta x \circ s^* + x^* \circ \Delta s &= 0. \end{aligned} \tag{4.9}$$

The first two equations in (4.9) imply that $x^* + \Delta x$ and $y^* + \Delta y$, $s^* + \Delta s$ satisfy $A(x^* + \Delta x) = b$ and $A^T(y^* + \Delta y) + s^* + \Delta s = c$.
In block notation the Jacobian of (4.9) is given by

$$J := \begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & A_1^T & I & & & & & \\ \vdots & \ddots & \vdots & 0 & 0 & \vdots & & \ddots & & & & \\ 0 & \cdots & 0 & 0 & 0 & A_p^T & & & I & & & \\ 0 & \cdots & 0 & 0 & 0 & A_I^T & & & & I & & \\ 0 & \cdots & 0 & 0 & 0 & A_N^T & & & & & I & \\ A_1 & \cdots & A_p & A_I & A_N & 0 & 0 & \cdots & 0 & 0 & 0 & \\ S_1 & & & & & 0 & X_1 & & & & & \\ & \ddots & & & & \vdots & & \ddots & & & & \\ & & S_p & & & 0 & & & X_p & & & \\ & & & S_N & & 0 & & & & X_I & & \\ & & & & S_I & 0 & & & & & X_N & \end{pmatrix}. \tag{4.10}$$

Here, $S_i := Arw(s_i)$, $X_i := Arw(x_i)$, $i = 1, \ldots, p$, S_I and S_N are block diagonal matrices, where the i-th diagonal block is given by $S_i = Arw(s_i)$ for i in I, respectively i in N. Likewise, X_I and X_N are block diagonal matrices, where the i-th diagonal block is given by $X_i = Arw(x_i)$ for i in I, respectively i in N. Note that $S_N = 0$, $X_N = 0$ and S_I and X_I are symmetric positive definite matrices. As mentioned before, $Arw(x_i)$ and $Arw(s_i)$ commute and thus share a system of eigenvectors.

We define P as the block diagonal matrix

$$P := (Q_1 \oplus \cdots \oplus Q_p \oplus I \oplus I) \oplus I \oplus (Q_1 \oplus \cdots \oplus Q_p \oplus I \oplus I),$$

where $Q_i = (\sqrt{2}w_i', \hat{Q}_i, \sqrt{2}w_i)$ is the matrix containing the eigenvectors of $Arw(x_i)$ for boundary blocks x_i for $i = 1, \ldots, p$.
Now, we form the matrix $P^T JP$. Uniqueness of the dual optimal solution implies primal nondegeneracy, i.e. the matrix

$$\hat{A} = (A_1\bar{Q}_1, \cdots, A_p\bar{Q}_p, A_I)$$

has linearly independent rows. Uniqueness of the primal optimal solution implies dual nondegeneracy, i.e. the matrix

$$\check{A} = (A_1 R_{n_1} s_1, \cdots, A_p R_{n_p} s_p, A_I)$$

has linearly independent columns. Strict complementarity implies that $x_i = \alpha_i R_{n_i} s_i$ for $i = 1, \ldots, p$ and hence also the matrix

$$\check{A}' = (A_1 x_1, \cdots, A_p x_p, A_I)$$

has linearly independent columns. Now, we take all $p + n_I$ columns of $\check{A}'$ together with some $m - p - n_I$ columns of $\hat{A}$ and form an $m \times m$ nonsingular matrix B_1. The remaining $n - m$ columns of $\hat{A}$ form a matrix B_2. We sort the columns of the last $n = n_B + n_I + n_N$ rows of $P^T JP$ according to the decomposition of B_1 and B_2.
Thus, we obtain matrices $V_1 \in \mathbb{R}^{m\times m}$, $V_2 \in \mathbb{R}^{(n-m)\times(n-m)}$ and $U_1 \in \mathbb{R}^{m\times m}$, $U_2 \in \mathbb{R}^{(n-m)\times(n-m)}$. Note that V_2 and U_1 contain the blocks of S_I, respectively X_I, and, furthermore, the columns of diagonal matrices arising from $Q_i^T S_i Q_i$, respectively $Q_i^T X_i Q_i$, for $i = 1, \ldots, p$ with columns corresponding to their zero eigenvalues removed. Thus, the matrices V_2 and U_1 are symmetric positive definite. The remaining columns of $Q_i^T S_i Q_i$ and $Q_i^T X_i Q_i$ together with S_N, respectively X_N, are assigned to V_1 and U_2. Then, these matrices are symmetric positive semidefinite. Observe, that V_1 and U_1 commute and likewise do V_2 and U_2. Hence, the matrix $P^T JP$ is a PDBC matrix and thus nonsingular. Therefore, J is nonsingular and thus (4.6) has a unique solution.

We have shown that system (4.6) has a unique solution. Therefore, the implicit function theorem can be applied to the system (4.5). As we have just seen, the linearization of (4.5) at x^*, y^*, s^* is nonsingular, and hence (4.5) has a differentiable and locally unique solution. □

4.3 A reformulation of the conic program

Below, we follow the concept of [25]. In [8] it is shown that regularity concepts do not translate in a straightforward way from semidefinite programs to second order cone programs, and here, as well, some of the necessary modifications are not straightforward.
We assume that (P^{SOC}) and (D^{SOC}) have feasible solutions and that (P^{SOC}) or (D^{SOC}) satisfies Slater's condition. In this case, finding an optimal solution for (P^{SOC}) and (D^{SOC}) is equivalent to finding $z = (x; s) \in K_1 \cap K_2$, where

$$K_1 := (\mathcal{L} + b) \times (\mathcal{L}^{\perp} + c) \cap \{(x; s) \mid \langle c, x\rangle + \langle b, s\rangle = \langle b, c\rangle\}$$

and

$$K_2 := \mathcal{Q}_{n+1} \times \mathcal{Q}_{n+1}^{D}.$$

Here, K_1 and K_2 are closed and convex, more precisely, K_1 is an affine subspace and K_2 a pointed, closed, convex cone with nonempty interior.

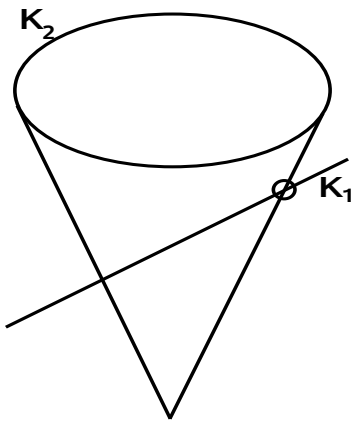

Figure 4.1: Intersection of K_1 with K_2

For a closed set $\mathcal{C}$ and a vector $\tilde{u}$ we denote the distance of $\tilde{u}$ to $\mathcal{C}$ by

$$d(\tilde{u}, \mathcal{C}) := \min\{\|u - \tilde{u}\|_2 \mid u \in \mathcal{C}\}.$$

This distance is given by

$$d(\tilde{u}, \mathcal{C}) = \|\tilde{u} - \Pi_{\mathcal{C}}(\tilde{u})\|_2,$$

where $\Pi_{\mathcal{C}}(\tilde{u})$ denotes the projection of $\tilde{u}$ onto the set $\mathcal{C}$. The projection onto the second order cone is given by (see section 4.2 in [43])

$$\Pi_{\mathcal{Q}_n}(x) = \begin{cases} \frac{1}{2}\left(1 + \frac{x_0}{\|\bar{x}\|_2}\right)(\|\bar{x}\|_2; \bar{x}) & \text{if} \quad |x_0| < \|\bar{x}\|_2, \\ (x_0, \bar{x}) & \text{if} \quad x_0 \geq \|\bar{x}\|_2, \\ 0 & \text{if} - x_0 \geq \|\bar{x}\|_2. \end{cases}$$

A discussion regarding the computation of the projection onto K_1 is given in [25]. We just recall that the projection of x onto $\mathcal{L} + b$ is given by

$$\Pi_{\mathcal{L}+b}(x) = x - A^T(AA^T)^{-1}A(x - b)$$

and that the projection of s onto $\mathcal{L}^{\perp} + c$ can be computed via

$$\Pi_{\mathcal{L}^{\perp}+c}(s) = s - (I - A^T(AA^T)^{-1}A)(s - c).$$

Now, solving (P^{SOC}) and (D^{SOC}) is equivalent to finding z such that

$$\phi(z) := \frac{1}{2}(d(z, K_1)^2 + d(z, K_2)^2) = 0,$$

i.e. such that ϕ is minimized. Function and gradient evaluations of ϕ can be computed efficiently analogously to [25], and the minimization of ϕ is possible by cg- or limited memory BFGS-type algorithms.

4.4 Solving (P^{SOC}) and (D^{SOC})

The affine space K_1 can be decomposed into $K_1 = z^{(0)} + N_1$ where $z^{(0)}$ is any fixed vector in K_1 and N_1 is a linear subspace. Following the approach in [25], we restrict ϕ to K_1 and define the function $\tilde{\phi}$ by

$$\tilde{\phi}(\tilde{z}) := \frac{1}{2}\|d(\tilde{z}, K_2)\|_2^2 \text{ for } \tilde{z} \in K_1.$$

The function $\tilde{\phi}$ is not defined outside K_1. For linear programs, i.e. for the case $K_2 = (\mathcal{Q}_1)^n$ this is a differentiable, convex, piecewise quadratic function similar to the function considered in chapter 3.
In the next subsection we give an example with a unique and strictly complementary optimal solution z^* of (P^{SOC}) and (D^{SOC}) such that there are directions $\Delta z \in N_1$, such that $\tilde{\phi}(z^* + \lambda \Delta z)$ grows in the order of λ^4 and other directions $\tilde{\Delta} z \in N_1$ such that $\tilde{\phi}(z^* + \lambda \tilde{\Delta} z)$ grows in the order of λ^2.

4.4.1 A small example

In this subsection we give a small example of a pair of second order cone programs (P^{SOC}) and (D^{SOC}) satisfying Assumption 2 such that the (generalized) Hessian of $\tilde{\phi}$ has an unbounded condition number for z near z^*.
Let $A = [1, 0, 1]$, $b = [1; 0; 0]$ and $c = [1; 0; 0]$ be the data of a primal-dual pair of second order cone programs. The primal-dual optimal solution $(x^*; s^*) = ((\frac{1}{2}; 0; \frac{1}{2}); (\frac{1}{2}; 0; -\frac{1}{2}))$ is unique and strictly complementary. The space $N_1 := \mathcal{L} \times \mathcal{L}^{\perp} \cap \{(\Delta x; \Delta s) | c^T \Delta x + b^T \Delta s = 0\}$ is given by

$$N_1 = \{\Delta z = (\Delta x; \Delta s) = ((\beta; \alpha; -\beta); (-\beta; 0; -\beta)) \mid \alpha, \beta \in \mathbb{R}\}.$$

By construction, $z^* + \Delta z \in K_1$ for $\Delta z \in N_1$. For small $|\alpha|$, $|\beta|$ it is easily verified that

$$d(z^* + \Delta z, K_2) = O(\alpha^2) \text{ if } \beta = 0, \qquad d(z^* + \Delta z, K_2) = O(|\beta|) \text{ if } \alpha = 0.$$

Thus, for directions $z^* + \lambda \Delta z$ the function $\tilde{\phi}$ grows in the order of λ^4 when $\beta = 0$ and in the order of λ^2 when $\alpha = 0$. Minimizing $\tilde{\phi}$ by some conjugate gradient scheme would result in a very slow algorithm. Therefore we derive a regularization of $\tilde{\phi}$.

4.4.2 A local regularization

Let

$$\hat{f}((x; s)) := \|x \circ s\|_2^2.$$

Then, $\hat{f}$ is minimized at the optimal solution $z^* = (x^*; s^*)$.

(P^{SOC}) and (D^{SOC}) can be solved in two stages (cf.[25]), the first one minimizing $\tilde{\phi}$ for $\tilde{z} \in K_1$, and when convergence of this stage is slow, starting a second stage minimizing $\tilde{\phi} + \hat{f}$ for $\tilde{z} \in K_1$. Note: In the following we will only consider points in K_1. The restriction of $\phi + \hat{f}$ to K_1 will be denoted by Ψ,

$$\Psi(z) := \phi(z) + \hat{f}(z), \text{ for } z \in K_1.$$

The gradient of Ψ is not everywhere differentiable. However, as stated in the next lemma, it satisfies some weaker smoothness properties. For the definition of semismoothness and the relevance of semismoothness for Newton-type algorithms, see e.g. [37, 46].

Lemma 5. *The gradient of Ψ is strongly semismooth and the generalized Hessian is positive definite at z^*.*

Proof. First, we prove strong semismoothness of the gradient of Ψ. Proposition 4.3 in [10] states that the projection of a vector x onto the second order cone is strongly semismooth. For $i = 1, \ldots, r$ let $f_i(x)$ be the projection of a vector $x = (x_1; \ldots; x_r)$ onto the second order cone for the i-th component of x and the projection onto the zero vector for the remaining components of x. Then, the projection of $x = (x_1; \ldots; x_r)$ onto the second order cone can be written as a sum of r projections, namely $f_1(x) + \cdots + f_r(x)$. Theorem 5 in [37] states, that the sum of semismooth functions is semismooth and so is the gradient of Ψ.
Let a perturbation $\Delta z = (\Delta x; \Delta s)$ with $z^* + \Delta z \in K_1$ and $\|(\Delta x; \Delta s)\|_2 = 1$ be given. It suffices to show that there exists a $\sigma > 0$ independent of Δz such that

$$\frac{1}{2} d(z^* + \lambda \Delta z, K_2)^2 + \hat{f}(z^* + \lambda \Delta z) \geq \lambda^2 \sigma. \tag{4.11}$$

As shown in Theorem 5, the following system for the unknowns $(\Delta x; \Delta y; \Delta s)$:

$$\begin{aligned} A\Delta x &= p, \\ A^T \Delta y + \Delta s &= q, \\ x^* \circ \Delta s + \Delta x \circ s^* &= r, \end{aligned} \tag{4.12}$$

is nonsingular. We eliminate the variable Δy from the second equation of (4.12). To this end let $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^{n-m}$ be a linear operator of full rank such that $\mathcal{F}(A^T y) = 0$ for all $y \in \mathbb{R}^m$.

Let $\tilde{q} := \mathcal{F}(q)$. Straightforward calculations show that the two systems (4.12) and

$$M \begin{pmatrix} \Delta x \\ \Delta s \end{pmatrix} := \begin{pmatrix} A\Delta x \\ \mathcal{F}(\Delta s) \\ \Delta x \circ s^* + x^* \circ \Delta s \end{pmatrix} = \begin{pmatrix} p \\ \tilde{q} \\ r \end{pmatrix} \tag{4.13}$$

are equivalent and thus system (4.13) has full rank as well.
Note that $\|(p;\tilde{q};r)\|_2 := \|M(\Delta x;\Delta s)\|_2 \geq \frac{1}{\|M^{-1}\|_2}$, since $\|(\Delta x;\Delta s)\|_2 = 1$. Given that $z^* + \Delta z \in K_1$, it follows that $p = 0$ and $\tilde{q} = 0$. Hence, $\|r\|_2 \geq \frac{1}{\|M^{-1}\|_2}$ which means that

$$\hat{f}(z^* + \lambda \Delta z) \geq \sigma \lambda^2,$$

where $\sigma = \frac{1}{\|M^{-1}\|_2^2}$. □

By Theorem 3.2 in [46], Lemma 5 implies local quadratic convergence of Newton's method for minimizing Ψ. Note that in contrast to the analysis in [25] the function $\tilde{\phi}$ is not needed to show the positive definiteness of the (generalized) Hessian of Ψ when considering second order cone programs.

Chapter 5

Application

In this chapter we present an application of the primal-dual method of the previous chapter. Our experiments deal with the cone of completely positive matrices. This cone and it's dual will be introduced first. Then, we will explain how the problem of deciding whether a given matrix is contained in the set of completely positive matrices can be related to a second order cone program and - in some cases - be solved via the primal-dual method. We provide an algorithm that either proves that a given matrix B is in C^* or converges to a matrix $\bar{S}$ that is 'close to' C^*. Moreover, a regularization step is presented to improve convergence of this algorithm, whenever it stagnates.

5.1 Completely Positive Matrices

The cone of *copositive matrices* is given by

$$\mathcal{C}_n = \{X \in I\!R^{n\times n} | X = X^T, y^T X y \geq 0 \ \forall \ y \in I\!R^n_+\}.$$

The dual cone of $\mathcal{C}_n$ is the cone of *completely positive matrices*

$$\mathcal{C}_n^* = \left\{ X \in I\!R^{n\times n} | X = \sum_{k\in K} v^k (v^k)^T \text{ for some finite } \{v^k\}_{k\in K} \in I\!R^n_+ \right\}.$$

Both, $\mathcal{C}_n$ and $\mathcal{C}_n^*$ are closed, convex cones of full dimension.
The dimension n will always be clear from the context, so the subscript will be dropped from now on.

With the definition of these cones, we may write the linear conic program over the cone of copositive matrices and it's dual as

$$(P^{COP}) \qquad \text{minimize} \quad c^T x \quad \text{s.t.} \quad x \in \mathcal{C} \cap (\mathcal{L} + b)$$

and

$$(D^{CPP}) \qquad \text{maximize} \quad \bar{b}^T y \quad \text{s.t.} \quad c - A^T y =: s \in \mathcal{C}^* \cap (\mathcal{L}^\perp + c).$$

As for the problems considered in chapter 2, if Slater's condition is satisfied for (P^{COP}) or (D^{CPP}), then the duality gap of the optimal values of (P^{COP}) and (D^{CPP}) is zero. Note, that in contrast to the cones considered in chapter 2, the cone of copositive matrices is not self-dual.

5.1.1 The cp-rank

By definition, for any $B \in C^*$ there exists a natural number p and a $n \times p$-matrix $X \geq 0$ such that $B = XX^T$.

For a given matrix $B \succeq 0$ the algorithm of the present chapter aims at generating a matrix $X \geq 0$ such that $B = XX^T$ holds true. If the algorithm succeeds then the matrix X provides a certificate for the statement $B \in C^*$. The dimension p of the $n \times p$ matrix X is discussed next. Evidently, when $B \in C^* \subset \mathcal{S}^n_+$, the Cholesky factor $L \in I\!R^{n \times p}$ of $B = LL^T$ can be computed with $p \leq n$. ($p < n$ when B has zero eigenvalues.) On the other hand, even when $B \in C^*$, the matrix L typically is not nonnegative, and, as discussed next, the choice $p \leq n$ is not suitable in general.

Given a matrix $B \in C^*$ the minimal number p for which there is a $n \times p$-matrix $X \geq 0$ such that $B = XX^T$ is called the cp-rank of B, see e.g. [4].

Let E be the $l \times l$ all-ones matrix and I the $l \times l$ identity matrix. Then, it is straightforward to verify that

$$\hat{S} := \begin{pmatrix} lI & E \\ E & lI \end{pmatrix}$$

has cp-rank l^2. Thus, for even numbers n there exist $n \times n$-matrices $B \in C^*$ with cp-rank $n^2/4$.

In fact, also matrices nearby $\hat{S}$ have a large cp-rank: Let $U_\epsilon(\hat{S}) := \{S = S^T \mid \|S - \hat{S}\|_F \leq \epsilon\}$. Then, for sufficiently small $\epsilon > 0$, all matrices in $C^* \cap U_\epsilon(\hat{S})$ have a cp-rank of at least l^2. As intuitively clear and confirmed by preliminary numerical experiments in Section 6.2.4 it is difficult to generate a C^*-certificate for such matrices with high cp-rank.

When n is not even, there also exist matrices with cp-rank $\geq \lfloor n^2/4 \rfloor$. On the other hand, by Caratheodorys theorem, the cp-rank of a matrix $B \in C^*$ always satisfies $p \leq n(n+1)/2$.

Remark 3. *For $1 \leq p < \lfloor n^2/4 \rfloor$ the set $C_p^* := \{XX^T \mid X \in I\!R^{n \times p},\ X \geq 0\}$ of matrices with cp-rank $\leq p$ is not convex.*

Proof. The proof is a trivial consequence of the observation that there exist matrices with cp-rank $> p$. Let S be one such matrix, then S is a convex combination of positive rank-1-matrices each of which is contained in C_p^*. □

This simple observation has implications on the selection of p in Algorithm 2 below.

5.2 Generating a starting point

5.2.1 The diagonal of B

The question under consideration is whether a given symmetric matrix B is in C^*. When B has a negative eigenvalue (or a negative matrix entry), then trivially $B \notin C^*$. Hence, we assume $B \succeq 0$ in this chapter. If the positive semidefinite matrix B has a zero diagonal element then the corresponding row and column of B is zero and the task of finding $X \geq 0$ with $XX^T = B$ can be reduced to a smaller dimensional problem. We therefore assume that B has strictly positive diagonal entries.

5.2.2 Criteria for the starting point

In this section a starting point $X^0 \in I\!R^{n \times p}$, $X^0 \geq 0$ is defined such that $S^0 := X^0(X^0)^T \approx B$. The algorithm in Section 5.3 generates iterates $S^k = X^k(X^k)^T$ where $X^k \geq 0$ and S^k lie in a neighborhood of the line segment $[S^0, B]$.

To faciliate the computation of X^k at each iteration the matrix S^0 is chosen in the interior of C^* (implying that the entire line segment $[S^0, B)$ is in the interior of C^* whenever $B \in C^*$). In a certain sense, S^0 is chosen "central" to C^*.

For a fixed matrix S^0, the choice of $X^0 \geq 0$ with $X^0(X^0)^T = S^0$ is far from unique. The particular choice of X^0 may be crucial in determining the efficiency of the algorithm in Section 5.3:

- When X^0 has two or more identical columns, these columns will remain identical throughout the algorithm. (This will only increase computation time.) Below we generate nonnegative columns that have pairwise a "large angle" to each other.
- When X^0 does not have full rank, the matrix $S^0 = X^0(X^0)^T$ lies on the boundary of C^*. To guarantee that $S^0 \in (C^*)^\circ$ the matrix X^0 below is generated such that it contains a strictly positive $n \times n$ submatrix whose smallest singular value is "large".
- The quality of the approximation $X^0(X^0)^T \approx B$ is less crucial; it is the goal of the algorithm to improve this approximation.

5.2.3 Rescaling to an "all-ones-diagonal"

Let D be the positive definite diagonal matrix such that D^{-2} coincides with the diagonal of B. Given a nonnegative factorization $DBD = \tilde{X}\tilde{X}^T$, it is trivial to recover the nonnegative factorization $B = (D^{-1}\tilde{X})(D^{-1}\tilde{X})^T$. Hence when defining a starting point X^0 such that $X^0(X^0)^T \approx B$ we may rescale $B := DBD$ to have a diagonal of all ones. When $B \in C^*$ this implies that $B_{i,j} \in [0, 1]$ for all i, j.

5.2.4 Two specific starting points

We consider two possible choices of p:

1. When n is large a choice of $p \geq n^2/4$ may be infeasible due to limitations in storage and computation time. In this case it may suffice to find an approximation $XX^T \approx B$ that improves the initial decomposition $X^0(X^0)^T \approx B$.

We then choose $n < p \leq 2n$ and a starting point X^0 is evaluated by the following steps:
By symmetric permutations the columns of B are reordered in increasing norm, $\tilde{B} := \Pi^T B \Pi$. Then, a Cholesky factorization $\tilde{B} = LL^T$ is computed (when $\tilde{B}$ is singular, L has less than n columns). When $L \geq 0$ stop ($B \in C^*$); else project L onto the set of nonnegative matrices. Finally, the rows of L are permuted back; $L := \Pi L$. Let $e \in I\!R^n$ be the vector of all ones and e_i be the i-th unit vector for $1 \leq i \leq n$. The first n columns of X^0 are set to $\frac{1}{2n}e + \frac{1}{\sqrt{n}}e_i$ $(1 \leq i \leq n)$. The remaining $p-n$ columns are set to the first $p-n$ columns of L. (Reduce p when L has less than $p-n$ nonzero columns.) Let $\hat{D}$ be the diagonal of $X^0(X^0)^T$. To match the diagonal of B and $X^0(X^0)^T$ we set $X^0 := \hat{D}^{-1/2}X^0$.

2. Second, $p = n(n+1)/2$. This second option is feasible only for small sizes of n, say $n \leq 50$. In this case, the following procedure generates a matrix $X^0(X^0)^T$ in the "center" of C^*:
As above, the first n columns of X^0 are set to $\frac{1}{2n}e + \frac{1}{\sqrt{n}}e_i$ $(1 \leq i \leq n)$. The remaining $n(n-1)/2$ columns are set to $\frac{1}{n^2}e + \frac{1}{n}(e_i + e_j)$ $(1 \leq i < j \leq n)$. It is easy to see that $X^0(X^0)^T = \lambda ee^T + \rho I$ where I is the identity matrix and λ, ρ are positive scalars. When changing the factors $\frac{1}{2n}, \frac{1}{\sqrt{n}}, \frac{1}{n^2}, \frac{1}{n}$ to other positive values, the numbers λ, ρ will change. $X^0(X^0)^T$ is in the interior of C^* if, and only if, $\lambda > 0$ and $\rho > 0$. Normalizing $X^0(X^0)^T$ to diagonal of all ones is simply achieved by setting $X^0 := \frac{1}{\sqrt{\lambda+\rho}}X^0$.

Whether or not the matrix B is scaled back $B := D^{-1}BD^{-1}$ (and likewise $X^0 := D^{-1}X^0$) before starting Algorithm 2 in the next section depends on the norm in which we would like to measure the distance between XX^T and B.

5.3 A Lyapunov type SOC-algorithm

Given a symmetric matrix $B \in I\!R^{n \times n}$ we wish to minimize $\|S - B\|$ for $S \in C^*$. As indicated in Section 5.2 we assume that $B \succeq 0$ and that an initial approximation $X = X^0 \geq 0$ is given such that $XX^T \approx B$ and XX^T is in the interior of C^*.

5.3.1 Motivation

The quadratic factorization heuristics of [7] can be adapted to the problem of generating a certificate of complete positivity: If B is in C^*, then there exists a matrix ΔX^* such that

$$(X + \Delta X^*)(X + \Delta X^*)^T = B \tag{5.1}$$

and $X + \Delta X^* \geq 0$. Neglecting the second order term $\Delta X^*(\Delta X^*)^T$ in (5.1) we obtain the linearized equation yielding an approximation ΔX for ΔX^*:

$$X\Delta X^T + \Delta X X^T = B - XX^T. \tag{5.2}$$

For a given $B \in C^*$ the set of ΔX^* satisfying (5.1) contains more than one element. The fact that the linearization error in (5.2) depends on $\|\Delta X\|_F$ suggests to determine an approximation ΔX for ΔX^* based on the linearized problem

$$\text{minimize} \;\; \|\Delta X\|_F \;\; | \;\; X\Delta X^T + \Delta X X^T = R, \;\; X + \Delta X \geq 0, \tag{5.3}$$

where $R = B - XX^T$. Problem (5.3) is the basis for an iterative process with repeated updates of the form $X \mapsto X + \Delta X$.

Problem (5.3) is a second order cone program (SOC problem) with np variables and $n(n+1)/2$ equality constraints. To be able to handle problems of the form (5.3) with a large number of variables and constraints, a specialized approach is discussed next.

5.3.2 Reformulation of the second order cone program

Problem (5.3) can be reformulated as

$$\text{minimize} \;\; x_0 \;\; | \, x_0 \geq \|x_1\|_2, \; x_2 \geq 0, \; \mathcal{A}(X_1) = R, \; -X_1 + X_2 = X, \tag{5.4}$$

where $X_1 := \Delta X$, $x_1 := \text{vec}\,(X_1)$, $X_2 := X + \Delta X$, $x_2 := \text{vec}\,(X_2)$, and

$$\mathcal{A}(\Delta X) := X\Delta X^T + \Delta X X^T$$

depends on the current iterate X. With the above notations we may also write problem (5.4) as

$$\text{minimize} \;\; x_0 \;\; | \, x = (x_0, x_1, x_2)^T \in \mathcal{K} \cap (\mathcal{L} + \bar{R}). \tag{5.5}$$

Here, $\mathcal{K}$ is the cone $\mathcal{K} = \mathcal{Q}_{np+1} \times \mathbb{R}^{np}_{+}$ with

$$\mathcal{Q}_{np+1} := \{x := (x_0; x_1) \in \mathbb{R} \times \mathbb{R}^{np} \mid x_0 \geq \|x_1\|_2\}$$

being the second order cone of dimension $np + 1$. The linear set $\mathcal{L}$ in (5.5) is given by $\mathcal{L} := \{x \mid \hat{A}x = 0\}$, where

$$\hat{A}x = \begin{bmatrix} 0 & A & 0 \\ 0 & -I & I \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}.$$

Here, A represents the linear operator $\mathcal{A}$ such that $Ax_1 = \text{vec}\,(\mathcal{A}(X_1))$ for $x_1 = \text{vec}\,(X_1)$. The linear equations of problem (5.5) can be written as

$$\hat{A}x = \begin{bmatrix} \text{vec}\,(R) \\ \text{vec}\,(X) \end{bmatrix} =: \hat{r} =: \hat{A}\bar{r}$$

for some suitable vector $\bar{r}$. This defines the element $\bar{R} = \text{mat}\,(\bar{r}) \in \mathbb{R} \times \mathbb{R}^{n\times p} \times \mathbb{R}^{n\times p}$ in (5.5). Problem (5.5) is given in the standard form of the apd-approach [25]. Note, that for problem (5.5) the apd-method is applied to a linear optimization problem with mixed cone constraints as pointed out in the beginning of this thesis.

5.3.3 Solution of the SOC problem

For small size problems, the subproblems (5.5) can be solved by interior-point approaches. However, due to the large number of equality constraints, another approach was used for the numerical results in Section 6.2.4:

As the correction $X_1 = \Delta X$ is subject to a linearization error (resulting from (5.2)), the subproblems (5.5) are not solved up to full precision in the implementation in Section 6.2.4. Instead, these subproblems are solved iteratively, and when the accuracy obtained for the subproblem is of the same magnitude as the linearization error, the algorithm for solving the subproblem is stopped.

Since the projection of a given iterate onto the cone $\mathcal{K}$ is trivial, the main computational effort in the apd-approach for solving (5.5) is the repeated computation of the projection of the current iterate onto the linear set $\mathcal{L}$.

As detailed below, this projection is computationally cheap as well. Moreover, as the required accuracy of the approximate solution of (5.5) is low, the apd-method in [49, 25] seems to be very well suited for solving the subproblems (5.5).

The projection of a point x onto $\mathcal{L}$ is given by

$$\Pi_{\mathcal{L}}(x) = x - \hat{A}^T(\hat{A}\hat{A}^T)^{-1}\hat{A}x.$$

Multiplications by $\hat{A}$ and $\hat{A}^T$ are cheap, the only critical part in the computation of this projection is the solution of a linear equation of the form

$$\hat{A}\hat{A}^T g = h \tag{5.6}$$

for a given right hand side $h = (h_1, h_2)^T \in \mathbb{R}^{n^2+np}$. Equation (5.6) is given by

$$\begin{bmatrix} AA^T & -A \\ -A^T & 2I \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \tag{5.7}$$

and its solution is obtained from

$$AA^T g_1 = 2h_1 + Ah_2 =: \hat{h}.$$

(Trivially, $g_2 = \frac{1}{2}(h_2 + A^T g_1)$.) Writing this equation in operator notation (with $n \times n$ matrices $G_1 := \text{mat}\,(g_1)$ and $\hat{H} := \text{mat}\,(\hat{h})$) leads to

$$\mathcal{A}\mathcal{A}^*(G_1) = \hat{H},$$

which is precisely the following Lyapunov equation

$$XX^T G_1^T + G_1 XX^T = \hat{H}. \tag{5.8}$$

Below, we discuss the solution of the above Lyapunov equation for the case that X has full row rank: To this end let

$$XX^T = U\Sigma U^T$$

be the eigenvalue decomposition of XX^T, i.e. $U \in I\!R^{n\times n}$ is an orthogonal matrix, and Σ is an $n \times n$ positive definite diagonal matrix.
Denoting $\widetilde{G_1} := U^T G_1 U$ and $\widetilde{H} := U^T \hat{H} U$, equation (5.8) is equivalent to

$$\Sigma\widetilde{G_1}^T + \widetilde{G_1}\Sigma = \widetilde{H}. \tag{5.9}$$

The solution of this system is given by $\widetilde{G_1}_{i,j} = \widetilde{H}/(\Sigma_{ii} + \Sigma_{jj})$ for $1 \le i, j \le n$. This yields the solution $G_1 = U\widetilde{G_1}U^T$ of (5.8).

5.3.4 Overall algorithm

Now, we summarize an algorithm based on (5.5):

Algorithm 2. *[Lyapunov type LP algorithm]*

1. *Input: A matrix* $X^0 \geq 0$ *of full row rank and a matrix* $B \succeq 0$.

   Set $k := 0$, $S^0 := P^0 := X^0(X^0)^T$.

2. *Set* $\hat{R}^k := \frac{1}{2}(B + P^k) - S^k$.

3. *Solve problem (5.5) for* $X = X^k$ *and determine a step size* α_k *such that* $\|(X^k + \alpha_k \Delta X)(X^k + \alpha_k \Delta X)^T - B\| < \|S^k - B\|$.

4. *Set* $X^{k+1} := X^k + \alpha_k \Delta X^k$, $S^{k+1} := X^{k+1}(X^{k+1})^T$, *and compute the projection* P^{k+1} *of* S^{k+1} *onto the straight line connecting* S^0 *and* B.

5. *Set* $k = k + 1$ *and go to Step 2.*

In order to limit the effect of the linearization error, Step *2.* aims not at the full step from S^k to B but only "half the way". More precisely, as the matrix X^0 can be chosen such that S^0 is a "central point" in C^*, Step *2.* aims back towards a point on the straight line connecting B and S^0.

5.3.5 Matrix completion

We point out that Algorithm 2 can be used with minor modifications to (approximately) solve the completely positive completion problem: "Given an index set $\mathcal{I} \subset \{1, \ldots, n\}^2$ and a symmetric matrix $B \in I\!R^{n \times n}$, find a matrix $S \in C^*$ such that $S_{i,j} = B_{i,j}$ for all $(i, j) \in \mathcal{I}$." In this case, the equality constraints in problems (5.3) or (5.4) that correspond to index pairs not in $\mathcal{I}$ are simply dropped. Unfortunately, the constraints then do not lend themselves any longer to the application of the apd-algorithm; the inverse of AA^T is not given by (5.8). For small size problems, of course, interior-point algorithms could be used in place of the apd-approach.

5.4 A regularization step

For a given matrix $S \in C^*$ of cp-rank $\leq p$ the set

$$\Xi_p(S) := \{X \in I\!R^{n\times p} \mid X \geq 0,\ XX^T = S\}$$

contains more than one element (unless $p = 1$). As shown in [15], for any $S \in (C^*)^\circ$, there exists a representation $S = XX^T$ satisfying

$$X = [X^1, X^2], \quad 0 < X^1 \in I\!R^{n\times n}, \quad \text{and} \quad X^1(X^1)^T \succ 0. \tag{5.10}$$

On the other hand, even when $S \in (C^*)^\circ$ there may also exist representations XX^T of S that violate (5.10). For example, $S = I + (n+2)E$ has the representations

$$S = XX^T = \hat{X}\hat{X}^T \quad \text{with} \quad X = [E + I, 0] \quad \text{and} \quad \hat{X} = [I, \sqrt{n+2}e].$$

The representation $\hat{X}\hat{X}^T$ not only violates (5.10), but, as will be detailed next, it is also less suitable for the computation of corrections $\Delta\hat{X}$:

Let us define the perturbation ΔS with entries $\Delta S_{i,j} = 0$ for all i, j except from $\Delta S_{1,2} = \Delta S_{2,1} = -1$. We consider corrections $\Delta\hat{X}$ and ΔX such that $\hat{X} + \Delta\hat{X} \geq 0$ and $X + \Delta X \geq 0$ satisfy the linearized equations

$$\Delta\hat{X}\hat{X}^T + \hat{X}\Delta\hat{X}^T = \Delta S \quad \text{and} \quad \Delta X X^T + X\Delta X^T = \Delta S. \tag{5.11}$$

Straightforward calculations show that the minimum norm solution $\Delta\hat{X}$ of (5.11) has a norm of about $\sqrt{2n-4}$. On the other hand, for any ΔS of norm $\sqrt{2}$ (including the above perturbation ΔS), the minimum norm solution ΔX of (5.11) is bounded by $\sqrt{2}$. In this example, the zero entries in $\hat{X}$ restrict the choice of corrections $\Delta\hat{X}$.

As the linearization error $\Delta X \Delta X^T$ increases with $\|\Delta X\|$ the representation XX^T appears to be more suitable as a starting point for linearized corrections $X \to X + \Delta X$ than the representation $\hat{X}\hat{X}^T$. This suggests to prefer strictly positive matrices X for starting the correction step of Algorithm 2. Below we present a heuristics for generating matrices $X \in \Xi_p(S)$ whose smallest entries are as large as possible.

Let $S = X^k(X^k)^T$ denote a certain iterate of Algorithm 2. Goal of this section is to compute a "central" element $\bar{X}$ of $\Xi_p(S)$ in the sense that $\bar{X} - \bar{\rho}E \geq 0$ for a large value of $\bar{\rho}$. Here, E is the 'all-ones-matrix'. (When $\bar{\rho} > 0$ and ΔX is given arbitrarily, this allows a correction $\bar{X} \mapsto \bar{X} + \epsilon\Delta X$ for some $\epsilon > 0$ without violating the nonnegativity constraints.) The regularization step can be applied after each iteration of Algorithm 2 replacing X^k with a "more central" matrix $\bar{X}^k$.

The following proposition is used to generate such a "central element".

Proposition 4. *Given $S \succeq 0$ with distinct eigenvalues $\lambda_i > \lambda_{i+1}$ for $1 \leq i \leq n-1$ and $X, \bar{X}$ with $XX^T = S = \bar{X}\bar{X}^T$ then $\bar{X} = X\hat{V}$ for some unitary matrix $\hat{V}$.*

Proof. Let $X = U\Sigma V$ and $\bar{X} = \bar{U}\bar{\Sigma}\bar{V}$ be the singular value decompositions of X and $\bar{X}$ where the singular values $\Sigma_{i,i} = \bar{\Sigma}_{i,i} = \sqrt{\lambda_i}$ are arranged in decreasing order. Comparing XX^T and $\bar{X}\bar{X}^T$ one obtains

$$U\Sigma\Sigma^T U^T = \bar{U}\bar{\Sigma}\bar{\Sigma}^T\bar{U}^T,$$

i.e. $\Sigma\Sigma^T = U^T\bar{U}\bar{\Sigma}\bar{\Sigma}^T\bar{U}^T U$. As $\Sigma\Sigma^T = \bar{\Sigma}\bar{\Sigma}^T$ is a diagonal matrix with strictly decreasing diagonal entries and $U^T\bar{U}$ is unitary, it follows that $U^T\bar{U} = I$, i.e. $U = \bar{U}$. Defining $\hat{V} := V^T\bar{V}$ the claim of the proposition follows. □

Remark 4. *When the eigenvalues of S are not pairwise distinct there might be additional degrees of freedom in the selection of X and $\bar{X}$. This possibility is not exploited in this chapter.*

Proposition 4 shall be used to change a given matrix $X \in \Xi(S)$ to a slightly 'more central' matrix $\bar{X}$. The change will be based on a "linearization" of the matrix $\hat{V}$, so that the equality $XX^T \approx \bar{X}\bar{X}^T$ only holds approximately.

By \$ we always denote a *skew symmetric* matrix, $\$ = -\$^T$. For small $\|\hat{V} - I\|$ it follows that there exists a skew symmetric matrix \$ such that

$$\hat{V} = I + \$ + O(\|\$\|^2).$$

(This equation defines the "linearization" referred to above.)

Given a matrix $X \in \Xi(S)$ we search for a small correction of the form

$$X \mapsto \bar{X} := X(I + \epsilon\$) \tag{5.12}$$

such that $\bar{X} \geq \bar{\rho}E$ for a large value of $\bar{\rho}$. To this end the matrix \$ is determined by the linear program

$$\text{maximize} \quad \rho \mid X(I + \$) \geq \rho E$$

which can be written in the dual form

$$\text{maximize} \quad \rho \mid \rho E - X\$ \leq X. \tag{5.13}$$

Whenever the optimal solution to (5.13) has an optimal value that is larger than $\min_{i,j} X_{i,j}$ an update of the form (5.12) with $\epsilon \in (0, 1)$ will increase the lower bound $\min_{i,j} \bar{X}_{i,j}$ – at the expense of a second order perturbation to XX^T. The solution of (5.13) can also be computed by the apd algorithm, and as in Section 5.3, the accuracy of the solution of the subproblems can be adjusted according to the linearization error and the distance of XX^T to B.

5.4.1 Standard form of the apd-algorithm

Let the mapping $\mathcal{A}^*$ be given by $\mathcal{A}^*(\$) = X\$$. $\mathcal{A}^*$ maps the space of skew symmetric $p \times p$-matrices to $\mathbb{R}^{n \times p}$. Its adjoint is given by

$$\mathcal{A}(Z) = \frac{1}{2}(X^T Z - Z^T X)$$

for $Z \in \mathbb{R}^{n \times p}$. With this notation, the primal of (5.13) is given by

$$\text{minimize} \quad X \bullet Z \mid E \bullet Z = 1, \quad \mathcal{A}(Z) = 0, \quad Z \geq 0.$$

Note that $Z^0 := X/(E \bullet X)$ is feasible for the primal problem.

To apply the apd-algorithm [25] to this LP we denote

$$\mathcal{L} := \{Z \mid E \bullet Z = 0, \quad \mathcal{A}(Z) = 0\}.$$

The primal problem can thus be written as

$$\text{minimize} \quad X \bullet Z \mid Z \in (\mathcal{L} + Z^0) \cap \mathbb{R}^{n \times p}_+.$$

Here, the iterate X is given data (it is the goal ot this LP to increase the minimum entry of X) and Z is the dual variable. We recall that the apd-algorithm is based on the availability of cheap projections onto $\mathcal{L}$. These will be discussed next. (The authors were not able to provide equally cheap solutions for the linear systems that arise in interior-point approaches for this problem.)

The KKT conditions for the projection ΔZ of a matrix Z onto $\mathcal{L}$ can be written as follows: There exists a $\rho \in I\!R$ and a skew symmetric \$ such that

$$\rho E \bullet E + E \bullet \mathcal{A}^*(\$) = E \bullet Z$$

$$\rho \mathcal{A}(E) + \mathcal{A}(\mathcal{A}^*(\$)) = \mathcal{A}(Z).$$

In the sequel the brackets as in $\mathcal{A}(E)$ will be omitted and we simply write $\mathcal{A}E$. The solution of the above system can be obtained via

$$\rho = \frac{Z \bullet (E - \mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^\dagger \mathcal{A}E}{E \bullet (E - \mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^\dagger \mathcal{A}E}$$

$$\$ = (\mathcal{A}\mathcal{A}^*)^\dagger \mathcal{A}(Z - \rho E).$$

Note that E is not contained in the range of $\mathcal{A}^*$ and hence, ρ is well-defined. (If E was contained in the range of $\mathcal{A}^*$ the linear program (5.13) would be unbounded.)

We briefly discuss the least squares solution of the system $\mathcal{A}\mathcal{A}^*\$ \approx R$ for some given skew symmetric right hand side $R \in I\!R^{p \times p}$. (This least squares solution coincides with $\$ = (\mathcal{A}\mathcal{A}^*)^\dagger R$.)

Observe that $\mathcal{A}\mathcal{A}^*\$ = \frac{1}{2}(X^T X\$ + \$X^T X)$. We obtain the equation

$$2\mathcal{A}\mathcal{A}^*\$ = X^T X\$ + \$X^T X \approx 2R$$

for the unknown matrix \$. We assume that the singular value decomposition of X is given, $X = U\Sigma V$ with unitary matrices U and V of suitable dimensions. Using the singular value decomposition and setting $\tilde{\$} = V\$V^T$ this is equivalent to

$$\Sigma^T \Sigma \tilde{\$} + \tilde{\$} \Sigma^T \Sigma \approx 2\tilde{R} := 2VRV^T.$$

The matrix $\tilde{\$}$ is skew symmetric as well, and the above unitary transformations do not change the least squares solution.

Here, $\Sigma^T \Sigma$ is a $p \times p$ diagonal matrix, only the leading n diagonal entries of it being nonzero (when X has maximum rank, else there are $r < n$ nonzero entries).

Solving this system for $\tilde{\$}$ in a least squares sense is trivial, yielding the desired solution $\$ = V^T\tilde{\$}V$. Above computations require about $O(p^3)$ operations. Note that $\mathcal{A}\mathcal{A}^*$ maps the skew symmetric $p \times p$ matrices into themselves; the inversion of a general map $\mathbb{R}^{p\times p} \to \mathbb{R}^{p\times p}$ may take $O(p^6)$ operations.

5.4.2 Recovering the primal variable

If problem (5.13) is solved by the apd-method, the last step of the algorithm can be chosen as the projection onto the affine hull of the primal dual feasible solutions. We obtain a primal dual solution in the apd-format satisfying all equality constraints. The dual solution N is a matrix in $\mathbb{R}^{n\times n}$ such that there exist variables $\$ and ρ with

$$\rho E - X\$ = R := X - N.$$

(Ideally, when also the primal dual inequalities are satisfied then $N \in \mathbb{R}^{n\times n}_+$, but due to the last projection this cannot be guaranteed.) We are then interested in the values of $\$ and ρ. Specifically, we need to solve a system of the form

$$\rho E - U\Sigma V\$ = R$$

where $U\Sigma V$ is the singular value decomposition of X. (For general R this system may not have a solution, but by our assumption that the apd method terminates with a projection on the primal dual feasible equations a solution must exist.) Setting $\tilde{E} := U^T E V^T$, $\tilde{\$} := V\$V^T$, and $\tilde{R} := U^T R V^T$, this system is equivalent to

$$\rho\tilde{E} - \Sigma\tilde{\$} = \tilde{R}.$$

Let $\Sigma = [D, 0]$. By the assumption $XX^T \in (C^*)^\circ$ it follows that X must have full rank and thus D is an $n \times n$ positive definite diagonal matrix. We obtain

$$\rho D^{-1}\tilde{E} - [I, 0]\tilde{\$} = D^{-1}\tilde{R}.$$

Note that $\tilde{\$}$ is skew symmetric and thus has a zero diagonal. Hence we may determine ρ such that $D^{-1}(\tilde{R} - \rho\tilde{E})$ has a zero diagonal. (In the presence of rounding errors a least squares solution ρ may be used.) Once ρ is given, the computation of $\$ is straightforward.

Chapter 6

Implementation and Numerical Results

In this chapter numerical results of the algorithms in chapter 3 and 5 are presented. We also explain basic elements of the implementation. First of all, numerical results for Algorithm 1 in chapter 3 are presented. Then, the implementation of the apd-algorithm introduced in chapter 4 for the application presented in chapter 5 is discussed. To this end we repeat the main results on the (limited memory) BFGS method. Afterwards, we briefly discuss the step size control used here. We conclude this chapter with numerical results for Algorithm 2 in chapter 5.

6.1 Numerical Examples for Linear Programs

We start our numerical examples with a Newton-type method for minimizing the piecewise quadratic functions in chapter 3. We have implemented Algorithm 1 with MATLAB in order to test the program for functions of the form (3.2) and (3.1). Here, our goal was not to find a competitive numerical algorithm for solving linear programs, but to obtain a better understanding of how many weakly active indices will be intersected by the generalized Newton path minimizing f of the form (3.2) or (3.1). To obtain some intuition about the worst-case behavior, we tested a large number of random examples and limited ourselves to small size problems.

The function $f^{(P),(D)}$ in (3.1) is not strictly convex. When the Hessian of f is singular, the generalized Newton path runs parallel to weakly active constraints, and, as seen in Section 3.1.2, it will typically run into points with more than one weakly active index. At such points a generalized Newton step is difficult to compute. We therefore added a perturbation ϵI to $\nabla^2 f(y)$ whenever $\nabla^2 f(y)$ was nearly singular. Unfortunately, the numerical results are biassed by rounding errors; the distinction of which constraints are active, weakly active, or inactive becomes unreliable.
In several examples the algorithm ended up with very short steps zigzagging between two weakly active indices, a behavior that cannot occur when exact arithmetic is used. In order to obtain a numerical implementation that might be competitive to other algorithms, one would not only need to control rounding errors but also use suitable rank-one update formulae when crossing weakly active indices.

For all numerical experiments we therefore used an exact line search along the (generalized) Newton direction. Since the function f is smooth, it is unlikely that the minimizer of the line search lies at a point with weakly active indices. (The zig-zagging was now indeed reduced to very few cases among 100000 test problems.) The exact line search can be carried out in order nm arithmetic operations. We counted both, the number of iterations (Newton steps) used and the total number of weakly active indices intersected along this path.

For our first set of examples we chose the function (3.2), where all data vectors b, a_i and γ are uniformly distributed in $[-0.5, 0.5]$, and the Hessian H of q as the product of a matrix Q and its transpose, Q having uniformly distributed entries in $[0, 1]$. The starting point is chosen uniformly distributed in $[-50, 50]$.

In Table 6.1, the results of the algorithm for such f are listed. We kept the dimension fixed at $n = 30$ and increased m by a factor of 3/2 for each row. In each row the results are listed for 10000 random examples. The first column displays the values of m. In the second column we list the average number of Newton steps, in the third column the maximum number of Newton steps, in the fourth column the average number of weakly active constraints intersected along the Newton path, and finally, in the last column we list the maximum number of weakly active constraints that were crossed along the path.

m	aver. Newt.	max. Newt.	aver. cross.	max. cross
4	3.45	16	2.63	8
6	3.53	15	4.26	12
9	3.58	16	6.63	17
14	3.80	24	10.58	24
21	4.03	11	15.80	35
32	4.22	8	23.89	46
48	4.36	8	35.12	61
72	4.38	7	50.09	81
108	4.33	7	72.07	100
162	4.23	6	102.41	137

Table 6.1: Random f as in (3.2)

The algorithm stopped when the norm of the gradient was less than 10^{-12} or when the Newton direction was not a descent direction. We note that in the first two rows the maximum number of Newton steps was higher than the number of intersections with weakly active constraints. This was due to rounding errors in the final iterations.

Note: Table 6.1 summarizes the results of a total of 100000 test problems. In none of the examples, the number of intersections of weakly active constraints exceeded $2m$. We do know, however, that $m^2/4$ or more intersections are possible for problems that are designed as in Section 3.1.2.

For our second set of examples we used functions $f^{(P),(D)}$ arising from random linear programs that have primal and dual feasible solutions. Whenever the Hessian of $f^{(P),(D)}$ had a condition number of more than 10^{12}, a regularization term ϵI was added to $f^{(P),(D)}$. The resulting step is a Levenberg-Marquardt step for the convex function $f^{(P),(D)}$. Table 6.2 lists the results with 1000 random examples for each row. Each problem (P) has $2m$ variables and m linear equality constraints. The resulting primal-dual function $f^{(P),(D)}$ has $4m$ "plus-squared"-terms. Again, the maximum number of crossing weakly active indices during the generalized Newton method is less than twice the number of "plus-squared"-terms.

m	aver. Newt.	max. Newt.	aver. cross.	max. cross
4	3.44	9	4.96	16
6	5.34	11	10.72	27
8	6.66	13	16.06	32
10	8.32	22	23.51	72
12	9.60	23	29.61	63
14	11.35	23	37.95	72
16	12.55	24	44.48	84
18	14.51	28	54.46	104
20	15.80	29	61.20	115

Table 6.2: $f^{(P),(D)}$ from random linear programs

Finally, in Table 6.3 we list the results for Klee-Minty problems of the form

$$\max \Big\{ \sum_{j=1}^{n} \epsilon^{n-j} x_j \;\Big|\; x_i + 2\sum_{j=1}^{i-1} \epsilon^{i-j} x_j \le 1 \quad \text{for} \quad 1 \le i \le n, \quad x \ge 0 \Big\},$$

where $\epsilon = 0.45$.

We have implemented both a primal-dual version and a dual-only version minimizing a function $f^{(D)}$ with $2n+1$ "plus-squared"-terms using the information that the optimal value of the above problem is 1. We list the results for the "dual-only" version since this version allowed problems of slightly larger dimension that were not biassed by rounding errors. Here each row lists the results with 1000 different starting points.

The Klee-Minty problems were designed specifically to trick a method of completely different nature (the Simplex method). As expected, one would need to find other examples to embarrass the generalized Newton approach as considered here.

Note: For functions $f = f^{(P),(D)}$ arising from linear programs the observations are very similar as for general f of the form (3.2). While we do not know whether there might be exponentially many intersections in the worst case, the results indicate that the average number of intersections might be fairly small.

n	aver. Newt.	max. Newt.	aver. cross.	max. cross
4	7.48	10	15.72	23
6	9.23	19	21.63	40
8	10.24	23	27.52	58
10	11.88	29	33.07	84
12	13.96	30	40.28	92
14	15.35	28	44.94	79

Table 6.3: $f^{(D)}$ from Klee-Minty problems

6.2 Numerical Experiments for Completely Positive Matrices

In this section numerical results for Algorithm 2 in chapter 5 are presented. This algorithm was implemented with MATLAB. For the computation of the search direction we used the limited memory BFGS method. First, a description of this method is given in sections 6.2.1 and 6.2.2 below. Then, a discussion regarding the line search that is used here is provided. Finally, this section is concluded with numerical results for the regularization step in section 5.4. As we are not aware of other approaches for solving the problem introduced in chapter 5 for matrices of moderate dimensions we cannot present comparisons with existing approaches.

6.2.1 Quasi-Newton Methods

The basic Newton iteration for the determination of a minimum of a convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is given as the Newton iteration for the gradient of f, $F(x) := \nabla f(x)$, i.e.

$$x^{k+1} = x^k - (\nabla F(x^k))^{-1} F(x^k),$$

where k is the iteration number and x^k is the actual iterate. Since the determination of the Hessian $\nabla^2 f(x^k)$ causes high computational costs in each iteration, our goal is to find a matrix B^k that approximates $\nabla^2 f(x^k)$, respectively a matrix H^k that approximates the inverse of $\nabla^2 f(x^k)$.

The quasi-Newton methods are used especially for problems, where the Hessian is a dense matrix. In such problems, a Cholesky factorization of the Hessian of f is computed, i.e. $\nabla^2 f(x^k) = LL^T$. The Cholesky-factor L can thus be corrected in $\mathcal{O}(n^2)$ operations to a Cholesky-factorization of B^{k+1}.
So, in quasi-Newton methods, we have to compute the next iterate by

$$x^{k+1} = x^k - \lambda_k (B^k)^{-1} \cdot f(x^k),$$

$$\text{resp. } x^{k+1} = x^k - \lambda_k H^k \cdot f(x^k),$$

where λ_k denotes the step size. For the remainder of this chapter we make the following assumptions.

Assumption 3.

- *The function F is continuously differentiable on $\mathcal{D} \in \mathbb{R}^n$.*
- *$\mathcal{D}$ is convex and open.*
- *There exists $x^* \in \mathcal{D}$ with $F(x^*) = 0$ and $\nabla F(x^*)$ is nonsingular.*
- *$\nabla F(x^*)$ is Lipschitz-continuous, i.e.*

$$\|\nabla F(x) - \nabla F(x^*)\| \leq L\|x - x^*\| \text{ for all } x \in \mathcal{D}.$$

With the definition of $s^k = x^{k+1} - x^k$ and $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$ we repeat Theorem 6.6.3 from [26] to present a convergence result for quasi-Newton methods.

Theorem 1 (Dennis, Moré)**.** *Let the following assumptions be fulfilled for a sequence of quasi-Newton iterates:*

- *B^k is nonsingular for all k.*
- *$\lambda_k = 1$ for all k.*
- *$\lim x^k = x^*$, $x^k \neq x^*$ and $x^k \in \mathcal{D}$ for all k.*

Then, the following statements are equivalent:

- $\lim_k \frac{\|x^{k+1}-x^*\|}{\|x^k-x^*\|} = 0,$
- $\lim_k \frac{\|(B^k-\nabla F(x^*))s^k\|}{\|s^k\|} = 0,$
- $\lim_k \frac{\|B^k s^k-y^k\|}{\|s^k\|} = 0.$

The first property states that the convergence of the iterates in a quasi-Newton method is Q-superlinear, what implies that for large k the new iterate x^{k+1} is much closer to the zero x^* than the last iterate x^k. To preserve this convergence, we have to fulfill the following *quasi-Newton condition*

$$B^{k+1}s^k = y^k.$$

One important quasi-Newton method is the BFGS method (Broyden, Fletcher, Goldfarb, Shanno), where the next approximation B^{k+1} is given by

$$B^{k+1} = B^k + \frac{y^k(y^k)^T}{(s^k)^T y^k} - \frac{B^k s^k (s^k)^T B^k}{(s^k)^T B^k s^k},$$

respectively

$$H^{k+1} = H^k + \frac{(s^k - H^k y^k)(s^k)^T + s^k(s^k - H^k y^k)^T}{(s^k)^T y^k} - \frac{(s^k - H^k y^k)^T y^k}{((s^k)^T y^k)^2} s^k (s^k)^T.$$

These formulae are rank-2-updates for the matrices B^k, respectively H^k.
In view of convergence, quasi-Newton methods outperform steepest descent methods in most cases. The convergence of steepest descent methods often is nonsatisfactory and so-called zigzagging may occur. Both types of algorithms, namely steepest descent and quasi-Newton, require only first derivatives, and both require a line search. The quasi-Newton algorithms require slightly more operations to calculate the search direction and somewhat more storage, but in almost all cases, these additional costs are outweighted by the advantage of superior convergence.
For large scale problems there arises a problem of memory storage, since the storage of the old approximation H^k exceeds the memory capacity. As the matrix H^k is symmetric, it is necessary to have $\frac{n(n+1)}{2}$ storage locations only for the approximation of the Hessian. Since our problems, when written as second order cone programs, become very large in n, we use the *limited memory BFGS method*.

6.2.2 Limited Memory BFGS

In this section, we first repeat the general BFGS algorithm. Then, we explain the idea of the limited memory BFGS method and the advantages of the same.

The BFGS method generates a sequence of iterates $\{x^k\}$ according to the following algorithm.

Algorithm 3. *[BFGS]*

1. *Input: a starting point* x^0 *and an initial approximation* $H^0 \succ 0$. *Set* $k = 0$. *while* $\|\nabla f(x^k)\| > 0$ *do*
2. *Compute the search direction* $d^{k+1} = -H^k \nabla f(x^k)$.
3. *Determine the step length* $\alpha_k = \arg\min_{\alpha>0} f(x^k + \alpha d^k)$.
4. *Set* $x^{k+1} = x^k + \alpha_k d^k$ *and compute* H^{k+1} *by the BFGS update formula.*
5. *Set* $k = k + 1$ *and go to Step* 2.

Now, if we had to save the approximation H^k of the Hessian of f, this would exceed the storage capacity in large scale problems. Hence, we have to save memory, and for this purpose we do not store the matrix H^{k+1} in each step of algorithm 3, but compute the product in Step 2. in algorithm 3 from the most recent m difference pairs $\{s^i, y^i\}$.
When $m \ll n$, this leads to a significant reduction in memory usage. If $k < m$, the method uses the k difference pairs it has available.

Note, that in our implementation we save only $m = 5$ difference pairs while in our problems there are about $n = 30.000$ unknowns, so the reduction of memory usage is tremendous.

With the definition of $\rho_k = \frac{1}{(y^k)^T s^k}$, see below the two-loop formula for the computation of the matrix-vector product $r = H^k v$ (see [19]).
Note, that for a sparse matrix H_0^k the memory usage within this algorithm is considerably lower. We determine the matrix H_0^k as stated below.

Algorithm 4 (Limited Memory BFGS Update)**.**

1. *Set* $q = v$.

2. **for** $i = k-1 : -1 : k-m$
   $\mu_i = \rho_i (s^i)^T q$
   $q = q - \mu_i y^i$
   end

3. *Set* $r = H_0^k q$.

4. **for** $i = k-m : k-1$
   $\beta = \rho_i r^T y^i$
   $r = r - (\beta - \mu_i) s^i$
   end

For H_0^k, we use the scaling suggested in [32], i.e. we determine the diagonal matrix D^k, that minimizes

$$\|D^k Y^{k-1} - S^{k-1}\|_F. \tag{6.1}$$

Here, $Y^{k-1} = [y^{k-1}, \ldots, y^{k-m}]$, $S^{k-1} = [s^{k-1}, \ldots, s^{k-m}]$ and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, i.e.

$$\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2}$$

for the $n \times n$- matrix A. The minimal argument of (6.1) is given by the diagonal matrix $D^k = \mathrm{diag}(d_k^i)$ with

$$d_k^i = \frac{s_{k-1}^i y_{k-1}^i + \cdots + s_{k-m}^i y_{k-m}^i}{(y_{k-1}^i)^2 + \cdots + (y_{k-m}^i)^2} \text{ for } i = 1, \ldots, n.$$

As recommended in [32], this formula is used only if the denominator is greater than 10^{-10}, and if all the diagonal elements satisfy $d_k^i \in [10^{-2}\gamma_k, 10^2\gamma_k]$; otherwise we set $d_k^i = \gamma_k$. Here, γ_k is given by

$$\gamma_k = \frac{(y^k)^T s^k}{\|y^k\|_2^2}.$$

Note, that if $k < m$ we use $H_0^k = \gamma_k I$.
In the general BFGS method, the positive definiteness of H^k is preserved, if $(y^k)^T s^k > 0$. This observation can be copied to the limited memory BFGS method. To guarantee the condition $(y^k)^T s^k > 0$, a suitable line search is necessary. In the next section we present the line search that is used in our implementation.

6.2.3 Line Search

Several line search methods are available for the determination of the step length α_k in Step 3. in algorithm 3. The best-known line search conditions are the *Wolfe conditions*.
Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth objective function, and d^k be a given search direction. A step length α_k is said to satisfy the Wolfe conditions if the following two inequalities hold.

$$\begin{aligned} i) \qquad f(x^k + \alpha_k d^k) &\leq f(x^k) + c_1 \alpha_k (d^k)^T \nabla f(x^k), \\ ii) \quad (d^k)^T \nabla f(x^k + \alpha_k d^k) &\geq c_2 (d^k)^T \nabla f(x^k), \end{aligned}$$

with $0 < c_1 < c_2 < 1$. Inequality $i)$ is known as the Armijo condition (or Goldstein condition, or Goldstein-Armijo condition) and $ii)$ as the curvature condition. Condition $i)$ ensures that α_k decreases f 'sufficiently', and $ii)$ ensures that the slope of the function $\phi(\alpha) = f(x^k + \alpha d^k)$ at α_k is greater than c_2 times than at $\alpha = 0$.
In [26] it is shown, that the curvature condition

$$(y^k)^T s^k > 0$$

is satisfied if condition $ii)$ is fulfilled and thus, the BFGS-update H^{k+1} of H^k is symmetric and positive definite.

Since the function evaluations in our implementation are cheap, we use an exact line search, i.e. in each iteration of algorithm 3 we determine the minimizer

$$\alpha_k = \arg\min_{\alpha > 0} f(x^k + \alpha d^k).$$

It is obvious that for the exact line search, condition $ii)$ is satisfied and thus the BFGS curvature condition $(y^k)^T s^k > 0$ is fulfilled in each step of algorithm 3.

6.2.4 Numerical Results

Algorithm 2 was tested for random examples with $p = 2n$ and $n = 10,\ 50,\ 200$. The matrix B was generated as $B = WW^T$ where W was chosen as a random (uniformly distributed entries in $(0, 1)$) matrix of dimension $n \times k$. For $k < n$ it follows that $B \in \partial C^*$.
We observe significantly longer solution times for the case $k < n$ with less accuracy in the final solution. The solution times reflect that the subproblems to be solved with the apd-method tend to require a higher number of iterations; the number of overall (outer) iterations does not vary to such extent.

It is somewhat surprising that the final accuracy reached for the "large scale" problems ($n = 200$) is higher than for the smaller problems.

The results of the algorithm without regularization step are given in Tables 6.4, 6.5, 6.6. The running times refer to a 1.6GZ PC (from 2003).

$n = 10$	$k = \frac{n}{2}$	$k = n$	$k = 2n$
minimal $\|B - X^0(X^0)^T\|_F$	$5.62 \cdot 10^{-2}$	$4.78 \cdot 10^{-2}$	$6.03 \cdot 10^{-2}$
maximal $\|B - X^0(X^0)^T\|_F$	$1.06 \cdot 10^{-1}$	$6.47 \cdot 10^{-2}$	$6.39 \cdot 10^{-2}$
minimal $\|B - X^{end}(X^{end})^T\|_F$	$3.44 \cdot 10^{-6}$	$1.06 \cdot 10^{-12}$	$4.36 \cdot 10^{-13}$
maximal $\|B - X^{end}(X^{end})^T\|_F$	$1.04 \cdot 10^{-4}$	$1.85 \cdot 10^{-12}$	$8.71 \cdot 10^{-13}$
minimal number of iterations	10	34	37
maximal number of iterations	30	39	38
average running time	$78.8s$	$5.8s$	$3.4s$

Table 6.4: Results of Algorithm 2 for $n = 10$

The effect of the regularization step

For $B = \begin{pmatrix} lI & E \\ E & lI \end{pmatrix}$ the results without regularization step were disappointing. As pointed out, this matrix has cp-rank $n^2/4$. To have any chance to prove complete positivity for this choice of B we applied Algorithm 2 with the expensive choice $p = n(n + 1)/2$. The example was tested first with Algorithm 2 without regularization.

$n = 50$	$k = \frac{n}{2}$	$k = n$	$k = 2n$
minimal $\|B - X^0(X^0)^T\|_F$	$6.46 \cdot 10^{-2}$	$3.74 \cdot 10^{-2}$	$1.62 \cdot 10^{-2}$
maximal $\|B - X^0(X^0)^T\|_F$	$7.94 \cdot 10^{-2}$	$5.02 \cdot 10^{-2}$	$2.28 \cdot 10^{-2}$
minimal $\|B - X^{end}(X^{end})^T\|_F$	$2.08 \cdot 10^{-8}$	$4.23 \cdot 10^{-14}$	$2.05 \cdot 10^{-14}$
maximal $\|B - X^{end}(X^{end})^T\|_F$	$1.19 \cdot 10^{-7}$	$7.94 \cdot 10^{-14}$	$3.43 \cdot 10^{-14}$
minimal number of iterations	27	41	40
maximal number of iterations	34	44	42
average running time	$764.5s$	$123.7s$	$34.2s$

Table 6.5: Results of Algorithm 2 for $n = 50$

$n = 200$	$k = \frac{n}{2}$	$k = n$	$k = 2n$
minimal $\|B - X^0(X^0)^T\|_F$	$1.04 \cdot 10^{-1}$	$6.74 \cdot 10^{-2}$	$2.80 \cdot 10^{-2}$
maximal $\|B - X^0(X^0)^T\|_F$	$1.13 \cdot 10^{-1}$	$7.09 \cdot 10^{-2}$	$3.02 \cdot 10^{-2}$
minimal $\|B - X^{end}(X^{end})^T\|_F$	$1.10 \cdot 10^{-10}$	$4.05 \cdot 10^{-15}$	$2.41 \cdot 10^{-15}$
maximal $\|B - X^{end}(X^{end})^T\|_F$	$1.86 \cdot 10^{-10}$	$4.99 \cdot 10^{-15}$	$2.49 \cdot 10^{-15}$
minimal number of iterations	35	50	57
maximal number of iterations	41	54	68
average running time	$27609.6s$	$1595.4s$	$1858.1s$

Table 6.6: Results of Algorithm 2 for $n = 200$

Due to slow convergence we stopped the algorithm after 260 iterations with a residual of $1.56 \cdot 10^{-2}$. For this example we also tested the regularized approach, allowing 10 regularization steps after each iteration. (Each regularization step solves a linear program with $p(p-1)/2$ variables and is thus very expensive.) After 65 iterations the regularized approach obtained an accuracy of $4.85 \cdot 10^{-5}$.

Since the results of Algorithm 2 for $n = 10$ and $k = n/2$ were disappointing, we also tested the regularized approach for these examples. The corresponding results for the regularized approach in comparison with the approach without regularization are given in Table 6.7.

$n = 10$	with regulariz.	without regulariz.
minimal $\|B - X^0(X^0)^T\|_F$	$5.62 \cdot 10^{-2}$	
maximal $\|B - X^0(X^0)^T\|_F$	$1.06 \cdot 10^{-1}$	
minimal $\|B - X^{end}(X^{end})^T\|_F$	$4.38 \cdot 10^{-8}$	$3.44 \cdot 10^{-6}$
maximal $\|B - X^{end}(X^{end})^T\|_F$	$3.25 \cdot 10^{-7}$	$1.04 \cdot 10^{-4}$
minimal number of iterations	17	10
maximal number of iterations	29	30

Table 6.7: Results of Algorithm 2 with/without regularization for $n = 10$

The approach without regularization was stopped whenever the algorithm stagnated or the iteration number exceeded twice the iteration number of the regularized approach.
The algorithm with regularization performs better when the constant ϵ in the regularization step is chosen in dependence of the distance of the current iterate $X^k(X^k)^T$ to B with smaller values of $\epsilon > 0$ when $X^k(X^k)^T$ is close to B.

Chapter 7

Summary and Outlook

This thesis deals with linear conic programs. We equated the solution of linear, respectively linear second order cone programs, with the minimization of a certain function.

In chapter 3, we recalled the equivalence of minimizing a certain convex, differentiable, piecewise linear function f with the problem of solving a linear program. We defined a generalized Newton path for minimizing f. This path is piecewise linear. The gradient $\nabla f(z)$ of this path forms a straight line from $\nabla f(z^0)$ to zero. We therefore considered the convex conjugate function f^* of f. The number of piecewise quadratic segments of the implicit function f^* along a given line therefore corresponds to the number of (generalized) Newton steps with line search for minimizing f. Closely related is another implicit function defined by the augmented Lagrangian. This function has a slightly different structure, and there are known examples where Newton's method for minimizing this function may take an exponential number of steps. While the discussion in chapter 3 concentrated on linear programs, similar considerations seem possible for convex quadratic objective functions.

In generalization of this approach, we considered linear second order cone programs in chapter 4. We started the analysis of second order cone programs by proving the equivalence of uniqueness of the optimal solution and nondegeneracy of the optimal solution for linear second order cone programs.

We have also shown that the solutions of linear second order cone programs with small perturbations in the data are differentiable functions of the perturbations and that the standard primal-dual system for second order cone programs is non-singular under uniqueness, strict complementarity and Slater's condition.
Based on these results, we extended the augmented primal-dual algorithm introduced in [25] to second order cone programs. Within this apd-method a certain function is minimized. This function is, as well as the function $\hat{f}$ considered in chapter 3, closely related to the augmented Lagrangian introduced in section 3.2.1.
This generalization can also be combined with the approach in [25] in order to solve programs with both, semidefiniteness and second order constraints. Such programs arise, for example, when transforming a semidefinite program with a convex quadratic objective function into a conic program with a linear objective function.

Semidefinite programs arise, for example, as a relaxation for the determination of a maximum stable set. When replacing the semidefinite constraint in this relaxation by a completely positive constraint, a strengthened relaxation is obtained. In this context the problem of deciding whether a given matrix is completely positive or not is of wide interest.

In chapter 5 we presented an application on completely positive matrices for the apd-method introduced in chapter 4. Within this chapter, the quadratic factorization heuristics – proposed in a different context in [7] – is used for the generation of a certificate of complete positivity of a given matrix B; or for completely positive completion problems.
The algorithm generates iterates that are determined by approximate solutions of certain subproblems. These subproblems can be reformulated as second order cone programs. Due to a linearization error, the exact solution of the subproblems does not generate the desired certificate but merely determines a step towards the next iterate. Because of this linearization error the implementation solves the subproblems only up to a precision of same the magnitude as the linearization error. For such approximate solutions the apd-method [25, 49] is well suited.

Both approaches were implemented with MATLAB. Numerical experiments and special features of the implementation are given in chapter 6.
Our numerical results for Algorithm 2 show a very promising convergence behavior of the algorithm for matrices B in the interior of C^* and of low cp-rank. The convergence slows down significantly, when B is on the boundary of C^* or when B has a large cp-rank. To accelerate the algorithm for this case we propose a novel regularization step after each iteration aiming at making all matrix entries of the current factor X^k as large as possible without changing the product $X^k(X^k)^T$. Numerical examples illustrate the positive effect of this regularization. It was possible to test this regularization – and to establish its positive effect on the overall algorithm – for small size problems. Due to limits in computation time, the application to large problems remains the topic of future research.

At this point I would like to thank some people who contributed to the success of this thesis.
First, I want to thank my supervisor Prof. Dr. F. Jarre whose help, stimulating suggestions and encouragement helped me in all the time of research for and writing of this thesis.
Special thanks go to my husband Thomas. This thesis is the result of your support, patience and love. Thanks must also go to my immediate family for being a constant source of love, support and strength all these years.
Particularly, I want to thank all former and current members at the Chair of Mathematical Optimization and the Chair of Applied Mathematics in Düsseldorf for the kind athmosphere during the time of my PhD studies.

Bibliography

[1] Alizadeh, F.; Goldfarb, D. (2001), Second-Order Cone Programming,Technical Report 51-2001, RUTCOR, Rutgers University.

[2] Alizadeh, F.; Schmieta, S. (2000), Symmetric Cones, Potential Reduction Methods and Word-by-Word Extensions, in: Wolkowicz, H.;, Saigal, R.; Vandenberghe, L. (Eds.), Handbook of Semidefinite Programming: Theory, Algorithms and Applications, Kluwer Academic Publishers, Dordrecht, pp. 195–233.

[3] Amenta, N.; Ziegler, G. (1996), Shadows and slices of polytopes, Proceedings of the Twelfth Annual Symposium on Computational Geometry, Association for Computing Machinery, Philadelphia Pennsylvania, pp. 10–19.

[4] Berman, A.; Shaked-Monderer, N. (2003), Completely positive matrices, World Scientific, Singapore.

[5] Bertsekas, D.P. (1999), Nonlinear Programming (2nd edition), Athena Scientific.

[6] Bomze, I.M.; Dür, M.; de Klerk, E.; Roos, C.; Quist, A.J.; Terlaky, T. (2000), On Copositive Programming and Standard Quadratic Optimization Problems, Journal of Global Optimization 18, pp. 301–320.

[7] Bomze, I.M.; Jarre, F.; Rendl, F. (2007), A quadratic factorization heuristics for copositive programs, Preprint, in preparation.

[8] Bonnans, J.F.; Ramirez C., H. (2005), Perturbation analysis of second-order cone programming problems, Math. Prog., vol. 104, pp. 205–227.

[9] Burer, S. (2007), On the copositive representation of binary and continuous nonconvex quadratic programs, Preprint, Univ. of Iowa, available at http://www.optimization-online.org/DB_HTML/2006/10/1501.html.

[10] X. D. Chen, D. Sun and J. Sun (2002), Complementarity Functions and Numerical Experiments on Some Smoothing Newton Methods for Second-Order-Cone Complementarity Problems, Comp. Opt. and Appl., vol. 25, pp. 39–56.

[11] Clarke, F.H. (1990), Optimization and nonsmooth analysis (2nd edition), Classics in Applied Mathematics 5, SIAM, Philadelphia.

[12] Conn, A.R.; Gould, N.I.M.; Sartenaer, A.; Toint, P.L. (1996), Convergence properties of an augmented lagrangian algorithm for optimization with a combination of general equality and nonlinear constraints, SIAM J. Opt. 6, pp. 674–703.

[13] Conn, A.R.; Gould, N.I.M.; Toint, P.L. (1991), A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds, SIAM J. Numerical Anal. 28, pp. 545–572.

[14] Dantzig, G.B. (1963), Linear Programming and Extensions, Princeton Landmarks in Mathematics and Physics series, Princeton University Press.

[15] Dür, M.; Still, G. (2007), Interior points of the completely positive cone, preprint, Universität Darmstadt.

[16] Ferris, M.C.; Munson, T.S.; Ralph, D. (2000), A Homotopy Method for Mixed Complementarity Problems Based on the PATH Solver, D.F. Griffiths and G.A. Watson (eds.), Numerical Analysis 1999, Research Notes in Mathematics, London: Chapman and Hall, pp. 143–167.

[17] Freund, R.W.; Jarre, F. (2004), A Sensitivity Result for Semidefinite Programs Oper. Res. Letters, vol. 32, pp. 126–132.

[18] Friedlander, M.P.; Saunders, M.A. (2005), A globally convergent linearly constrained Lagrangian method for nonlinear optimization, SIAM J. Opt. 15 (3), pp. 863–897.

[19] Frimannslund, L.; Steihaug, T. (2006), A class of Methods Combining L-BFGS and Truncated Newton, reports in informatics, report no 319.

[20] Golub, G.H.; Van Loan, C.F. (1989), Matrix Computations, 2nd Edn., The John Hopkins University Press, Baltimore, Maryland.

[21] Hall Jr., M.; Newman, M. (1963), Copositive and Completely Positive Quadratic Forms, Proceedings of the Cambridge Philosophical Society 59, pp. 329–339.

[22] Hauk, K.; Jarre, F. (2007), Linear Programs and Implicit Functions, Pacific J. of Opt., Vol. 3, No. 1, pp. 53–72.

[23] Hestenes, M.R. (1969), Multiplier and gradient methods, J. of Opt. Theory and Appl., vol. 4, pp. 303–320.

[24] Hiriart-Urruty, J.B.; Lemarechal, C. (1996), Convex analysis and minimization algorithms, Volume 1, Springer.

[25] Jarre, F.; Rendl, F. (2007), An Augmented Primal-Dual Method for Linear Conic Programs, Preprint, Univ. of Klagenfurt, available at http://www.optimization-online.org/DB_HTML/2007/04/1628.html to appear in: SIAM Journal on Optimization.

[26] Jarre, F.; Stoer, J. (2004), Mathematische Optimierung, Springer.

[27] Karmarkar, N. (1984), A new polynomial-time algorithm for linear programming, Combinatorica 4, pp. 373–395.

[28] Khachiyan, L.G. (1979), A polynomial algorithm in linear programming, Soviet Mathematics Doklady 20, pp. 191–194.

[29] Klee, V.; Minty, G.J. (1972), How Good is the Simplex Algorithm?, in: O. Shisha, editor, Inequalities III, Academic Press, New York, pp. 159–175.

[30] de Klerk, E.; Pasechnik, D.V. (2002), Approximating the stability number of a graph via copositive programming, SIAM Journal on Optimization, Volume 12, Number 4, pp. 875–892.

[31] Kojima, M.; Megiddo, N.; Mizuno, S. (1992), Theoretical convergence of large-step primaldual interior point algorithms for linear programming, Math. Prog. 59, pp. 1–21.

[32] Liu, D.C.; Nocedal, J. (1989), On the Limited Memory BFGS Method for Large Scale Optimization, Math. Prog. 45, pp. 503–528.

[33] Lovasz, L. (1979), On the Shannon capacity of a graph, IIIE Transactions on Information Theory 25, pp. 1–7.

[34] Maxfield, J.E.; Minc, H. (1962/1963), On the Matrix Equation $X'X = A$, Proceedings of the Edinburgh Mathematical Society, 13, pp. 125–129.

[35] Megiddo, N. (1984), Linear programming in linear time when the dimension is fixed, J. ACM 31, pp. 114–127.

[36] Mehrotra, S. (1992), On the implementation of a primal-dual interior-point method, SIAM J. Optim., vol. 2, pp. 575–601.

[37] Mifflin, R. (1977), Semismooth and Semiconvex Functions in Constrained Optimization, SIAM Journal on Control and Optimization, vol. 15, pp. 957–972.

[38] Monteiro, R.; Tsuchiya, T. (2003), A variant of the Vavasis-Ye layered-step interior-point algorithm for linear programming, SIAM J. Opt. 13 Nr. 4, pp. 1054–1079.

[39] Murty, K.G.; Kabadi, S.N. (1987), Some NP-Complete Problems in Quadratic and Linear Programming, Mathematical Programming 39, pp. 117–129.

[40] Nesterov, Y.; Nemirovskii, A. (1994), Interior-Point Polynomial Algorithms in Convex Programming, SIAM, Philadelphia.

[41] Nocedal, J. (1980), Updating Quasi-Newton Matrices With Limited Storage, Mathematics of Computation, Volume 35, Number 151, pp. 773–782.

[42] Nocedal, J.; Wright, S. (1999), 'Numerical Optimization', Chapter 17.

[43] Pang, J.-S.; Sun, D.; Sun, J. (2002), Semismooth Homeomorphisms and Strong Stability of Semidefinite and Lorentz Complementarity Problems, Math. Oper. Res., vol 28, pp. 39–63.

[44] Pietrzykowski, T. (1970), The potential method for conditional maxima in the locally compact metric spaces, Numer. Math. 14 Nr. 4, pp. 325–329.

[45] Powell, M.J.D. (1969), A method for nonlinear constraints in minimization problems, Fletcher, R. (ed), Optimization, Academic Press, New York, pp. 283–298.

[46] Qi, L.; Sun, J. (1993), A nonsmooth version of Newton's method, Math. Prog., vol. 58, pp. 353–367.

[47] Rendl, F. (2004), Solving Semidefinite Programs using Bundle Methods and the Augmented Lagrangian Approach, Plenary talk at Veszprem Optimization Conference: Advanced Algorithms (VOCAL), Veszprem, Hungary, December 13-15, 2004.

[48] Rockafellar, R.T. (1970), Convex analysis, Princeton mathematical series 28, Princeton University Press.

[49] Schmallowsky, K. (2008), On the Regularity of Second Order Cone Programs and an Application to Solving Large Scale Problems, to appear in: Mathematical Methods of Operations Research.

[50] Tardos, E. (1986), A strongly polynomial algorithm to solve combinatorial linear programs, Oper. Res. 34 Nr. 2, pp. 250–256.

[51] Vavasis, S.; Ye, Y. (1996), A primal dual accelerated interior point method whose running time depends only on A, Math. Progr. 74, pp. 79–120.

[52] Ye, Y.; Todd, M.J.; Mizuno, S. (1994), An $O(\#nL)$-iteration homogeneous and self-dual linear programming algorithm, Math. of Oper. Res. 19, pp. 53–67.

Statement of Originality

Die hier vorgelegte Disseration habe ich eigenständig und ohne unerlaubte Hilfe angefertigt. Die Dissertation wurde von der vorgelegten oder in ähnlicher Form noch bei keiner anderen Institution eingereicht. Ich habe bisher keine erfolglosen Promotionsversuche unternommen.

I do herewith declare that the material contained in this dissertation is an original work performed by me without illegitimate help. The material in this thesis has not been previously submitted for a degree in any University.

Düsseldorf, den 27.05.2008

Katrin Schmallowsky