Numerical Treatment of Nonlinear Semidefinite Programs

Inaugural-Dissertation

zur
Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Christoph Helmut Vogelbusch

aus Ratingen

Dezember 2006

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. Helmut Ratschek

Tag der mündlichen Prüfung: 31.01.2007

Zusammenfassung

Die Dissertation *Numerical Treatment of Nonlinear Semidefinite Programs* diskutiert zwei Algorithmen zum Lösen von nichtlinearen, nicht konvexen semidefiniten Programmen (SDPs).

Ausgangspunkt für die Forschungen waren mathematische Probleme, die bei der Schaltkreissimulation[1] auftreten. Für diese als nichtlineare SDPs formulierbaren Probleme existierte kein brauchbarer Löser.

Ein nichtlineares SDP wird in dieser Dissertation gegeben durch

$$\min\{\ C \bullet X \mid F(X) = 0,\ X \in \mathcal{S}_+\ \}$$

Hierbei sind $C, X \in \mathcal{S}^n$ symmetrische Matrizen und $F : \mathcal{S}^n \to \mathbb{R}^m$ eine nichtlineare Abbildung. $C \bullet X$ ist das Standard-$\mathbb{R}^{n\times n}$-Skalarprodukt, gegeben durch die Spur des Produkts $C^{\mathrm{T}}X$,

$$C \bullet X := \operatorname{tr}(C^{\mathrm{T}}X).$$

Die Menge $\mathcal{S}^n_+$ ist der Teilkegel der reellen Matrizen in $\mathbb{R}^{n\times n}$, der die symmetrischen Matrizen mit nicht negativen Eigenwerten umfasst. Weiter bezeichnen wir mit $\mathcal{S}^n_{++}$ den Kegel der symmetrischen Matrizen, deren Eigenwerte positiv sind.

Eine Semidefinitheitsbedingung lässt sich über die Unterdeterminanten auch als nichtlineare Bedingung schreiben. Eine Berücksichtigung solcher nichtlinearen Nebenbedingungen führt allerdings zu entarteten Problemen und ist nicht effizient lösbar. In [St05] beschreibt Stingl eine alternative Methode zum Lösen nichtlinearer SDPs, die zur gleichen Zeit wie diese Dissertation entstand. Die Methode in [St05] basiert auf einem modifizierten Barriereansatz und wurde ebenfalls implementiert. In der Praxis werden daher Algorithmen verwendet, die die Semidefinitheitsbedingung als Kegelbedingung berücksichtigen.

Der erste hier vorgestellte Ansatz solche Probleme effizient zu lösen ist das SSP ("Sequential Semidefinite Programs") Verfahren. Dieses Verfahren lehnt sich an das SQP Verfahren an, das die Lösung nichtlinearer Programme durch das sequentielle Lösen quadratischer Programme der Form

$$\min_{\Delta X}\{\ C \bullet \Delta X + \frac{1}{2}B[\Delta X, \Delta X] \mid F(X) + \mathrm{D}F(X)[\Delta X] = 0,\ X + \Delta X \in \mathcal{S}_+\ \}$$

annähert. Beim SSP Verfahren wird die Lösung eines nichtlinearen semidefiniten Programms durch eine Folge von linearen SDPs angenähert. Erste analytische Ergebnisse bescheinigen dem SSP Verfahren die gleichen theoretischen Konvergenz Ergebnisse wie dem SQP Verfahren. Diese Analysen verwenden für die linearisierten Probleme die exakte Hessematrix der Lagrangefunktion. Leider gibt es für die daraus entstehenden approximativen Probleme nur dann effiziente Löser, wenn diese Matrix positiv semidefinit ist. Für den SQP Fall gibt es unter milden Voraussetzungen positiv semidefinite Approximationen, die auf den linearisierten Nebenbedingungen mit der Hessematrix der Lagrangefunktion übereinstimmen.

[1] Unsere Beispiele wurden gestellt von den Lucent/Bell Laboratorien

In dieser Dissertation wird gezeigt, dass es solche positiv semidefiniten Approximationen der Hessematrix der Lagrangefunktion im SSP Fall nicht immer gibt. Die Hessematrix von nichtlinearen SDPs kann auf den linearisierten Nebenbedingungen negative Eigenwerte haben. Der Grund hierfür ist, dass die Randkrümmung des semidefiniten Kegels nicht in der Lagrangefunktion eingeht.

Weiter zeigen wir an Hand eines Beispiels, dass das SSP Verfahren für jede beschränkte positiv semidefinite Approximation der Hessematrix nicht schneller als linear konvergieren kann. Dies bildet einen überraschenden Kontrast zu dem SQP-Verfahren.

Im Rahmen dieser Dissertation wurde das SSP Verfahren auch implementiert, um die Schaltkreis-Probleme zu lösen. Es zeigt sich, dass das SSP Verfahren eine gute globale Konvergenz für die getesteten Probleme hat.

Im Kapitel über die SSP-Implementierung stellen wir eine neue Schrittweitenkontrolle vor: die erweiterten Filter. Das SSP Verfahren hat für alle uns vorliegenden Beispiele mit den erweiterten Filtern eine deutlich schnellere Konvergenz als eine Penalty-Linesearch oder die standard Filter Methode. Eine besondere Eigenschaft dieser erweiterten Filter ist, dass sie einen guten Indikator liefern, wann wir Nahe am Optimum sind, wann es also sinnvoll wäre auf einen Löser mit schneller lokal Konvergenz zu wechseln.

Um die Stärken, die schnelle globale Konvergenz, des SSP Verfahrens zu nutzen und Schwächen des SSP Verfahrens, die lineare lokale Konvergenz, zu vermeiden schlagen wir in dieser Arbeit einen hybriden Löser vor. Dieser hybride Löser verwendet das SSP Verfahren um dem Optimum nahe zu kommen und wechselt dann zu einen schnellen lokalen Löser.

Als schnellen lokalen Löser betrachten wir eine Innere-Punkte-Methode (IPM) .

Für diese IPM, wird zunächst ein zentraler Pfad in der Nähe der Optimal Lösung definiert. Wir stellen einen Prädiktor-Korrektor Algorithmus vor, der diesem Pfad folgt. Um beim Folgen dieses Pfades eine Abstiegsrichtung zu erhalten, muss die Summe

$$\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$$

auf den linearisierten Nebenbedingungen positiv definit sein. Hierbei ist $\mathcal{H}$ die Hessematrix der Lagrangefunktion und $\mathcal{F}^{-1}\mathcal{E}$ ein Symmetrisierungsterm. Da der Term $\mathcal{F}^{-1}\mathcal{E}$ bereits positiv definit ist, ist ein naheliegender Ansatz, eine positiv semidefinite Approximation von $\mathcal{H}$ zu verwenden. Da die Suchschritte der IPM denen des SSP Verfahrens sehr ähneln, folgt auch für das IPM bei einer positiv semidefiniten Approximation von $\mathcal{H}$ eine langsame Konvergenz.

Wir zeigen, dass eine positiv definite Approximation von $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \in \mathcal{S}^n_{++}$ gefunden werden kann, für die die quadratische Konvergenz erhalten bleibt, falls eine schwache Barrierebedinung erfüllt ist.

Theoretisch wird für den Prädiktorschritt eine Approximation benötigt, die in nur gewisse Richtungen positiv definit ist. Für eine generelle positiv definite Approximation, ergeben aber einige Vorteile. Da eine solche Approximation sowohl für Prädiktor- als auch den Korretkorschritt verwendet werden kann, ist es möglich Rang-1 oder Rang-2 Updates für $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ zu definieren. Diese können dann gegen eine solche positiv definite Approximation von $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ konvergieren und eine superlineare Konvergenz erreichen. Zusätzlich ist die Dekomposition einer positiv definiten Matrix effizienter, da für diese das Choleskiverfahren verwendet werden kann.

Abstract

The dissertation *Numerical Treatment of Nonlinear Semidefinite Programs* discusses two algorithms that solve nonlinear, nonconvex semidefinite programs (SDPs).

Basis for this thesis were problems that occur in circuit simulation[2]. A good model for these problems are nonlinear SDPs. There was no suitable solver for such problems. In [St05] Stingl describes another method for solving nonlinear SDPs that was developed simultaneously to this thesis. The method in [St05] is based on a modified barrier approach and also includes a numerical implementation.

We consider nonlinear SDPs of the form

$$\min\{\ C \bullet X \mid F(X) = 0,\ X \in \mathcal{S}_+\ \}$$

where $C, X \in \mathcal{S}^n$ are symmetric matrices and $F : \mathcal{S}^n \to \mathbb{R}^m$ is a nonlinear mapping. $C \bullet X$ is the standard-$\mathbb{R}^{n\times n}$- scalar product, given by the trace of the product $C^{\mathrm{T}}X$,

$$C \bullet X := \operatorname{tr}(C^{\mathrm{T}}X).$$

The cone $\mathcal{S}^n_+$ is the set of real matrices $\mathbb{R}^{n\times n}$, that are symmetric and only have non negative eigenvalues. $\mathcal{S}^n_{++}$ denotes the cone of symmetric matrices and have only positive eigenvalues.

A semidefinite constraint can be reformulated as nonlinear inequality constraints for the determinants of the principal submatrices. This formulation leads to degenerated problems that are not efficiently solvable. In practical applications algorithms are used that consider the semidefinite constraint directly as a cone constraint.

The first approach presented here is the SSP ("Sequential Semidefinite Programs") method. This method is similar to the SQP method. The SSP method solves nonlinear SDPs by approximating them with a sequence of quadratic SDPs

$$\min_{\Delta X}\{\ C \bullet \Delta X + \frac{1}{2}B[\Delta X, \Delta X] \mid F(X) + \mathrm{D}F(X)[\Delta X] = 0,\ X + \Delta X \in \mathcal{S}_+\ \}.$$

It can be shown that the SSP method processes nearly the same theoretical convergence properties as SQP methods. The analysis uses the exact Hessian for the linearized programs. Efficient solvers for these linearized programs only exist, if the approximation of the Hessian of the Lagrangian is positive semidefinite. Under mild conditions there exist positive semidefinite approximations that lead to fast convergence for the SQP approach. These approximations are identical with the Hessian of the Lagrangian on the set that satisfies the linearized constraints.

In this thesis we prove, that such approximations of the Hessian of the Lagrangian in general do not exist for the SSP approach. The Hessian of the Lagrangian for nonlinear SDPs can have negative eigenvalues on set that satisfies the linearized constraints. The reason for this is that the curvature of the boundary of the semidefinite cone is not represented in the Lagrangian.

[2]Numerical examples of such problems were given by Lucent/Bell Laboratories.

We use an example to prove that for the SSP approach we cannot expect more than linear convergence for any choice of bounded semidefinite approximation of the Hessian of the Lagrangian. This is a surprising contrast to the SQP approach.

For this dissertation the SSP approach has been implemented to solve the given problems. The implementation shows that the SSP approach has a good global convergence speed for the given problems.

In the chapter "Implementation" we present a new step length control: the augmented filter. For the given examples the SSP approach using the augmented filter had a much faster convergence speed than with a penalty line search or the standard filter method. A special property of the augmented filter method is that it provides a good indicator for closeness to the optimum. This indicator can be used to switch to a fast local solver. This allows us to define a hybrid solver. This hybrid solver uses the SSP approach for the global convergence and then switches to a fast local solver when the current iterate is close to the optimum.

We propose an interior point method (IPM) as fast local solver.

We first introduce a central path for this IPM close to the optimal solution. We then present a predictor corrector algorithm, that follows this central path. To obtain a descent direction from the predictor step the sum

$$\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$$

has to be positive definite on the set that satisfies the linearized constraints. The matrix $\mathcal{H}$ is the Hessian of the Lagrangian and $\mathcal{F}^{-1}\mathcal{E}$ is a term that comes from matrix symmetrization. The analysis shows that $\mathcal{F}^{-1}\mathcal{E}$ is positive definite. An obvious approach would be to use a positive semidefinite approximation for $\mathcal{H}$. Since the IPM and the SSP method have both very similar search steps, using a positive semidefinite approximation for $\mathcal{H}$ would again lead to slow convergence.

We show that a positive definite approximation for $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \in \mathcal{S}^n_{++}$ exists, that preserves the quadratic convergence if a weak barrier condition is satisfied.

Theoretically this approximation for the predictor step only needs to be positive definite along certain directions. A general positive definite approximation that leads to quadratic convergence yields some advantages. This approximation can be used for the predictor as well as for the corrector steps. It is possible to define a low rank update for $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$. This update can converge to such a positive definite approximation of $\mathcal{H}+\mathcal{F}^{-1}\mathcal{E}$ and yield superlinear convergence. The decomposition of the approximation matrix is the main effort of the algorithm. A positive definite matrix can be decomposed using Cholesky's algorithm, thus make an implementation more efficient.

Contents

1 Overview

This thesis is written in such a way that most chapters can be read separately, in particular the SSP and IPM results have their own introduction.

In this chapter we will discuss the landscape of optimization and relate it to the work of this thesis. We summarize and discuss the results of this thesis. Finally, we present an outline of this thesis.

Please note that this chapter does not introduce notations. These follow in the next chapter.

1.1 Optimization classes and current approaches

The problem class focused on in this thesis has a linear objective function and nonlinear constraints as well as positive semidefinite cone constraints. Such problems evolve naturally from real world problems. The solver we present contains strategies that come from the following classes.

A first very well known class is the class of unconstrained nonlinear programs. Especially interesting is the minimization over three times continuously differentiable functions

$$\min\{\ f(x) \mid x \in \mathbb{R}^n\ \} \quad \text{with } f : \mathbb{R}^n \to \mathbb{R},\ f \in \mathcal{C}^3. \tag{1.1}$$

A typical solver for such problems if f is convex is Newton's algorithm[1] for $\mathrm{D}f(x) = 0$. The IPM we present here is based upon Newtons algorithm and we will refer to the quadratic convergence result several times. For non convex function we use variants of this algorithm that use positive definite approximations of $\mathrm{D}^2 f(X)$ to generate descent directions. We will show a similar descent property for our predictor step in section 7.4.

Another important optimization class is the class of linear programs. These have a linear objective function and linear constraints and are typically restricted to the cone of positive variables. One formulation of linear programs is

$$\min\{\ c^{\mathrm{T}} x \mid Ax = b,\ x \in \mathbb{R}^n_+\ \} \quad \text{with } A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n. \tag{1.2}$$

A first method to find an exact solution is the simplex method (see [Da66]). This method is typically fast in practical applications but has exponential convergence in worst case scenarios and cannot be generalized to nonlinear or conic programs. Another method, that is in opposite to the simplex method easy to generalize, is the interior point method (IPM). IPM methods with very good theoretical properties and convergence results exist. Mainly short-step methods were used for theoretical results while implementation were made using predictor-corrector approaches. Today efficient IPM solvers are competitive to the simplex method in the average case and better than the simplex method in worst case scenarios. Polynomial lower bounds for the convergence rate of specific IPM solvers are known.

[1] The general algorithm is to calculate a zero x^* of a function $g(x)$ by setting the next iterate $x_{k+1} = x_k - \mathrm{D}g(x)^{-1} g(x)$ starting a x_0. The main result used here is local quadratic convergence towards x^* for any function $g \in \mathcal{C}^2$ for that $\det \mathrm{D}g(x^*) \neq 0$.

One generalization of linear programs are linear conic programs where the cone of positive real vectors is replaced by another convex cone. We focus here on linear semidefinite programs (SDP) that are linear programs over variables from the cone of positive semidefinite (PSD) matrices $\mathcal{S}^n_+ \subset \mathcal{S}^n = \{\ X \in \mathbb{R}^{n\times n} \mid X = X^{\mathrm{T}}\ \}$ of the form

$$\min\{\ C \bullet X \mid \mathcal{A}[X] = b,\ X \in \mathcal{S}^n_+\ \} \quad \text{with } \mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m \text{ (linear)}, C \in \mathcal{S}^n, \tag{1.3}$$

with $C \bullet X$ being the scalar product $C \bullet X = \mathrm{trace}(C^{\mathrm{T}}X)$.

Typically such problems are solved by IPM that respect the semidefinite cone. Several papers were published on SDPs as they arise naturally from practical interesting problems. One well known paper on SDPs is [VB96] and a robust solver for linear conic programs including SDPs is e.g. SeDuMi see [St99].

These IPMs are based on two properties of $\mathcal{S}^n_+$. The first is the equivalence

$$X \bullet S = 0,\ X, S \in \mathcal{S}^n_+ \quad \Leftrightarrow \quad \frac{1}{2}(XS + SX) = 0,\ X, S \in \mathcal{S}^n_+ \tag{1.4}$$

and second is its convexity. We will discuss these properties in detail in chapter 6.

Another generalization of linear programs are constrained nonlinear programs. A typical representative is

$$\min\{\ c^{\mathrm{T}}x \mid F(x) = 0,\ x \in \mathbb{R}^n_+\ \} \quad \text{with } F : \mathbb{R}^n \to \mathbb{R}^m,\ F \in \mathcal{C}^3, c \in \mathbb{R}^n. \tag{1.5}$$

A well known solver is the sequential quadratic programming (SQP) algorithm (see e.g. [BT95]). The SQP algorithm generates a series of quadratic programs. These quadratic subprograms can be solved by a conical linear program solver mentioned above, if the quadratic term is positive semidefinite. We will show in chapter 4 that this condition can be fatal for generalizations of the SQP algorithm.

The class we examine in this thesis is the class of nonlinear semidefinite programs of the form

$$\min\{\ C \bullet X \mid F(X) = 0,\ X \in \mathcal{S}^n_+\ \} \quad \text{with } F : \mathcal{S}^n \to \mathbb{R}^m,\ F \in \mathcal{C}^3, C \in \mathbb{R}^{n\times n}. \tag{1.6}$$

This is a hybrid of the previous two classes, having nonlinear constraints as well as PSD cone constraints. Please note that $\mathbb{R}^n_+$ is a special case of $\mathcal{S}^n_+$ where all non diagonal elements are zero. The boundary manifolds of $\mathcal{S}^n_+$ has a non zero curvature, while the sub-cone $\mathbb{R}^n_+$ has zero curvature manifolds.

Some papers have been published (see e.g. [CR04],[FJV06]) to solve nonlinear SDPs by using a SQP like approach called SSP. The SSP approach creates sequence of quadratic semidefinite programs, hence the name SSP. The cited papers prove quadratic convergence under conditions that are too weak to solve the generated subproblems with existing linear SDP solvers. We will analyze this algorithm for applicability. We also present and analyze an algorithm that evolves by extending an IPM for linear SDPs to nonlinear SDPs.

1.2 Outline of this thesis

In the next chapter we introduce basic notations and tools. Then we introduce the idea of the SSP solver in detail in chapter 3. Here we also introduce the sequential linear conic programs (SLCP) solver that technically solves the same problems as the SSP, but with reduced sizes for sub-cones of the semidefinite cone.

One main result of this thesis is that in practical applications only linear convergence for the SSP is guaranteed. We will present this result in chapter 4 by giving a counter-example. On the other hand our implementation of the SSP algorithm shows fast global convergence. In consequence we suggest a hybrid algorithm with a fast local algorithm. We describe our implementation in chapter 5.

The second part of this thesis presents such a fast local algorithm. It is a IPM solver for that we show local quadratic convergence. The central path of the IPM is defined in chapter 6. In chapter 7 we presend an an algorithm that follows this central path. Finally, in chapter 8 we discuss the local quadratic convergence.

2 Notations and Basics

In this chapter we will discuss the class of problems considered this thesis. We will start by introducing notations and naming conventions. Then we will present the class in its standard forms. Finally, we will introduce the optimality conditions for this class.

2.1 Notations

A matrix $A \in \mathbb{R}^{n\times n}$ is called symmetric positive (semi-)definite if $A = A^{\mathrm{T}}$ and its eigenvalues are all positive (or zero). We use $\mathcal{S}^n_+$ for symmetric positive semidefinite matrices and S^n_{++} is the open cone of symmetric positive definite matrices.

We also use the following notation for semidefinite variables

$$X \in \mathcal{S}^n_+ \Leftrightarrow X \succeq 0. \tag{2.1}$$

The notation $\succeq$ can be extended to be a half-order by defining

$$A \succeq B \Leftrightarrow A - B \succeq 0. \tag{2.2}$$

In this sense we can write $X \preceq 0$ for negative semidefinite variables.

For nonlinear functions, $\mathcal{C}^\alpha$ denotes the class of functions that are α times continuously differentiable. Domain and range of such a function are given separately. Typically we focus on functions

$$F : \mathcal{S}^n_+ \to \mathbb{R}^m, F \in \mathcal{C}^3 \tag{2.3}$$

for the nonlinear constraints in (1.6).

Throughout this thesis we will use $\langle \cdot, \cdot \rangle$ for the standard scalar product for vectors or matrices. For an $n \times n$ matrix A let $\mathrm{tr}(A)$ denote the trace of A. The standard scalar product for matrices is

$$\langle A, B \rangle = A \bullet B = \mathrm{tr}(A^{\mathrm{T}} B). \tag{2.4}$$

If we want to point out that this is the scalar product of two matrices $A, B \in \mathbb{R}^{n\times n}$ we write $A \bullet B$ instead of $\langle A, B \rangle$. This notation is standard in semidefinite programming.

When applying a linear operator in semidefinite programming we use brackets $[\,\cdot\,]$. For variables x in a vector representation a linear operator A can be represented as a matrix. But when a variable X is a matrix the operator $\mathcal{A}$ is typically represented by a set of matrices $\mathcal{A}_i$ or $\mathcal{A}_{ij}$ that is applied via a scalar product

$$\mathcal{A}[X] = \begin{pmatrix} \mathcal{A}_1 \bullet X \\ \vdots \\ \mathcal{A}_m \bullet X \end{pmatrix} \quad \text{or} \quad \mathcal{A}[X] = \begin{pmatrix} \mathcal{A}_{11} \bullet X & \dots & \mathcal{A}_{1n} \bullet X \\ \vdots & \ddots & \\ \mathcal{A}_{n1} \bullet X & & \mathcal{A}_{nn} \bullet X \end{pmatrix}. \tag{2.5}$$

Operators that apply to matrices are written with square brackets $[\,\cdot\,]$ e.g. $\mathrm{D}F(X)[\Delta X]$ means the linear operator $\mathrm{D}F(X)$ is applied to ΔX, while $DF(x)\Delta x$ is the application that is equivalent to the matrix multiplication. For numerical computations the columns of an $n \times n$-matrix X are stacked on each other resulting in a vector $x = vec(X) \in \mathbb{R}^{n^2}$.

For such a $x = vec(X) \in \mathbb{R}^{n^2}$ the matrix $A \in \mathbb{R}^{m \times n^2}$ is defined by the relation $Ax = \mathcal{A}[X]$ to the operator $\mathcal{A}[\ ]$. For the scalar product $C \bullet X$ this notation leads to the standard vector scalar product as $\mathrm{tr}(C^{\mathrm{T}}X) = c^{\mathrm{T}}x$ for the vector representation $x = vec(X)$ and $c = vec(C)$. We will explicitly point out when we use this transformation to ease the reading in this thesis.

For vector valued functions $F(x)$ we distinguish between $\mathrm{D}F(x)$ and $\nabla F(x)$. We use $\mathrm{D}F(x)$ when we use it as a linear operator to apply a multiplication from the right hand side, like $\mathrm{D}F(x)[\Delta x] = \mathrm{D}F(x)\Delta x$. The notation $\nabla F(x)$ simply means the transpose of the derivative such that $\nabla F(x) = \mathrm{D}F(x)^{\mathrm{T}}$ and is just used for convenience.

Definition 2.1.1. *A cone $\mathcal{K} \subset \mathbb{R}^n$ is called selfdual if the dual cone*

$$K^D := \{\ x \in \mathbb{R}^n \mid \langle x, y \rangle \geq 0,\ \forall y \in \mathcal{K}\ \} \tag{2.6}$$

and the cone itself coincide $K = K^D$.

The cone $\mathcal{S}^n_+$ is a convex, selfdual cone. A proof for the selfduality is well known as Féjer's Theorem (see e.g. [HJ85]). In linear programs the selfduality is used to define a primal dual starting point problem. Here the selfduality reflects in the convenience that the dual cone variable has the same properties as the primal variable.

In section 8.3.1 we will show some properties on the more general scale of Jordan Algebras[1]. We will also show that $\mathcal{S}^n_+$ is a set of squares over a specific Jordan Algebra. This property allows us to formulate a stronger complementarity condition.

The cone $\mathcal{S}^n_+$ also includes two other important cones that are treated separately often throughout this thesis. The first is the Lorentz cone also known as quadratic cone[2] $\mathcal{Q}$ that is defined by

$$\mathcal{Q} := \left\{ \begin{pmatrix} x_0 \\ \bar{x} \end{pmatrix} \in \mathbb{R}^{n+1} \;\middle|\; x_0 \geq \|\bar{x}\|_2 \right\}. \tag{2.7}$$

The relation

$$\begin{pmatrix} x_0 \\ \bar{x} \end{pmatrix} \in \mathcal{Q}^{n+1} \Leftrightarrow \begin{pmatrix} x_0 & \bar{x}_1 & \dots & \bar{x}_n \\ \bar{x}_1 & x_0 & & \\ \vdots & & \ddots & \\ \bar{x}_n & & & x_0 \end{pmatrix} \in \mathcal{S}^{n+1}_+ \tag{2.8}$$

between $\mathcal{Q}$ and $\mathcal{S}^n_+$ is easy to verify.

On the other hand if we force all non diagonal entries of a Matrix $X \succeq 0$ to be zero $\mathcal{S}^n_+$ is equivalent to $X_{ii} \in \mathbb{R}_+$.

The cone of positive variables is selfdual by definition. The following result is well known, we give a short proof.

Proposition 2.1.2. *The Lorentz cone is selfdual.*

Proof. Let $a \in \mathbb{R}^n$ with

$$a^{\mathrm{T}}b \geq 0 \quad \forall b \in \mathcal{Q}. \tag{2.9}$$

As $\mathcal{Q}$ includes

$$\tilde{b} := \begin{pmatrix} \|\bar{a}\| \\ -\bar{a} \end{pmatrix} \tag{2.10}$$

we have

$$0 \leq a^{\mathrm{T}}\tilde{b} = a_0\|\bar{a}\| - \bar{a}^{\mathrm{T}}\bar{a} = \|\bar{a}\|(a_0 - \|\bar{a}\|) \tag{2.11}$$

[1] For details on Jordan Algebras see [WSV00]

[2] also called second order cone and ice cream cone

thus $\|\bar{a}\| \leq a_0$ and $a \in \mathcal{Q}$. On the other hand when we have $a \in \mathcal{Q}$ and any $b \in \mathcal{Q}$ it follows

$$a_0 b_0 + \bar{a}^{\mathrm{T}} \bar{b} \geq a_0 b_0 - \|\bar{a}\|\|\bar{b}\| \geq a_0 b_0 - a_0 b_0 = 0. \tag{2.12}$$

□

2.1.1 Naming conventions

If not stated separately we use the following naming conventions thoughout this thesis.

When minimizing with respect to a variable x or X we often omit the name of the variable and write shortly $\min\{ \dots \}$ in place of

$$\min_x\{ \dots \} \text{ or } \min_X\{ \dots \}. \tag{2.13}$$

A small letter x denotes a vector and a capital letter X is used to indicate a matrix. The same applies for s and S respectively. s and S are used as dual cone variable. The dual variable for the equality constraints is the vector y. We assume X and S to be symmetric.

Linear constraints are typically represented by $Ax = b$ and $A[X] = b$ respectively. Nonlinear constraints are notated as $F(x) = 0$ and $F(X) = 0$ respectively. Conic constraints come as $X \in \mathcal{S}^n_+$ or $X \succeq 0$ or in the general case $x \in \mathcal{K}$ and $X \in \mathcal{K}$ respectively. $\mathcal{K}$ usually denotes a cartesian product of $\mathbb{R}_+$, $\mathcal{Q}$ and $\mathcal{S}^n_+$.

For the theoretical analysis we focus on a single iterate, thus we omit the iteration index k. Thus we write x for the current iterate, Δx for the current step and x_+ for the next iterate. We apply the naming convention to the other variables such as s and y as well.

If we describe a complete algorithm we use $x^{(k)}$, $s^{(k)}$ etc. for the current iterate and $k-1$ or $k+1$ for the previous and next respectively.

With $\mathcal{L}(x, y, s)$ and $\mathcal{L}(X, y, S)$ respectively we denote the Lagrangian of the optimization problem a section is currently focussing on. We also use

$$\begin{aligned} \mathrm{g}(x, y, s) &:= \mathrm{D}_x \mathcal{L}(x, y, s), & \mathrm{g}(X, y, S) &:= \mathrm{D}_X \mathcal{L}(X, y, S), \\ \mathcal{H}(x, y, s) &:= \mathrm{D}^2_{xx} \mathcal{L}(x, y, s), & \mathcal{H}(X, y, S) &:= \mathrm{D}^2_{XX} \mathcal{L}(X, y, S). \end{aligned}$$

For the augmented Lagrangian we use $\Lambda(x, y, s)$ and $\Lambda(X, y, S)$ respectively. Its derivatives are

$$\begin{aligned} g_+(x, y, s) &:= \mathrm{D}_x \Lambda(x, y, s), & g_+(X, y, S) &:= \mathrm{D}_X \Lambda(X, y, S), \\ \mathcal{H}_+(x, y, s) &:= \mathrm{D}^2_{xx} \Lambda(x, y, s), & \mathcal{H}_+(X, y, S) &:= \mathrm{D}^2_{XX} \Lambda(X, y, S). \end{aligned}$$

As we fixed the names of A and $F(X)$, etc. we often omit the argument. Thus we write F instead of $F(X)$ or $F(X^{(k)})$. This short notation will always be pointed out explicitely.

2.2 The class examined here

In this thesis we present a solver for nonlinear semidefinite programs (SDP). Nonlinear semidefinite programs have nonlinear equality constraints as well as positive semidefinite (PSD) cone constraints.

In some real-world problems PSD conditions occur naturally. Many constraints such as classes of polynomial inequality constraints can be reformulated as PSD constraints. A nonlinear formulation of a PSD cone constraint would be very CPU time expensive as well as bare several other problems. Keeping the PSD cone constraints in linear program has proven to be very efficient as the PSD cone is convex.

There are different formulations for SDPs. We keep consistency by using one problem-formulation and its generalizations throughout this thesis.

The nonlinear SDP version is

$$\min\{\ C \bullet X \mid F(X) = 0,\ X \in \mathcal{S}_+^n\ \} \tag{2.14}$$

it is a generalization of the following linear SDPs

$$\min\{\ C \bullet X \mid A[X] = b,\ X \in \mathcal{S}_+^n\ \}, \tag{2.15}$$

which is a generalization of the standard form for linear programs

$$\min\{\ c^{\mathrm{T}}x \mid Ax = b,\ x \geq 0\ \}. \tag{2.16}$$

As we stated in section 2.1 the semidefinite cone also contains the quadratic cone, the cone of positive vectors, as well as free variables. The results for the algorithms presented in this thesis can easily be generalized to $X \in \mathcal{K}$ with $\mathcal{K}$ being a cartesian product of the cones mentioned above. We obtain the generalized problem

$$\min\{\ c(x) \mid F(x) = 0,\ x \in \mathcal{K}\ \} \text{ or } \min\{\ c^{\mathrm{T}}x \mid F(x) = 0,\ x \in \mathcal{K}\ \}. \tag{2.17}$$

For the SQP approach this cone condition is simply shifted to the subproblems and taken into account by the linear conic solvers, such as SeDuMi (see [St99]). SeDuMi supports a cartesian product of these cones.

For the IPM presented in this thesis these cones have to respect specifically only two properties. First for the complementary condition, where every cone has its special multiplication and one-element. These multiplications and one-elements come from the Jordan Algebra for which the specified cones are sets of squares. We will cover Jordan Algebras in sections 8.2.2 and 8.3.1. Second we have to consider the cone specifically for the maximal step length, namely to guarantee that we do not hit or pass the boundary of the cone.

2.3 Optimality conditions

Throughout this thesis we use the Karush Kuhn Tucker (KKT) first order optimality conditions. We consider the conic program

$$\min_x\{\ c(x) \mid F(x) = 0,\ x \in \mathcal{K}\ \}, \tag{2.18}$$

which is the most general form we use in this thesis. Under standard assumptions (existence of a solution and the MFCQ condition) an optimal solution (x^*, y^*, s^*) exists and fullfills the KKT conditions (see [JS03]) in the optimum, i.e.

$$\begin{aligned} x^* &\in \mathcal{K}, \\ s^* := -\mathrm{D}c(x^*) - y^{*\mathrm{T}}\mathrm{D}F(x^*) &\in \mathcal{K}^D, \\ F(x^*) &= 0, \\ \langle s^*, x^* \rangle &= 0. \end{aligned} \tag{2.19}$$

Let $\mathcal{L}(x, y, s)$ be the Lagrangian of (2.18)

$$\mathcal{L}(x, y, s) := c(x) + y^{\mathrm{T}}\mathrm{D}F(x) + \langle x, s \rangle. \tag{2.20}$$

The KKT conditions include the original cone constraint $x \in \mathcal{K}$ and the first order condition for a critical point of the Lagrangian

$$
\begin{aligned}
&- \mathrm{D}c(x) - y^{\mathrm{T}}\mathrm{D}F(x) \in \mathcal{K}^D \Leftrightarrow \\
&\qquad \mathrm{g}(x,y,s) := \mathrm{D}_x\mathcal{L}(x,y,s) = \mathrm{D}c(x) + y^{\mathrm{T}}\mathrm{D}F(x) + s = 0, \quad s \in \mathcal{K}^D. \quad (2.21)
\end{aligned}
$$

The equation $F(x) = 0$ of the KKT conditions is the nonlinear constraint from (2.18). The last condition in 2.19 is the complementarity condition, stating that there always exists a primal dual pair $x \in \mathcal{K}$, $s \in \mathcal{K}^D$ that is orthogonal.

These optimality conditions are the basis, for both the SSP approach and the IPM approach for solving nonlinear SDPs. The conditions $x \in \mathcal{K}$ and $-\mathrm{D}c(x) - y^{\mathrm{T}}\mathrm{D}F(x) \in \mathcal{K}^D$ are not respected by Newton type algorithms. The SSP method generates quadratic subproblems that are similar to Newton steps, but include the cone condition. The IPM introduces a relaxation to the last equation $\langle s, x\rangle = 0$ to force the search step to point to a cone-feasible point.

3 Basis of the SSP

The SSP algorithm solves nonlinear non convex semidefinite programs. It is a generalization of the sequential quadratic programming (SQP) algorithm.

In the following we will introduce the SQP algorithm on which the SSP algorithm is based. Then we will present current work on the SSP method followed by our research in the next chapter.

3.1 The SQP approach

The SQP algorithm (see [BT95]) solves nonlinear programs of the form

$$\min_x \{\ c^{\mathrm{T}}x \mid F(x) = 0,\ x \in \mathbb{R}^n_+\ \}. \tag{3.1}$$

With $c \in \mathbb{R}^n$ and $F(x) : \mathbb{R}^n \to \mathbb{R}^m$ a nonlinear C^3 function[1].

The Lagrangian for (3.1) is

$$\mathcal{L}(x, y, s) := c^{\mathrm{T}}x + y^{\mathrm{T}}F(x) + \langle x, s\rangle. \tag{3.2}$$

An optimal pair (x^*, y^*, s^*) of problem (3.1) satisfies the following conditions equivalent to the first order optimality (KKT) conditions

$$\begin{aligned} \mathrm{g}(x^*, y^*, s^*) := \mathrm{D}_x\mathcal{L}(x^*, y^*, s^*) = c + \mathrm{D}F(x^*)^{\mathrm{T}}y^* + s^* &= 0,\\ F(x^*) &= 0,\\ S^*x^* &= 0,\\ s^*, x^* &\in \mathbb{R}^n_+, \end{aligned} \tag{3.3}$$

with $S^* = \mathrm{Diag}(s^*)$ being a square matrix with s^* on its diagonal. Note that $s_i^*x_i^* = 0$ for $1 \le i \le n$ since $s^* \ge 0$, $x^* \ge 0$ elementwise and $\langle s^*, x^*\rangle = 0$. Thus the conditions

$$S^*x^* = 0,\ s^* \ge 0,\ x^* \ge 0$$

are equivalent to the conditions

$$\langle x^*, s^*\rangle = 0,\ s^* \ge 0,\ x^* \ge 0$$

as used in the KKT conditions. The KKT conditions consist of the following conditions: x^* is a feasible point, (x^*, y^*, s^*) is a critical point of the Lagrangian, s^* is in the dual cone and the primal/dual cone variables x^*, s^* are orthogonal to each other $\langle s^*, x^*\rangle = 0$.

For these conditions the dimension of the variables as well as the dimension of the equality constraints are $2n + m$.

Dropping the cone conditions $x \in \mathbb{R}^n_+$, $s \in \mathbb{R}^n_+$ one could use Newton's algorithm or a variant to find a solution of (3.3), but this could lead to an infeasible solution, with $x_i < 0$ or $s_i < 0$.

[1] The condition $f \in C^3$ is stronger than in [BT95].

To respect the cone constraints the SQP algorithm defines a sequence of quadratic "sub"problems to solve (3.1) thus the name sequential quadratic programming. For a given iterate x these subproblems have the form

$$\min_{\Delta x}\{\ c^{\mathrm{T}}\Delta x + \frac{1}{2}\Delta x^{\mathrm{T}} B\Delta x \mid F(x) + DF(x)\Delta x = 0,\ x + \Delta x \in \mathbb{R}^n_+\ \}. \tag{3.4}$$

The symmetrical $n \times n$ matrix B is an approximation of the Hessian of the Lagrangian. We omit the index when we focus on a single subproblem and the variable is associated to the current k-th iterate. We add the subscript $+$ (such as x_+) if we associate this variable with the next iterate $k+1$. A letter Δ in front of the variable is used for the current step thus $\Delta x = x_+ - x$.

Using these notations the Lagrangian for subproblem (3.4) is given by

$$\tilde{\mathcal{L}}(\Delta x, y_+, s_+) := c^{\mathrm{T}}\Delta x + \frac{1}{2}\Delta x^{\mathrm{T}} B\Delta x + y_+^{\mathrm{T}}(F(x) + DF(x)\Delta x) + \langle x + \Delta x, s_+\rangle, \tag{3.5}$$

which we can use to formulate the modified KKT conditions for (3.4)

$$\begin{aligned} \nabla\tilde{\mathcal{L}}_{\Delta x}(\Delta x, y_+, s_+) = c + B\Delta x + DF(x)^{\mathrm{T}} y_+ + s_+ &= 0,\\ F(x) + \mathrm{D}F(x)\Delta x &= 0,\\ S_+(x + \Delta x) &= 0,\\ x + \Delta x, s_+ &\in \mathbb{R}^n_+. \end{aligned} \tag{3.6}$$

By defining $\Delta s := s_+ - s$ as well as $\Delta y := y_+ - y$ and setting $B = \mathcal{H}(x, y, s) := \mathrm{D}_{xx}\mathcal{L}(x, y, z)$ it can be shown that the solution $(\Delta x, \Delta y, \Delta s)$ is almost a Newton step for (3.3).

$$\begin{array}{rr} \text{SQP step:} & \text{Newton step:}\\ c + B\Delta x + DF(x)^{\mathrm{T}} y + s_+ = 0 & c + B\Delta x + \mathrm{D}F(x)^{\mathrm{T}}(\Delta y + y) + \Delta s + s = 0\\ F(x) + \mathrm{D}F(x)\Delta x = 0 & F(x) + \mathrm{D}F(x)\Delta x = 0\\ S_+(x + \Delta x) = 0 & S(\Delta x + x) + \Delta S x = 0\\ x + \Delta x, s_+ \in \mathbb{R}^n_+ & \end{array} \tag{3.7}$$

The noticeable differences are that the SQP solution includes the quadratic term $\Delta S\Delta x$ and respects the cone conditions $x, s \in \mathbb{R}^n_+$. If the active indices are guessed correctly for the Newton step, then these steps are identical. Note that for such a step $\Delta S\Delta x = 0$.

It is well known that the SQP approach with $B = \mathcal{H}(x, y, s)$ converges quadratically (see [JS03]). The idea behind the proof is, that if the iterate is close enough to the optimal solution, the active set is the optimal set. Thus the SQP steps are Newton steps. When the cone $\mathbb{R}^n_+$ is replaced with some more general non polyhedral cone, this line of reasoning needs significant modifications.

The SQP algorithm is shown is algorithm 1. The stopping criterion is typically precision based depending on e.g. derivatives of the KKT conditions as described in section 5.7.

If B is positive definite then there exists a symmetric square root $\sqrt{B}$ with $\sqrt{B}^2 = B$. Note that in practical applications instead of a symmetric square root a Cholesky decomposition is used. Using this square root we can replace the quadratic objective function by a linear objective function and an extra variable and an additional quadratic cone constraint (see sections 4 and 5.4 for details). Today there exist several efficient solvers for such problems such as SeDuMi [SeWWW] and SDPT3 [TTT03]. On the other hand if B is not positive semidefinite finding the global minimum is a NP complete

Algorithm 1 The SQP-Algorithm

let $(x^{(0)}, y^{(0)}, s^{(0)})$ be a given starting point and $k = 0$
let $B^{(0)} \approx \mathcal{H}(x^{(0)}, y^{(0)}, s^{(0)})$ be an approximation of the Hessian of the Lagrangian
while not [stopping criterion] **do**
calculate a Kuhn-Tucker point $(\Delta x, y, s)$ for

$$\min_{\Delta x^{(k)}} \left\{ c^{\mathrm{T}} \Delta x^{(k)} + \frac{1}{2} \Delta x^{(k)\mathrm{T}} B^{(k)} \Delta x^{(k)} \;\middle|\; \begin{array}{r} F(x^{(k)}) + DF(x^{(k)}) \Delta x^{(k)} = 0, \\ x^{(k)} + \Delta x^{(k)} \in \mathbb{R}^n_+ \end{array} \right\}$$

set the next iterate $x^{(k+1)} = x^{(k)} + \Delta x$, $y^{(k+1)} = y$ and $s^{(k+1)} = s$
update $B^{(k+1)} \approx \mathcal{H}(x^{(k+1)}, y^{(k+1)}, s^{(k+1)})$ (e.g. BFGS update)
set $k = k + 1$
end while

problem. There exist solvers that yield good approximation to a local minimum , but these are not generalized for conic programs. Thus if $\mathcal{H}$ is not positive semidefinite a damped BFGS approach or the Hessian of an augmented Lagrangian is used to generate a positive semidefinite B. For these approximations still superlinear convergence holds (convergence for damped BFGS see [Po78], for the augmented Lagrangian [BS00]).

3.2 The SSP and SLCP approach

One approach to solve nonlinear SDPs is to extend the SQP algorithm. The SQP algorithm solves a nonlinear program by a series of approximating simpler programs that preserve the $\mathbb{R}_+$ cone condition. Solvers for the simpler programs are e.g. interior point methods (IPMs).

IPMs have been developed for different classes including linear semidefinite programs such as

$$\min\{\ C \bullet X \mid A[X] = b,\ X \in \mathcal{S}^n_+\ \}. \tag{3.8}$$

Recall that the quadratic cone $\mathcal{Q}$ is a partial cone of the PSD cone. For a positive semidefinite B the quadratic term can be represented by a variable in the quadratic cone $\mathcal{Q}$. Thus we can reformulate the quadratic semidefinite subprogram into a linear semidefinite program. To reduce variable size, implementations such as SeDuMi [SeWWW] support the PSD subcones $\mathbb{R}^n_+$ and $\mathcal{Q}$ directly.

In the SQP approach the $\mathbb{R}^n_+$ cone conditions are shifted to the subproblems. In the sequential semidefinite programming (SSP) approach the positive semidefinite cone $\mathcal{S}^n_+$ constraints are shifted to the subproblems. Thus the sequential semidefinite programs are now generated by linearizing the nonlinear constraints and preserving the conic constraints. The objective function of the subproblem for both cases (SQP and SSP) include a quadratic term that is either the Hessian of the Lagrangian or an approximation.

This means that we can solve nonlinear semidefinite problems such as

$$\min\{\ C \bullet X \mid F(X) = 0,\ X \in \mathcal{S}^n_+\ \} \tag{3.9}$$

by solving sequential programs of the form

$$\min_{\Delta x}\{\ C \bullet \Delta X + \frac{1}{2} B[\Delta X, \Delta X] \mid F(X) + DF(X)[\Delta X] = 0,\ X + \Delta X \succeq 0\ \}. \tag{3.10}$$

With a similar argument as in the last section it has been shown (see e.g. [CR04], [FJV06]) that the SSP approach, converges quadratically if the optimal solution X^* satisfies a nondegenerate condition and $B = \mathcal{H}(X, y, S)$ holds. Analogously to the SQP approach the quadratic condition of the SSP subproblem can be respected easily if B is symmetric positive semidefinite. For such a $B \in PSD$ we use solvers for linear SDPs as stated in previous paragraphs. We go into more detail on the subproblem generation at the beginning of the next chapter and in section 5.4 where we discuss the implementation.

Theoretically the SSP approach can be generalized to any cone $\mathcal{K}$ for that quadratic programs are easy to solve. In the following we will present the SSP as well as the more general case of sequential conic linear programming (SLCP). While in theory the semidefinite cone includes the quadratic cone and cone of positive variables, these are supported explicitly by solver implementation to gain lower variable dimensions. Thus for theoretical analysis we consider sequential semidefinite programs (SSP). For examples and the discussion of the implementation we consider sequential linear conic programs (SLCP) to ease the reading and gain efficiency.

For a general cone $\mathcal{K}$ this leads to the following problem

$$\min_x \{\ c^{\mathrm{T}}x \mid F(x) = 0,\ x \in \mathcal{K}\ \}, \tag{3.11}$$

with the associated subproblems

$$\min_{\Delta x} \{\ c^{\mathrm{T}}\Delta x + \frac{1}{2}\Delta x B \Delta x \mid F(x) + DF(x)\Delta x = 0,\ x + \Delta x \in \mathcal{K}\ \}. \tag{3.12}$$

Recall the comparison (3.7) between SQP steps and Newton steps. The situation is similar here, but we have the cone $\mathcal{K}$ instead of the $\mathbb{R}^n_+$ and the multiplication that is derived from the orthogonality of the vectors is different for every $\mathcal{K}$. But it basically comes down to a single quadratic term that makes the difference between the steps once the iterate is close enough to an optimal point. If the cone is a quadratic set over a Jordan algebra, such as the SDP cone, this multiplication is the multiplication from that Jordan algebra:

$$X, S \in \mathcal{S}^n_+,\ X \bullet S = 0 \quad \Leftrightarrow \quad \frac{1}{2}(XS + SX) = 0 \tag{3.13}$$

$$\begin{pmatrix} x_0 \\ \bar{x} \end{pmatrix}, \begin{pmatrix} s_0 \\ \bar{s} \end{pmatrix} \in \mathcal{Q},\ x^{\mathrm{T}}s = 0 \quad \Leftrightarrow \quad \begin{pmatrix} x_0 s_0 + \bar{x}^{\mathrm{T}}\bar{s} \\ x_0\bar{s} + s_0\bar{x} \end{pmatrix} = 0 \tag{3.14}$$

$$x, s \in \mathbb{R}_+,\ x^{\mathrm{T}}s = 0 \quad \Leftrightarrow \quad Sx = 0 \qquad \text{with } S = \operatorname{diag}(S) \tag{3.15}$$

The cones that occur in real-world problems are mostly $\mathcal{Q}$ or $\mathcal{S}^n_+$. We focus especially on the PSD cone $\mathcal{S}^n_+$ for two reasons. One reason is of course that variables of the quadratic cone can be written as positive semidefinite variables. The other reason is that nonlinear SDP occur naturally in problems from reduced circuit simulation that were part of the motivation for this thesis.

A difference between the quadratic cone/positive semidefinite cone and $\mathbb{R}^n_+$ is that the boundaries of first ones have a non zero curvature. Thus for the SQP all curvature informations are represented by the Hessian of the Lagrangian. The SSP has some curvature information preserved in its conic constraints. The curvature of the conic constraint's boundary is not represented by the Lagrangian. Thus it cannot be assumed that the Hessian of this Lagrangian is positive semidefinite on the active set. On the other hand we have to use positive semidefinite approximations to solve the subproblems with existing solvers. In our analysis in the next chapter we take in account that it is necessary to have a positive semidefinite approximation B for the Hessian in oder to be able to find a solution of the SSP subproblem. We will show that this assumption leads to linear convergence.

For the SLCP algorithm 2 given here we already expect B to be positive semidefinite and thus have easy to solve subproblems.

Algorithm 2 The SLCP-Algorithm

let $(x^{(0)}, y^{(0)}, s^{(0)})$ be a given starting point and $k = 0$
let $B^{(0)} \approx \mathcal{H}(x^{(0)}, y^{(0)}, s^{(0)})$ be an approximation of the Hessian of the Lagrangian
while not [stopping criterion] **do**
calculate a Kuhn-Tucker point $(\Delta x, y, s)$ for

$$\min_{\Delta x_k, z_0, \bar{z}} \left\{ c^{\mathrm{T}} \Delta x_k + z_0 \;\middle|\; \begin{array}{ll} F(x^{(k)}) + DF(x^{(k)})\Delta x^{(k)} = 0, & x^{(k)} + \Delta x^{(k)} \in \mathcal{K} \\ \sqrt{B^{(k)}}\Delta x^{(k)} = \bar{z} & \frac{1}{2}\|\bar{z}\|_2^2 \leq z_0 \end{array} \right\}$$

set the next iterate $x^{(k+1)} = x^{(k)} + \Delta x$, $y^{(k+1)} = y$ and $s^{(k+1)} = s$
update $B^{(k+1)} \approx \mathcal{H}(x^{(k+1)}, y^{(k+1)}, s^{(k+1)})$, $B^{(k+1)} \in \mathcal{S}_+^n$
set $k = k + 1$
end while

4 Linear Convergence for the SLCP Approach

This chapter contains results from a joint work with F. Jarre and M. Diehl. The results have been published in [DJV06]. The SQP as well as the SSP/SLCP method presented in section 3.2 solve difficult nonlinear programs by solving a series of simpler subproblems. These subproblems are quadratic programs with linear constraints and in case of the SSP/SLCP additionally with some conical constraints.

If the quadratic term is positive semidefinite then the subproblem is efficiently solvable. Recall that for such a quadratic term B one could define a matrix $\sqrt{B}$ with $\sqrt{B}^2 = B$. This allows a reformulation of the problem (3.12) as a linear program with conic constraints

$$\min_{\Delta x, z_0, \bar{z}} \left\{ \mathrm{D}c(x)\Delta x + z_0 \;\middle|\; \begin{array}{r} F(x) + \mathrm{D}F(x)\Delta x = 0, \\ \sqrt{B}\Delta x - \bar{z} = 0, \\ x + \Delta x \in \mathcal{K} \\ \begin{pmatrix} \frac{1}{2}(1+z_0) \\ \frac{1}{2}(1-z_0) \\ \bar{z} \end{pmatrix} \in \mathcal{Q} \end{array} \right\}, \tag{4.1}$$

hence the name sequential linear conical program (SLCP).

In order to see that those programs are equivalent consider

$$\frac{1}{2}(1+z_0)^2 \geq \frac{1}{2}(1-z_0)^2 + \|\bar{z}\|_2^2 \Leftrightarrow z_0 \geq \frac{1}{2}\|\bar{z}\|_2^2 \tag{4.2}$$

and

$$\Delta x^{\mathrm{T}} B \Delta x = (\sqrt{B}\Delta x)^{\mathrm{T}}(\sqrt{B}\Delta x) = \bar{z}^{\mathrm{T}}\bar{z} = \|\bar{z}\|_2^2. \tag{4.3}$$

Thus minimizing z_0 leads to

$$z_0 = \frac{1}{2}\Delta x^{\mathrm{T}} B \Delta x. \tag{4.4}$$

While having a positive semidefinite approximation is necessary for efficient solving, it can destroy superlinear convergence. For necessary and sufficient second order optimality conditions for linear semidefinite progams we refer to [BS00]. For nonlinear programs over $\mathbb{R}^n_+$ the second order optimality condition uses the Hessian of the Lagrangian. For non polyhedral conic programs an additional term is necessary that represents the curvature of the cones boundary. This is because the Lagrangian represents the curvature of the nonlinear constraint only, but not the conic constraint's curvature. In consequence, the Hessian of the Lagrangian (e.g. for semidefinite programs) is not necessarily positive semidefinite on the active set. Thus for some problems the positive semidefinite approximations $B^{(k)}$ can't satisfy the following necessary condition for superlinear convergence

$$\lim_{k\to\infty} \frac{\|(B^{(k)} - \mathcal{H}(x^{(k)}, y^{(k)}, s^{(k)}))\Delta x^{(k)}\|}{\Delta x^{(k)}} = 0. \tag{4.5}$$

Using the following example we will prove that the SLCP method cannot generally converge faster than linearly when a bounded positive semidefinite approximation B is used.

4.1 An NLP and its conic reformulation

In this section we will present a nonlinear program (NLP) and its conic reformulation. This nonlinear program satisfies the strong second order conditions for local optimality. Thus a damped BFGS-SQP approach would converge superlinearly.

Consider the NLP

$$\min \left\{ -x_1^2 - (x_2 - 1)^2 \;\middle|\; \|\hat{x}\|_2^2 \leq 1, \quad \hat{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2 \right\}. \tag{4.6}$$

Let $\hat{\mathcal{L}}$ be the Lagrangian of problem (4.6)

$$\hat{\mathcal{L}}(x_1, x_2, y) := -x_1^2 - (x_2 - 1)^2 + y(x_1^2 + x_2^2) - 1. \tag{4.7}$$

Problem (4.6) has the optimal solution $(0, -1)^{\mathrm{T}}$ with corresponding multiplier $y = 2$. This solution satisfies the strong second order sufficiency conditions

$$\forall \xi \in S, \quad \xi^{\mathrm{T}} \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \xi > 0, \quad \text{with } S := \{\ \xi \in \mathbb{R}^2 \mid (0, -2)\xi = 0,\ \xi \neq 0\ \}. \tag{4.8}$$

Let $\mathcal{K}$ be the second order cone in three dimensions, i.e.

$$\mathcal{K} := \left\{ (x_0, x_1, x_2)^{\mathrm{T}} \in \mathbb{R}^3 \mid x_0 \geq \sqrt{x_1^2 + x_2^2} \right\}.$$

We extend the vector

$$\hat{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2 \qquad \text{to} \qquad x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^3.$$

With this definitions, problem (4.6) allows an equivalent conic reformulation

$$\min\{c(x) \mid F(x) = 0, \quad x \in \mathcal{K}\} \tag{4.9}$$

where $c : \mathbb{R}^3 \to \mathbb{R}$ is defined by

$$x \mapsto c(x) = -x_1^2 - (x_2 - 1)^2$$

and $F : \mathbb{R}^3 \to \mathbb{R}$ is defined by

$$x \mapsto F(x) = x_0 - 1.$$

The Lagrangian $\mathcal{L}$ of (4.9) with Lagrangian multiplier $y \in \mathbb{R}$ and the dual variable $s \in \mathbb{R}^3$ is given by

$$\mathcal{L}(x, y, s) := c(x) - yF(x) - \langle s, x \rangle,$$

where s lies in the dual cone $\mathcal{K}^D$,

$$\mathcal{K}^D = \mathcal{K}.$$

The gradient g of $\mathcal{L}$ with respect to x is given by

$$\begin{aligned} \mathrm{g}(x,y,s) &:= \nabla_x \mathcal{L}(x,y,s) = \nabla c(x) - \begin{pmatrix} y \\ 0 \\ 0 \end{pmatrix} - s \\ &= \begin{pmatrix} -y & -s_0 \\ -2x_1 & -s_1 \\ -2(x_2-1) & -s_2 \end{pmatrix} \end{aligned}$$

and the Hessian with respect to x is given by

$$\begin{aligned} \mathcal{H}(x,y) &:= \mathrm{D}_x^2 \mathcal{L}(x,y,s) = D_x(\nabla c(x)) \\ &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{pmatrix}. \end{aligned}$$

The global minimizer is

$$x^* = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix},$$

with multipliers

$$s^* = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ 4 \end{pmatrix} \quad \text{and} \quad y^* = -4$$

satisfying $\mathrm{g}(x^*, y^*, s^*) = 0$. Observe that the Hessian $\mathcal{H}(x,y)$ is negative semidefinite and independent of x, y.

4.2 Linear convergence with the projected Hessian

In the following we analyze the local convergence properties of a basic SLCP algorithm. To simplify the notation we define

$$c^{(k)} := \nabla c(x^{(k)}).$$

The algorithm approximates the solution by the iterates $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$, where $\Delta x^{(k)}$ solves the approximation

$$\min \left\{ c^{(k)\mathrm{T}} \Delta x + \frac{1}{2} \Delta x^{\mathrm{T}} B^{(k)} \Delta x \;\middle|\; \mathrm{D}F(x)(x^{(k)})[\Delta x] = -F(x)(x^{(k)}) \quad x^{(k)} + \Delta x \in \mathcal{K} \right\} \tag{4.10}$$

of the conic problem (4.9). Here, $B^{(k)}$ is an approximation of the Hessian $\mathcal{H}(x^{(k)}, y^{(k)})$ of $\mathcal{L}$. Because $\mathcal{H}$ is constant and no other part of the above conic problem depends on $y^{(k)}$, we need not regard multiplier iterates here.

Note that the linear equality constraint in (4.10) implies $x_0^{(k+1)} = x_0^{(k)} + \Delta x_0^{(k)} = 1$. Thus we can assume that $x_0^{(k)}$ is fixed to 1 for all $k > 0$ and simplify (4.10) by replacing the cone $\mathcal{K}$ with the inequality constraint:

$$\min \left\{ \begin{pmatrix} -2x_1 \\ -2(x_2-1) \end{pmatrix} \Delta x + \frac{1}{2} \Delta x^{\mathrm{T}} B \Delta x \;\middle|\; \|x + \Delta x\|_2 \leq 1 \right\}.$$

For simplicity of notation, we omit the iteration index k. We denote the projections on the (x_1, x_2)-space of the exact Hessian $\mathcal{H}$ and its approximation B by $\mathcal{H}$ and B.

If the exact Hessian $B = \mathcal{H}$ is used we obtain the nonconvex problem

$$\min \left\{ \begin{pmatrix} -2x_1 \\ -2(x_2 - 1) \end{pmatrix}^{\mathrm{T}} \Delta x + \frac{1}{2} \Delta x^{\mathrm{T}} \begin{pmatrix} -2 & 0 \\ 0 & -2 \end{pmatrix} \Delta x \;\Big|\; \|x + \Delta x\|_2 \leq 1 \right\}. \tag{4.11}$$

This subproblem is equivalent to the initial nonlinear program (4.6) and has thus the same optimal solution as the initial problem.

The idea of SQP-type algorithms is that they approximate the solution of a hard to solve problem with a sequence of easier to solve subproblems. However, nonconvex quadratic conic problems are about as difficult to solve as general nonlinear conic problems. Given efficient software packages like SeDuMi [SeWWW] or SDPT3 [TTT03], that solve convex conic programs, we search for a suitable approximation B of $\mathcal{H}$. This approximation has to be positive semidefinite to reformulate the subproblems as a linear conic problems and it should yield rapid convergence.

The Hessian of the Lagrangian in (4.11) is $\mathcal{H} = -2I$. The orthogonal projection of $\mathcal{H}$ onto the cone of positive semidefinite matrices is simply given by $B = 0$.

We first consider the choice $B = 0$, for which the optimal solution is always on the boundary of the cone. We start close to the optimal solution at the point

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{pmatrix}$$

where

$$0 < \alpha \ll 1. \tag{4.12}$$

Without loss of generality we will keep this choice of α also in sections 4.3 and 4.4. The case $-1 \ll \alpha < 0$ can be treated analogously.

For $B = 0$ and α as in (4.12) the conic SLCP subproblem is equivalent to

$$\min \left\{ \begin{pmatrix} -2\sin(\alpha) \\ 2\,(1 + \cos(\alpha)) \end{pmatrix}^{\mathrm{T}} (x + \Delta x) \Big| \; \|x + \Delta x\|_2 \leq 1 \right\}. \tag{4.13}$$

The solution of (4.13) is given by

$$x + \Delta x = \begin{pmatrix} \sin\left(\frac{\alpha}{2}\right) \\ -\cos\left(\frac{\alpha}{2}\right) \end{pmatrix}.$$

Hence, the (local) convergence for $B = 0$ is linear, with a convergence rate of $\frac{1}{2}$. As indicated in the next section this result does not imply linear convergence for all choices of positive semidefinite B.

4.3 Superlinear convergence for unbounded $B \in \mathcal{S}^n_+$

It is well known (see e.g. [Ja03]) that the orthogonal projection of the Hessian as used in Section 4.2 is not affine invariant. Other semidefinite approximations of $\mathcal{H}$, for example the Hessian of the augmented Lagrangian, may be better suitable to obtain rapid local convergence of a SQP-type method.

In fact, as we will show in this section, we can present a sequence of positive semidefinite matrices $B^{(k)}$ for which the iterates $x^{(k)}$ generated by solution of the SLCP subproblem

$$\min \left\{ c^{\mathrm{T}} \Delta x + \frac{1}{2} \Delta x^{\mathrm{T}} B \Delta x \;\middle|\; \|x + \Delta x\|_2 \leq 1 \right\} \tag{4.14}$$

converge quadratically to $(0, -1)^{\mathrm{T}}$. The SLCP subproblems based on the matrices $B^{(k)}$ presented here have unique solutions on the boundary of the constraint set of (4.14). Thus we use again

$$x(\alpha) = \begin{pmatrix} x_1(\alpha) \\ x_2(\alpha) \end{pmatrix} = \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{pmatrix} \tag{4.15}$$

and prove the quadratic convergence with respect to α as defined in (4.12) .

For $B = 0$ the result of the previous section states that the search step $\Delta x = v$ is approximately equal to

$$v := \begin{pmatrix} \sin\left(\frac{\alpha}{2}\right) \\ -\cos\left(\frac{\alpha}{2}\right) \end{pmatrix} - \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{pmatrix} \approx \frac{\alpha}{2} \begin{pmatrix} -1 \\ -\frac{3}{4}\alpha \end{pmatrix}. \tag{4.16}$$

The method is locally superlinearly convergent, if and only if this direction is perturbed to $\Delta x \approx v^*$ with

$$v^* := x^* - x(\alpha) = \begin{pmatrix} 0 \\ -1 \end{pmatrix} - \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{pmatrix} \approx \alpha \begin{pmatrix} -1 \\ -\frac{1}{2}\alpha \end{pmatrix}. \tag{4.17}$$

In Figure 4.1 the steps v and v^* are visualized.

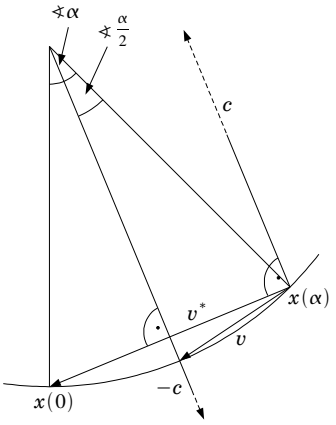

Figure 4.1: Visualization of the SLCP subproblem.

Note that the direction v^* leading to the optimal solution $x^* = x(0)$ is orthogonal to the vector c. Hence c is the direction we have to penalize, but it is also the gradient of the objective function of (4.14) at $\Delta x = 0$. In consequence, the SLCP subproblems (4.14) using

$$B := \frac{1}{\alpha^4} v v^{*\mathrm{T}} = \frac{1}{\alpha^4} c c^{\mathrm{T}} \tag{4.18}$$

do not necessarily produce superlinearly convergent steps. Let $\mathcal{N}$ be the null space of B. For B as in (4.18) the space of optimal solutions of (4.14) is the intersection of the feasible set and

$$V := \mathcal{N} - \frac{\alpha^4 c}{16} + \mathcal{O}(\alpha^6). \tag{4.19}$$

The affine space $x(\alpha)+V$ includes a point on the boundary of the constraint set of (4.14) of the form $x(\alpha)+\big(v^*+\mathcal{O}(\alpha^3)\big)$. In the following we denote with v^* as a point "close" to the optimum.

In order to obtain an unique optimal solution for each SLCP subproblem we use a rotation with a small angle $\beta>0$ of the vector c to form a matrix B_β. We define

$$\mathrm{rot}_\beta := \begin{pmatrix}\cos(\beta) & -\sin(\beta)\\ \sin(\beta) & \cos(\beta)\end{pmatrix}, \quad c_\beta := \mathrm{rot}_\beta c, \quad \text{and} \quad B_\beta := \frac{1}{\alpha^4} c_\beta c_\beta^{\mathrm{T}}.$$

Note that for $\beta\in(0,\frac{\alpha}{4})$ the objective value of (4.14) can be improved in a direction orthogonal to the penalty direction c_β. This implies that the optimal solution of (4.14) is a unique point on the boundary of the constraint set of (4.14). For $\beta=0$ we have the case of (4.18) while for $\beta=\frac{\alpha}{4}$ we obtain the same SLCP iterates as for $B=0$.

In the following we assume $\beta\in(0,\frac{\alpha}{4})$ and consider the problem

$$\min\left\{c^{\mathrm{T}}\Delta x+\frac{1}{2}\Delta x^{\mathrm{T}}B_\beta\Delta x \;\middle|\; \|x+\Delta x\|_2\le 1\right\}. \tag{4.20}$$

Recall that v^* is in the null space $\mathcal{N}$ of B and $x(\alpha)+v^*$ lies on the boundary of the constraint set of (4.14).

Let $\mathcal{N}_\beta$ be the null space of B_β and let v_β^* be the unique solution of (4.20). Note that the angle between $\mathcal{N}_\beta$ and the objective function c is less than the angle of the null space $\mathcal{N}=\mathcal{N}_{\beta=0}$ of B and c. Therefore, as in (4.19), the vector v_β^* lies on the boundary of the constraint set of (4.20) and is $\mathcal{O}(\alpha^4)$-"close" to points of the null space $\mathcal{N}_\beta$ of B_β.

The null space $\mathcal{N}_\beta$ intersects the boundary of the constraint set of (4.20) twice. Let $\tilde{v}_\beta^*$ be the intersection that is close to v^*. Then $\tilde{v}_\beta^*$ satisfies

$$\tilde{v}_\beta^* = v_\beta^* + \mathcal{O}(\alpha^3)$$

and is given by

$$\tilde{v}_\beta^* = \begin{pmatrix}\sin(2\beta)-\sin(\alpha)\\ \cos(2\beta)+\cos(\alpha)\end{pmatrix} = x(2\beta)-x(\alpha).$$

Thus, for a sequence $\beta^{(k)}$ the points

$$x(\alpha^{(k)+1}) = x(2\beta^{(k)}) + \mathcal{O}({\alpha^{(k)}}^3)$$

converge superlinearly to $x(0)$, if and only if the angles $\beta^{(k)}$ converge superlinearly to zero, too.

Summarizing, the method is quadratically convergent if we choose $\beta^{(k)}=\frac{{\alpha^{(k)}}^2}{2}$ and accordingly $c_{\beta^{(k)}}$ as well as $B^{(k)}:=\frac{1}{{\alpha^{(k)}}^4}c_{\beta^{(k)}}c_{\beta^{(k)}}^{\mathrm{T}}$ with $\alpha^{(k)}:=\arcsin(x_1^k)$.

The main advantage of the augmented Lagrangian over other penalty functions is the fact that under standard assumptions a finite value for the barrier parameter is sufficient to guarantee exactness. In the above analysis, however, the matrices B_k are unbounded. In the next section we show that we cannot obtain superlinear convergence when $B^{(k)}$ is bounded.

4.4 Linear convergence for any choice of bounded $B\in\mathcal{S}_+^n$

In the following theorem we will show that we cannot gain more than linear convergence when we bound our series of positive semidefinite approximations.

Theorem 4.4.1. *Suppose we solve problem* (4.6) *with the SLCP method where the Hessian approximations are given by* any *globally bounded sequence of positive semidefinite matrices. Then it is not possible to have a faster than linear convergence.*

Proof. We assume from now on that the positive semidefinite matrix $\breve{B} = \breve{B}^{(k)}$ for the SLCP subproblem (4.10) is bounded independently of k, $\|\breve{B}\|_2 \leq M$, and prove by contradiction that for any such $\breve{B}$ the rate of convergence can not be better than $\alpha^{(k+1)} = \frac{\alpha^{(k)}}{2} + \mathcal{O}({\alpha^{(k)}}^2)$.

We denote the solution of the SLCP subproblem (4.10) by $\breve{v}$ and use v^* as defined in (4.17) for the vector to the global optimum of (4.6).

If $x + \breve{v}$ is not on the unit circle, i.e., the boundary of the feasible set, it follows that $\breve{B}\breve{v} = -c$. Since $\breve{v} \to 0$ and $\|c\|_2 \to 4$ this implies $\|\breve{B}\|_2 \to \infty$ in contradiction to our assumption $\|\breve{B}\|_2 \leq M$. Thus we can assume without loss of generality that $x + \breve{v}$ is on the unit circle and use again the notation

$$x(\alpha) = \begin{pmatrix} \sin(\alpha) \\ -\cos(\alpha) \end{pmatrix}$$

with α as in (4.12).

Let $x(\alpha)$ be the k-th iterate and let $x(\gamma)$ be the $k+1$-st iterate, thus we have $\breve{v} = x(\gamma) - x(\alpha)$. We assume for contradiction that

$$0 \leq \gamma \leq \frac{\alpha}{2} - \varepsilon\alpha + \mathcal{O}(\alpha^2)$$

with $0 < \varepsilon \leq \frac{1}{2}$ independent of α (The SLCP -method is quadratically convergent if and only if $\varepsilon = \frac{1}{2}$). As in (4.16) the optimal solution of the linear SLCP subproblem (4.10) using $B = 0$ is denoted by v.

As $\breve{v}$ is the optimal solution of the SLCP subproblem (4.10) using $B = \breve{B}$, the objective value of v is greater than the objective value of $\breve{v}$:

$$\begin{array}{rccc} & c^{\mathrm{T}}v + \frac{1}{2}v^{\mathrm{T}}\breve{B}v & \geq & c^{\mathrm{T}}\breve{v} + \frac{1}{2}\breve{v}^{\mathrm{T}}\breve{B}\breve{v} \\ \Leftrightarrow & c^{\mathrm{T}}(v - \breve{v}) & \geq & \frac{1}{2}(\breve{v} - v)^{\mathrm{T}}\breve{B}(\breve{v} - v) + v^{\mathrm{T}}\breve{B}(\breve{v} - v) \end{array} . \tag{4.21}$$

In the following we will evaluate the terms of (4.21) up to $\mathcal{O}(\alpha^4)$ to show that (4.21) cannot be true.

First we analyze the linear term on the left hand side of (4.21) and obtain from Figure 4.2

$$\begin{aligned} c^{\mathrm{T}}(v - \breve{v}) &= -\|c\|_2\|v - \breve{v}\|_2 \cos\left(\frac{\pi}{2} - \frac{\alpha}{4} + \frac{\gamma}{2}\right) \\ &= -\|c\|_2\|v - \breve{v}\|_2 \sin\left(\frac{\alpha}{4} - \frac{\gamma}{2}\right) \\ &= -\|c\|_2 2 \sin\left(\frac{\alpha}{4} - \frac{\gamma}{2}\right) \sin\left(\frac{\alpha}{4} - \frac{\gamma}{2}\right) \\ &= -8 \sin^2\left(\frac{\alpha}{4} - \frac{\gamma}{2}\right) + \mathcal{O}(\alpha^4) \\ &\leq -2\varepsilon^2\alpha^2 + \mathcal{O}(\alpha^4). \end{aligned}$$

For the evaluation of the right hand side of (4.21) note that

$$\frac{1}{2}(\breve{v} - v)\breve{B}(\breve{v} - v) \geq 0$$

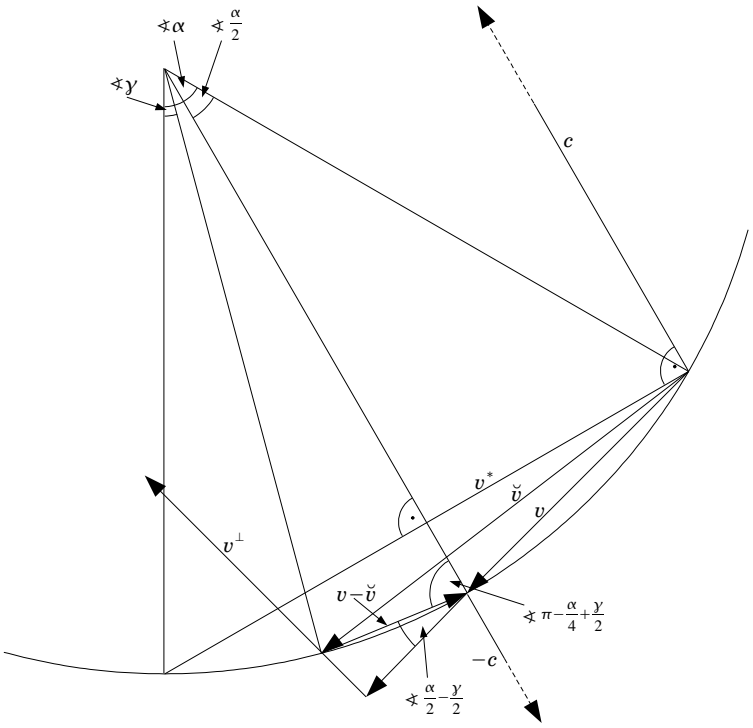

Figure 4.2: Angles and vectors used in the proof.

since $\breve{B}$ is positive semidefinite.

Denote by $v^\perp$ the orthogonal complement of v with norm $\|v\|_2 = \|v^\perp\|_2$ that is obtained by a clockwise rotation of v by 90°. To evaluate the last term of the right hand side of (4.21) note again by Figure 4.2 that

$$\breve{v} - v = \frac{\|\breve{v} - v\|_2}{\|v\|_2} v \cos\left(\frac{\alpha}{2} - \frac{\gamma}{2}\right) + \frac{\|\breve{v} - v\|_2}{\|v\|_2} v^\perp \sin\left(\frac{\alpha}{2} - \frac{\gamma}{2}\right).$$

Thus we have

$$\begin{array}{rcccl} & v^{\mathrm{T}}\breve{B}(\breve{v} - v) & & & \\ = & \frac{\|\breve{v}-v\|_2}{\|v\|_2} \cos\left(\frac{\alpha}{2} - \frac{\gamma}{2}\right) v^{\mathrm{T}}\breve{B}v & + & \frac{\|\breve{v}-v\|_2}{\|v\|_2} \sin\left(\frac{\alpha}{2} - \frac{\gamma}{2}\right) v^{\mathrm{T}}\breve{B}v^\perp & \\ \geq & 0 & - & \sin\left(\frac{\alpha}{2} - \frac{\gamma}{2}\right) \|\breve{v} - v\|_2 \|\breve{B}\|_2 \|v^\perp\|_2 & \\ \geq & -\sin\left(\frac{\alpha}{2}\right) 2\sin\left(\frac{\alpha}{4}\right) M 2 \sin\left(\frac{\alpha}{4}\right) & & & \\ = & -\frac{M}{8}\alpha^3 + \mathcal{O}(\alpha^5). & & & \end{array}$$

From (4.21) it would therefore follow that

$$\begin{array}{rc} & -2\varepsilon^2\alpha^2 + \mathcal{O}(\alpha^4) \geq c^{\mathrm{T}}(v - \breve{v}) \\ \geq & \frac{1}{2}(v - \breve{v})\breve{B}(v - \breve{v}) + v^{\mathrm{T}}\breve{B}\breve{v} \geq -\frac{M}{8}\alpha^3 + \mathcal{O}(\alpha^5) \end{array}$$

i.e, $-2\varepsilon^2\alpha^2 + \mathcal{O}(\alpha^3) \geq 0$, which is not true for α sufficiently small and fixed $\varepsilon > 0$. Therefore, the assumption of faster than linear convergence is led to a contradiction. □

4.5 Conclusion

This result shows that the SSP/SLCP approach is not suitable for local convergence. In practical applications however it can be seen that it is very useful for fast global convergence. We will present details about our implementation of the SLCP method in the next chapter. To overcome the local convergence problem we will discuss how to build a hybrid solver in section 5.9. This solver switches between a solver for global and one for local convergence. The difficult part is to guess when to switch to the local solver.

In chapter 7 we will present an algorithm that does have quadratic local convergence. As solver for fast global convergence we suggest the SSP/SLCP solver.

5 Implementation

We have implemented an SLCP solver. In numerical experiments the implementation displays fast global convergence, even though we've proven unattractive local convergence properties for the SSP algorithm in the last chapter.

Our focus while implementing was on practical results for given problems. We investigated into optimizing the different elements of our SLCP solver. This chapter sums up the results of our experiments, including the choices of search steps and a discussion on step length controls. In this chapter we also present a new approach we call "augmented filter".

In the next section we present an example with an important application. Then we show the algorithm and its extensions. We then continue with a special kind of step control, the filters, and our extension to it. Finally, we give a perspective on an hybrid algorithm that switches to a different solver for a faster local convergence. Such a solver is presented in the following chapters.

A problem that is rarely discussed in mathematical papers is the implementation of nonlinear problem solvers. Existing solver are often either hard to read, have a complicated input format or are restricted to a small problem class. We conclude this chapter by giving a few details about our solver. We describe how problems as mentioned above can be avoided by using Matlab's object oriented programming technique.

5.1 A practical example

In circuit board simulation a data model is generated by using a Lanczos process (see e.g. [BF00],[BF01], [Fr03]). Due to the truncated precision of such a process the generated reduced problem typically is not conform with the law of conservation of energy. In [FJ04] system (5.1) has been introduced. This system can be used to verify whether the law of conservation of energy is respected.

As part of this thesis, a SSP solver has been used to create a perturbation for a reduced order model approximation, such that the perturbed data set respects the law of conservation of energy.

The data model is given by the matrices $B_1, B_2 \in \mathbb{R}^{n\times m}$ and $G, C \in \mathbb{R}^{n\times n}$. In real applications these matrices are low dimensional approximations of a higher dimensional model. The law of conservation of energy is given if a matrix X exists such that

$$\begin{aligned} X^{\mathrm{T}}B_1 &= B_2, \\ G^{\mathrm{T}}X + X^{\mathrm{T}}G &\succeq 0, \\ X^{\mathrm{T}}C &\succeq 0, \\ X^{\mathrm{T}}C &= C^{\mathrm{T}}X. \end{aligned} \tag{5.1}$$

This can be easily verified with the linear SDP

$$\min \left\{ -\lambda I \;\middle|\; \begin{array}{rl} X^{\mathrm{T}} B_1 = B_2, & \\ X^{\mathrm{T}} G + G^{\mathrm{T}} X \succeq \lambda I, & X \in \mathbb{R}^{n \times n} \\ X^{\mathrm{T}} C + C^{\mathrm{T}} X \succeq \lambda I, & \\ X^{\mathrm{T}} C - C^{\mathrm{T}} X = 0, & \end{array} \right\}. \tag{5.2}$$

If the solution is $\lambda \geq 0$ then we know that the data respects the law of conservation of energy.

The problem with formulation (5.2) is that for certain combinations of B_1, B_2 (5.2) has a nearly degenerate feasible set depending on C, G or even no feasible point at all. Nearly dengenerated means that the solver cannot distinguish it from a problem with no feasible point for numerical reasons. A nearly degenerate feasible set results in a slow convergence and a lack of accuracy. For our nonlinear formulation we need so solve similar problems, where the lack of a solution for the linear problem forced the algorithm to stop even though there is a solution for the nonlinear problem.

To avoid these problems we used another formulation of (5.2), namely

$$\min \left\{ \|S\|_2^2 \;\middle|\; \begin{array}{rl} X^{\mathrm{T}} B_1 + S = B_2, & \\ X^{\mathrm{T}} G + G^{\mathrm{T}} X \succeq 0, & X \in \mathbb{R}^{n \times n} \\ X^{\mathrm{T}} C + C^{\mathrm{T}} X \succeq 0, & \\ X^{\mathrm{T}} C - C^{\mathrm{T}} X = 0, & \end{array} \right\}. \tag{5.3}$$

Equivalent to (5.2) where $\lambda = 0$ indicates whether the law of conservation of energy is respected or not $S = 0$ indicates this here. An advantage of this formulation is, that we always have at least the feasible point $S = B_2, X = 0$. In practical applications this approach seems to be faster and more accurate to solve. The disadvantage of this approach is, that it is desirable to find a $\lambda \geq \varepsilon > 0$ for (5.2). It is possible to add such a restriction for $\varepsilon > 0$ to (5.3) by replacing the second and third condition with

$$X^{\mathrm{T}} G + G^{\mathrm{T}} X \succeq \varepsilon I \tag{5.4}$$

$$X^{\mathrm{T}} C + C^{\mathrm{T}} X \succeq \varepsilon I, \tag{5.5}$$

but this approach requires a a priori election of the unknown quantity ε.

The problem with this modification is that we again might not have a feasible point or have a nearly degenerate feasible set. In our approach we solve the problem for $\varepsilon = 0$ up to a high precision and try to increase ε until either ε is reasonably "large" or no solution exists.

At this point we can verify whether the reduced order model approximation respects the law of conservation of energy or not. We want to find an approximation that does. This approximation should be "close" to the original one. We therefore introduce perturbations P_G, P_C and try to find a pair $\hat{G} = G + P_G, \hat{C} = C + P_C$ for that the solution of (5.3) is $S = 0$.

Introducing these perturbations we get the following nonlinear semidefinite program

$$\min \left\{ s_0 \;\middle|\; \begin{array}{rl} X^{\mathrm{T}} B_1 + S - B_2 = 0, & X, P_G, P_C \in \mathbb{R}^{n \times n}, \\ X^{\mathrm{T}}(G + P_G) + (G + P_G)^{\mathrm{T}} X - Z_G = 0, & Z_G, Z_C \preceq 0, \\ X^{\mathrm{T}}(C + P_C) + (C + P_C)^{\mathrm{T}} X - Z_C = 0, & \begin{pmatrix} s_0 \\ S \end{pmatrix} \in \mathcal{Q} \\ X^{\mathrm{T}}(C + P_C) - (C + P_C)^{\mathrm{T}} X = 0, & \end{array} \right\}. \tag{5.6}$$

To solve general problems fast we implemented the SLCP algorithm that supports the same set of cones as SeDuMi. We will list these cones in the next section. The algorithm is kept very flexible. Intense numerical test led to an efficient steplength control and search step formulations. We present specific speedups for the reduced order model problem in section 5.8.

5.2 The SLCP Algorithm

In this section we present the SLCP algorithm. The elements of this algorithm will be discussed throughout the next section. Chapter 3 presents the SLCP algorithm regarding theoretical aspects. Our focus in these sections is on the implementations side of the SLCP algorithm.

We use SeDuMi (see [St99] or [SeWWW] for details) as solver for the subproblems. Besides being known as a robust solver for conical linear programs, SeDuMi has other properties that are very useful to us. First of all it has a straightforward standard form

$$\min \left\{ \; c^{\mathrm{T}} x \; | Ax = b, \; x \in \mathcal{K} \; \right\}. \tag{5.7}$$

Second the cone $\mathcal{K}$ includes the most common cones. More precisely any Cartesian product of the following cones

$$\begin{aligned}
\mathbb{R}^{\mathcal{K}_f} &\quad \text{free variables,} \\
\mathbb{R}_+^{\mathcal{K}_l} &\quad \text{positive variables,} \\
\mathcal{Q}^{\mathcal{K}_{q_1}} \times \cdots \times \mathcal{Q}^{\mathcal{K}_{q_i}} &\quad \text{quadratic cone variables,} \\
{\mathcal{Q}_{\mathcal{R}}}^{\mathcal{K}_{r_1}} \times \cdots \times {\mathcal{Q}_{\mathcal{R}}}^{\mathcal{K}_{r_j}} &\quad \text{rotated quadratic cone variables,} \\
\mathcal{S}_+^{\mathcal{K}_{s_1}} \times \cdots \times \mathcal{S}_+^{\mathcal{K}_{s_k}} &\quad \text{positive semidefinite cone variables.}
\end{aligned} \tag{5.8}$$

The quadratic cone is the cone of vectors

$$\mathcal{Q}^{n+1} \ni \begin{pmatrix} x_0 \\ \bar{x} \end{pmatrix} \quad \begin{matrix} x_0 \in \mathbb{R}_+ \\ \bar{x} \in \mathbb{R}^n \end{matrix} \quad \text{with} \quad x_0 \geq \|\bar{x}\|_2. \tag{5.9}$$

The rotated quadratic cone is the cone of vectors

$${\mathcal{Q}_{\mathcal{R}}}^{n+2} \ni \begin{pmatrix} x_0 \\ x_1 \\ \bar{x} \end{pmatrix} \quad \begin{matrix} x_0, x_1 \in \mathbb{R} \\ \bar{x} \in \mathbb{R}^n \end{matrix} \quad \text{with} \quad x_0 x_1 \geq \frac{1}{2}\|\bar{x}\|_2^2, \quad x_0 + x_1 \geq 0. \tag{5.10}$$

It is called rotated quadratic cone as a vector $\begin{pmatrix} x_0 \\ x_1 \\ \bar{x} \end{pmatrix} \in {\mathcal{Q}_{\mathcal{R}}}^{n+2}$ can be easily transformed to be a vector of the quadratic cone

$$\begin{aligned}
& x_0 x_1 \geq \frac{1}{2}\|\bar{x}\|_2^2, \quad x_0 + x_1 \geq 0 \\
\Leftrightarrow \quad & \frac{1}{2}(x_0 + x_1)^2 \geq \left\| \begin{pmatrix} \frac{1}{\sqrt{2}}(x_0 - x_1) \\ \bar{x} \end{pmatrix} \right\|_2^2 \quad \frac{1}{\sqrt{2}}(x_0 + x_1) \geq 0 \\
\Leftrightarrow \quad & \begin{pmatrix} \frac{1}{\sqrt{2}}(x_0 + x_1) \\ \frac{1}{\sqrt{2}}(x_0 - x_1) \\ \bar{x} \end{pmatrix} \in \mathcal{Q}^{n+2}.
\end{aligned} \tag{5.11}$$

We used the rotated quadratic cone to convert a quadratic program into a linear conic program, when we introduced the SLCP (sequential *linear conic* program) algorithm.

Instead of writing a sequential semidefinite programming (SSP) algorithm, we used the supported cones of SeDuMi to implement the slightly more general SLCP algorithm. Of course all the supported cones lie in $\mathcal{S}_+^n$, but supporting these cones directly allows much smaller program sizes.

As a result of using SeDuMi our algorithm solves problems of the form

$$\min \left\{ \; c^{\mathrm{T}} x \mid F(x) = 0, \; x \in \mathcal{K} \; \right\} \tag{5.12}$$

where $\mathcal{K}$ is the cartesian product of any SeDuMi cones from (5.8).

Algorithm 3 The SLCP-Implementation-Algorithm

let $(x^{(0)}, y^{(0)}, s^{(0)})$ be a given starting point and $k = 0$
let $B^{(0)} \approx \mathcal{H}(x^{(0)}, y^{(0)}, s^{(0)})$ be an approximation of the Hessian of the Lagrangian
while *[stopping criterion]* (see section 5.7) **do**
  [create conical linear program] (see section 5.4)
  retrieve search step $(\Delta x, y, s)$, by solving the conical program with SeDuMi
  generate next iterate $(x^{(k+1)}, y^{(k+1)}, s^{(k+1)})$ via [line search/filter] (see section 5.5)
  [post corrections]
  [B update] $B^{(k+1)} \approx H(x^{(k+1)}, y^{(k+1)}, s^{(k+1)})$ (see section 5.3)
  set $k = k + 1$
end while

Here we present an algorithm that focuses on the implementation, while the general concept of the SLCP algorithm was introduced in section (3.2). Algorithm 3 includes typical elements such as step length control and a problem specific intervention point called *post correction*.

The post correction lets us use additional knowledge about a specific problem to speedup the convergence. We used this post correction e.g. for the reduced order model problem. As the nonlinear reduced order model problem evolves from a linear problem we solved such a much smaller linear problem to generate a feasible point from the current non feasible iterate. This modification led to a significant speedup in our numerical experiments. For some examples that converged very slow this speedup leads to convergence within a few steps. We will go into more detail about the reduced order model problem in section 5.8.

In the following specific elements of the algorithm are described in more detail.

5.3 Approximation of $\mathcal{H}$

To obtain a real square root of $B^{(k)}$ we need $B^{(k)}$ to be positive semidefinite.

The Hessian of the augmented Lagrangian

$$\begin{aligned} \Lambda_r(x, y, s) := c^{\mathrm{T}} x + \frac{r}{2} \sum_{i=1}^{m} (F_i(x) + \frac{y_i}{r})^2 \qquad & -\frac{1}{2r} y^{\mathrm{T}} y \\ + \frac{r}{2} [\frac{s}{r} - x]_+ \bullet [\frac{s}{r} - x]_+ \qquad & -\frac{s^{\mathrm{T}} s}{2r}. \end{aligned} \tag{5.13}$$

is one possible choice for such an approximation $B^{(k)}$. The operator $[\,\cdot\,]_+$ is the orthogonal projection onto the cone $\mathcal{K}$.

A weakness of this approximation is while Λ_r is convexified with an increasing r the radius of quadratic convergence decreases. Additionally this parameter r has to be estimated.

The magnitude of such a projection is $\mathcal{O}(n^3)$. It is possible to save the additional time of calculating and projecting the Hessian of the augmented Lagrangian Λ_r and just project the Hessian of the Lagrangian $\mathcal{L}$.

Cheaper approaches are quasi Newton update strategies for the approximation of the Hessian that are typically $\mathcal{O}(n^2)$. A well known update is the BFGS update. While the BFGS update does not guarantee to keep the approximations $B^{(k)}$ positive semidefinite a simple variant called damped BFGS update does. The damped BFGS update still leads to super linear convergence of the SQP approach (see [Po78]). Another benefit of this update is that it can be applied directly to the square root of $B^{(k)}$.

Numerical experiments show that using a BFGS approach often slows down the convergence. Since the additional subproblems are more difficult to solve than a projection it is cheapter to use the Hessian of the Lagrangian an project it, if the Hessian is calculatable in a reasonable time.

If the Hessian is hard to calculate an efficient variant might be to alternate both in the following way. Let $QDQ^{\mathrm{T}} = \mathcal{H}^{(k)}$ be the eigenvalue decomposition of the Hessian of the Lagrangian and let d_i be the eigenvalues of $\mathcal{H}^{(k)}$. Instead of using a projection onto the cone of positive semidefinite matrices we make the matrix slightly positive. Let $\varepsilon > 0$ be a small constant. We then set

$$B^{(k)} := QD^{++}Q^{\mathrm{T}} \in \mathcal{S}_{++} \quad \text{with} \quad D_{ii}^{++} = \max(\varepsilon, d_i). \tag{5.14}$$

For the next few iterates we apply a quasi Newton update to the square root of $B^{(k)}$ such as the BFGS update. After a few iterations we start again with a positive semidefinite "projection" of the Hessian.

With this approach the convergence is fast and its calculation is cheap.

5.4 Search Steps

Again for abbreviation we will omit all iteration indices k. Recall that the search step is the solution Δx of the subproblem

$$\min_{\Delta x}\{\ c^{\mathrm{T}}\Delta x + \frac{1}{2}\Delta x B \Delta x \mid F(x) + \mathrm{D}F(x)\Delta x = 0,\ x + \Delta x \in \mathcal{K}\ \}. \tag{5.15}$$

Let n be the dimension of x and m be the dimension of $F(x)$. Thus (3.12/5.15) has n variables, namely Δx, and m linear constraints, $F(x) + \mathrm{D}F(x)\Delta x = 0$.

To find an optimum for problem (5.15) with a solver for conical linear programs, we use that a positive semidefinite approximation B of $\mathcal{H}$ has a square root $\sqrt{B}$. Using this squareroot $\sqrt{B}$ we formulate problem (5.16) that is equivalent to (3.12/5.15).

$$\min_{\Delta x, z_0, \bar{z}} \left\{ c^{\mathrm{T}}\Delta x + z_0 \;\middle|\; \begin{array}{ll} \mathrm{D}F(x)\Delta x = -F(x), & x + \Delta x \in \mathcal{K}, \\ \sqrt{B}\Delta x - \bar{z} = 0, & \begin{pmatrix} z_0 \\ 1 \\ \bar{z} \end{pmatrix} \in \mathcal{Q}_{\mathcal{R}} \end{array} \right\} \tag{5.16}$$

This increases the number of variables to $2n + 1$ and the linear constraints to $m + n$.

SeDuMi only allows variable conic constraints, thus (5.16) can not be solved with SeDuMi directly. We therefore introduce a new variable $x_+ = x + \Delta x$ and $z_1 = 1$ and get the conical linear program

$$\min_{\Delta x, x_+, z_0, z_1 \bar{z}} \left\{ c^{\mathrm{T}} \Delta x + z_0 \;\middle|\; \begin{array}{rl} \mathrm{D}F(x)\Delta x = -F(x), & x_+ \in \mathcal{K}, \\ \sqrt{B}\Delta x - \bar{z} = 0, & \Delta x \in \mathbb{R}^n, \\ x_+ - \Delta x = x, & \begin{pmatrix} z_0 \\ z_1 \\ \bar{z} \end{pmatrix} \in \mathcal{Q}_{\mathcal{R}} \\ z_1 = 1, & \end{array} \right\}. \tag{5.17}$$

The dimensions of (5.17) are $3n+2$ variables and $m+2n+1$ non-conic constraints. When x is close to the optimal solution of (3.11) then the correction Δx of (5.17) is small and avoids cancellation in errors. Thus this formulation yields high accuracy for Δx. If a high accuracy is unnecessary e.g. when we are far away from the solution, the following equivalent problem[1] can be solved faster as it has a reduced dimension, but with a lower accuracy as a trade off.

Instead of having x_+ and Δx we set $\Delta x = x_+ - x$ and define the problem

$$\min_{x_+, z_0, z_1 \bar{z}} \left\{ c^{\mathrm{T}} x_+ + z_0 \;\middle|\; \begin{array}{rl} \mathrm{D}F(x)x_+ = -F(x) + \mathrm{D}F(x)x, & x_+ \in \mathcal{K}, \\ \sqrt{B}x_+ - \bar{z} = \sqrt{B}x, & \begin{pmatrix} z_0 \\ z_1 \\ \bar{z} \end{pmatrix} \in \mathcal{Q}_{\mathcal{R}} \\ z_1 = 1, & \end{array} \right\}. \tag{5.18}$$

This formulation has reduced dimensions of $2n + 2$ variables and $m + n + 1$ non-conical constraints. To analyze the accuracy difference of (5.17) and (5.18) we focus on the linearized constraint

$$\begin{array}{ll} \overset{(5.17)}{} & \overset{(5.18)}{} \\ DF(x)\Delta x = -F(x) & DF(x)x_+ = -F(x) + DF(x)x. \end{array} \tag{5.19}$$

For problem (5.17), the accuracy of Δx is about the machine precision. For problem (5.18) where we calculate $\Delta x = x_+ - x$ on the other hand we have an accuracy that is the division of the machine accuracy multiplied by the magnitude of x_+ divided by the magnitude of Δx. The subproblem solver does not yield results with machine accuracy, but a few digits less. Thus it is crucial to switch to the solver with high accuracy when the iterate is close to x^*, if an accurate approximation of x^* is needed. In our numerical experiments the search step resulting from (5.17) showed a faster convergence very close to x^*.

The formulation (5.17) that yields high accuracy can easily be reformulated to use a trust region r for Δx without any noticeable loss of speed

$$\min_{\Delta x, x_+, z_0, z_1 \bar{z}} \left\{ c^{\mathrm{T}} \Delta x + z_0 \;\middle|\; \begin{array}{rl} \mathrm{D}F(x)\Delta x = -F(x), & x_+ \in \mathcal{K}, \\ \sqrt{B}\Delta x - \bar{z} = 0, & \begin{pmatrix} \Delta x_0 \\ \Delta x \end{pmatrix} \in \mathcal{Q}, \\ x_+ - \Delta x = x, & \\ z_1 = 1, & \begin{pmatrix} z_0 \\ z_1 \\ \bar{z} \end{pmatrix} \in \mathcal{Q}_{\mathcal{R}} \\ \Delta x_0 = r, & \end{array} \right\}. \tag{5.20}$$

This approach uses a trust region and has a high accuracy. The only disadvantage is an increase in dimension compared to (5.18) since problem (5.20) has $3n + 3$ variables and $m + 2n + 2$ linear constraints.

[1] It is equivalent concerning $\Delta x = x_+ - x$, but has a different target value

Redundant equality constraints can theoretically be removed using a QR decomposition. In practical applications it turns out that this often leads to a increased computation time for solving the subproblem as SeDuMi needs more IPM steps. Structure and order of constraints have a significant influence on the speed of the algorithm even for a robust solver like SeDuMi. For instance in (5.3) the term $X^{\mathrm{T}}C$ is forced to be symmetric. Yet using $X^{\mathrm{T}}C + C^{\mathrm{T}}X \succeq 0$ leads to a faster overall convergence than $X^{\mathrm{T}}C \succeq 0$.

For our examples we use mainly two search steps. The algorithm starts using (5.18) and switches to (5.20) for more accuracy. The decision for a switch between these search steps is made by the step length control. Typically this is done when the penalty line search or the filter does make only a minimal step. In this case the solver switches from the fast (5.18) to a high accuracy search step (5.17/5.20).

5.5 Step length control

For our examples the search step is a crucial part for fast convergence. On one hand small steps lead to a slow convergence, on the other hand too large steps lead to a strong violation of the restriction $F(x) = 0$ which again leads to a slow convergence. With the right step length it is possible to use steps just as long such that the constraint violation does not lead to slow convergence.

Line search

A first idea to determine an efficient step length is a penalty line search. A penalty line search does a line search over a merit function $p(x)$. Often $p(x)$ is a weighted sum of the objective function $c^{\mathrm{T}}\Delta x$ and a penalty term $\|F(x)\|$. In our implementation we used

$$p(x) := c^{\mathrm{T}}x + M\|F(x)\| \tag{5.21}$$

where $M \approx \zeta\|y\|$ for a constant ζ.

Using such a merit function $p(x)$ an efficient step length is an approximation of

$$\lambda^* := \mathrm{argmin}_\lambda\{\ p(x + \lambda\Delta x),\ \lambda \in [\varepsilon, 1]\} \tag{5.22}$$

using $\varepsilon > 0$ to guarantee a minimal step.

The KKT conditions seem naturally to induce a merit function. In practical applications it shows that such a merit function is not suitable for the reasons to follow. When using a KKT conditions based merit function three factors have to be considered. First the violation of the derivative of the Lagrangian $\mathrm{g} = 0$, second the violation of the nonlinear constraints $F(X) = 0$ and third the complementarity $\langle X, S\rangle$. By construction the cone constraint violation of the primal variable can be assumed to be zero. The cone constraint violation for the dual variable has to be measured, too. It is not enough to measure the complementarity. It is difficult to weigh all three terms correctly to get a fast convergence.

It is numerically impossible to find a global minimum of (5.22) unless $F(x)$ satisfies a certain smoothness property. Typically we are interested in finding at least an approximation to a local minimum. On one hand $F(x)$ might be expensive to evaluate on the other evaluating $F(x)$ is typically by far cheaper than to solve an additional SLCP-subproblem. Thus the question is how to gain a "useful" step with as little evaluations as possible. In this implementation chapter this "usefulness" of a line search is determined by the global convergence, especially the global convergence we experienced with the reduced order model examples that motivated this work.

An alternative that is often used and easy to implement is the Armijo line search. Let $j = 0..J$ and ρ be a constant typically $\rho \in [0.75, 0.95]$ and $\rho^J \approx \varepsilon$. Using the Armijo line search a local "minimum" is found by evaluating the points $\lambda = \rho^j$ until $p(x + \lambda\Delta x)$ increases. This is only a rough approximation of a local minimum, but as M is also only an estimated parameter there is no reason to be more precise for λ than for M. For an $F(X)$ that is expensive to obtain the parameter ρ can be reduced. This line search also tends to yield longer steps than other line searches. Additionally one can enforce a monotony condition $p(x) > p(x + \lambda\Delta x)$ to avoid local minima that are worse approximations than the current iterate.

For most of our examples the convergence using a line search irrespective of the estimator we tried is slow. The method generated very short steps, when $F(x)$ got too large. Next we adaptively adjusted M not only according to y but also to $F(x)$. This increased the convergence speed a little bit.

As the violation of the constraints seemed to be the main problem for the slow convergence a quadratic correction step was introduced into the line search. For a step $x + \Delta x$ we define the following minimization problem for the line search parameter

$$\xi^* := \operatorname{argmin}\{\ \|\xi\| \mid F(x + \Delta x) + \mathrm{D}F(x)\xi = 0\ \}. \tag{5.23}$$

With ξ^+ we do a line search along

$$x^+(\lambda) = x + \lambda\Delta x + \lambda^2\xi^* \quad \text{with} \quad \lambda \in [0, 1]. \tag{5.24}$$

We also experimented with using $\mathrm{D}F(x+\Delta x)$ instead of $\mathrm{D}F(x)$. Using $\mathrm{D}F(x)$ avoids an additional calculation of $\mathrm{D}F$ and it has proven in practical applications to yield more efficient steps, for our examples. With the quadratic correction again only a small convergence improvement was noticeable.

Filter

A very different class of step length controls are filters. The filter approach is a heuristic approach that seems unsatisfactory, but yields good convergence results in practical applications. Moreover, theoretical analyses of various filter approaches have established global convergence properties to stationary points (see e.g. [FLT02]). They can easily be introduced using the SLCP subproblem with a trust region. A filter is a set Σ that holds pairs of objective values and constraint violations measures for a subset $\{x_{k_i}\}$ for the past iterates $x^{(k)}$

$$\Sigma := \left\{\ \left(c^{\mathrm{T}}x_{k_i}, \|F(x_{k_i})\|\right)\ \right\}. \tag{5.25}$$

Let $0 < \alpha \ll 1$ and $0 < \beta \ll 1$ be two constants. In the standard filter approach a new step $x^+ = x + \Delta x$ is accepted if for every k_i one of the following conditions is fullfilled

$$\begin{aligned} &a)\ c^{\mathrm{T}}x^+ + \alpha F(x^+) \le c^{\mathrm{T}}x_{k_i} \\ &b)\ F(x^+) \le \beta F(x)_{k_i}. \end{aligned} \tag{5.26}$$

In other words a step is accepted if either the reduction of the objective function is longer by a factor α than the increase of the constraint violation or if the constraint violation decreases at least by a factor β.

If a new point is accepted then

- x^+ is the new iterate $x^{(k+1)}$,

- the trust region r is adapted (either through an estimation or set to a multiple of the current trust region or the length of the step),
- x^+ is added to the filter,
- and all surplus filter elements x_{k_j} are removed, that is all x_{j_i} for which

$$c^{\mathrm{T}}x^+ + \alpha F(x^+) \leq c^{\mathrm{T}}x_{k_j} \quad \text{and} \quad F(x^+) \leq \beta F(x_{k_j}). \tag{5.27}$$

  holds

On the other hand, if the element is not accepted

- the trust region r is reduced (e.g. by a fixed factor),
- the search step Δx^+ is dropped,
- and the iterate stays the same $x^{(k+1)} = x^{(k)}$.

Figure 5.1 shows a filter and its acceptable area. One can also see the points of the filter sets and some points that are obsolete to the filter, when the point x^+ is added.

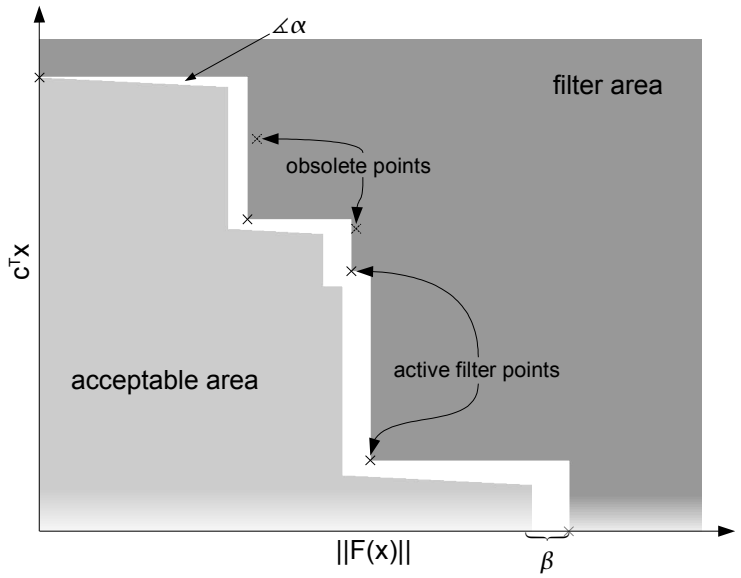

Figure 5.1: A filter.

Recall that our main problem for fast convergence seemed to be getting short steps once the non-conic constraint violation was too large. Filters now allow us to limit that violation by defining a start set. The start set contains the pair $(c^{\mathrm{T}}x_0, \|F(x_0)\|)$ for the starting point x_0 and for some $\gamma > \|F(x_0)\|$

$$\text{a constraint violation limiting pair} \quad (-\infty, \gamma). \tag{5.28}$$

With this initial filter set, $\beta\gamma$ is the largest acceptable nonlinear constraint violation.

By using the filter approach a faster convergence is achieved. The downside of filters is that some search steps are dropped even though the calculation of a search step is the

most CPU time expensive part. In the following chapter we will present an augmented filter approach. This approach does not drop any search step and still it yields convergence in even less steps. Additionally it supports another way of guessing the closeness to the optimal solution.

5.6 Augmented filter

In the last section we have introduced various step length controls. It turns out that the filter method was superior over the line search approach for the NLCPs (nonlinear conic programs) we tested with our algorithm.

On one hand the calculation of a search step is the most expensive part of the algorithm. On the other hand every search step either reduces the constraint violation or the objective value or both, except for the inaccuracy of the search step due to floating point rounding errors. Thus dropping a search step is undesirable.

A simple but effective alternative to avoid this is, to use this search step with a reduced step length. Hence the traditional trust region concept is modified in such a fashion that a search step resulting from a trust region problem is subject to a "filter line search". We could use a penalty line search, but there is no guarantee that the resulting step would be acceptable. Moreover such a line search would possibly destroy our convergence speed gain. Therfore we use the longest step that is acceptable for the filter.

Finding an acceptable reduced step can be done with an Armijo-like "line search" meaning the longest step of $x + \rho^i \Delta x$ that is accepted by the filter is used.

To further improve efficiency we treat search directions with $c^{\mathrm{T}} \Delta x > 0$ differently. For such a direction we search for a local minimum of $\|F(x + \lambda \Delta x)\|$. We use an "inverse" Amjijo line search that tests the points $x + \frac{\lambda_{\min}}{\rho^i} \Delta x$ for $i = 0 \,..\, \log_\rho \lambda_{\min}$.

A line search for $c^{\mathrm{T}} \Delta x \leq 0$ is shown in figure 5.2.

If α and β are sufficiently small we can find such a step, assuming our search step calculation does not suffer too severely from rounding errors.

In practical applications α and β might not always be small enough. If we cannot find an acceptable reduced step we first change our subproblem formulation to a more accurate one. If we already use the most accurate subproblem formulation we go a minimal step and reduce first α and if necessary then β accordingly. Note that reducing α and β does not render any filter elements obsolete. We experienced that the reduction of α and β is only necessary if the iterate is very close to the optimum. Note that the parameters α and β are less important in the augmented filter approach. In the standard trust region approach α and β directly influence wether a step is dropped and thus influence the trust region parameter. In the augmented filter approach α and β only influence which step length is accepted.

This filter approach does not depend on a trust region. Thus we can use the cheaper search step (5.18) that is less accurate. If the reduced step is very small (e.g. less than 25% of the search step's length) then it is appropriate to switch to a high accuracy search step either with or without a trust region.

If a trust region is used a simple but effective way to determine a trust region size r is to set it a little larger than the last step, $r = \vartheta \|x^{(k+1)} - x^{(k)}\|$ with for instance $\vartheta = \frac{3}{2}$. It turns out that for a large ϑ almost never a full step in the filter line search is done, but always a very large one. We observe that almost all iterates lie close to the border of the constraint violation limit until there is no feasible point for the subproblem that still decreased the objective value. Thereafter the filter will only accept points that decrease the constraint violation.

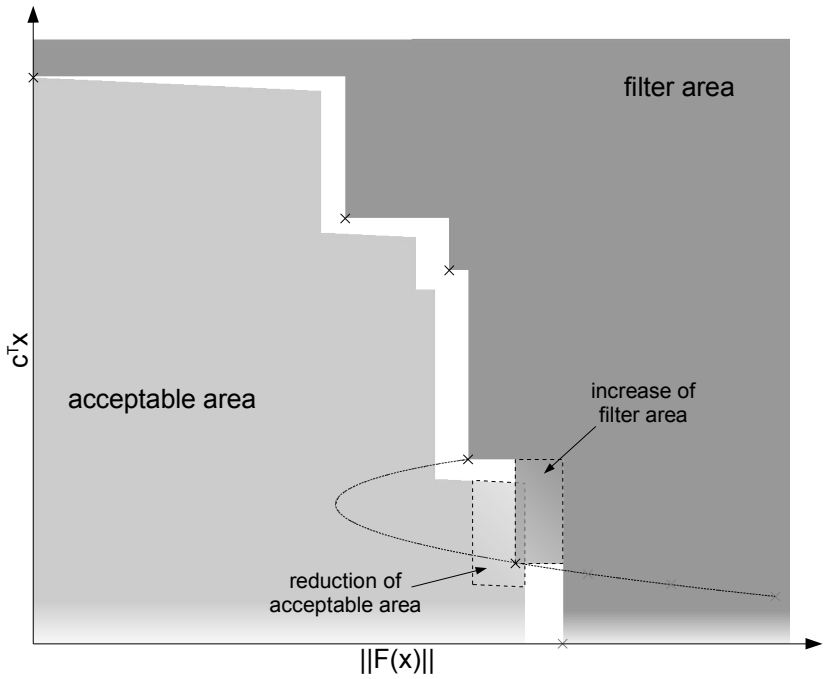

Figure 5.2: A reduced step that is accepted by the filter.

The convergence of an SLCP algorithm using such a filter is shown in Figure 5.3. The algorithm starts with a feasible point. Initially the constraint violation increases while the objective function is reduced. In iterations 8 to 11 the constraint violation is reduced again, in iteration 10 the objective value increases slightly. This figure shows a hard to solve example we used for testing. The first implementation with a penalty line search took over 60 iterations.

This behavior of iteration 8 to 11 could be observed in all of our problems and we suggest this as a switching characteristic for a hybrid solver in section 5.9.

Another shortcoming of the standard filter method is, that the limitation γ of the constraint violation from (5.28) is problem dependent and has to be guessed for each NLCP. Such a γ does not always arise naturally from the problem itself. An easy solution is to add a constraint limiting pair $(-\infty, \gamma)$ in a later iteration.

In our numerical experiments a large constraint violation resulted in a short acceptable step, compared to the step length of the calculated search step. Therefore we add a constraint violation limiting pair $(-\infty, \gamma)$ to the filter based on the constraint violation of the last iterate, e.g. $\gamma = \frac{1}{2}\|F(x^{(k-1)})\|$. Note that when $\|F(x^{(k)})\|$ is large the next steps can increase the objective value.

5.7 Stopping criteria

5.7.1 Abort criteria

For some problems a subproblem might not have any feasible point or the solver cannot find a feasible point. As we said before SeDuMi is a very robust solver, but a nearly

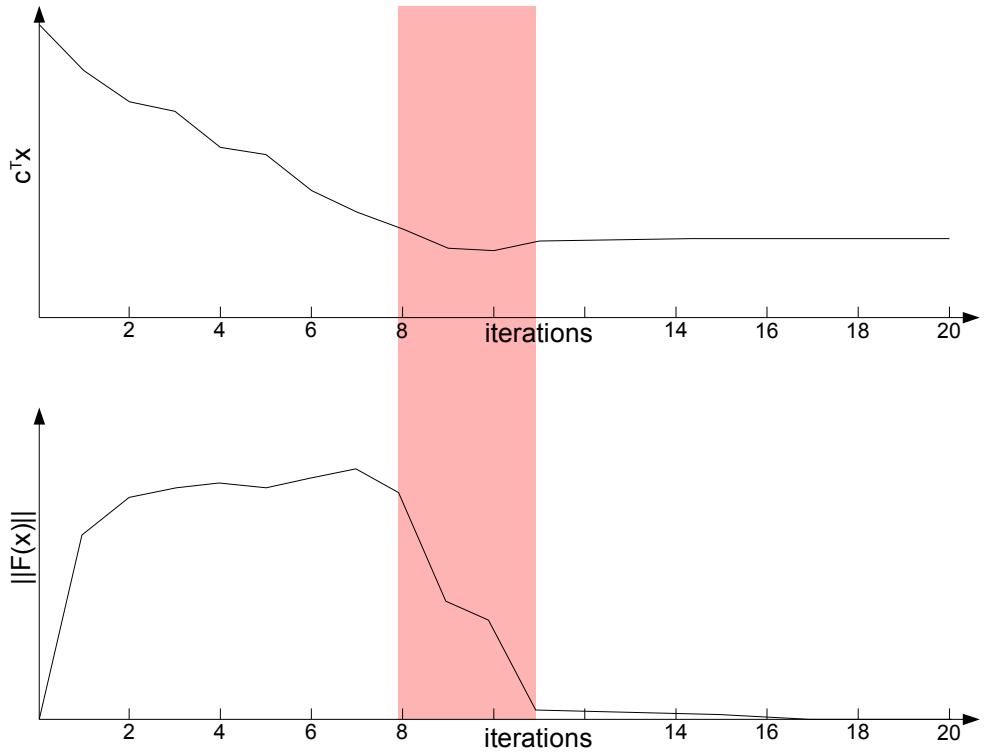

Figure 5.3: Convergence of an SLCP algorithm using an augmented filter.

degenerated feasible set often results in an increased computation time for SeDuMi and in a less accurate solution or premature stopping. There are different ways to deal with such a problem.

One way is to abort the overall solution process. For the reduced order model problem this abort indicated to us that a less degenerated formulation should be found. In section 5.1 we presented such a reformulation.

Another way is to drop the last step and to add a constraint violation limiting pair. If a line search instead of a filter is used then the penalty parameter M is adjusted.

Another abort criterion is the iteration count. This is typically introduced to abort in cases of a program or problem design error. For an hybrid-algorithm an iteration count threshold can be useful to switch between the solvers. For more details on an hybrid solver see section 5.9

5.7.2 No "improving" step

Let $x^{(k)}$, $x^{(k+1)}$ be two consecutive iterates of the SLCP algorithm. A step $x^{(k+1)} - x^{(k)}$ is called an improving step if

$$c^{\mathrm{T}}(x^{(k+1)} - x^{(k)}) < 0 \quad \text{or} \quad F(x^{(k+1)}) < F(x^{(k)}). \tag{5.29}$$

is satisfied.

One might think of an improving step as a step that is acceptable for a filter for small enough filter parameters α and β.

The high accuracy search step formulation is used if the direction of the last search step could not be used to find an improving step. If the high accuracy search step formulation was used then the program is aborted.

This criterion is very helpful, because it simply stops when "we cannot do any better". This avoids cycling, especially when other stopping criteria like the KKT stopping criterion from section 5.7.3 cannot be satisfied in the course of the algorithm.

One way to implement this stopping criterion is, to stop, when the step is not accepted by the filter even for very small values $\alpha_{\min}$ and $\beta_{\min}$.

5.7.3 The KKT conditions

The KKT optimality conditions is the classical candidate for a stopping criterion.

Let (x^*, y^*) be an optimal pair for (5.12). Recall the KKT conditions from section 2.3

see (2.19)
$$\begin{aligned} x^* &\in \mathcal{K}, \\ s^* := -\mathrm{D}c(x^*) - {y^*}^{\mathrm{T}}\mathrm{D}F(x^*) &\in \mathcal{K}^D, \\ F(x^*) &= 0, \\ \langle s^*, x^* \rangle &= 0. \end{aligned}$$

Let $(x, y) = (x^{(k)}, y^{(k)})$ be the k-th iterate of the SLCP algorithm. For this pair (x, y) we define $s := -c^{\mathrm{T}} - y^{\mathrm{T}}\mathrm{D}F(x)$. If (x, y) is not the optimal pair (x^*, y^*) then (under Robinson's constraint qualification) we know that one of the conditions (2.19) for $(x^{(k)}, y^{(k)}, s^{(k)})$ is violated.

We can assume that $x^{(k)} \in \mathcal{K}$ is satisfied, which is trivially true if any $j < k$ satisfies $x_j \in \mathcal{K}$. The other conditions might be violated and violation measures will be discussed in the following.

Violation of $F(x) = 0$

$F(x)$ might be a set of functions that might be differently scaled. Thus we consider the violation of each $F_j(x) = 0$ separately. To measure the violation we define

$$F_j^{\max} := \max\{\ |F_j(x^{(i)})|,\ i \in 0, .., k\ \} \tag{5.30}$$

and a scaled violation measure

$$\tilde{F}_j(x) := \frac{F_j(x)}{F_j^{\max}}. \tag{5.31}$$

The overall violation of the non-conic constraints is measured by either average constraint violation, a weighted Euclidean constraint violation or the maximum constraint violation, i.e. by

$$\frac{1}{m}\|\tilde{F}(x)\|_1 \quad \text{or} \quad \frac{1}{\sqrt{m}}\|\tilde{F}(x)\|_2 \quad \text{or} \quad \|\tilde{F}(x)\|_\infty. \tag{5.32}$$

Since we want all constraint violations to be equally small we used $\|\tilde{F}(x)\|_\infty$ for most examples.

Violation of $s \in \mathcal{K}^D$

Let x be the Cartesian product of x_j. The vectors x_j lie in the following cones

see (5.8)
$$\mathbb{R}^n,\ \mathbb{R}^n_+,\ \mathcal{Q}^{n+1},\ {\mathcal{Q}_{\mathcal{R}}}^{n+2},\ \mathcal{S}^n_+.$$

Let accordingly s be the Cartesian product of s_j.

All cones listed above are selfdual, except for $\mathbb{R}$ which has $\{0\}$ as dual cone.

Let $\hat{s}_j$ be the orthogonal projection onto the dual cone containing x_j. A good measure of the conic constraint violation σ_j is

$$\sigma_j := \frac{\|s_j - \hat{s}_j\|_2}{\|s_j\|_2}. \tag{5.33}$$

Orthogonal Projections

The orthogonal projection of s_j onto $\mathbb{R}_+^{n_j}$ is trivially

$$\hat{s}_j := \max(s_j, 0) := \begin{pmatrix} \max((s_j)_1, 0) \\ \vdots \\ \max\left((s_j)_{n_j}, 0\right) \end{pmatrix}. \tag{5.34}$$

The orthogonal projection of $s_j = \begin{pmatrix} (s_j)_0 \\ \bar{s}_j \end{pmatrix}$ onto $\mathcal{Q}^{n_j+1}$ is

$$\hat{s}_j = \begin{cases} s_j & \text{for} \quad (s_j)_0 \geq \|\bar{s}_j\| \\ \begin{pmatrix} \tau\|\bar{s}_j\| \\ \tau(\bar{s}_j)_1 \\ \vdots \\ \tau(\bar{s}_j)_{n_j} \end{pmatrix} & \text{else} \end{cases} \qquad \text{with } \tau := \left(\frac{(s_j)_0}{2\|\bar{s}_j\|} + \frac{1}{2}\right). \tag{5.35}$$

For the projection onto ${\mathcal{Q}_\mathcal{R}}^{n_j+1}$ let $\tilde{s}_j$ be the rotated vector of s_j

see (5.11)
$$\tilde{s}_j := \begin{pmatrix} \frac{1}{\sqrt{2}}(s_{j0} + s_{j1}) \\ \frac{1}{\sqrt{2}}(s_{j0} - s_{j1}) \\ \bar{s}_j \end{pmatrix}.$$

Note that $\tilde{\cdot}$ is a mapping from $s_j \in {\mathcal{Q}_\mathcal{R}}^{n_j+2}$ to $\tilde{s}_j \in \mathcal{Q}^{n_j+2}$. This mapping is unitary as it is a rotation. Thus the projections for $s_j \in {\mathcal{Q}_\mathcal{R}}^{n_j+2}$ are the same as for $\tilde{s}_j \in \mathcal{Q}^{n_j+2}$.

For the projection onto $\mathcal{S}_+^n$ we assume that s_j is a symmetric matrix. Let $QDQ^{\mathrm{T}} = s_j$ be the eigenvalue decomposition of s_j with $D = \mathrm{Diag}(d)$ and d_i the eigenvalues of s_j. Further let d_+ and $D_+ = \mathrm{Diag}(d_+)$ respectively have the entries $\max(d_i, 0)$. With these definitions the orthogonal projection $\tilde{s}_j$ of s_j is

$$\tilde{s}_j := QD_+Q^{\mathrm{T}}. \tag{5.36}$$

Overall norm of σ_j

Recall the overall measure of the violations of $F_j = 0$. Analogously let σ be the vector $(\sigma_j)_j$. Again we can measure the overall cone violation using

$$\frac{1}{m}\|\tilde{\sigma}\|_1 \quad \text{or} \quad \frac{1}{\sqrt{m}}\|\tilde{\sigma}\|_2 \quad \text{or} \quad \|\tilde{\sigma}\|_\infty. \tag{5.37}$$

Again we used the ∞-norm for our examples.

Violation of the complementarity

The complementarity is given by

$$\langle s, x \rangle = 0, \quad x \in \mathcal{K}, \quad s \in \mathcal{K}^D. \tag{5.38}$$

Recall x and s are cartesian products of

see (5.8) $\qquad \mathbb{R}^n$ respectively $\{0\}$, $\mathbb{R}^n_+$, $\mathcal{Q}^{n+1}$, ${\mathcal{Q}_{\mathcal{R}}}^{n+2}$, $\mathcal{S}^n_+$.

Let again be x_j and s_j be the components of the cartesian product of x and s respectively. We already consider the cone constraint violation, here we discuss how to measure the orthogonality of x_j and s_j.

Note that for $\mathbb{R}^n$, $\{0\}$, $\mathbb{R}^n_+$, $\mathcal{Q}^{n+1}$, ${\mathcal{Q}_{\mathcal{R}}}^{n+2}$ the standard scalar product is $\langle s_j, x_j \rangle = s_j^{\mathrm{T}} x_j$. We can use the same scalar product for $x_j, s_j \in \mathcal{S}^n_+$ by using the vector representation $\operatorname{vec}(x_j)$ and $\operatorname{vec}(s_j)$ since

$$x_j \bullet s_j = \operatorname{vec}(x_j)^{\mathrm{T}} \operatorname{vec}(s_j). \tag{5.39}$$

Again we get one measure $x_j^{\mathrm{T}} s_j$ for each j and have to consider a overall norm. Since all scalar products are the same an alternative is measure $x^{\mathrm{T}} s$.

To normalize our measure we use $\frac{s^{\mathrm{T}} x}{\|s\|_2 \|x\|_2}$ and analogously for the separate measure x_j and s_j.

Final remarks on the KKT condition based stopping criterion

For most examples we let the SSP approach calculate until the accuracy can no longer be improved, i.e. until the search step given cannot be used for an "improving" step (see section 5.7.2). We use the KKT conditions afterwards to check how accurate the result is.

Examples for which a high accuracy is unnecessary use the KKT condition as a stopping criterion. Even though the measurements mentioned are scaled based on the problem's data, it might be useful to introduce weights for the different measurements.

As mentioned before a line search based on the KKT conditions is problematic as it has local minima with $\|F(X)\| > 0$.

5.8 Speed ups for the reduced order model example

The reduced order model example from section 5.1 is one important application of the algorithms presented in this thesis. In this section we will describe in more detail how this specific problem is solved and how we speed up the convergence.

Recall that the reduced order model problem is given by

see (5.6) $$\min \left\{ s_0 \;\middle|\; \begin{array}{rl} X^{\mathrm{T}} B_1 + S - B_2 = 0, & X, P_G, P_C \in \mathbb{R}^{n \times n}, \\ X^{\mathrm{T}}(G + P_G) + (G + P_G)^{\mathrm{T}} X - Z_G = 0, & Z_G, Z_C \preceq 0, \\ X^{\mathrm{T}}(C + P_C) + (C + P_C)^{\mathrm{T}} X - Z_C = 0, & \begin{pmatrix} s_0 \\ S \end{pmatrix} \in \mathcal{Q} \\ X^{\mathrm{T}}(C + P_C) - (C + P_C)^{\mathrm{T}} X = 0, & \end{array} \right\}.$$

Let n be the dimension of the matrices $X, G, C, Z_G, Z_C \in \mathbb{R}^{n \times n}$. Let $m < n$, $B_1, B_2 \in \mathbb{R}^{m \times n}$. And let l be the number of free entries of P_C and P_G. For the symmetric conditions

of (5.6)

$$\begin{aligned} X^{\mathrm{T}}(G+P_G)+(G+P_G)^{\mathrm{T}}X-Z_G=0,\\ X^{\mathrm{T}}(C+P_C)+(C+P_C)^{\mathrm{T}}X-Z_C=0 \end{aligned} \tag{5.40}$$

only $\frac{1}{2}n(n+1)$ conditions are necessary for each equation and for

$$X^{\mathrm{T}}(C+P_C)-(C+P_C)^{\mathrm{T}}X=0 \tag{5.41}$$

only $\frac{1}{2}n(n-1)$ conditions are necessary. Moreover, due to details in the implementation of SeDuMi, it turns out that not omitting the redundant symmetric conditions of $X^{\mathrm{T}}(C+P_C)-(C+P_C)^{\mathrm{T}}X=0$ leads to a faster convergence of SeDuMi.

With this formulation the nonlinear program has

$$\begin{aligned} 3n^2+nm+2l+1 \text{ variables and}\\ nm+n(n+1)+n(n-1) \text{ constraints.} \end{aligned} \tag{5.42}$$

Please recall that when solving the SLCP subproblem with high accuracy (5.17/5.20) we have

$$\begin{aligned} 3(3n^2+nm+2l+3)+2[+1] \text{ variables and}\\ (nm+n(n+1)+n(n-1))+2(3n^2+nm+2l+1)[+2] \text{ constraints.} \end{aligned} \tag{5.43}$$

That is more than $9n^2$ variables and more than $8n^2$ equality constraints.

For problems that occur in practical applications we have up to $n=40$. This results in a matrix A of the linear constraints of the subproblem with a dimension larger than 14400×12800. Thus storing only the already set up matrix for the subproblem would take 1,5GB of RAM if we wouldn't use a sparse matrix format. Note that even though C, G, P_C and P_G are not sparse, A is sparse, thus these problems can be calculated on a personal computer.

Please note that the linear reduced order model problem 5.3 only needs

$$\begin{aligned} 3n^2+nm+1 \text{ variables and}\\ nm+n(n+1)+n(n-1) \text{ conditions.} \end{aligned} \tag{5.44}$$

This means that calculating a linear problem is considerably cheaper compared with the generated SLCP subproblem. In the following we will show how to use this fact.

We gain such a linear reduced order model problem, by fixing the perturbations P_C and P_G. Thus we can determine a X, Z_C and Z_G by solving the linear reduced order model problem such that we have a feasible point for the nonlinear program. So the first approach is to increase the rate of convergence by projecting the new iterate onto the feasible set after each step. This is done by solving an additional linear SDP. This increased the rate of convergence drastically, the necessary SLCP steps reduced from over 50 steps to less than 10 steps. Theoretically, this modification could lead to some form of cycling, but it never happened for our data.

Projecting after each step only works if

$$\begin{aligned} X^{\mathrm{T}}(G+P_G)+(G+P_G)^{\mathrm{T}}X-Z_G=0,\\ X^{\mathrm{T}}(C+P_C)+(C+P_C)^{\mathrm{T}}X-Z_C=0, \end{aligned} \tag{5.45}$$

but what we need in the end is to satisfy

$$\begin{aligned} X^{\mathrm{T}}(G+P_G)+(G+P_G)^{\mathrm{T}}X-Z_G &= \lambda_\varepsilon I, \\ X^{\mathrm{T}}(C+P_C)+(C+P_C)^{\mathrm{T}}X-Z_C &= \lambda_\varepsilon I. \end{aligned} \tag{5.46}$$

We achieved this by including this projection onto the feasible set in our filter-line search. If the objective function Δs_0 satisfies $\Delta s_0 \leq 0$ we used the longest step, that still had a solution. These linear SDPs are much cheaper to solve than the subproblems from the nonlinear problem. Nevertheless we aim to solve only few of them. We decided to reduce the Armijo factor ρ to solve less linear SDPs.

Calculating initial values is easy as $P_C = P_G = 0$ are good initial values, so we only have to solve the simple linear program once and get a feasible starting point. Often this starting point is close close to the optimum.

For problems that had a singular C the process typically converged to a non singular solution $C + P_C$. The convergence is very slow until a matrix P_C was found such that the matrix $C + P_C$ is non singular. Once such a P_C is found the solver converges within a few steps. To avoid the slow convergence part, we chose the initial matrix P_C such that we have a non singular sum $C + P_C$. This led to fast convergence for such examples[2].

As a small restricted perturbation is desired we replaced

$$P_C, P_G \in \mathbb{R}^{n\times n} \quad \text{with} \quad \begin{pmatrix} p_C \\ P_C \end{pmatrix}, \begin{pmatrix} p_G \\ P_G \end{pmatrix} \in \mathcal{Q}^{n^2+1} \tag{5.47}$$

and used p_C, p_G to restrict the maximal allowed violation. The approach for an initial matrix P_C with $C + P_C$ non singular however might be infeasible for this condition. We use the following approach to overcome this problem.

Running the SLCP algorithm with the infeasible P_C can lead to an unsolvable subproblem. If this subproblem is solvable still a necessary reduced step could result in a point that was infeasible again. A solution is to start with an appropriate p_C and reduce it over time until the desired perturbation limit is reached.

Let $\hat{p}_C$ be such a desireable perturbation limit. If a penalty term $\varrho\|p_C - \hat{p}_C\|$ is included in the target function[3], then steps are preferred, that reduce the perturbation size to the wanted amount.

To conclude this section we present and discuss some results shown in table 5.1. The tabel lists the dimensions n of the reduced order model problem, as well as the NLSDP variable size $N(n)$ and the number of nonlinear constraints $M(n)$. The number of SSP steps is listed under "iter", while CPU time lists the number of minutes it took to compute the example. The results presented are all based on examples with a non singular C. With the trick mentioned above singular ones take about 2 to 4 additional steps, if an appropriate starting point P_C is found. Of course we can construct examples, where the initial starting point satisfies $\|P_C\| \gg p_C$, and the algorithm takes a lot more iterates. This situation did not occur in the practical relevant cases in our examples.

As mentioned before the time SeDuMi takes for solving a subproblem depends on different factors, including the degree of degeneracy of the feasible set. A few examples stand out in table 5.1 by using more CPU time, while having smaller dimensions with only few SLCP steps, see examples $n = 24$ and $n = 32$. SeDuMi takes more CPU as the occurring subproblems have an almost degenerated feasible set. Table 5.1 does only include examples up to $n = 35$, due to a lack of memory in our computer.

[2] The solution of the circuit design problem, however, typically is singular.

[3] This term can either be included temporarily until we replace p_C with $\hat{p}_C$ or its weight ϱ has to be increased over time.

n	$M(n)$	$N(n)$	iter	CPU time	n	$M(n)$	$N(n)$	iter	CPU time
8	118	285	5	3.71	22	783	1713	4	416.31
9	146	348	5	4.43	23	853	1862	5	151.33
10	177	417	7	8.06	24	926	2013	6	683.54
11	211	492	5	7.18	25	1002	2176	3	145.60
12	248	573	8	16.05	26	1081	2337	5	612.22
13	288	660	4	10.88	27	1163	2508	7	518.92
14	331	753	6	20.77	28	1248	2685	5	789.41
15	377	852	7	30.12	29	1336	2868	4	475.52
16	426	957	6	34.38	30	1427	3057	7	4213.50
17	478	1068	5	37.40	31	1521	3252	4	784.34
18	533	1185	10	91.17	32	1618	3455	6	4659.64
19	591	1308	4	47.61	33	1718	3660	5	1130.44
20	652	1437	5	83.66	34	1821	3877	2	630.53
21	716	1572	4	289.48	35	1927	4092	6	1799.36

Table 5.1: Numerical results for the reduced order model example.

5.9 A hybrid solver

The SLCP method only guarantees linear convergence while the global convergence is fast. Consider again figure 5.3. The second half of the steps where used to gain a high accuracy. This was a typical result with the problems we tested. Note that these steps are the most time consuming part, because we have to use the high accuracy search step formulation that has a larger problem dimension for the last few steps. A quadratically convergent solver would take much less steps. Moreover the solver we present here consumes much less CPU time. Note that the behavior of increasing objective function and decreasing constraint violation is a good indication for closeness to the optimum. Using the augmented filter approach this behavior can be observed for all of our test cases.

In nonlinear programming, the convergence path can be indefinitively long for an adequate[4] constraint $F(x)$. It seems from practical experiments that the SLCP approach gets close to an optimal solution quickly. Once we have a point close to the optimum the local convergence property of the algorithm is crucial.

Using the indicator that we get by the augmented filter or based on the KKT conditions one can formulate a hybrid solver. This hybrid solver uses one algorithm to get close to the optimum, that we will call "global solver" and another to gain accuracy that we will call "local solver". The hybrid solver can even decide to switch back to the global solver when the closeness estimate increases again.

While the SLCP algorithm qualifies as a "global solver" we want to propose an interior point method (IPM) as a "local solver". The combination of these two solvers is especially interesting, as they are based on similar primal and dual data. Using the SLCP method we get a dual equality constraint variable from SeDuMi and can then easily calculate the dual conic constraint variable. All we have to do is to shift these slightly inside the cone[5] and compute an IPM step without having to solve any additional starting point problem.

In the following chapters we will discuss such a primal dual IPM method. We especially focus on the conditions that guarantee local quadratic convergence.

[4] e.g. for bounded problems without a finite optimal solution
[5] This might increase the equality constraint violation

5.10 A Matlab OOP implementation

In this section we want to add a few technical details about the implementation. As we focused on usable results for the reduced order model problem, we did not implement one step length control and one search step, but experimented with many different, that can be specified together with the given nonlinear program. So in this chapter we want to describe our approach. Our aim was to maintain maximal flexibility and easy extendability.

The basis for our approach was the object oriented programming (OOP) support provided by Matlab.

The OOP feature to combine functions and data was used to define an abstract class for the objects to be solved. As we solved problems of the class

see (5.12) $$\min \left\{ \; c^{\mathrm{T}} x \mid F(x) = 0, \; x \in \mathcal{K} \; \right\}$$

we build an abstract class called NLCP (nonlinear conic program[6]).

This abstract class already included a numerical derivative $\mathrm{D}F(x)$ and a numerical Hessian of the Lagrangian $\mathcal{H}$. Thus an inherited class does not need an implementation of these functions for first tests. Once the analytical derivatives are implemented, this class allows to verify the analytical derivatives based on its numerical derivatives. This class includes a void function $F(x)$ for problems without any nonlinear constraints and a placeholder function "Postcorrection(x)" that can be overloaded if necessary.

The main benefit of OOP for us is that we defined every element of the algorithm as a class. Thus the solver has objects for getting a search step, an $\mathcal{H}$ update, for a step length control and a stopping criterion. Then again these classes can have other classes they depend on such as a line search has a separate classes for different merit functions. In a trust region approach the search step can even come with its own line search.

When the SLCP solver is called the solver gets the problem object and can be given additional objects for step length control, etc. The solver automatically adds the default choices for elements that are not given, e.g. the "calculate $\mathcal{H}$ and update it with BFGS" hybrid from section 5.3 update for the $\mathcal{H}$-update and the augmented filter approach as a step length control. For a problem where the second derivative is cheap to calculate a $\mathcal{H}$-update could be given, that gives back a projection of the exact Hessian $\mathcal{H}(x, y)$ onto the cone of positive semidefinite matrices, in every iteration.

Another strong benefit of the OOP implementation is that the solver is as readable as the algorithm itself, as it only calls the abstract functions without knowing how the single elements are handled or implemented. We will even use this same solver to implement our IPM in the future as it mainly has different search steps. So all we need to do is automatically choose between centering and predictor steps. We could even comunicate with the step length control to use different merit functions depending on the search step being a centering or predictor step.

The main difference between our solver and the SLCP algorithm is that our solver is already hybrid. So far we switch between two different search steps that have different accuracies.

As we had practical results in mind we tried different combinations of search steps and step length controls etc. Table 5.10 shows the main classes implemented for our solver. The stopping criteria are not listed. We used the standard stopping criteria from section 5.7 and the KKT conditions. Additionally every element such as step length control or the search step subproblems can force the loop to quit. This is useful if the given direction given can not be used for an "improving" step (see section 5.7.2) or if SeDuMi could

[6]For supported cones see (5.8).

not solve the subproblem. In the second case a problem description allows the solver to continue by e.g. using the other search step given to the hybrid solver. This means that the solver switches to a more accurate search step once the fast search step can not be used for an "improving" step.

Element Type	**Description**
search step	SD formulation using only $x^{(k+1)}$ (low accuracy)
	SD formulation using Δx and $x^{(k+1)}$
	SD with Trust Region
	SD with weighted constraint violation
	SD with $x^{(k+1)}$ and reduced size (through svd)
Step Length Control	Armijo Line Search
	Armijo Line Search with quadratic correction
	Golden Section Line Search
	Filter approach
	augmented Filter approach
$\mathcal{H}$-Update	exact $\mathcal{H}$ (with projection)
	$\mathcal{H}_+$ of the augmented Lagrangian
	BFGS Update
	damped BFGS Update
	hybrid damped BFGS Update
merit functions	$\|F(x+\Delta x)\| + Mc^{\mathrm{T}}\Delta x$
	KKT merit function (see sections 5.7.3 and 5.5)

Additionally to the OOP approach for a general solver, we implemented a detailed messaging system. This messaging system allowed us to get different level of details to the screen as well to different files. Information that is not printed is collected or calculated, this is in opposite to other solutions that divert the output into "/dev/null" or similar. The output can be formated accordingly to the needs of the reader. It might be organized in blocks for each iterate, which is used when a lot of output is given. Or it is organized as a table for a quick overview of the algorithm's development over time. While for a normal useage a minimal screen output is enough, a more detailed output allowed us to easily choose the right components for all the different problems we solved and helped us finding bugs quickly.

For debugging the common convergence characteristics are shown on screen while all additional information is split up into several files for in depth analysis.

This messaging system also handles the plotting output. This plotting system can be adjusted to the information needed and supports multiple plots as well as collecting plots in a few windows. It supports tasks like continues per-iterate drawing that is used for the development comparison of target function value and constraint violation. Additionally line searches and filter can draw their own plots for each iteration, on request or under predefined circumstances. Additional informational plots are drawn for iterates that lead to an abort of the algorithm.

The plots in this thesis were created by the solver itself.

6 Interior point methods

In this section we present the basics of IPMs needed in the following chapters. We start with some general notations and continue with the optimality conditions and the basic idea of IPMs like the one presented here.

We conclude this chapter by presenting a central path. In the next chapter we present an algorithm that is following this central path.

6.1 About IPMs

When interior point methods (IPMs) for linear programs where first introduced in 1984 by Karmarkar (see [Ka84]) they had to compete with the simplex method. The simplex method solved linear programs by moving along the edges from corner to corner checking whether there was an edge left that led to a corner with a smaller objective value. The simplex method was acceptably efficient in practical applications, but had an $\mathcal{O}(n!)$ worst case complexity.

The IPM introduced by Karmarkar takes $\gamma\sqrt{n}\log\frac{1}{\varepsilon}$ steps to archive an accuracy of ε. From Khachian it has been known (see [Kh79]) that the exact solution can be constructed from a sufficiently close approximation of the optimal solution when the data is given by rational numbers. The convergence of IPMs is obviously superior to the worst case of the simplex method, but Karmarkar's approach had to be modified before it became practically relevant.

Efficient IPM implementations are competitive to the simplex method. One well known and often used algorithm is Mehrotra's long step predictor-corrector approach (see [Me92] and [LMS92]). Today it is still a matter discussion whether the simplex or an IPM implementation is "better" for linear programs.

As the IPM approach is not based on the special structure of linear programs, but on the structure of the optimality conditions, the IPM approach can easily be generalized to other problem classes. The IPM approach has been generalized for example for different cones. These cones include the quadratic cone as well the cone of positive semidefinite (PSD) matrices, as used in semidefinite programs (SDP).

In this thesis we further generalize the IPM for linear SDP to an approach for nonlinear (non convex) SDPs.

6.2 Notation and conventions

We typically consider a nonlinear semidefinite program (NLSDP) of the form

$$\min\{\ C \bullet X\ \mid F(X) = 0,\ X \succeq 0\} \tag{6.1}$$

or a more general form

$$\min\{\ c^{\mathrm{T}}x\ \mid F(x) = 0,\ x \in \mathcal{K}\}$$

where the cone $\mathcal{K}$ is either the cone of positive variables $\mathbb{R}^n_+$, the quadratic cone $\mathcal{Q}$, the cone of positive semidefinite matrices $\mathcal{S}^n_+$, or any cartesian product of these.

During the next chapters we use the following notations and conventions. Large letters indicate matrices for variables e.g. C, X or vector valued functions for mappings like $F(X)$. We assume that these matrices are of dimension n such that $C, X \in \mathbb{R}^{n\times n}$. Note that linear operators that are applied to $n \times n$-matrices like $A : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ can be written as $N \times N$ matrices with $N = n^2$. When we later discuss the complexity of operations we call $\mathcal{O}(n^2)$, $\mathcal{O}(n^3)$ negligible while the important operations are either $\mathcal{O}(N^2)$ or $\mathcal{O}(N^3)$. The operations of the latter are the one that determine the speed of the overall solver. The nonlinear constraint $F(X) \in \mathcal{C}^3$ is typically $F : \mathbb{R}^{n\times n} \to \mathbb{R}^m$.

Throughout this thesis we assume that every positive definite matrix is also symmetric. If we refer to non symmetric positive semidefinite matrices we point this out specifically.

We typically consider points that are close to a local solution (X^*, y^*, S^*) and assume that the problem is not degenerate.

6.3 Optimality conditions revised

In section 2.3 the aspects of the KKT optimality conditions that we needed for the SSP approach are discussed. Now we add some aspects that we need for our IPM.

Recall the Lagrangian for (6.1)

$$\mathcal{L}(X, y, S) := C \bullet X - y^{\mathrm{T}} F(X) - S \bullet X. \tag{6.2}$$

as well as its first derivative with respect to X that we need for the KKT conditions

$$\mathrm{g}(X, y, S) := \mathrm{D}_X \mathcal{L}(X, y, S) = C - \mathrm{D}F(X)^* y - S. \tag{6.3}$$

We also need its second derivative for later analysis

$$\mathcal{H}(X, y, S) := \mathrm{D}_{XX} \mathcal{L}(X, y, S) = -\mathrm{D}_X^2(y^{\mathrm{T}} F(X)). \tag{6.4}$$

As discussed before, an optimal solution is a feasible critical point for the Lagrangian that satisfies a complementarity condition.

Let (X^*, S^*, y^*) be a primal-dual pair of optimal solutions of (6.1). The KKT conditions then are

$$\begin{aligned} \mathrm{g}(X^*, y^*, S^*) &= 0, \\ F(X^*) &= 0, \\ X^* \bullet S^* &= 0, \\ X^*, S^* &\in \mathcal{S}^n_+. \end{aligned} \tag{6.5}$$

Leaving the cone conditions aside the system (6.5) has less equations than variables. It is well known that from $X^*, S^* \in \mathcal{S}^n_+$ and $X^* \bullet S^* = 0$ follows $X^* S^* = 0$. Exchanging the condition $X^* \bullet S^* = 0$ with $X^* S^* = 0$ leads to a system that has as many equations as variables. We need this to get a well-determined linearization for our relaxation.

The property $X^* S^* = 0$ has been used to define the class of MZ-symmetrizations

$$\mathbf{S}_P(XS) := \frac{1}{2}\left(PXSP^{-1} + \left(PXSP^{-1}\right)^{\mathrm{T}}\right). \tag{6.6}$$

A reason to use these different symmetrizations is that using such a symmetrization the solutions ΔX, ΔS for the linearized relaxed system (6.12) are symmetric.

One element of this class is the AHO symmetrization, it has $P = I$. We focus on this symmetrization as it has necessary properties for our IPM, that we discuss in chapter 8.

In practical applications of linear SDPs many other symmetrizations have been considered for their decomposition properties. It turned out that these are not suitable for the IPM presented here.

The AHO symmetrization $\frac{1}{2}(XS + SX) = 0$ is also the multiplication of a Jordan algebra over symmetric matrices $\mathcal{S}$. The cone of positive semidefinite matrices $\mathcal{S}^n_+$ is the space of squares over this algebra. This allows us to analyze certain properties in the space of Jordan algebras. Other cones such as the quadratic cone or the cone of positive variables can also be considered as set of squares over Jordan algebras. More details about the Jordan algebra over $\mathcal{Q}$ and $\mathcal{S}^n_+$ are given in sections 8.2.2 and 8.3.1 respectively.

Using a MZ-symmetrization (6.5) is equivalent to

$$\begin{aligned} \mathrm{g}(X^*, y^*, S^*) &= 0 \\ F(X^*) &= 0 \\ \mathbf{S}_P(X^* S^*) &= 0 \\ X^*, S^* &\in \mathcal{S}^n_+. \end{aligned} \tag{6.7}$$

6.4 Formulations of Complementarity conditions

In the case of the cone of positive variables $\mathcal{K} = \mathbb{R}^n_+$ it is obvious that for an optimal solution (x^*, y^*, s^*) of (6.2) either $x^*_i = 0$ or $s^*_i = 0$ or both for every $1 \leq i \leq n$.

A similar complementarity for the quadratic cone is shown in section 8.2.2 in proposition 8.2.3. The following proposition shows the complementarity condition for the positive semidefinite cone as mentioned in the last section.

Proposition 6.4.1. *Let $X, S \in \mathcal{S}^n_+$ if $X \bullet S = 0$ holds then it follows $XS = 0$.*

Proof. As we have

$$0 = X \bullet S = tr(XS) = tr(SX) \tag{6.8}$$

we know that $tr(\frac{1}{2}(XS + SX)) = 0$. From proposition 8.2.2 we know that the Jordan-product of two semidefinite matrices is again a semidefinite matrix. $\frac{1}{2}(XS + SX)$ is a Jordan multiplication for which the positive semidefinite matrices are the space of squares, thus $\frac{1}{2}(XS + SX) \succeq 0$. The trace of such a symmetric positive semidefinite matrix A is always the sum of its eigenvalues. This can be seen from the eigenvalue decomposition $QDQ^{\mathrm{T}} = A$ implying that $tr(QDQ^{\mathrm{T}}) = tr(Q^{\mathrm{T}}QD) = tr(D)$. It follows that $XS + SX = 0$.

Using the eigenvalue decomposition $Q_X D_X Q_X^{\mathrm{T}} = X$ we have

$$0 = XS + SX \quad \Leftrightarrow \quad D_X Q_X^{\mathrm{T}} S Q_X + Q_X^{\mathrm{T}} S Q_X D_X = 0. \tag{6.9}$$

Let λ_i be the eigenvalues of D_X then we have

$$0 = (\lambda_i + \lambda_j)(Q_X^{\mathrm{T}} S Q_X)_{i,j} \tag{6.10}$$

so either $(\lambda_i + \lambda_j) = 0$ (thus $\lambda_i = 0$ and $\lambda_j = 0$) or $(Q_X^{\mathrm{T}} S Q_X)_{i,j} = 0$. Thus it follows

$$0 = D_X Q_X^{\mathrm{T}} S Q_X \Leftrightarrow 0 = Q_X D_X Q_X^{\mathrm{T}} S = XS. \tag{6.11}$$

□

Note that in contrast to linear programming the solution in nonlinear programming does not necessarily converge towards a strict complementary solution.

6.5 A central path

As we mentioned before, applying the Newton approach to the equality constraints of (6.7) might lead to an infeasible solution. In IPMs for convex programs a central path depending on μ is defined. For each $\mu > 0$ we gain a solution that is in the inner of the primal and dual cone. The limit of these solutions for $\mu \to 0$ is a solution of (6.5).

We can use the symmetrized form (6.7) to formulate a central path close to a local solution

$$\begin{aligned} \mathrm{g}(X, y, S) &= 0, \\ F(X) &= 0, \\ \mathbf{S}_P(XS) &= \mu I, \\ X, S &\in \mathcal{S}^n_+. \end{aligned} \tag{6.12}$$

Note that the identity matrix I is also the identity of the Jordan algebra over positive matrices with the cone $\mathcal{S}^n_+$ as set of squares. For the Jordan algebra that has the quadratic cone $\mathcal{Q}$ as set of squares the identity is the vector $(1, 0, .., 0)^{\mathrm{T}}$.

Let (X_μ, S_μ, y_μ) be a point on the central path, thus solutions of (6.12). If a limit for $\mu \to 0$ exists then this limit is a solution to problem (6.7) since (6.12) is continuous with respect to μ.

For linear SDPs all conditions except for the complementarity are satisfied after one full Newton step. Thus the only characteristic needed is "points on the central path". For nonlinear SDP we need different characteristics as no condition is satisfied trivially after one step.

We call solutions of (6.12) points on the central path. While we refer to solutions of (6.12) that violate $F(X) = 0$ as points on an infeasible central path. We write (infeasible) in brackets when it is unimportant whether $F(X) = 0$ is satisfied or not. Finally, we call the condition $\mathbf{S}_P(XS) = \mu I$ relaxed complementarity and solutions X, S relaxed complimentary points.

7 An IPM algorithm for nonlinear SDPs

In the last chapter we described the central path of our IPM. In this chapter we present and discuss an IPM algorithm that follows this path.

A first theoretical algorithm based on Newtons method converges towards the central path. After every Newton step the relaxation factor μ is reduced as much as possible without leaving the neighborhood of quadratic convergence. This type of algorithm is called short step algorithm and has been used to analyze convergence properties of convex programs (see e.g. [JS03]). The short step algorithm has only theoretical relevance, especially for the following two properties. For the short step algorithm for linear programs it is well known that it takes $6\sqrt{n}\log\frac{n\mu_0}{\varepsilon}$ steps to reach a precision of ε. Where n is the dimension of the variable and μ_0 a relaxation factor for the first iterate of the IPM such that the starting point is in the radius of quadratic convergence for Newtons algorithm. The second property is that the short step algorithm for linear programs converges towards a strictly complementary solution.

A second approach is the predictor-corrector approach. It consists of two alternating step types. The first one is a predictor step that is a step "parallel" to the central path towards $\mu = 0$. The second one consists of a series of centering steps, that converge towards the central path for a fixed μ. Such algorithms for linear programming and linear semidefinite programming have proven to be fast in practical applications.

As we focus on practical results we describe and analyze a predictor-corrector approach in this chapter.

When we talk about local quadratic convergence we focus on convergence towards the central path, introduced in the last chapter. These "centering steps" are very similar to the steps from the SSP/SLCP approach, except that they must maintain strict feasibility with respect to the cone constraints. We discuss their similarity in section 8.1. In the next chapter we present a major difference between these steps, that allows a superior convergence of the centering steps over the SSP steps.

7.1 A predictor-corrector algorithm

In this section we first present our predictor-corrector algorithm. Subsequently we discuss the separate elements of the algorithm and the necessary conditions that have to be satisfied.

Let (X_k, y_k, S_k) be the current iterate. Throughout the rest of this chapter we use the following abbreviations

$$\begin{aligned} F_k &:= F(X_k), \\ \mathrm{D}F_k &:= \mathrm{D}F(X_k), \\ \mathrm{g}_k &:= \mathrm{g}(X_k, y_k, S_k), \\ \mathcal{H}_k &:= \mathcal{H}(X_k, y_k, S_k), \\ \mathcal{E}_k &: X_k \mapsto \mathbf{S}_P(X_k S_k), \\ \mathcal{F}_k &: S_k \mapsto \mathbf{S}_P(X_k S_k). \end{aligned} \tag{7.1}$$

We again assume that $F \in \mathcal{C}^3$. As the objective function is only represented in the Lagrangian, the following approach can be used for a linear objective function $C \bullet X$ as well as a nonlinear objective function $C(X)$ with $C \in \mathcal{C}^3$.

The predictor-corrector IPM discussed in this thesis is presented in algorithm 4 and discussed in the following section.

7.1.1 Comments on our IPM algorithm

About line 1: We assume that a starting point is given. For linear programs phase one problems exist that find a feasible point to start with. For nonlinear programs it is difficult to define such a phase one problem. Nonlinear programs might have multiple local minima. A suitable starting point might be crucial to converge towards a solution that is acceptable for the given application. For the reduced order model problem from section 5.1 we wanted to find a perturbation that is close to zero. Thus a starting point was naturally given by that fact.

The quantity μ_0 is given by the starting point via $\mu_0 := \frac{1}{n} X_0 \bullet S_0$. It might be necessary to start with a few centering steps, if the starting point is not close enough to the feasible central path.

About line 2: To gain an approximation that is close to the optimal solution μ has to be smaller than a certain threshold. In this case the stopping criteria are the same as in section 5.7. For the "improving" step from section 5.7.2 the violation of the complementarity should now also be considered. This can be done by using the measurement

$$\frac{\|X \bullet S\|}{\|X\|\|S\|}. \tag{7.2}$$

We also show that $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ has to be invertible. While $\mathcal{H} \succeq 0$ would yield solutions for the IPM-step it would also lead to a similar convergence as the SSP.

About line 6: If the used step length control suggests a very short step it might be useful to return to step 4 and to calculate it for a $\mu > 0$. As we see in section 7.2 such problems are cheap to solve as we can use the same decompositions and all operations to be done are $\mathcal{O}(N^2)$. While $\mu = 0$ was a good choice for linear programs, it might not be for some non-linear programs.

About line 10: This distance is measured by the KKT conditions as presented in section 5.7.3. The only difference is the complementarity, where not $X \bullet S$ is measured, but

$$\frac{1}{\mu} \left(\|\mathbf{S}_P(XS) - \mu I\| \right). \tag{7.3}$$

About lines 11 to 13: In section 7.2 we discuss the complexity for solving such a system of equations. Here we just want to point out again that solving a system anew where only the right hand side changed is very cheap. Thus we can use the corrector to reduce the quadratic error for the complementarity with minimal effort. This corrector was introduced with Mehrotra's predictor-corrector approach [Me92, LMS92]. While for Mehrotra's algorithm one centering and corrector step was enough, we might need more steps in our problem depending on the nonlinear constraint $F(X) = 0$ and the distance to the feasible central path. Technically the corrector steps might only be needed when the violations $\|F(X)\|$ are very small and $\|\mathbf{S}_P(XS) - \mu I\|$ is large. In practical applications we suggest for most NLSDP to use them for all centering steps as they are "almost" for free and might save an additional centering step.

Note that in practical applications it might be efficient to introduce a line search for the result from the corrector step. When the violation of the nonlinear constraint is larger

Algorithm 4 A nonlinear semidefinite predictor-corrector IPM approach

let (x_0, y_0, s_0) be a given starting point, $k = 0$
while not [stopping criterion] **do**
  predictor step:
  solve

$$\begin{bmatrix} \mathcal{H}_k & \mathrm{D}F_k^* & -I \\ \mathrm{D}F_k & & \\ \mathcal{E}_k & & \mathcal{F}_k \end{bmatrix} \begin{bmatrix} \Delta X_k \\ \Delta y_k \\ \Delta S_k \end{bmatrix} = \begin{pmatrix} -\mathrm{g}_k \\ -F_k \\ -\mathbf{S}_P(X_k S_k) \end{pmatrix}$$

  calculate longest possible step $\lambda_{\max} := \min\{\lambda_X, \lambda_S\}$ with

$$X_k + \lambda \Delta X_k \succeq 0, \quad S_k + \lambda \Delta S_k \succeq 0$$

  [steplength control]: find suitable steplength $\lambda < c\lambda_{\max}$ (with $0.8 < c < 0.98$)
  predicted step:

$$X_{k_0} := X_k + \lambda \Delta X_k, \quad y_{k_0} := y_k + \lambda \Delta y_k, \quad S_{k_0} := S_k + \lambda \Delta S_k$$

  $\mu_{k+1} = \frac{1}{n} X_{k_0} \bullet S_{k_0}$, set $i = 0$
  centering/corrector step(s):
  while [too large distance to central path] **do**
    calculate (theoretical) centering step

$$\begin{bmatrix} \mathcal{H}_{k_i} & \mathrm{D}F_{k_i}^* & -I \\ \mathrm{D}F_{k_i} & & \\ \mathcal{E}_{k_i} & & \mathcal{F}_{k_i} \end{bmatrix} \begin{bmatrix} \Delta \tilde{X}_{k_i} \\ \Delta \tilde{y}_{k_i} \\ \Delta \tilde{S}_{k_i} \end{bmatrix} = \begin{pmatrix} -\mathrm{g}_{k_i} \\ -F_{k_i} \\ \mu_{k+1} I - \mathbf{S}_P(X_{k_i} S_{k_i}) \end{pmatrix}$$

    use centering step for corrector step

$$\begin{bmatrix} \mathcal{H}_{k_i} & \mathrm{D}F_{k_i}^* & -I \\ \mathrm{D}F_{k_i} & & \\ \mathcal{E}_{k_i} & & \mathcal{F}_{k_i} \end{bmatrix} \begin{bmatrix} \Delta X_{k_i} \\ \Delta y_{k_i} \\ \Delta S_{k_i} \end{bmatrix} = \begin{pmatrix} -\mathrm{g}_{k_i} \\ -F_{k_i} \\ \mu_{k+1} I - \mathbf{S}_P(X_{k_i} S_{k_i}) - \mathbf{S}_P(\Delta \tilde{X}_{k_i} \Delta \tilde{S}_{k_i}) \end{pmatrix}$$

    corrected step

$$X_{k_{i+1}} := X_{k_i} + \Delta X_{k_i}, \quad y_{k_{i+1}} := y_{k_i} + \Delta y_{k_i}, \quad S_{k_{i+1}} := S_{k_i} + \Delta S_{k_i}$$

    set $i = i + 1$
  end while
  $X_{k+1} := X_{k_i}$, $y_{k+1} := y_{k_i}$, $S_{k+1} := S_{k_i}$
  set $k = k + 1$
end while

than for the complementarity an additional line search for the centering step makes sense, especially when the nonlinear constraint has a strong curvature. This line search ensures that the corrector is calculated for a point that is possibly close to the step that will be used.

7.2 System solving

In algorithm 4 all systems have the same structure and only differ on the right hand side. In this section we discuss the complexity for solving such a system.

We consider a vector representation $c = vec(C)$, $x = vec(X)$ and $s = vec(S)$ introduced in section 2.1 to ease the reading of the following equations. For these vector the operators $\mathrm{D}F$ and $\mathcal{H}$ can be represented as matrices.

The systems from algorithm 4 in vector notation are of the form

$$\begin{pmatrix} \mathcal{H} & \mathrm{D}F^* & -I \\ \mathrm{D}F & 0 & 0 \\ \mathcal{E} & 0 & \mathcal{F} \end{pmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}. \tag{7.4}$$

It is easy to recalculate that the solution of equation (7.4) is

$$\begin{aligned} \Delta s &= \mathcal{F}^{-1}(r_3 - \mathcal{E}\Delta x) \\ \Delta x &= (\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}(r_1 - \mathrm{D}F^{\mathrm{T}}\Delta y + \mathcal{F}^{-1}r_3) \\ \Delta y &= -(\mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}\mathrm{D}F^{\mathrm{T}})^{-1}(r_2 - \mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}(r_1 + \mathcal{F}^{-1}r_3)). \end{aligned} \tag{7.5}$$

While matrix multiplications and sums are relatively cheap the main effort is inverting matrices and decomposing matrices combined with back solving respectively.

As we show in section 8.3.1 inverting $\mathcal{F}$ is done easily if we know the eigenvalues and eigenvectors of X which are very cheap to calculate. In section 7.4 we show that we get a descent direction if $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ has positive eigenvalues on the tangential cone of the nonlinear constraints. We give an equivalent barrier-formulation convergence condition in section 8.3.2. We also show that a strong condition, like $\mathcal{H} \succeq 0$ as in section 4 leads to slow convergence again. We further show that an approximation that satisfies such a condition is unnecessary for solving the system efficiently.

We show in the next chapter that quadratic convergence can be achieved even if we force $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \succ 0$. Having $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \succ 0$ would additionally allow us to use a Cholesky decomposition. In practical applications a rank one update (similar to the BFGS update) for $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ will be used. As neither $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ nor $\mathcal{H}$ is used directly to solve system (7.4) the cheapest way is to use an update that directly updates the inverse. It is still an open question how to define a suitable low rank update.

As inverting $\mathcal{F}$ is cheap and $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ is typically replaced with some form of a low rank update the main main effort is decomposing $\mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}\mathrm{D}F^{\mathrm{T}}$. For A with maximal rank and $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \succ 0$ a Cholesky decomposition can be used.

A first comparison between the nonlinear SDPs presented here and the equivalent linear SDPs shows that the complexity for solving the IPM systems has the same magnitude. The complexity of the algorithm's global convergence is much harder to compare, since the convergence of nonlinear problems depends strongly on the curvature of the nonlinear equations. This curvature can influence the convergence towards the central path also it limits the steplength of the predictor step as this depends on the violations of the equality constraints.

Note if we use the Hessian of the Lagrangian for $\mathcal{H}_{k_i}$ we have a normal Newton approach for the "centering steps". Since $X \in \mathcal{S}_+$ is invertible, $\mathcal{F}$ is invertible. If $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ is invertible and $\mathrm{D}F(x)$ has maximal rank then equation (7.5) has a solution. Thus under these conditions we get the quadratic convergence from Newton's approach.

7.3 Tangential step

In this section we motivate, that it is appropriate to follow a perturbed central path with a tangential step as predictor step.

For abbreviation we define

$$z := \begin{pmatrix} X \\ y \\ S \end{pmatrix}, \qquad \Phi(z) := \begin{pmatrix} \mathrm{g} \\ F \\ \mathbf{S}_P(XS) \end{pmatrix}. \tag{7.6}$$

For the tangential direction on the central path we consider dependence of z on μ through

$$\Phi(z) \stackrel{!}{=} \begin{pmatrix} 0 \\ 0 \\ \mu I \end{pmatrix}. \tag{7.7}$$

Note that the tangential step does not necessarily start on the central path, but only close to it. Thus it would not be a "tangential" step, but a mixture of a step towards the central path and parallel to it. This step is

$$\Delta z(\mu) := D\Phi(z)^{-1}\Big(\begin{pmatrix} 0 \\ 0 \\ \tilde{\mu} I \end{pmatrix} - \Phi(z) \Big) = D\Phi(z)^{-1}\Big(\begin{pmatrix} -\mathrm{g} \\ -F \\ \tilde{\mu} I - \mathbf{S}_P(XS) \end{pmatrix} \Big) \tag{7.8}$$

with $\tilde{\mu} = \mu + \Delta\mu$ and $0 > \Delta\mu > -\mu$.

This "tangential" direction is targeting at the central path. Using the target direction for a predictor step shows a slow convergence in linear programming examples. We describe in the following the tangential direction to a perturbed central path that leads to better results in practical applications.

Again we omit the iteration index and define for the k-th iteration

$$Q := \frac{1}{\hat{\mu}} \mathbf{S}_P(\hat{X}\hat{S}). \tag{7.9}$$

Let $\hat{\mu} := \mu^{(k)}$, $\hat{X} := X^{(k)}$ and $\hat{S} := S^{(k)}$ be fixed. For $\mu \to 0$ we get an alternative path to the optimal solution of the original NLSDP, given by the solutions z of

$$\Phi(z) \stackrel{!}{=} \begin{pmatrix} 0 \\ 0 \\ \mu Q(z, \mu) \end{pmatrix}. \tag{7.10}$$

A Newton step for (7.10) is given by

$$\Delta z(\mu) := D\Phi(z)^{-1}\Big(\begin{pmatrix} -\mathrm{g}(z) \\ -F(z) \\ \frac{\tilde{\mu}}{\hat{\mu}} \mathbf{S}_P(\hat{X}\hat{S}) - \mathbf{S}_P(XS) \end{pmatrix} \Big). \tag{7.11}$$

As we want to examine a tangential step we consider $\tilde{\mu} = 0$ and set $\hat{\mu} = \mu_k$, $\hat{X} = X_k$ and $\hat{S} = S_k$, which defines the tangential step

$$\Delta z_k := D\Phi(z_k)^{-1}\Big(\begin{pmatrix} -\mathrm{g}(z_k) \\ -F(z_k) \\ \frac{\Delta\mu_k}{\mu_k}\mathbf{S}_P(X_kS_k) \end{pmatrix} \Big). \tag{7.12}$$

We can reduce μ after each step by a fixed factor ν, thus set $\mu_{k+1} := \nu\mu_k$. It yet has to be shown how small ν can be such that we don't leave the area of quadratic convergence for the centering steps. In linear programming typically μ is reduced by a percentile of 95% meaning $\nu = 0.05$ which is currently used as a heuristic value for linear SDPs. For nonlinear SDPs we might have to reduce μ much less depending on the curvature of $F(X)$ as larger steps might lead to a strong violation of $F(X) = 0$. An implementation will show wether $\mu = 0$ is appropriate to predict a step.

For the convergence of the tangential step we simply need

$$\mu_{k+1}Q_k(z_k, \mu_k) = \nu\mu_k\frac{1}{\mu_k}\mathbf{S}_P(X_kS_k) = \nu^{k+1}\mu_0\mathbf{S}_P(X_kS_k) \overset{k\to\infty}{\longrightarrow} 0. \tag{7.13}$$

This is obvious, if $\|\mathbf{S}_P(X_kS_k)\|$ is bounded. If we use sufficiently many centering steps to converge towards the central path we keep $\|\mathbf{S}_P(X_kS_k)\|$ small, as we have $\mathbf{S}_P(XS) = \mu I$ on the central path. Thus, for an implementation the main concern is the size of μ as mentioned above.

Figure 7.1 illustrates the position of tangential and perturbed tangential direction. Both steps are illustrated for $\mu > 0$. The step that is using the direction that is tangential to the perturbed path, targets closer towards the optimal solution. This behavior is known for linear SDPs and is the reason why this predictor step was introduced.

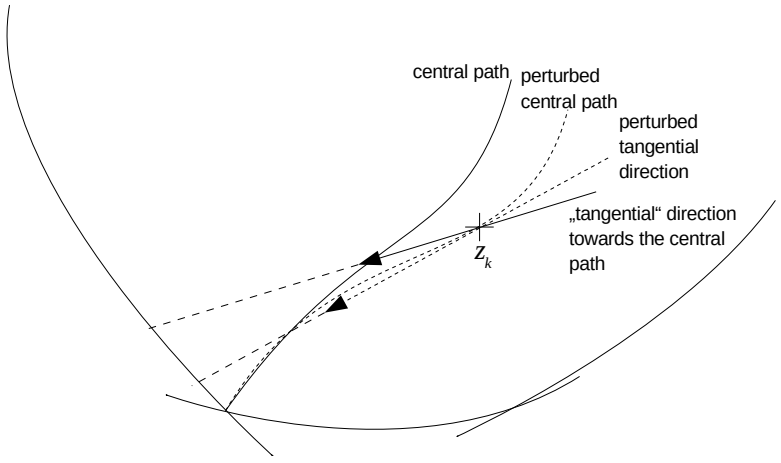

Figure 7.1: Comparison of tangential steps

7.4 Descent property

We consider a single iterate k and omit the index k for abbreviation. Let $\mathcal{H} = \mathcal{H}(X, y, S)$, $\mathrm{g} = \mathrm{g}(X, y, S)$, $F = F(X)$, $\mathrm{D}F = \mathrm{D}F(X)$, $\mathcal{E}\Delta X = \mathbf{S}_P(\Delta XS)$, and $\mathcal{F}\Delta S = \mathbf{S}_P(X\Delta S)$.

Note that $\mathcal{H}$, $\mathcal{E}$, $\mathcal{F}$ and DF are operators $\mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ and $\mathbb{R}^{n\times n} \to \mathbb{R}^m$ respectively. To ease the reading we omit the braces $[\cdot]$. This notation is very similar to the vector representation in section 7.2.

We consider a single step (see (7.4)) for (X, y, S)

$$\begin{pmatrix} \mathcal{H} & \mathrm{D}F^* & -I \\ \mathrm{D}F & 0 & 0 \\ \mathcal{E} & 0 & \mathcal{F} \end{pmatrix} \begin{bmatrix} \Delta X \\ \Delta y \\ \Delta S \end{bmatrix} = \begin{pmatrix} -\mathrm{g} \\ -F \\ \mu I - \mathbf{S}_P(XS) \end{pmatrix}. \tag{7.14}$$

For further abbreviation we define $R := \mu I - \mathbf{S}_P(XS)$ and write the solution of (7.14) as

$$\begin{aligned} \Delta S &= \mathcal{F}^{-1}(R - \mathcal{E}\Delta X), \\ \Delta X &= (\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}(-g - \mathrm{D}F^*\Delta y + \mathcal{F}^{-1}R), \\ \Delta y &= (\mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}\mathrm{D}F^*)^{-1}(F - \mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}(g - \mathcal{F}^{-1}R)). \end{aligned} \tag{7.15}$$

Note that

$$\mathcal{F}^{-1}R = \mathcal{F}^{-1}\mu I - \mathcal{F}^{-1}\mathbf{S}_P(XS) = \mu X^{-1} - S \tag{7.16}$$

and recall

$$\mathrm{g}(X, y, S) = C - \mathrm{D}F(X)^* y - S. \tag{7.17}$$

Thus (7.15) is equivalent to

$$\begin{aligned} \Delta S &= \mu X^{-1} - S - \mathcal{F}^{-1}\mathcal{E}\Delta X, \\ \Delta X &= (\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}(-c + \mathrm{D}F^* y - \mathrm{D}F^*\Delta y + \mu X^{-1}), \\ \Delta y &= (\mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}\mathrm{D}F^*)^{-1}(F - \mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}(c - \mathrm{D}F^* y - \mu X^{-1})). \end{aligned} \tag{7.18}$$

The following theorem gives a necessary condition to yield a descent direction, when μ is reduced.

Theorem 7.4.1. *Let X, S be a feasible relaxed complementary pair, thus $F(X) = 0$, $X, S \in \mathcal{S}_+$ with $\mathbf{S}_P(XS) = \bar{\mu}I > 0$. Let c be not orthogonal to the nullspace of* DF *and* DF *have maximal rank. We consider the solution ΔX of (7.14), with $\mu = 0$. If $(\tilde{\mathcal{H}} + \mathcal{F}^{-1}\mathcal{E}) \succ 0$ then the solution ΔX is a descent direction for the objective function, thus $C \bullet \Delta X < 0$.*

Proof. From the assumptions we have X, S with

$$\begin{aligned} F(X) &= 0, \\ \mathbf{S}_P(XS) &= \bar{\mu}I, \\ S &\succ 0, \\ X &\succ 0. \end{aligned} \tag{7.19}$$

We show that ΔX from (7.14) with $\mu = 0$ is a descent direction by showing $C \bullet X < 0$.

We examine

$$\begin{aligned} \Delta y &= (\mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}\mathrm{D}F^*)^{-1}(-\mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}(g - \mathcal{F}^{-1}(-\bar{\mu}I)), \\ C \bullet \Delta X &= tr(C^{\mathrm{T}}(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}(-g - \mathrm{D}F^*\Delta y + \mathcal{F}^{-1}(-\bar{\mu}I))) \end{aligned} \tag{7.20}$$

As $\mathbf{S}_P$ is linear and $\mathbf{S}_P(XS) = \bar{\mu}I$ it follows $\mathcal{F}^{-1}(-\bar{\mu}I) = -S$.

For abbreviation we define $M := (\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})$.

$$\begin{aligned} C \bullet \Delta X \; &= tr(- C^{\mathrm{T}} M^{-1}(I - \mathrm{D}F^*(\mathrm{D}FM^{-1}\mathrm{D}F^*)^{-1}\mathrm{D}FM^{-1})C \\ &\quad + C^{\mathrm{T}} M^{-1} \left(I - \mathrm{D}F^*(\mathrm{D}FM^{-1}\mathrm{D}F^*)^{-1}\mathrm{D}FM^{-1}\right) \underbrace{(S + \mathcal{F}^{-1}(-\bar{\mu}I)}_{=0 \text{ (see (7.16))}} \\ &\quad + C^{\mathrm{T}} M^{-1} \underbrace{(\mathrm{D}F^* y - \mathrm{D}F^*(\mathrm{D}FM^{-1}\mathrm{D}F^*)^{-1}(\mathrm{D}FM^{-1}\mathrm{D}F^*)y))}_{=0} \end{aligned} \tag{7.21}$$

The matrix M is by definition (symmetric) positive definite. Thus it has a (symmetric) positive definite root $M^{\frac{1}{2}}$.

Let Π_R be the orthogonal projection onto the range $R(M^{-\frac{1}{2}}\mathrm{D}F^*)$ and Π_N be the orthogonal projection onto the nullspace $N(\mathrm{D}FM^{-\frac{1}{2}})$. For Π_R we have

$$\Pi_R = M^{-\frac{1}{2}}\mathrm{D}F^*(\mathrm{D}FM^{-1}\mathrm{D}F^*)^{-1}\mathrm{D}FM^{-\frac{1}{2}}. \tag{7.22}$$

Thus we can write (7.21) as

$$C \bullet \Delta X \; = tr(-C^{\mathrm{T}} M^{-\frac{1}{2}} \underbrace{(I - \Pi_R)}_{=\Pi_N} M^{-\frac{1}{2}} C) \leq 0. \tag{7.23}$$

We show by contradiction $C \bullet X < 0$. We now assume

$$tr(-C^{\mathrm{T}} M^{-\frac{1}{2}} \Pi_N M^{-\frac{1}{2}} C) = 0. \tag{7.24}$$

It follows[1] $\Pi_N M^{-\frac{1}{2}} C = 0$ which is equivalent to $M^{-\frac{1}{2}} C$ being orthogonal to the nullspace of $\mathrm{D}FM^{-\frac{1}{2}}$. Thus we have

$$\begin{aligned} &\forall \Xi \in \mathbb{R}^{n \times n}, \text{ with } \mathrm{D}FM^{-\frac{1}{2}}[\Xi] = 0 \Rightarrow tr(C^{\mathrm{T}} M^{-\frac{1}{2}} \Xi) = 0 \\ \Leftrightarrow\; &\forall \Xi \in \mathbb{R}^{n \times n}, \text{ with } DF[\Xi] = 0 \Rightarrow tr(C^{\mathrm{T}} \Xi) = 0. \end{aligned} \tag{7.25}$$

This is a contradiction to the assumption that C is not orthogonal to the nullspace of $\mathrm{D}F$ thus we have

$$tr(-C^{\mathrm{T}} M^{-\frac{1}{2}} \Pi_N M^{-\frac{1}{2}} C) < 0. \tag{7.26}$$

□

Note since (7.18) is continuous in X and S, an area around feasible complementary points exists for that ΔX is still a descent direction for the objective function.

Theorem 7.4.1 has one major assumptions $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \succ 0$.

It is well known that for the family of MZ-symmetrizations $\mathcal{F}^{-1}\mathcal{E}$ satisfies

$$\langle \mathcal{F}^{-1}\mathcal{E}(X), X \rangle > 0, \quad \text{for } X \neq 0, \tag{7.27}$$

but it is not necessarily symmetric. In the next section we show that the AHO symmetrization satisfies $\mathcal{F}^{-1}\mathcal{E} \in \mathcal{S}^n_{++}$ on the central path.

Using a positive semidefinite approximation to $\mathcal{H}$ would guarantee $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \succ 0$. This assumption led to the linear convergence for the SSP. For the IPM presented here the centering steps suffer from a similar problem for $\mu \to 0$ as we show in section 8.1.

[1] For $A \in \mathcal{S}^n_+$, $x \in \mathbb{R}^n$ are equivalent $A^2 x = 0 \Leftrightarrow Ax = 0 \Leftrightarrow x^{\mathrm{T}} A^{\mathrm{T}} Ax = 0$. The first equation follows from the singular value decomposition $A = QDQ^{\mathrm{T}}$ and $A^2 = QDQ^{\mathrm{T}}QDQ^{\mathrm{T}} = QD^2Q^{\mathrm{T}}$.

In section 8.3.2 we show that under weak assumptions on the central path $\mathcal{H}+\mathcal{F}^{-1}\mathcal{E}$ is positive definite on the tangential cone[2]. We further show that there exists a convexified approximation of $\mathcal{H}+\mathcal{F}^{-1}\mathcal{E}$ that we can use to gain quadratic convergence. It is an open question wether there is a cheap low rank update for $\mathcal{H}+\mathcal{F}^{-1}\mathcal{E}$ that leads to superlinear convergence.

7.5 The AHO symmetrization

The AHO symmetrization is the symmetrization $\mathbf{S}_P(XS)$ with $P = I$

$$\mathbf{S}_I(XS) = \frac{1}{2}(XS + SX) \tag{7.28}$$

For a fixed matrix X we define

$$L_X(S) = XS + SX. \tag{7.29}$$

For the following propositions the matrices X are assumed to be symmetric. In our application these matrices are positive semidefinite. The operator $L_X(S)$ is the AHO symmetrization except for the constant $\frac{1}{2}$, i.e. it is a Jordan multiplication over symmetric matrices. Some results from the following propositions can be shown on a more general Jordan multiplication basis. These results will be discussed in sections 8.2.1 and 8.3.1.

Proposition 7.5.1. *If $A, B \in \mathcal{S}^n$ can be simultaneously diagonalized, then L_A and L_B commute.*

Proof. Let $A = QD_AQ^{\mathrm{T}}$ and QD_BQ^{T} be the eigenvalue decompositions of A and B. Then A and B commute, as the diagonal matrices D_A and D_B commute

$$AB = QD_AQ^{\mathrm{T}}QD_BQ^{\mathrm{T}} = QD_AD_BQ^{\mathrm{T}} = QD_BD_AQ^{\mathrm{T}} = QD_BQ^{\mathrm{T}}QD_AQ^{\mathrm{T}} = BA. \tag{7.30}$$

Thus we have

$$\begin{aligned} L_A(L_B(X)) &= L_A(BX + XB) = ABX + BXA + AXB + XBA = \\ &\quad BAX + AXB + BXA + XAB = L_B(AX + XA) \\ &= L_B(L_A(X)). \end{aligned} \tag{7.31}$$

□

Proposition 7.5.2. *If $A, B \in \mathcal{S}^n_+$ can be simultaneously diagonalized, then L_A and L_B^{-1} commute.*

Proof. We have

$$\begin{aligned} L_A(L_B^{-1}(X)) &= L_B^{-1}(L_B(L_A(L_B^{-1}(X)))) \\ &= L_B^{-1}(L_A(L_B(L_B^{-1}(X)))) = L_B^{-1}(L_A(X)). \end{aligned} \tag{7.32}$$

□

Proposition 7.5.3. *If X is symmetric, L_X is symmetric.*

[2]The tangential cone will be introduced in section 8.3.2, roughly it is the set of directions ΔX with $\mathrm{D}F[\Delta X] = 0$

Proof. We have

$$\begin{aligned}\langle L_X(A), B\rangle &= \langle XA, B\rangle + \langle AX, B\rangle \\ &= Tr((XA)^{\mathrm{T}}B) + Tr((AX)^{\mathrm{T}}B) \\ &= Tr(A^{\mathrm{T}}XB) + Tr(XA^{\mathrm{T}}B) \\ &= Tr(A^{\mathrm{T}}(XB)) + Tr(A^{\mathrm{T}}(BX)) \\ &= \langle A, XB\rangle + \langle A, BX\rangle = \langle A, L_X(B)\rangle.\end{aligned} \tag{7.33}$$

□

Corollary 7.5.4. *$L_X^{-1}L_S$ is symmetric, if X and S are symmetric and can be simultaneously diagonalized.*

Proof. This follows directly from L_X^{-1} and L_S commuting and each being symmetric:

$$L_X^{-1}L_S = L_S L_X^{-1} = L_S^* L_X^{-*} = (L_X^{-1}L_S)^*. \tag{7.34}$$

□

Proposition 7.5.5. *Relaxed complementary matrices X and S can be simultaneously diagonalized.*

Proof. We have

$$\begin{aligned}&\mathbf{S}_P(XS) = \mu I \\ \Leftrightarrow &XS = \mu I\end{aligned} \tag{7.35}$$

and thus for $X = QD_XQ^{\mathrm{T}}$

$$\begin{aligned}&XS = \mu I \\ \Leftrightarrow\ &D_XQ^{\mathrm{T}}SQ = \mu I \\ \Leftrightarrow\ &Q^{\mathrm{T}}SQ = \mu D_X^{-1}.\end{aligned} \tag{7.36}$$

Let D_S be the diagonal matrix $D_S := Q^{\mathrm{T}}SQ$, then we have $S = QD_SQ^{\mathrm{T}}$. □

Finally we have the necessary condition $\mathcal{F}^{-1}\mathcal{E} = (\mathcal{F}^{-1}\mathcal{E})^{\mathrm{T}}$ for relaxed complementary points X, S, for the AHO symmetrization.

Corollary 7.5.6. *For relaxed complementary points X, S the operator $\mathcal{F}^{-1}\mathcal{E}$ is symmetric, when using the AHO symmetrization.*

Proof. On the central path X and S can be simultaneously diagonalized. Thus the result follows from (7.5.4). □

In consequence we now know that for relaxed complementary points $\mathcal{F}^{-1}\mathcal{E} + \mathcal{H}$ is symmetric which is a major condition of theorem 7.4.1.

8 Superiority of the IPM over SSP method

In chapter 4 it can be seen that the condition $\mathcal{H} \succeq 0$ is too strong to guarantee quadratic convergence for the SSP case. In this chapter we analyze wether the same problem holds for the IPM case. Finally, we present a weaker condition that leads to quadratic convergence.

We start in the next section by showing the similarities of the IPM's centering steps to the SSP steps. We have seen in section 7.2 and 7.4 that the condition $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \succ 0$ is sufficient in order to solve the occurring systems and to have a descent direction. We show in section 8.2.5 for the quadratic cone there exists an approximation to $\mathcal{H}$ such that $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \succ 0$ and that is identical to $\mathcal{H}$ on the central path in all directions of the linearized feasible set. We illustrate this result on the counter-example from chapter 4 for which the SSP had linear convergence.

Finally, we generalize the results for the cone of semidefinite matrices.

8.1 Similarities to the SSP method

Again we consider a nonlinear semidefinite program (NLSP) of the form

see (6.1). $$\min \left\{ C \bullet X \;\middle|\; F(X) = 0,\ X \in \mathcal{S}^n_+ \right\}.$$

For a given iterate $\tilde{X}$ we define the SSP subproblem

$$\min\{\ C \bullet \Delta X + \frac{1}{2}\mathcal{H}[\Delta X, \Delta X] \mid \mathrm{D}F(\tilde{X})[\Delta X] = -F(\tilde{X}),\ \tilde{X} + \Delta X \in \mathcal{S}^n_+\ \}, \tag{8.1}$$

where $\mathcal{H} := \mathcal{H}(\tilde{X}, \tilde{y}, \tilde{S})$ is again the Hessian of the Lagrangian of (6.1).

We further define $X := \Delta X + \tilde{X}$ and get a program that is equivalent to (8.1)

$$\min\{\ C \bullet X - \mathcal{H}[\tilde{X}, X] + \frac{1}{2}\mathcal{H}[X, X] \mid F(\tilde{X}) + \mathrm{D}F(\tilde{X})[X - \tilde{X}] = 0,\ X \in \mathcal{S}^n_+\ \}. \tag{8.2}$$

As we are looking at a fixed subproblem we abbreviate $F := F(\tilde{X})$ and $\mathrm{D}F := \mathrm{D}F(\tilde{X})$.

The Lagrangian for problem (8.2) is

$$\begin{aligned} L^{SSP}(X, y, S) &:= C \bullet X - \mathcal{H}[\tilde{X}, X] + \frac{1}{2}\mathcal{H}[X, X] + (F + \mathrm{D}F[X - \tilde{X}])^{\mathrm{T}} y + X \bullet S, \\ g^{SSP}(X, y, S) &:= D_X L^{SSP}(X, y, S) = C + \mathcal{H}[\Delta X] + \mathrm{D}F^* y + S, \end{aligned} \tag{8.3}$$

and the optimality conditions are

$$\begin{aligned} C + \mathcal{H}[\Delta X] + \mathrm{D}F^* y + S &= 0 \\ \mathrm{D}F[\Delta X] &= -F \\ \frac{1}{2}(XS + SX) &= 0 \\ X, S &\in \mathcal{S}^n_+. \end{aligned} \tag{8.4}$$

As shown in chapter 4 the SSP-method may show only linear convergence when a positive semidefinite approximation of $\mathcal{H}$ is used.

We now want to compare this to a centering step of an IPM for (6.1). We use the IPM presented in chapter 6 and expand (7.14) to

$$\begin{aligned} \mathcal{H}[\Delta X] + \mathrm{D}F^*\Delta y + \Delta S &= -C - \mathrm{D}F^*\tilde{y} - \tilde{S}, \\ \mathrm{D}F[\Delta X] &= -F, \\ \mathcal{F}\Delta X + \mathcal{E}\Delta S &= \mu I - \mathbf{S}_P(\tilde{X}\tilde{S}). \end{aligned} \tag{8.5}$$

Apart from equation (8.5) the iterates generated by our IPM satisfy the conic constraints. Thus we have

$$\begin{aligned} C + \mathcal{H}[\Delta X] + \mathrm{D}F^*y + S &= 0, \\ \mathrm{D}F[\Delta X] &= -F, \\ \mathbf{S}_P(XS) &= \mu I + \mathbf{S}_P(\Delta X \Delta S), \\ X, S &\in \mathcal{S}^n_{++} \subset \mathcal{S}^n_+. \end{aligned} \tag{8.6}$$

The difference between (8.4) and (8.6) is the term $\mu I + \mathbf{S}_P(\Delta X \Delta S)$ in the last equation. Due to the linearization the quadratic term $\mathbf{S}_P(\Delta X \Delta S)$ is not included and the relaxation term μI is added. Ignoring the quadratic term, for $\mu \to 0$ the centering steps converge towards the steps of the SSP method that can be linearly convergent for any bounded choice of a positive definite $\mathcal{H}$ approximation.

In the following we show for the example from chapter 4, that the IPM presented here, cannot have more than linear convergence if a positive semidefinite approximation of $\mathcal{H}$ is used.

Linear convergence for any positive semidefinite approximation of $\mathcal{H}$

To abbreviate the notation we define

$$z := \begin{pmatrix} x \\ y \\ s \end{pmatrix}, \qquad \Phi(z) := \begin{pmatrix} \mathrm{g} \\ F \\ 2\mathbf{S}_P(XS) - \mu I \end{pmatrix}. \tag{8.7}$$

We consider the Newton algorithm for $\Phi(\,\cdot\,)$ with the solution z^* that fullfills $\Phi(z^*) = 0$. For a given iterate z_k we define

$$\Delta z := -D\Phi(z_k)^{-1}\Phi(z_k). \tag{8.8}$$

The Newton step is

$$z_{k+1} := z_k + \Delta z. \tag{8.9}$$

For the Newton algorithm local quadratic convergence is known.

When $D\Phi(z_k)$ is now replaced with an approximation we can still achieve quadratic or at least super linear convergence. One well known criterion, that is equivalent to superlinear convergence (see e.g [JS03]) for a given approximation Ψ_k of $D\Phi(z_k)$, is

$$\begin{aligned} &\lim_k \frac{\|(\Psi_k - D\Phi(z_k))\Delta z\|}{\|\Delta z\|} = 0 \\ \Leftrightarrow\ &\lim_k \frac{\|z_{k+1} - z^*\|}{\|z^k - z^*\|} = 0. \end{aligned} \tag{8.10}$$

Let $\tilde{\mathcal{H}}$ be any bounded positive semidefinite approximation of $\mathcal{H}$. $\tilde{\mathcal{H}}$ is a submatrix of Ψ_k. For the difference between the Hessian of the Lagrangian $\mathcal{H}$ and an approximation $\tilde{\mathcal{H}}$ we have

$$\tilde{\mathcal{H}} - \mathcal{H} \succeq \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}. \tag{8.11}$$

The set of constraints $\mathcal{Z}$ is

$$\mathcal{Z} := \{\ x = (x_0, x_1, x_2) \in \mathbb{R}^3 \mid x_0 = 1,\ x \in \mathcal{Q}\ \} \tag{8.12}$$

and all steps within that cone are of the form $(0, *, *)$.

It follows for those steps Δx_k

$$\lim_{k\to\infty, \Delta x_k \in \mathcal{Z}} \frac{\|(\tilde{\mathcal{H}} - \mathcal{H})\Delta x_k\|}{\|\Delta x_k\|} \geq 2. \tag{8.13}$$

Since $\mathcal{H}$ is a submatrix of Ψ_k that applies to Δx we have

$$2 \leq \frac{\|(\tilde{\mathcal{H}} - \mathcal{H})\Delta x_k\|}{\|\Delta x_k\|} \leq \frac{\|(\Psi_k - D\Phi(z_k))\Delta z\|}{\|\Delta z\|}. \tag{8.14}$$

It follows that the IPM cannot converge faster than linearly for this example if a positive semidefinite approximation is used.

But while the SSP method needs to have a positive semidefinite $\mathcal{H}$ to have appropriately fast solvable subproblems, we show in the next section that we can have a much weaker condition for $\mathcal{H}$ when using an IPM. This weaker condition still guarantees invertibility and the descent property.

8.2 Eigenvalues of "$\mathcal{F}^{-1}\mathcal{E}$" for the Lorentz cone $\mathcal{Q}$

8.2.1 Jordan algebras

In the following we use Jordan algebras to analyze certain properties of the quadratic cone and the semidefinite cone.

A Jordan algebra is an algebra over a n-dimensional vector space V with a multiplication $\cdot \times \cdot$ that maps $V^2 \to V : (x, y) \to x \times y$. This mapping is bilinear, thus for $x, y, z \in V$ and $a, b \in \mathbb{R}$ it satisfies:

$$\begin{aligned} (ax + by) \times z &= a(x \times z) + b(y \times z), \\ x \times (ay + bz) &= a(x \times y) + b(x \times z). \end{aligned} \tag{8.15}$$

Additionally this multiplication is commutative

$$x \times y = y \times x \tag{8.16}$$

and satisfies

$$x \times (x^2 \times y) = x^2 \times (x \times y) \quad \text{with} \quad x^2 = x \times x. \tag{8.17}$$

Note that a Jordan algebra is not necessarily associative.

The reason why we focus on Jordan algebras is that they allow us to analyze the symmetrization in our IPM on a more general scope. The cones discussed in this paper $\mathbb{R}^n$, $\mathcal{Q}$, and $\mathcal{S}^n_+$ are all sets of squares over specific Jordan algebras.

For a first example we consider the Jordan algebra over the vector space $\mathbb{R}^n$ defined by the multiplication

$$x \times y = \begin{pmatrix} x_0 y_0 \\ \vdots \\ x_n y_n \end{pmatrix}. \tag{8.18}$$

The unit element of this algebra is the vector

$$I_{\mathbb{R}_+} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{8.19}$$

It is easy to see that the cone of squares over this algebra is $\mathbb{R}^n_+$.

For a more interesting Jordan algebra over $\mathbb{R}^{n+1}$ let x be composed of $x_0 \in \mathbb{R}$ and $\bar{x} \in \mathbb{R}^n$ such that

$$x = \begin{pmatrix} x_0 \\ \bar{x} \end{pmatrix}. \tag{8.20}$$

The multiplication $\cdot \times \cdot$ is defined by

$$a \times b := \begin{pmatrix} a^{\mathrm{T}} b \\ a_0 \bar{b} + b_0 \bar{a} \end{pmatrix} \tag{8.21}$$

and the identity I_Q is given by

$$I_Q := \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{8.22}$$

In the following paragraphs we cite some results that we need later, for proofs we refer to [WSV00].

The Quadratic cone $\mathcal{Q}$ is the set of squares of this Jordan algebra.

Proposition 8.2.1. *The Jordan-product of two vectors $a, b \in \mathcal{Q}$ is again in $\mathcal{Q}$.*

A final important Jordan algebra is over the vector space of symmetric matrices $\mathcal{S}^n$ defined by the multiplication

$$X \times S = \frac{1}{2}(XS + SX) \tag{8.23}$$

with the matrix identity

$$I_{\mathcal{S}^n_+} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \tag{8.24}$$

as identity element.

The cone of positive semidefinite matrices $\mathcal{S}^n_+$ is the set of squares of this Jordan algebra.

Proposition 8.2.2. *The Jordan-product of two positive semidefinite matrices is a semidefinite matrice.*

8.2.2 Jordan algebra for the Lorentz cone $\mathcal{Q}$

We consider the Jordan algebra on $\mathbb{R}^{n+1}$, where we partition $\theta \in \mathbb{R}^{n+1}$ into $\theta = \begin{pmatrix} \theta_0 \\ \bar{\theta} \end{pmatrix}$ with $\theta_0 \in \mathbb{R}$ and $\bar{\theta} \in \mathbb{R}^n$. We then have the multiplication with its unit element

$$x \times y = \begin{pmatrix} x^{\mathrm{T}} y \\ x_0 \bar{y} + y_0 \bar{x} \end{pmatrix} \quad \text{and} \quad I_Q = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{8.25}$$

The Lorentz cone

$$\mathcal{Q} := \{\ a \mid a_0 \geq \|\bar{a}\| \ \} \tag{8.26}$$

is the cone that contains the squares of this algebra.

In the following we always assume $a \neq 0$.

Proposition 8.2.3. *Let $x, s \in \mathcal{Q}$ from $\langle x, s \rangle = 0$ follows $x \times s = 0$.*

Proof. We have

$$x \times s = \begin{pmatrix} x^{\mathrm{T}} s \\ x_0 \bar{s} + s_0 \bar{x} \end{pmatrix} = \begin{pmatrix} 0 \\ x_0 \bar{s} + s_0 \bar{x} \end{pmatrix}. \tag{8.27}$$

From proposition 8.2.1 we know that $x \times s \in \mathcal{Q}$. It follows $0 \geq \|x_0 \bar{s} + s_0 \bar{x}\|_2$ thus $x_0 \bar{s} + s_0 \bar{x} = 0$. □

Proposition 8.2.4. *For interior points $a \in \mathcal{Q}^\circ$ of the Lorentz cone there exists an element $a^\dagger$ that satisfies*

$$a^\dagger \times a = I_{\mathcal{Q}} \qquad \text{and} \qquad a \times a^\dagger = I_{\mathcal{Q}}. \tag{8.28}$$

Proof. Recalculate

$$a^\dagger := \frac{1}{a_0^2 - \|\bar{a}\|^2} \begin{pmatrix} a_0 \\ -a_1 \\ \vdots \\ -a_n \end{pmatrix}. \tag{8.29}$$

□

For the following propositions we define the operator

$$T_a[\theta] := a \times \theta. \tag{8.30}$$

Proposition 8.2.5. *The linear operator T_a has the eigenvalues a_0, $a_0 + \|\bar{a}\|$ and $a_0 - \|\bar{a}\|$.*

Proof. It is easy to calculate that the vectors

$$\begin{pmatrix} \|\bar{a}\| \\ \bar{a} \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} -\|\bar{a}\| \\ \bar{a} \end{pmatrix} \tag{8.31}$$

are eigenvectors of T_a with eigenvalues $a_0 + \|\bar{a}\|$ and $a_0 - \|\bar{a}\|$ respectively.

All vectors orthogonal to this have the eigenvalue a_0. Given the following basis this is easy to see:

$$\left\{ \begin{pmatrix} 0 \\ -a_2 \\ a_1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -a_3 \\ 0 \\ a_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ -a_{n-1} \\ 0 \\ \vdots \\ 0 \\ a_1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -a_n \\ 0 \\ \vdots \\ \vdots \\ 0 \\ a_1 \end{pmatrix} \right\}. \tag{8.32}$$

(8.32) is a basis as long as $a_1 \neq 0$. If $a_1 = 0$ one can build a similar basis as long as any $a_i \neq 0$. If all $a_i = 0$ then the operator is the scalar multiplication with a_0. □

Corollary 8.2.6. *It follows that $T_{a^\dagger}^{-1}$ has the eigenvalues $\frac{a_0^2 - \|\bar{a}\|^2}{a_0}$, $a_0 + \|\bar{a}\|$ and $\frac{a_0^2 - \|\bar{a}\|^2}{a_0 + \|\bar{a}\|}$.*

Corollary 8.2.7. *The eigenvalues of $T_{a^\dagger}^{-1} T_a$ are $a_0^2 - \|\bar{a}\|^2$, $(a_0 + \|\bar{a}\|)^2$ and $(a_0 - \|\bar{a}\|)^2$.*

Proof. Please note that $T_{a^\dagger}^{-1}$ and T_a have the same eigenspaces. This follows from

$$\frac{1}{a_0^2 - \|\bar{a}\|^2} \begin{pmatrix} -\|\bar{a}\| \\ \bar{a} \end{pmatrix} = - \begin{pmatrix} \|\bar{a}^\dagger\| \\ \bar{a}^\dagger \end{pmatrix} \quad \text{and} \quad \frac{1}{a_0^2 - \|\bar{a}\|^2} \begin{pmatrix} \|\bar{a}\| \\ \bar{a} \end{pmatrix} = - \begin{pmatrix} -\|\bar{a}^\dagger\| \\ \bar{a}^\dagger \end{pmatrix}. \tag{8.33}$$

Thus the result can easily be recalculated. □

8.2.3 A conical program over $\mathcal{Q}$

We consider the problem

$$\min\{\ c(x) \mid F(x) = 0, \quad x \in \mathcal{Q}\ \}. \tag{8.34}$$

It has the Lagrangian

$$\begin{aligned} \mathcal{L}(s, x, y) &:= c(x) - F(x)^{\mathrm{T}} y - x^{\mathrm{T}} s, \\ \mathrm{g}(s, x, y) &:= \mathrm{D}_x \mathcal{L}(s, x, y) = \mathrm{D}c(x) - \mathrm{D}_x(F(x)^{\mathrm{T}} y) - s, \\ \mathcal{H}(s, x, y) &:= \mathrm{D}_x^2 \mathcal{L}(s, x, y) = \mathrm{D}^2 c(x) - \mathrm{D}_x^2(F(x)^{\mathrm{T}} y). \end{aligned} \tag{8.35}$$

Using proposition 8.2.3 it is easy to see that

$$\begin{aligned} \mathrm{g}(s, x, y) = \mathrm{D}c(x) + \mathrm{D}_x(F(x)^{\mathrm{T}} y) - s &= 0, \\ F(x) &= 0, \\ x \times s &= 0, \\ x, s &\in \mathcal{Q} \end{aligned} \tag{8.36}$$

is equivalent to the KKT conditions.

We use these conditions to define a central path

$$\begin{aligned} \mathrm{D}c(x) + \mathrm{D}_x(F(x)^{\mathrm{T}} y) - s &= 0, \\ F(x) &= 0, \\ x \times s &= \mu I_Q, \\ x, s &\in \mathcal{Q}. \end{aligned} \tag{8.37}$$

For a step towards this central path we focus on the linearization and omit again the parameters for abbreviation

$$\begin{pmatrix} \mathcal{H} & -\mathrm{D}F^* & -I \\ \mathrm{D}F & & \\ \mathcal{E} & & \mathcal{F} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} -g \\ -F \\ \mu I_{\mathcal{Q}} - (x \times s) \end{pmatrix} \tag{8.38}$$

where $\mathcal{E}$ and $\mathcal{F}$ being the matrices for the operations

$$\mathcal{E}\theta := s \times \theta \qquad \mathcal{F}\theta := x \times \theta. \tag{8.39}$$

Comparing (8.38) with (7.14) shows that they are identical except for dimension and different multiplications $\mathcal{E}\Delta x$, $\mathcal{F}\Delta s$. We show in this section that both have the same properties

$$\forall z \in N(\mathrm{D}F) \Rightarrow z^{\mathrm{T}}\mathcal{F}^{-1}\mathcal{E}z \geq 0 \tag{8.40}$$

for relaxed complementary points (x, y, s). $N(\mathrm{D}F)$ is the null space of $\mathrm{D}F$.

Please recall that all examinations about the SDP-IPM were done purely symbolical. Thus all results of the PSD cone that did not use specific properties of the PSD cone transfer for IPMs over the quadratic cone. In particular the descent property if we use an approximation of $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ that is again positive definite on the linearized equality constraints. This guarantees invertibility of this approximation that we need for the centering step.

As we had the same system to solve as in section 7.4 the solution $(\Delta s^{\mathrm{T}}, \Delta x^{\mathrm{T}}, \Delta y^{\mathrm{T}})^{\mathrm{T}}$ is given by

$$\begin{aligned} \Delta s &:= \mathcal{F}^{-1}(\mu I - \mathcal{E}\Delta x), \\ \Delta x &:= (\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}(-g + \mathcal{F}^{-1}\mu I - \mathrm{D}F^{\mathrm{T}}\Delta y), \\ \Delta y &:= (\mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}\mathrm{D}F^{\mathrm{T}})^{-1}(F + \mathrm{D}F(\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})^{-1}(-g + \mathcal{F}^{-1}\mu I). \end{aligned} \tag{8.41}$$

Analogously to the SDP descent property from section 7.4 we need

$$\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \succ 0 \tag{8.42}$$

for feasible relaxed complementary points.

In the following we examine $\mathcal{F}^{-1}\mathcal{E}$ to see which eigenvalues of $\mathcal{H}$ can be negative, so that $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ is still positive definite.

On relaxed complementary points we have

$$x \times s = \mu I \quad \Leftrightarrow \quad x = \mu s^{\dagger}. \tag{8.43}$$

Recall the eigenvalues of $\mathcal{F}^{-1}\mathcal{E}$ given by corollary 8.2.7 in the last section.

Corollary 8.2.8. *Necessary conditions for $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \succ 0$ are*

$$\begin{aligned} \begin{pmatrix} \|\bar{s}\| \\ \bar{s} \end{pmatrix}^{\mathrm{T}} \mathcal{H} \begin{pmatrix} \|\bar{s}\| \\ \bar{s} \end{pmatrix} &> -\frac{1}{\mu}(s_0 + \|\bar{s}\|)^2, \\ \begin{pmatrix} \|\bar{x}\| \\ \bar{x} \end{pmatrix}^{\mathrm{T}} \mathcal{H} \begin{pmatrix} \|\bar{x}\| \\ \bar{x} \end{pmatrix} &> -\frac{1}{\mu}(s_0 - \|\bar{s}\|)^2, \\ \forall z, z\perp \begin{pmatrix} \|\bar{s}\| \\ \bar{s} \end{pmatrix}, z\perp \begin{pmatrix} \|\bar{x}\| \\ \bar{x} \end{pmatrix}, \quad z^{\mathrm{T}}\mathcal{H}z &> -\frac{1}{\mu}(s_0^2 - \|\bar{s}\|^2). \end{aligned} \tag{8.44}$$

Proof. With corollary 8.2.7 follows directly that the eigenvalues of $\mathcal{F}^{-1}\mathcal{E}$ are

$$\frac{1}{\mu}(s_0^2 - \|\bar{s}\|^2), \quad \frac{1}{\mu}(s_0 + \|\bar{s}\|)^2, \quad \text{and} \quad \frac{1}{\mu}(s_0 - \|\bar{s}\|)^2. \tag{8.45}$$

□

8.2.4 Limits of $\mathcal{F}^{-1}\mathcal{E}$'s eigenvalues towards (x^*, y^*, s^*)

In this section we assume that x^* and s^* lie on the boundary of the cone. For the case x^* in the interior of the cone we could reduce the problem to a nonlinear program without conic constraints. For s^* in the interior we have the trivial case $x^* = 0$.

Note that for the following propositions all variables x_μ and s_μ are dependent on μ. To ease the reading we omit the index μ for x_μ and s_μ and simply write x and s.

Proposition 8.2.9. *On relaxed complementary points we have* $\frac{1}{\mu}(s_0^2 - \|\bar{s}\|^2) = \frac{s_0}{x_0}$.

Proof. Let

$$\eta = \frac{x_0}{s_0}. \tag{8.46}$$

Note that we know from $x \times s = \mu I_{\mathcal{Q}}$

$$\begin{aligned} x^{\mathrm{T}} s &= \mu \\ \text{and} \qquad x_i &= -\frac{x_0}{s_0} s_i \text{ for } i \geq 1. \end{aligned} \tag{8.47}$$

This implies

$$\mu = x_0 s_0 + \sum_{i=1}^{n-1} x_i s_i = \eta s_0^2 - \eta \sum_{i=1}^{n-1} s_i^2 = \eta(s_0^2 - \|\bar{s}\|^2). \tag{8.48}$$

It follows that $(s_0^2 - \|\bar{s}\|^2) = \frac{\mu}{\eta}$. □

Proposition 8.2.10. *The eigenvalue* $\frac{1}{\mu}(s_0 - \|\bar{s}\|)^2$ *of* $\mathcal{F}^{-1}\mathcal{E}$ *on relaxed complementary points is converging to* 0 *for* $\mu \to 0$.

Proof. We know that on relaxed complementary points

$$\frac{1}{\mu}(s_0^2 - \|\bar{s}\|^2) = \frac{1}{\eta} \quad \text{with} \quad \eta = \frac{x_0}{s_0} \tag{8.49}$$

it follows

$$\frac{1}{\mu}(s_0 - \|\bar{s}\|)^2 = \frac{1}{\mu}(s_0^2 - \|\bar{s}\|^2)\frac{(s_0 - \|\bar{s}\|)}{(s_0 + \|\bar{s}\|)} = \frac{1}{\eta}\frac{(s_0 - \|\bar{s}\|)}{(s_0 + \|\bar{s}\|)} \to 0. \tag{8.50}$$

Note that the last result is only true if $s^* \neq 0$. For $s^* = 0$ the result is trivially true. □

Please note that the eigenvalue $\frac{1}{\mu}(s_0^2 + \|\bar{s}\|)^2$ converges to ∞ for $\mu \to 0$.

Corollary 8.2.11. *Using corollary 8.2.8 a necessary condition for* $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \succ 0$ *is that in the optimum the matrix* $\mathcal{H}$ *is positive definite in direction* x^* *and has larger eigenvalues than* $-\frac{s_0^*}{x_0^*}$ *for all vectors orthogonal to* x^* *and* s^*.

8.2.5 Approximation of $\mathcal{H}$ from chapter 4

The Hessian of the example from chapter 4 does **not** satisfy the necessary condition from corollary 8.2.11:

$$x^* = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \qquad \mathcal{H} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{pmatrix}. \tag{8.51}$$

Recall that a positive approximation of $\mathcal{H}$ leads to linear convergence.

In the example from chapter 4 we have $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} \not\succ 0$. In this section we show that for this example a perturbed $\tilde{\mathcal{H}}$ can be found that leads to quadratic convergence and thus does not satisfy $\tilde{\mathcal{H}} \succeq 0$, but $\tilde{\mathcal{H}} + \mathcal{F}^{-1}\mathcal{E} \succ 0$.

After presenting the results for this example we continue by generalizing the results to nonlinear SDP and give a general condition for the existence of such a perturbation.

Recall the example from chapter 4 given by

$$\begin{aligned} &\min \left\{ -x_1^2 - (x_2 - 1)^2 \mid \|x\|_2^2 \leq 1, \quad x \in \mathbb{R}^2 \right\} \\ \Leftrightarrow &\min \left\{ -x_1^2 - (x_2 - 1)^2 \;\middle|\; x_0 - 1 = 0, \quad \begin{pmatrix} x_0 \\ \bar{x} \end{pmatrix} \in \mathcal{Q} \right\}. \end{aligned} \tag{8.52}$$

with the Lagrangian

$$\begin{aligned} \mathcal{L}(x, y, s) &:= -x_1^2 - (x_2 - 1)^2 - y(x_0 - 1) - x^{\mathrm{T}} s, \\ \mathrm{g}(x, y, s) &:= \mathrm{D}_x \mathcal{L}(x, y, s) = \begin{pmatrix} -y & -s_0 \\ -2x_1 & -s_1 \\ -2(x_2 - 1) & -s_2 \end{pmatrix}, \\ \mathcal{H}(x, y, s) &:= \mathrm{D}_x^2 \mathcal{L}(x, y, s) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{pmatrix}. \end{aligned} \tag{8.53}$$

The optimal solution (x^*, y^*, s^*) is

$$x^* = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \quad y^* = -4, \quad s^* = \begin{pmatrix} 4 \\ 0 \\ 4 \end{pmatrix}. \tag{8.54}$$

As $x_0 = 1$ is fixed, the feasible set is the disc with radius one around $(0, 0)$, considering only the second and third vector entry.

We here present an indefinite $\tilde{\mathcal{H}}$ for that the sum $\tilde{\mathcal{H}} + \mathcal{F}^{-1}\mathcal{E}$ is positive definite and that is equivalent to $\mathcal{H}$ on the set that satisfies the active constraint $x_0 - 1 = 0$.

We define $\eta \in [0, 1[$ implicitly by

$$\mu = 2\left(\frac{1}{1 - \eta} - 1 + 3\eta - \eta^2 \right). \tag{8.55}$$

Please note that (8.55) is a monotone function of η, so that η is uniquely defined and for a given η the point on the central path is

$$x = \begin{pmatrix} 1 \\ 0 \\ -1 + \eta \end{pmatrix}, \quad s = \begin{pmatrix} \frac{2(2-\eta)}{1-\eta} \\ 0 \\ 4 - 2\eta \end{pmatrix}, \quad y = -\frac{2(2 - \eta)}{1 - \eta}. \tag{8.56}$$

To gain quadratic convergence $\tilde{\mathcal{H}}$ may not be changed in the direction Δx within the feasible set, that is $\tilde{\mathcal{H}}[\Delta x] \equiv \mathcal{H}[\Delta x]$ for any

$$\Delta x = \begin{pmatrix} 0 \\ \Delta x_1 \\ \Delta x_2 \end{pmatrix}. \tag{8.57}$$

We consider $\mathcal{F}^{-1}\mathcal{E}$ only for points that satisfy a relaxed complementarity that is

$$s = \frac{\mu}{x_0^2 - x_1^2 - x_2^2} \begin{pmatrix} x_0 \\ -x_1 \\ -x_2 \end{pmatrix}. \tag{8.58}$$

To simplify the following results we sometimes use radial coordinates and set

$$\begin{aligned} x_1 &= \sqrt{1-d}\sin(\alpha) \\ x_2 &= -\sqrt{1-d}\cos(\alpha). \end{aligned} \tag{8.59}$$

The scalar d satisfies $d = 1 - x_1^2 - x_2^2$. The optimal solution of (8.52) is $d = 0$ and $\alpha = 0$.

The lower right 2×2 matrix of the sum $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ is

$$M := \begin{pmatrix} -2 + \mu\left(\frac{1+x_1^2-x2^2}{(1-x_1^2-x_2^2)^2}\right) & \frac{2\mu x_2 x_1}{(1-x_1^2-x_2^2)^2} \\ \frac{2\mu x_2 x_1}{(1-x_1^2-x_2^2)^2} & -2 + \mu\left(\frac{1-x_1^2+x2^2}{(1-x_1^2-x_2^2)^2}\right) \end{pmatrix} \tag{8.60}$$

and has the eigenvalues

$$\lambda_1 = -2 + \frac{\mu}{1 - x_1^2 - x_2^2}, \quad \lambda_2 = -2 + \mu\frac{x_1^2 + x_2^2 + 1}{1 - 2x_1^2 - 2x_2^2 + (x_1^2 + x_2^2)^2}. \tag{8.61}$$

It is easy to see that $\lambda_1 > 0$ if

$$d = (1 - x_1^1 - x_2^2) > \sqrt{\frac{\mu}{2}}. \tag{8.62}$$

Note that according to (8.55) and (8.56) for η close to 0 we have $\mu \approx 6\eta$. So (8.62) is true for points close enough to the central path.

For the second eigenvalue one again can use $d := (1 - x_1^1 - x_2^2)$ and get a positive eigenvalue λ_2 for

$$0 < -2 + \frac{\mu(2-d)}{-1 + d + (1-d)^2} \quad \Leftrightarrow \quad d < -\frac{\mu}{4} + \frac{1}{4}\sqrt{\mu^2 + 16\mu}. \tag{8.63}$$

To see whether this is given close to the central path, we write this in terms of η and get

$$d < \eta\frac{(2-\eta)^2}{2(1-\eta)}\left(\sqrt{1 + 8\frac{1-\eta}{\eta(2-\eta)^2}} - 1\right) \tag{8.64}$$

which is true for a small area around the central path, as this has a distance of η to the boundary for any $\eta \in]0, 1[$. Furthermore for $\eta \to 0$ we have approximately $d < const\ \sqrt{\eta}$.

It follows that we can change $\mathcal{H}$ for the first component to get a positive sum $\mathcal{H}+\mathcal{F}^{-1}\mathcal{E} \succ 0$ if we are close enough to points that satisfy the relaxed complementarity. Furthermore for any feasible point (except $(1, 0, 0)$) we can choose a η and μ accordingly, such that we are "close enough" to points that satisfy a relaxed complementarity.

Let

$$\tilde{\mathcal{H}} := \mathcal{H} + \Delta\mathcal{H}, \quad \Delta\mathcal{H} := \begin{pmatrix} h & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{8.65}$$

We search for a number h so that $\tilde{\mathcal{H}} + \mathcal{F}^{-1}\mathcal{E} \succ 0$. Depending on h,d and μ the matrix $\tilde{\mathcal{H}} + \mathcal{F}^{-1}\mathcal{E}$ has the eigenvalues

$$\begin{aligned}
\lambda_1 &:= -2 + \frac{\mu}{d} \\
\lambda_2 &:= \frac{1}{2d^2}((h-2)d^2 + 2\mu(2-d) + \sqrt{d^4(h+2)^2 + 16\mu^2(1-d)}) \\
\lambda_3 &:= \frac{1}{2d^2}((h-2)d^2 + 2\mu(2-d) - \sqrt{d^4(h+2)^2 + 16\mu^2(1-d)})
\end{aligned} \tag{8.66}$$

While λ_1 is the same eigenvalue as in (8.61), λ_2 and λ_3 now depend on h and are positive for

$$h > \mu\frac{4 - 2d - \mu}{2\mu - 2d^2 - \mu d}. \tag{8.67}$$

For such an h the solution λ_2 has a zero, while λ_3 has a zero of higher multiplicity.

If $d = \eta$ which is given on the central path, then we have

$$h > 2\frac{(7\eta - 5\eta^2 - 2 + \eta^3) * (4 - 4\eta + \eta^2)}{(13\eta - 7\eta^2 + \eta^3 - 8)(1 - \eta)} \tag{8.68}$$

which has no singularities for $\eta \in]0, 1[$.

Thus for every $\eta \in]0, 1[$ we can find a h such that close to the central path $\tilde{\mathcal{H}} + \mathcal{F}^{-1}\mathcal{E} \succ 0$. For this approximation of $\mathcal{H}$ we get the same steps as for the exact $\mathcal{H}$.

Note that the condition $x_0 - 1 = 0$ is satisfied after the first step. Thus the Newton steps for the convexified $\tilde{\mathcal{H}}$ are exactly the same as for the original Hessian $\mathcal{H}$, thus we have quadratic convergence. This of course follows from the constraints being linear. We give a more general result for SDPs in section 8.3.2.

8.3 Eigenvalues of $\mathcal{F}^{-1}\mathcal{E}$ for semidefinite programs

8.3.1 Jordan algebra for the cone of semidefinite matrices $\mathcal{S}^n_+$

In section 8.2.1 we introduced the Jordan algebra over the symmetric matrices $\mathcal{S}^n$ by defining the multiplication $\cdot \times \cdot$

$$A \times B := \frac{1}{2}(AB + BA). \tag{8.69}$$

The unit element of this algebra is the identity matrix

$$I_{\mathcal{S}^n_+} := \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}. \tag{8.70}$$

The cone of positive semidefinite matrices $\mathcal{S}^n_+$ is the set that contains the squares of this algebra.

Proposition 8.3.1. *Let $A \in \mathcal{S}^n_{++}$ be an interior point of the positive semidefinite cone then A^{-1} has the following property*

$$A^{-1} \times A = I_{\mathcal{S}^n_+} \qquad \text{and} \qquad A \times A^{-1} = I_{\mathcal{S}^n_+}. \tag{8.71}$$

Proof. This result is trivial to recalculate. □

Let

$$L_A[X] := \frac{1}{2}(AX + XA) \tag{8.72}$$

and let $A = QDQ^{\mathrm{T}}$ be the eigenvalue decomposition of A then we can write $L_A[X]$ write as

$$\tilde{L}_D[\tilde{X}] := \frac{1}{2}Q(D\tilde{X} + \tilde{X}D)Q^{\mathrm{T}} = L_A[X] \quad \text{with} \quad \tilde{X} := Q^{\mathrm{T}}XQ. \tag{8.73}$$

This allows us to easily examine the eigenvalues and eigenvectors of L_A.

We use L_D if we want to examine L for a diagonal matrix D with diagonal elements d_i.

With $\Delta^{(i,j)}$ we denote a $n \times n$ matrix that has all entries zero,except for the entry (i, j) that one is 1.

Proposition 8.3.2. *L_D has the eigenvalues $\frac{1}{2}(d_i + d_j)$ for $1 \leq i, j \leq n$ with eigenvectors $\Delta^{(i,j)}$.*

Proof. This result is trivial to recalculate. □

Corollary 8.3.3. *Let $A = QDQ^{\mathrm{T}}$ be the eigenvalue decomposition of a matrix $A \in \mathcal{S}$ then L_A has the eigenvalues $\frac{1}{2}(d_i + d_j)$ for $1 \leq i, j \leq n$ and the eigenvectors $Q\Delta^{(i,j)}Q^{\mathrm{T}}$*

Proof. Recall $L_A[X] = \tilde{L}_D[\tilde{X}]$:

$$L_A[Q\Delta^{(i,j)}Q^{\mathrm{T}}] = \tilde{L}_D[\Delta^{(i,j)}] = QL_D[\Delta^{(i,j)}]Q^{\mathrm{T}} = \frac{1}{2}(d_i + d_j)Q\Delta^{(i,j)}Q^{\mathrm{T}} \tag{8.74}$$

□

Theorem 8.3.4. *Let $S = QDQ^{\mathrm{T}}$ be the eigenvalue decomposition of an IPM iterate variable $S \in \mathcal{S}$ with d_i being the eigenvalues. For relaxed complementary points $X = \mu S^{-1}$ the operator $\mathcal{F}^{-1}\mathcal{E}$ has the eigenvalues $\frac{d_i d_j}{\mu}$ for the eigenvectors $Q\Delta^{(i,j)}Q^{\mathrm{T}}$.*

Proof. Recall $\mathcal{F}^{-1} = L_X^{-1} = L_{\mu S^{-1}}^{-1}$. Thus $\mathcal{F}^{-1}$ has the eigenvalues $\frac{2}{\mu}\frac{d_i d_j}{d_i + d_j}$ for the eigenvectors $Q\Delta^{(i,j)}Q^{\mathrm{T}}$. Thus $\mathcal{F}^{-1}\mathcal{E}$ has the eigenvalues $\frac{d_i d_j}{\mu}$ with the eigenvectors $Q\Delta^{(i,j)}Q^{\mathrm{T}}$. □

We assume we have a bounded strictly complementary pair X^*, S^*. Let $d_i(\mu)$ be the eigenvalues of the solution S_μ on the central path. Those $d_i(\mu)$ that converge to 0 are $d_i(\mu) \in \mathcal{O}(\mu)$ as long as the according eigenvalue for X_μ converges.

Let k be the number of eigenvalues of S that converge to 0, then $\mathcal{F}^{-1}\mathcal{E}$ has an eigenspace of size k^2 that converges to 0, an eigenspace of size $2((n-k)k)$ that converges toward positive constants and an eigenspace of $(n-k)^2$ that is unbounded. This follows directly from the eigenvalues given in theorem 8.3.4 and $d_i(\mu) \in \mathcal{O}(\mu)$.

The curvature that is described by the eigenvalues of S is the one that makes $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ positive definite for the active set. In the next section we give a weak barrier problem condition that is equivalent to $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ on the linearized equality constraints $F(X)$.

8.3.2 A condition that leeds to quadratic convergence

We consider the nonlinear program

$$\min\{\ C \bullet X - \mu \ln(\det(X))\ |F(X) = 0\ \}. \tag{8.75}$$

The second derivative of $-\mu \ln(\det(X))$ for relaxed complementary points $XS0\mu I$ is $\mu\mathcal{F}^{-1}\mathcal{E}$. We show this result in corollary 8.3.7. Please note that (8.75) is similar to a barrier problem for linear SDPs, but with nonlinear constraints. Since (8.75) is a hybrid of the original problem and a barrier formulation one may expect that the minima of (8.75) converge towards the minimum of (6.1).

The Lagrangian of (8.75) and its derivatives are

$$\begin{aligned}
\breve{\mathcal{L}}(X,y) &:= C \bullet X + F(X)^* y - \mu \ln(\det(X)),\\
\breve{\mathrm{g}}(X,y) &:= \mathrm{D}_X \breve{\mathcal{L}}(X,y) = C + \mathrm{D}F(X)^* y - \mu X^{-1},\\
\breve{\mathcal{H}}(X,y) &:= \mathrm{D}_X^2 \breve{\mathcal{L}}(X,y) = \mathrm{D}_X^2(F(X)^* y) - \mu \mathrm{D}X^{-1}.
\end{aligned} \tag{8.76}$$

Corollary 8.3.5. *The feasible critical points of* (8.75) *can be interpreted as points on the central path of* (6.1).

Proof. On the central path we have $S = \mu X^{-1}$ and thus $\mathrm{g}(X,y,\mu X^{-1}) = \breve{\mathrm{g}}(X,y)$ and for both we have $F(X) = 0$. □

Let $\Delta^{(i,j)}$ be a square $n \times n$ matrix with all entries 0, except for the entry (i,j) that is $\Delta_{i,j} = 1$.

Note

$$\begin{aligned}
0 = \mathrm{D}I[H] = \mathrm{D}(XX^{-1})[H] &= \overbrace{\mathrm{D}X[H]}^{=H} X^{-1} + X\mathrm{D}X^{-1}[H]\\
\Leftrightarrow \mathrm{D}X^{-1}[H] &= -X^{-1}HX^{-1}.
\end{aligned} \tag{8.77}$$

We can also write the operator $\mathrm{D}X^{-1} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ as

$$\begin{aligned}
\mathrm{D}X^{-1}[H] &= -\left[\Xi^{(i,j)} \bullet H\right]_{i,j}\\
\text{with } \Xi^{(i,j)} &:= \left[\left(X^{-1}\right)_{j,k}\left(X^{-1}\right)_{l,i}\right]_{k,l} = X^{-\mathrm{T}}\Delta^{(j,i)}X^{-\mathrm{T}}.
\end{aligned} \tag{8.78}$$

As we always assume X to be symmetric. We have $\Xi^{(i,j)} = X^{-1}\Delta^{(j,i)}X^{-1}$.

Proposition 8.3.6. *Let $X = QDQ^{\mathrm{T}}$ be the eigenvalue decomposition of X, then the eigenvalues of $-\mathrm{D}X^{-1}$ are $\frac{1}{d_k d_l}$ for the eigenvectors $Q\Delta^{(k,l)}Q^{\mathrm{T}}$.*

Proof. Note

$$(Q\Delta^{(k,l)}Q^{\mathrm{T}})_{i,j} = Q_{k,i}Q_{l,j}. \tag{8.79}$$

For the eigenvector $Q\Delta^{(k,l)}Q^{\mathrm{T}}$ we have

$$\begin{aligned}
\left(-\mathrm{D}X^{-1}[Q\Delta^{(k,l)}Q^{\mathrm{T}}]\right)_{i,j} &= Tr(QD^{-1}Q^{\mathrm{T}}\Delta^{(j,i)}QD^{-1}Q^{\mathrm{T}}Q\Delta^{(k,l)}Q^{\mathrm{T}})\\
&= Tr(Q^{\mathrm{T}}\Delta^{(j,i)}QD^{-1}\Delta^{(k,l)}D^{-1})\\
&= \frac{1}{d_k d_l} Tr(Q^{\mathrm{T}}\Delta^{(j,i)}Q\Delta^{(k,l)})\\
&= \frac{1}{d_k d_l} Q_{k,i}Q_{l,j}\\
&= \frac{1}{d_k d_l} (Q\Delta^{(k,l)}Q^{\mathrm{T}})_{i,j}
\end{aligned} \tag{8.80}$$

□

Corollary 8.3.7. *For relaxed complementary points* $\mathcal{F}^{-1}\mathcal{E} \equiv \mathrm{D}X^{-1}$.

Proof. Recall the eigenvalues and eigenvectors of $\mathcal{F}^{-1}\mathcal{E}$ from theorem 8.3.4 and note that $S = \mu X^{-1}$ for relaxed complementary points. □

Definition 8.3.8. *Let* $\mathcal{Z}$ *be the feasible set of a minimization problem*

$$\min\{\ f(X) \mid X \in \mathcal{Z}\ \}. \tag{8.81}$$

We define the **tangential cone** $T(\mathcal{Z}, \bar{x})$ *by*

$$T(\mathcal{Z}, \bar{X}) := \Big\{ Z \in \mathbb{R}^{n\times n} \mid \exists\{X^{(k)}\}_k : \lambda_k \geq 0, X^{(k)} \in \mathcal{Z}, \lim_{k\to\infty} X^{(k)} = \bar{X}, \lim_{k\to\infty} \lambda_k(X^{(k)} - \bar{X}) = Z \Big\}. \tag{8.82}$$

Note that if (8.75) satisfies Robinson's regularity condition then the tangential cone at a feasible point $\bar{X}$ is

$$T(\{X|F(X) = 0\}, \bar{X}) = \{\ \Delta X \mid \mathrm{D}F(\bar{X})[\Delta X] = 0\ \}. \tag{8.83}$$

Proposition 8.3.9. *Let* $\mathcal{Z}$ *be the feasible set of* (8.75)*. If a critical point* $(\bar{X}_\mu, \bar{y}_\mu)$ *of* (8.75) *is a minimum then the according point* $(\bar{X}_\mu, \bar{y}_\mu, \mu\bar{X}_\mu^{-1})$ *on the central path of* (6.1) *satisfies*

$$\forall \Delta X \in T(\mathcal{Z}, \bar{X}),\ (\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})[\Delta X, \Delta X] \geq 0. \tag{8.84}$$

Proof. This follows directly from the 2nd order necessary conditions of (8.75) and as $\bar{X}_\mu$ is on the central path we have

$$\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} = \breve{\mathcal{H}}. \tag{8.85}$$

□

Corollary 8.3.10. *If the point of proposition 8.3.9 is a strict minimum then we have*

$$\forall \Delta X \in T(\mathcal{Z}, \bar{X}_\mu) \setminus \{0\},\ (\mathcal{H} + \mathcal{F}^{-1}\mathcal{E})[\Delta X, \Delta X] > 0. \tag{8.86}$$

The following proposition is known as "Finsler's Lemma" (see e.g. [JS03]):

Proposition 8.3.11. *Let* $U \in \mathcal{S}^n$*, let* $V \in \mathbb{R}^{m\times n}$ *and let* $\Upsilon := \{\ s \mid Vs = 0,\ s \neq 0\ \}$*. If* $s^{\mathrm{T}}Us > 0\ \forall s \in \Upsilon$ *then a* $\rho_0 \geq 0$ *exists such that* $U + \rho V^{\mathrm{T}}V \in \mathcal{S}^n_{++}$ *for all* $\rho \geq \rho_0$*.*

Theorem 8.3.12. *Let* (X_μ, y_μ, S_μ) *be a point on the central path which satisfies condition* (8.86) *strictly with* $X_\mu = \bar{X}_\mu$ *and let* μ *be close enough to 0. Then a positive semidefinite approximation of* $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ *exists that can be used to calculate a centering step that converges quadratically towards* (X_μ, y_μ, S_μ)*.*

Proof. We consider the convergence from a point (X, y, S) towards the central path point (X_μ, y_μ, S_μ). The central path point (X_μ, y_μ, S_μ) is a zero for

$$\begin{pmatrix} g \\ F \\ (X \times S) - \mu I_{\mathcal{Q}} \end{pmatrix}. \tag{8.87}$$

We can use Newton's approach to solve such a system and get quadratic convergence. The system for these steps are

$$\begin{pmatrix} \mathcal{H} & -\mathrm{D}F^* & -I \\ \mathrm{D}F & 0 & 0 \\ \mathcal{E} & 0 & \mathcal{F} \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta y \\ \Delta S \end{pmatrix} = \begin{pmatrix} -g \\ -F \\ \mu I_{\mathcal{Q}} - (X \times S) \end{pmatrix}. \tag{8.88}$$

It is well known that we maintain quadratic convergence if we exchange $\mathcal{H}$ and $\mathrm{D}F$ with $\bar{\mathcal{H}}$, $\mathrm{D}\bar{F}$ at the solution $(\bar{X}, \bar{y}, \bar{S})$. With these derivatives we get

$$\begin{pmatrix} \bar{\mathcal{H}} & -\mathrm{D}\bar{F} & -I \\ \mathrm{D}\bar{F} & 0 & 0 \\ \mathcal{E} & 0 & \mathcal{F} \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta y \\ \Delta S \end{pmatrix} = \begin{pmatrix} -g \\ -F \\ \mu I_{\mathcal{Q}} - (X \times S) \end{pmatrix}. \tag{8.89}$$

Note that for $F = 0$ we can add any perturbation $\Delta\mathcal{H}$ to $\bar{\mathcal{H}}$ satisfying

$$\Delta\mathcal{H}[\Delta X] = 0, \quad \text{for all } x \text{ with, } \mathrm{D}F[\Delta X] = 0 \tag{8.90}$$

and still get the same solution $(\Delta X, \Delta y, \Delta S)$.

In other words we could use $\Delta\mathcal{H} = \rho\mathrm{D}F^{\mathrm{T}}\mathrm{D}F$. Thus for $F = 0$ when can use proposition 8.3.11 to gain a positive semidefinite approximation $\bar{\mathcal{H}} := \Delta\mathcal{H} + \mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$ of $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$.

The condition $F(X_k) = 0$ is typically not satisfied, so we simply add $-\rho\mathrm{D}F^{\mathrm{T}}F(X_k)$ to the right hand and get the system

$$\begin{pmatrix} \bar{\mathcal{H}} + A\mathrm{D}F & -\mathrm{D}\bar{F} & -I \\ \mathrm{D}\bar{F} & & \\ \mathcal{E} & & \mathcal{F} \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta y \\ \Delta S \end{pmatrix} = \begin{pmatrix} -g - \rho\mathrm{D}F^{\mathrm{T}}F \\ -F \\ \mu I_{\mathcal{Q}} - (X \times S) \end{pmatrix}. \tag{8.91}$$

This system has the same solution as the original system, thus converges quadratically. □

8.3.3 Applying the results

In the following we discuss how to apply the result of quadratic convergence.

The proven quadratic convergence can not be used directly in practical applications as we do not know the point on the central path and cannot get the Hessian for that unknown point. Instead we can use the Hessian $\mathcal{H}$ and derivative $\mathrm{D}F$ at the current iterate.

It is only possible to convexify $\mathcal{H} + \mathcal{E}^{-1}\mathcal{F}$ by adding $\rho\mathrm{D}F^{\mathrm{T}}\mathrm{D}F$ if we're close enough to the central path[1]. We also need that μ is small enough and we need to converge to a strict minimum of (8.75).

Even though we focus on a local solver here, we might still be too far away from the central path to convexify $\mathcal{H} + \mathcal{E}^{-1}\mathcal{F}$ by adding $\rho\mathrm{D}F^{\mathrm{T}}\mathrm{D}F$. One alternative for such points is to use the pseudo inverse[2] for the "almost" convexified matrix $\mathcal{H} + \mathcal{E}^{-1}\mathcal{F} + \rho\mathrm{D}F^{\mathrm{T}}\mathrm{D}F$. Note that the pseudo inverse is exact on eigenspace where the eigenvalue is not too close to zero. For the predictor step we can use the pseudo inverse of the projection of the sum $\mathcal{H} + \mathcal{E}^{-1}\mathcal{F}$.

Another alternative is first convexifying then getting a correction term as follows. Let $\Phi(X, y, S)$ be a convexified approximation of $\mathcal{H}(X, y, S) + \mathcal{E}^{-1}\mathcal{F}$, e.g. using the eigenvalue decomposition. From this we can get the approximation of $\tilde{\mathcal{H}}$ as

$$\tilde{\mathcal{H}} := \Phi(X, y, S) - \mathcal{E}^{-1}\mathcal{F} \tag{8.92}$$

[1] Note that that the main argument here is that we have a strict minimum and the eigenvalues depend continuously on X, y, S and $F \in \mathcal{C}^3$.

[2] The pseudo inverse can be generated by "inverting" the diagonal of the eigenvalue decomposition. More precisely the inverse elements on the diagonal are used if they are not too small, else zero is used.

and define the perturbation $\Delta\mathcal{H}$ as

$$\Delta\mathcal{H} := \tilde{\mathcal{H}} - \mathcal{H}(X, y, S). \tag{8.93}$$

As we cannot expect to get a correction term $\rho \mathrm{D}F^{\mathrm{T}}F$ as in the proof of theorem 8.3.12, we try to get a least square approximation

$$A := \operatorname{argmin}\{\ \|A\mathrm{D}F - \Delta\mathcal{H}\|\ \}. \tag{8.94}$$

We can do so by solving

$$A\mathrm{D}F\mathrm{D}F^{\mathrm{T}} = \Delta\mathcal{H}\mathrm{D}F^{\mathrm{T}}. \tag{8.95}$$

If $\mathrm{D}F$ has maximal rank we can use a Cholesky decomposition, if not we can use the pseudo inverse.

An implementation has to show what the most efficient way is. Again we use the word "efficient" as a measure for the overall CPU time for the given set of problems.

9 Conclusion

In this last chapter we summarize and discuss our results. This thesis focuses on a solver for nonlinear SDPs.

9.1 On the SSP

The SSP algorithm has been analyzed before (see [FJV06], [CR04]), in this thesis we discuss its relevance in practical applications. We focus on necessary aspects to implement the algorithm. It turns out that in practical applications the SSP algorithm cannot yield more than linear local convergence. The reason is that the Lagrangian does not respect the curvature of the cone's boundary. In consequence this curvature is not represented in the KKT optimality conditions on which the SSP is based on.

An implementation however showed a nice global convergence behavior. Thus combined with a solver that has a fast local convergence properties, like the IPM presented in this thesis, we can define a hybrid algorithm.

9.2 On the SSP-implementation

We developed a SSP implementation to solve the given problems arising from industrial applications. The implementation presents the following features:

1. We introduced a new step length control: The augmented filter. The augmented filter like the filter is a purely heuristic approach that yields good results in practical applications. The augmented filter has proven to be more efficient for our examples than the standard filter approach. One reason is that it does not discard any search step. Additionally it can be used as an indicator for a hybrid solver switch and it can be used for a stopping criterion.

2. We have compared several different step length controls for our examples. We also discussed different implementations for theoretically equivalent search steps that have different practical properties. They differ in problem size, speed, and accuracy.

3. The implementation is build for readability and flexibility while being easy to handle. It turned out that with the right output, weaknesses and bugs are easily revealed. This solver may be considered as a solver construction kit, because it allows the user to freely choose between different algorithms to calculate the search step, step length control, and stopping criteria. It helps to easily determine the right elements to use. The solver can be used instantly. Components, that are typically the best, are chosen automatically. The solver detects when analytical derivations are missing and replaces them by numerical derivatives.

9.3 On the IPM presented here

We also presented an IPM algorithm for nonlinear SDPs. Different aspects had to be considered, such as the descent property and the invertibility of $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$. The results show that there is a correlation between these conditions and a certain condition from a barrier-like formulation.

We generated a matrix for that the invertibility is guaranteed. This matrix is positive definite and thus is easier to decompose using the Cholesky factorization. While we presented this result as a condition for quadratic convergence, it has practical consequences:

1. Such an approximation can be used for a centering step as well as for a predictor step. This allows us to define a short step algorithm. Short step algorithms are not interesting for practical applications, but allow further analysis of nonlinear IPMs using the central path presented here.

2. It allows analysis of (damped) low rank updates for $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E}$. We showed that a positive definite matrix exists for that we can maintain quadratic convergence. This result contrasts the counter example of chapter 4. We can now formulate a low rank update that converges against this matrix and yields superlinear convergence.

The matrix $\mathcal{H} + \mathcal{F}^{-1}\mathcal{E} + \rho \mathrm{D}F^{\mathrm{T}}\mathrm{D}F$ can be seen as a convexified version of $\mathcal{H}$. While the convexification of the directions orthogonal to DF reminds of the Hessian of the augmented Lagrangian, the convexification in cone boundary direction is done by a barrier term. Note this term does not exist at the optimal solution and the term $\mathcal{F}^{-1}\mathcal{E}$ is badly conditioned close to it.

It turns out that the AHO symmetrization gives us the interesting term $\mathcal{F}^{-1}\mathcal{E}$ to describe the curvature of the cone. This term is the second derivative of a barrier term when examining relaxed complementary points. Other symmetrizations might have similar properties, but the most common ones do not yield symmetric operators on relaxed complementary points.

Finally, an implementation will have to show whether the current results are good enough to yield a fast convergence of an IPM approach in practical applications. Even though we have quadratic convergence for the centering step, we might have a very small radius of convergence for the given examples.

Bibliography

[Da66] *G.B. Dantzig (1966):*
Lineare Programmierung und Erweiterungen,
Springer, Berlin

[Po78] *M.J.D. Powell (1978):*
The convergence of variable metric methods for nonlinearly constrained optimization calculations,
Nonlinear Programming p. 27–63, 3. Academic Press, New York

[Kh79] *L.G. Khachiyan (1979):*
A polynomial algorithm in linear programming,
Soviet Mathematics Doklady, 20 p.191–194

[Ka84] *N. Karmarkar (1984):*
A new polynomial-time algorithm for linear programming,
Combinatorica, 4, p. 373–395

[HJ85] *R.A. Horn, C.R. Johnson (1985):*
Matrix Analysis,
Cambridge University Press, New York

[Me92] *S. Mehrotra (1992):*
On the implementation of a primal-dual interior-point method,
SIAM J. Optimization, 2 p.575–601

[LMS92] *I.J. Lustig, R.E. Marsten, D.F. Shanno (1992):*
On implementing Mehrotra's predictor-corrector interior-point method for linear programming,
SIAM J. Optimization, 2 p.435–449

[BT95] *P.T. Boggs, J.W. Tolle (1995):*
Sequential Quadratic Programming,
Acta Numerica, 4, pp.1–51

[VB96] *L. Vandenberghe, S. Boyd (1996):*
Semidefinite Programming,
Siam Review, 38(1), pp. 49–95

[St97] *J.F. Sturm (1997):*
Primal-Dual Interior Point Approach to Ssemidefinite Programming,
Thesis Publishers Amsterdam

[St99] *J.F. Sturm (1999):*
Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,
Optimizational Methods Software 11–12, 625–653

[BF00] *Z. Bai, R.W. Freund (2000):*
Eigenvalue-based characterization and test for positive realness of scalar transfer functions,
IEEE Trans. Automat. Control 45, pp 2396–2402

[BS00] *J. F. Bonnans, A. Shapiro (2000):*
Perturbation analysis of optimization problems,
Springer, New York

[WSV00] *edited by H. Wolkowicz, R. Saigal, L. Vandenberghe (2000):*
Handbook of Semidefinite Programming, pp 199–213,
Kluwer Academic Publishers

[BF01] *Z. Bai, R.W. Freund (2001):*
A partial Padé-via-Lanczos method for reduced-order modeling,
Linear Algebra Appl. 332–334, pp.139–164

[FLT02] *R. Fletcher, S. Leyffer, Ph. L. Toint (2002):*
On the global convergence of a filter-SQP algorithm,
SIAM J. Optimization, 13(1):44–59

[Fr03] *R.W. Freund (2003):*
Model reduction method based on Krylov subspaces,
Acta Number 12, pp- 267–319

[Ja03] *F. Jarre (2003):*
On an Approximation of the Hessian of the Lagrangian,
http://www.optimization-online.org/DB_HTML/2003/12/800.html

[JS03] *F. Jarre, J. Stoer (2003):*
Mathematische Optimierung,
Springer

[TTT03] *R.H Tutuncu, K.C. Toh, and M.J. Todd (2003):*
Solving semidefinite-quadratic-linear programs using SDPT3,
Mathematical Programming Ser. B, 95 , pp. 189–217

[CR04] *R. Correa, H. C. Ramirez (2004):*
A Global Algorithm for Nonlinear Semidefinite Programming,
SIAM J. Optimization, Volume 15 Issue 1, pp. 303–318

[FJ04] *R.W. Freund,F. Jarre (2004):*
A sensitivity result for semidefinite programs,
Oper. Res. Lett. 32, pp 126–132

[St05] *Michael Stingl (2005):*
On the Solution of Nonlinear Semidefinite Programs by Augmented Lagrangian Methods,
Dissertation at Friedrich-Alexander-Universität Erlangen-Nürnberg

[DJV06] *M. Diehl, F. Jarre, C. H. Vogelbusch (2006):*
Loss of superlinear convergence for an SSP-type method with conic constraint,
SIAM J. Optimization

[FJV06] *R. Freund, F. Jarre, C. H. Vogelbusch (2006):*
Nonlinear semidefinite programming: sensitivity, convergence, and an application in passive reduced-order modeling,
Special Volume of Mathematical Programming Ser. B.

[SeWWW] *J.F. Sturm, et al. (WWW):*
SeDuMi: Let SeDuMi seduce you, too,
http://sedumi.mcmaster.ca/ — McMaster University