High Accuracy Algorithms for the Solutions of Semidefinite Linear Programs

by

Serge G. Kruk

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2001

I hereby declare that I am the sole author of this thesis.

I authorize the University of Waterloo to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the University of Waterloo to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

We present a new family of search directions and of corresponding algorithms to solve conic linear programs. The implementation is specialized to semidefinite programs but the algorithms described handle both nonnegative orthant and Lorentz cone problems and Cartesian products of these sets. The primary objective is not to develop yet another interior-point algorithm with polynomial time complexity. The aim is practical and addresses an often neglected aspect of the current research in the area, accuracy. Secondary goals, tempered by the first, are numerical efficiency and proper handling of sparsity.

The main search direction, called Gauss-Newton, is obtained as a least-squares solution to the optimality condition of the log-barrier problem. This motivation ensures that the direction is well-defined everywhere and that the underlying Jacobian is well-conditioned under standard assumptions. Moreover, it is invariant under affine transformation of the space and under orthogonal transformation of the constraining cone. The Gauss-Newton direction, both in the special cases of linear programming and on the central path of semidefinite programs, coincides with the search directions used in practical implementations. Finally, the Monteiro-Zhang family of search directions can be derived as scaled projections of the Gauss-Newton direction.

Acknowledgements

This adventure officially started years ago, on a winter day, when Henry suggested he supervise my graduate studies. Long before this conversation, the contagious enthusiasm he showed during an introductory course in optimization had convinced me to work with him. His unbridled energy often overwhelmed me, especially at first, when every email contained references to papers I should already have read; but the same enthusiasm never failed to bring me back on track after the unavoidable lulls and the moments of boredom, of depression. Thank you Henry.

Along the way, a number of people made the trek enjoyable and fruitful. My thesis cannot do justice to their influence for it extends far beyond the boundaries of a monograph. Thank you Jack and Kathie for your hospitality, in particular for one very long winter night, seated around a kitchen table, when our discussion of mathematics shaped my view of research. Thank you Dana and Steve, the believers, who saw a mathematician in me on very little evidence. Your faith helped me, more often than I usually care to acknowledge. And thank you Ada, gentle soul, for being there.

Finally, I need to mention my model teachers, those researchers who stand out from the usual pretentious and blasé academic crowd by the care they take to share, not only their expertise, but also their insights, and by the respect they show their students. In the classroom and around students I will always emulate Bill Cunningham, Steve Furino, Ian Goulden and Alan George. I chose academia for they made it appealing.

À la mémoire de mon père

Contents

List of Tables

List of Figures

Chapter 1

Semidefinite Programming

The expression used as the title of this chapter appeared in the early nineties [4, 41] and is attributed to Alizadeh [2], although some of the roots are older. Barely ten years after the development of linear programming, some researchers were thinking of generalizations to symmetric matrices [9]. Even older is the history of Linear Matrix Inequalities, a close parent of importance to control theorists, Yakubovich during the sixties [87, 88], and even Lyapunov at the beginning of the last century. (See the historical section of the Handbook of Semidefinite Programming [85] and the bibliography therein for details.)

Semidefinite Programming, as a topic of optimization, is barely ten years old [83] and sits at the boundary between linear and nonlinear programming. The functions involved in a standard formulation are linear. This similarity with linear programming explains why several semidefinite algorithms arose as extensions of standard linear programming algorithms [3]. But the cone constraint, maintaining nonnegative eigenvalues of the variable matrices, is on the other hand, nonlinear. Moreover, the major applications of semidefinite programming are relaxations of nonlinear programs and nonlinear control problems and the solution techniques by interior-points methods are modernizations of classical results of nonlinear programming [24].

The first chapter introduces the problem and the major concepts (primal-dual pair, optimality conditions, feasible set, interior, central path, barrier) in a classical manner. Every result derived in this chapter is well-known and included here to contextualize the work, express the basic definitions, and establish the notation.

We use semidefinite programming as the prototypical example of cone linear programs because it offers the right level of generality. The two other self-dual cones of any practical value at this time, the nonnegative orthant and the Lorentz cone, have specialized algorithms. Moreover, they are easily embedded in semidefinite cones as we will see in Chapter 4 where we discuss implementation and where we explain how our algorithms handle problems over Cartesian products of the three cones.

The main objective of this work is to introduce a new search direction based on the solution of a least-squares problem and to implement an interior-point algorithm that takes advantage of the strengths of this direction. Historically, interior-point algorithms for semidefinite programs were first developed by extending algorithms for linear programs. They were then given strong and more general foundations by the work of Nesterov and Nemirovskii [63]. More recently, an abstract approach to interior-point algorithms based on Euclidean Jordan algebras [7] has produced unified convergence results by showing how one can extend "word-by-word" certain linear programming algorithms to semidefinite problems, at the cost, we should note, of symmetrizing the complementarity condition. Another generalization of linear programming to conic programs, via the v-space approach, is developped in [82]. By contrast, we motivate the search direction from a classical nonlinear perspective. We investigate the main characteristics of the search direction, compare and contrast with the best practical directions. The final experiments exhibit a robust and accurate algorithm.

1.1 Standard Dual Pair

The problem, in the formulation we call the *primal*, is

$$(\mathfrak{Primal}) \qquad \min \left\{ \langle C, X \rangle \;\middle|\; \mathcal{A}(X) = b, X \in \mathbb{S}^n_+ \right\}, \tag{1.1}$$

where $b \in \mathbb{R}^m$, the standard Euclidean space; $\mathbb{S}^n$ is the space of real symmetric matrices of order n equipped with the inner product

$$\langle X, Y \rangle := \operatorname{trace}(XY) = \sum_{i=1}^{n} \sum_{j=1}^{n} X_{ij} Y_{ij}.$$

For $X, Y \in \mathbb{R}^{m \times n}$, the inner product is $\langle X, Y \rangle := \operatorname{trace}(X^t Y)$ and $\|X\|$ denotes the Frobenius norm, the norm induced by the inner product, ($\|X\| := \langle X, X \rangle^{\frac{1}{2}}$). The constraints are expressed by a linear operator $\mathcal{A}$,

$$\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m, \quad \mathcal{A}(X) := \begin{bmatrix} \langle A_1, X \rangle \\ \vdots \\ \langle A_m, X \rangle \end{bmatrix},$$

constructed from symmetric matrices A_i, for $1 \le i \le m$. Finally, $\mathbb{S}^n_+ \subset \mathbb{S}^n$ represents the cone of positive semidefinite matrices, the closure of $\mathbb{S}^n_{++} \subset \mathbb{S}^n_+$, the cone of positive definite matrices. A set C is a *cone* if

$$\lambda \ge 0, c \in C \Rightarrow \lambda c \in C.$$

The following properties are useful.

- $\mathbb{S}^n_+$ has non-empty interior;
- $\mathbb{S}^n_+$ is convex: $X_1, X_2 \in \mathbb{S}^n_+, \lambda \in [0, 1] \Rightarrow \lambda X_1 + (1 - \lambda) X_2 \in \mathbb{S}^n_+$;
- $\mathbb{S}^n_+$ contains no lines;
- $\mathbb{S}^n_+$ equals its polar cone, $\{Y \mid \langle Y, X \rangle \ge 0, \forall X \in \mathbb{S}^n_+\}$. (Note: this is either called self-duality or self-polarity.);
- $\mathbb{S}^n_+$ is homogeneous: for any pair $X_1, X_2 \in \mathbb{S}^n_{++}$, there is an element of the automorphism group of $\mathbb{S}^n_+$ that will map X_1 to X_2.

The last two properties are equivalent to what Nesterov and Todd [62], called *self-scaled*. This is crucial to the development of an abstract approach to interior-points algorithms using self-scaled barriers. This equivalence of self-polarity and homogeneity to self-scalability was noticed by Guler

[39] who also pointed out that homogeneous self-polar cones had been classified by certain Jordan algebras.

Few cones possess all the s above propertie. Only three have found significant practical applications at this time:

1. The Lorentz cone, an extension of relativistic space-time: $\mathbb{L}^n_+ := \{x \in \mathbb{R}^{n+1} \mid x_0 \geq \sqrt{x_1^2 + \ldots + x_n^2}\}$;
2. The positive semidefinite cone: $\mathbb{S}^n_+ := \{X \in \mathbb{S}^n \mid \forall x \in \mathbb{R}^n, x^t X x \geq 0\}$;
3. The nonnegative orthant: $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n \mid x \geq 0\}$.

The other cones are the positive semidefinite matrices with complex or with quaternion entries and an exceptional one, as well as direct sums of all of these cones.

We denote the *primal feasible region* of (1.1) by

$$\mathcal{F}^p := \left\{X \mid \mathcal{A}(X) = b, X \in \mathbb{S}^n_+\right\},$$

and the *strictly feasible primal* region by

$$\mathcal{F}^p_{++} := \left\{X \mid \mathcal{A}(X) = b, X \in \mathbb{S}^n_{++}\right\}.$$

This is also known as the *interior*, though it should properly be called the *relative interior*.

Problem (1.1) arises naturally in Control Theory [15, 83] but came to prominence as a relaxation of hard Combinatorial Problems [31], after the introduction of the Lovász theta function [53], the Stable-Set relaxation of Lovász and Schrijver [52], and the breakthrough approximation of the Maxcut problem by Goemans and Williamson [32]. Another application of semidefinite programming is the convex approximation of continuous non-convex problems. We briefly return to this topic in chapter 5.

Since the primal problem (1.1) is convex, the techniques of convex duality described by Rockafellar [71, 70] apply and we can derive a dual program via the Lagrangean function,

$$\mathscr{L}(X, y) := \langle C, X\rangle + \langle y, b - \mathcal{A}(X)\rangle,$$

where the second inner product is the standard inner product on $\mathbb{R}^m$, namely

$$\langle x, y\rangle := x^t y = \sum_{i=1}^{m} x_i y_i.$$

Following Rockafellar ([70] section 4), a dual problem is given by

$$\begin{aligned}
\max_y \left\{ \min_{X \in \mathbb{S}^n_+} \left\{ \mathscr{L}(X, y) \right\} \right\} &= \max_y \left\{ \min_{X \in \mathbb{S}^n_+} \left\{ \langle C, X\rangle + \langle y, b - \mathcal{A}(X)\rangle \right\} \right\} \\
&= \max_y \left\{ \min_{X \in \mathbb{S}^n_+} \left\{ \langle C, X\rangle + \langle y, b\rangle - \langle \mathcal{A}^*(y), X\rangle \right\} \right\} \\
&= \max_y \left\{ \min_{X \in \mathbb{S}^n_+} \left\{ \langle C - \mathcal{A}^*(y), X\rangle + \langle y, b\rangle \right\} \right\},
\end{aligned}$$

where $\mathcal{A}^*$ is the adjoint operator of $\mathcal{A}$, defined by

$$\mathcal{A}^* : \mathbb{R}^m \to \mathbb{S}^n, \quad \mathcal{A}^*(y) := \sum_{i=1}^{m} y_i A_i.$$

This definition of $\mathcal{A}^*$ results from the following

$$\langle y, \mathcal{A}(X)\rangle = \sum_{i=1}^{n} y_i \langle A_i, X\rangle = \left\langle \sum_{i=1}^{n} y_i A_i, X \right\rangle = \langle \mathcal{A}^*(y), X\rangle.$$

Since the inner minimization $\min_{X \in \mathbb{S}^n_+} \{\langle C - \mathcal{A}^*(y), X\rangle + \langle y, b\rangle\}$ is bounded only if $C - \mathcal{A}^*(y) \in \mathbb{S}^n_+$, the inner product $\langle C - \mathcal{A}^*(y), X\rangle$ attains its minimum at zero and we can simplify the dual program to

$$(\mathfrak{Dual}) \qquad \max \left\{ \langle b, y\rangle \;\middle|\; \mathcal{A}^*(y) + Z = C, Z \in \mathbb{S}^n_+ \right\}. \tag{1.2}$$

In program (1.2), which we call the dual, we introduced a slack variable Z to obtain an equality, the customary transformation. We denote the *dual feasible region* by

$$\mathcal{F}^{D} := \left\{ (y, Z) \mid \mathcal{A}^*(y) + Z = C, y \in \mathbb{R}^m, Z \in \mathbb{S}^n_+ \right\},$$

and the *strictly feasible dual region* by

$$\mathcal{F}^{D}_{++} := \left\{ (y, Z) \mid \mathcal{A}^*(y) + Z = C, y \in \mathbb{R}^m, Z \in \mathbb{S}^n_{++} \right\}.$$

Under the Slater convex constraint qualification for both problems, ($\mathcal{F}^{P}_{++} \neq \emptyset$ and $\mathcal{F}^{D}_{++} \neq \emptyset$), it is well-known that the optimal values of the primal and dual problems are equal and attained, a result we derive from our treatment of the central path. Moreover the set of optimal solutions of both primal and dual problem is bounded. We therefore obtain a *complementarity condition*: Let $X \in \mathcal{F}^P$ and $(y, Z) \in \mathcal{F}^D$ and consider

$$\begin{aligned} \langle C, X\rangle - \langle b, y\rangle &= \langle C, X\rangle - \langle \mathcal{A}(X), y\rangle \\ &= \langle C, X\rangle - \langle \mathcal{A}^*(y), X\rangle \\ &= \langle C - \mathcal{A}^*(y), X\rangle = \langle Z, X\rangle \geq 0. \end{aligned}$$

The last inequality is obtained from $X, Z \in \mathbb{S}^n_+$ and self-polarity of $\mathbb{S}^n_+$. For triple (X^*, y^*, Z^*), an optimal solution to a primal and dual pair with no duality gap, we therefore have that $\langle Z^*, X^*\rangle = 0$ and write a set of equations describing optimal solutions $(X, y, Z) \in \mathbb{S}^n_+ \times \mathbb{R}^m \times \mathbb{S}^n_+$,

$$\mathcal{A}(X) = b, \qquad \text{(primal feasibility)}; \tag{1.3a}$$

$$\mathcal{A}^*(y) + Z = C, \qquad \text{(dual feasibility)}; \tag{1.3b}$$

$$\langle Z, X\rangle = 0, \qquad \text{(complementarity)}. \tag{1.3c}$$

Note that the complementarity equation could as well be written as $ZX = 0$.

1.2 Derivatives

Before we go further in the development of semidefinite programming theory, we need to fix the concepts and the notation for the derivative of matrix functions. For abstract spaces, the reader is referred to Wouk [86], and more specifically for matrix functions, Graham [36] or Magnus and Neudecker [55].

Consider a function $F : V \to W$ where, in our case, the spaces V and W usually are symmetric matrix spaces, standard Euclidean vector spaces ($\mathbb{R}^n$) or Cartesian products of those inner product spaces. Whenever we speak of the derivative of such a function we mean the Fréchet derivative of F evaluated at v, the unique linear operator we denote $[\mathfrak{D}F(v)]$, satisfying, for all $d \in V$,

$$F(v + d) = F(v) + [\mathfrak{D}F(v)]d + o(\|d\|).$$

We use the notation $[\mathfrak{D}F(v)]$ to highlight the operator nature of the derivative ($[\mathfrak{D}F(v)] : V \to W$). The second derivative is often defined as a map $V \to [V \to W]$ but, following [86], we choose to view the second derivative as an operator $[\mathfrak{D}^2F(v)] : V \times V \to W$ satisfying, for all $d \in V$,

$$F(v + d) = F(v) + [\mathfrak{D}F(v)]d + \frac{1}{2}[\mathfrak{D}^2F(v)](d, d) + o(\|d\|^2).$$

Higher derivatives are defined and denoted similarly. The i^{th}-derivative, is denoted $[\mathfrak{D}^iF(v)]$.

For a function $F : U \times V \to W$ we use the following notation for the partial derivative of F with respect to variable $u \in U$ (respectively, variable $v \in V$)

$$[\mathfrak{D}_uF(u, v)], \qquad ([\mathfrak{D}_vF(u, v)]),$$

to indicate the linear operators such that

$$\begin{aligned} F(u + d_u, v) &= F(u, v) + [\mathfrak{D}_uF(u, v)]d_u + o(\|d_u\|), \qquad \text{and} \\ F(u, v + d_v) &= F(u, v) + [\mathfrak{D}_vF(u, v)]d_v + o(\|d_v\|). \end{aligned}$$

In the special case of particular interest where F is a functional ($F : V \to \mathbb{R}$), the first derivative is an element of the dual space and we use the gradient notation, $\nabla F(v)$, to identify the unique element of the primal space V satisfying, for every $d \in V$,

$$\langle \nabla F(v), d \rangle = [\mathfrak{D}F(v)]d. \tag{1.4}$$

Similarly for the second derivative,

$$\left\langle d, \nabla^2 F(v) d \right\rangle = [\mathfrak{D}^2 F(v)](d, d).$$

In the practical matter of calculating derivatives, it is sometimes easier to compute $\nabla F(v)$ than to find an expression for the operator $[\mathfrak{D}F(v)]$. As an example, consider the standard barrier function for the cone of semidefinite matrices,

$$F : \mathbb{S}^n_{++} \to \mathbb{R}, \qquad F(X) = -\log\det X.$$

By a result of Lewis [51], for a spectral function $F : \mathbb{S}^n \to \mathbb{R}$, the values of which can be expressed as $F(X) = f(\lambda(X))$ for some function $f : \mathbb{R}^n \to \mathbb{R}$, and eigenvalue function $\lambda : \mathbb{S}^n \to \mathbb{R}^n$, the gradient can be found via

$$\nabla F(X) = U^t \mathrm{Diag}\left(\nabla f(\lambda(X))\right) U, \quad \text{where} \quad U^t \mathrm{Diag}\left(\lambda(X)\right) U = X, U^t U = I.$$

In this case,

$$\begin{aligned} F(X) &= -\log\det(X) \\ &= -\log \prod \lambda_i(X) \\ &= -\sum_{i=1}^{n} \log \lambda_i(X) \\ &= -f(\lambda(X)), \quad \text{where} \quad f(x) := \sum_{i=1}^{n} \log x_i. \end{aligned}$$

The gradient is therefore given by

$$\begin{aligned}
\nabla F(X) &= -U^t \mathrm{Diag}\,(\nabla f(\lambda(X)))U \\
&= -U^t \mathrm{Diag} \begin{bmatrix} 1/\lambda_1(X) \\ \vdots \\ 1/\lambda_n(X) \end{bmatrix} U \\
&= -U^t [\mathrm{Diag}\,(\lambda(X))]^{-1} U \\
&= -X^{-1}.
\end{aligned}$$

For future reference, the derivative, obtained from (1.4), and the gradient of $F(X) = -\log\det(X)$ are

$$[\mathfrak{D}F(X)](\cdot) = -\langle X^{-1}, (\cdot) \rangle, \qquad \nabla F(X) = -X^{-1}. \tag{1.5}$$

Some more involved calculations [72] produce the second derivative and Hessian. Their expressions are

$$[\mathfrak{D}^2F(X)](\cdot,\cdot) = \langle (\cdot), X^{-1}(\cdot)X^{-1} \rangle, \qquad \nabla^2F(X)(\cdot) = X^{-1}(\cdot)X^{-1}. \tag{1.6}$$

Note that, for every $Y \in \mathbb{S}^n_+$, $[\mathfrak{D}^2F(X)](Y,Y) = \langle Y, X^{-1}YX^{-1} \rangle = \|YX^{-1}\|^2 \geq 0$. Moreover if $Y \neq 0$, $[\mathfrak{D}^2F(X)](Y,Y) = \|YX^{-1}\|^2 > 0$. This means that $F(X) = -\log\det(X)$ is a strictly convex function on its domain.

More properties of this barrier derive from its derivatives, but the above suffices for our exposition.

1.3 Central Path

Since the work of Fiacco and McCormick [24] on barrier techniques, later specialized by Nesterov and Nemirovskii [63] to self-concordant barriers, the preferred algorithms for our problem fall within the class of interior-point methods. These bear a striking resemblance to homotopy methods of differential equations.

Using a barrier on the cone $\mathbb{S}^n_{++}$, for example $-\log\det(X)$, we construct a family of strictly convex primal-dual pairs parameterized by the scalar $\mu > 0$,

$$\inf \left\{ \langle C, X \rangle - \mu \log \det(X) \;\middle|\; \mathcal{A}(X) = b, X \in \mathbb{S}^n_+ \right\}, \tag{1.7a}$$

$$\sup \left\{ \langle b, y \rangle + \mu \log \det(Z) \;\middle|\; \mathcal{A}^*(y) + Z = C, Z \in \mathbb{S}^n_+ \right\}. \tag{1.7b}$$

These two programs are dual in the sense of Fenchel, as we now proceed to exhibit. A concise version of this derivation is found in [51]. Consider the following transformation of the primal,

$$\begin{aligned} \nu(P) &= \inf \left\{ \langle C, X \rangle - \mu \log \det(X) \;\middle|\; \mathcal{A}(X) = b, X \in \mathbb{S}^n_+ \right\}, \\ &= \inf \left\{ f(X) + g(\mathcal{A}(X)) \;\middle|\; X \in \mathbb{S}^n_+ \right\}, \end{aligned}$$

where

$$f(X) := \langle C, X \rangle - L(X), \quad L(X) := \mu \log \det(X), \quad g(v) := i_{\{b\}}(v),$$

the last equation representing the indicator function of the set $\{b\}$. More generally, the indicator function of a set C is

$$i_C(x) := \begin{cases} 0 & \text{if } x \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

We calculate Fenchel conjugates,

$$\begin{aligned} L^*(Z) &:= \sup \left\{ \langle Z, X \rangle - \mu \log \det(X) \;\middle|\; X \in \mathbb{S}^n_+ \right\}; \\ g^*(z) &:= \sup \left\{ \langle z, y \rangle - i_{\{b\}}(y) \;\middle|\; y \in \mathbb{R}^m \right\} \\ &= \sup \left\{ \langle z, b \rangle \;\middle|\; y \in \mathbb{R}^m \right\} \\ &= \langle z, b \rangle; \\ f^*(Z) &:= \sup \left\{ \langle Z, X \rangle - \langle C, X \rangle - L(X) \;\middle|\; X \in \mathbb{S}^n_+ \right\} \\ &= \sup \left\{ \langle Z - C, X \rangle - L(X) \;\middle|\; X \in \mathbb{S}^n_+ \right\} \\ &= L^*(Z - C). \end{aligned}$$

Using these conjugates, we express a dual program,

$$-\inf\left\{f^*(-\mathcal{A}^*(y))+g^*(y)\;\middle|\;y\in\mathbb{R}^m\right\}=-\inf\left\{L^*(-\mathcal{A}^*(y)-C)+\langle y,b\rangle\;\middle|\;y\in\mathbb{R}^m\right\}.$$

Consider the inner supremum,

$$L^*(-\mathcal{A}^*(y)-C)=\sup\left\{\langle-\mathcal{A}^*(y)-C,X\rangle+\mu\log\det(X)\;\middle|\;X\in\mathbb{S}^n_+\right\}. \tag{1.8}$$

Its optimality conditions yield

$$0=-\mathcal{A}^*(y)-C+\mu X^{-1},\quad\text{or}\quad X=\mu(\mathcal{A}^*(y)+C)^{-1}.$$

Since we have a closed form for the solution of the inner supremum, we simplify (1.8),

$$\begin{aligned}
L^*(-\mathcal{A}^*(y)-C) &= \sup\left\{\langle-\mathcal{A}^*(y)-C,X\rangle+\mu\log\det(X)\;\middle|\;X\in\mathbb{S}^n_+\right\}\\
&= \left\langle-\mathcal{A}^*(y)-C,\mu(\mathcal{A}^*(y)+C)^{-1}\right\rangle+\mu\log\det(\mu(\mathcal{A}^*(y)+C)^{-1})\\
&= -\mu n+\mu\log\det(\mu(\mathcal{A}^*(y)+C)^{-1})\\
&= -\mu n+\mu n\log\mu-\mu\log\det(\mathcal{A}^*(y)+C).
\end{aligned}$$

We discard the constant term to obtain a simplified dual program,

$$\begin{aligned}
\nu(D) &= -\inf\left\{-\mu\log\det(\mathcal{A}^*(y)+C)+\langle y,b\rangle\;\middle|\;y\in\mathbb{R}^m\right\}\\
&= \sup\left\{\mu\log\det(\mathcal{A}^*(y)+C)-\langle y,b\rangle\;\middle|\;y\in\mathbb{R}^m\right\}\\
&= \sup\left\{\mu\log\det(C-\mathcal{A}^*(y))+\langle y,b\rangle\;\middle|\;y\in\mathbb{R}^m\right\}\\
&= \sup\left\{\langle y,b\rangle+\mu\log\det(Z)\;\middle|\;\mathcal{A}^*(y)+Z-C=0,y\in\mathbb{R}^m,Z\in\mathbb{S}^n_+\right\},
\end{aligned}$$

where we introduced the dual slack Z to make explicit the implicit cone constraint and to highlight the primal-dual symmetry. We now see that the two families of barrier problems (1.7) introduced in this section are indeed dual to each other.

The solutions to the pair of programs (1.9), parameterized by μ, is of crucial importance to the development of interior-point algorithms. We explore them further.

$$(\mathrm{P}_\mu) \qquad \inf\left\{\langle C, X\rangle - \mu \log\det(X) \;\middle|\; \mathcal{A}(X) = b\right\}, \tag{1.9a}$$

$$(\mathrm{D}_\mu) \qquad \sup\left\{\langle b, y\rangle + \mu \log\det(Z) \;\middle|\; \mathcal{A}^*(y) + Z = C\right\}. \tag{1.9b}$$

We observed before that the objective function of the primal is strictly convex, while that of the dual is strictly concave. From this convexity, we can show [57] that the existence of interior points $\overline{X} \in \mathcal{F}^{P}_{++}$ and $(\overline{y}, \overline{Z}) \in \mathcal{F}^{D}_{++}$ implies that the primal-dual pair (1.9a,1.9b) has a unique solution for each $\mu > 0$. To see this, consider that for each feasible X,

$$\begin{aligned}
\langle \overline{Z}, X\rangle &= \langle C - \mathcal{A}^*(\overline{y}), X\rangle \\
&= \langle C, X\rangle - \langle \overline{y}, \mathcal{A}(X)\rangle \\
&= \langle C, X\rangle - \langle \overline{y}, b\rangle.
\end{aligned}$$

Therefore $\langle \overline{Z}, X\rangle$ and $\langle C, X\rangle$ differ by a constant. Moreover, since $\overline{X}$ is feasible we can restrict the primal feasible set without affecting the optimal solution to obtain

$$\min\left\{\langle \overline{Z}, X\rangle - \mu \log\det(X) \;\middle|\; \mathcal{A}(X) = b, \langle \overline{Z}, X\rangle - \mu\log\det(X) \le \langle \overline{Z}, \overline{X}\rangle - \mu \log\det(\overline{X})\right\}.$$

Since the feasible set of this program is compact, we conclude that (1.9a) attains its optimal solution (and we justifiably write min instead of inf). Moreover, since the objective function is strictly convex, this solution is unique.

A similar argument may be developed for (1.9b). Therefore under the Slater constraint qualification, the pair of programs (1.9) has a unique solution for each barrier parameter $\mu > 0$. We stress that it is possible to express this solution using the optimality conditions of either programs.

For example, from the Lagrangeans,

$$\begin{aligned}\mathscr{L}_{\mathrm{P}}(X,y) &:= \langle C,X\rangle - \mu \log\det(X) + \langle y, b-\mathcal{A}(X)\rangle,\\ \mathscr{L}_{\mathrm{D}}(X,y,Z) &:= \langle b,y\rangle + \mu\log\det(Z) + \langle X, C-Z-\mathcal{A}^*(y)\rangle,\end{aligned}$$

we express the optimality conditions of the parameterized primal family as

$$0 = \nabla\mathscr{L}_{\mathrm{P}}(X,y) = \begin{bmatrix} C-\mu X^{-1}-\mathcal{A}^*(y)\\ b-\mathcal{A}(X)\end{bmatrix}, \tag{1.10}$$

and of the dual family as

$$0 = \nabla\mathscr{L}_{\mathrm{D}}(X,y,Z) = \begin{bmatrix} C-Z-\mathcal{A}^*(y)\\ b-\mathcal{A}(X)\\ \mu Z^{-1}-X\end{bmatrix}. \tag{1.11}$$

Without transforming the solution set we add $Z := \mu X^{-1}$ to (1.10) to obtain from either of the log-barrier problems,

$$\mathcal{A}^*(y)+Z = C, \tag{1.12a}$$

$$\mathcal{A}(X) = b, \tag{1.12b}$$

$$Z = \mu X^{-1}. \tag{1.12c}$$

The reader must be careful here. In a sense, the optimal solutions of (1.9a) are equivalent to the optimal solutions of (1.9b) since both are described by (1.12). But an implementation based on an attempt to solve (1.9a) by some iterative scheme leads to what is known as a primal interior-point method and is less desirable than an approach based on (1.12).

For all $\mu > 0$, the set of unique solutions to (1.12), which we denote (X_μ, y_μ, Z_μ) is called the *primal-dual central path.* (See Figure 1.1.) Note that

$$\langle Z_\mu, X_\mu \rangle = \left\langle \mu X_\mu^{-1}, X_\mu \right\rangle = \mu n.$$

This last relation provides one link between the solution to our problem (1.1,1.2) and the parameterized family of programs (1.9a,1.9b). As the parameter μ tends to 0, the sequence of solutions to (1.12) converges to a point where the complementarity equation (1.3c) is satisfied and the original problem is solved. This is the basis of all primal-dual path-following interior-point algorithms.

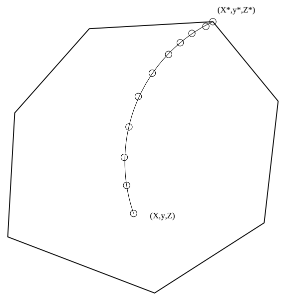


Figure 1.1: Central Path

They solve, more or less accurately, the system (1.12), or an algebraically equivalent formulation of this system, for decreasing values of μ.

1.4 Nonlinearity and Smoothing

The perturbed complementarity equation (1.12c) is written in terms of an inverse matrix. An algebraically equivalent formulation may be preferable. In practice, two transformations are used. The first one is to transform (1.12c) by a multiplication by X to obtain

$$ZX = \mu I. \tag{1.13}$$

This transformation is not inconsequential. Informally, it reduces the nonlinearity of the system with the aim of accelerating Newton-like methods. More precisely, it enlarges the radius of quadratic convergence.

Consider a function F to which we apply a Newton-like method to find ν^* such that $F(\nu^*) = 0$. We know ([20] Theorems 5.2.1 and 10.2.1) that the radius of quadratic convergence is bounded by a measure of the *relative nonlinearity* of F given by $\frac{k}{\beta\gamma}$. In this expression k is a small constant that depends on the method used (Newton or Gauss-Newton, for example), the scalar β provides a bound,

$$\|[\mathfrak{D}F(\nu^*)]^{-1}\| \leq \beta,$$

and γ is a Lipschitz continuity constant for $[\mathfrak{D}F(\nu)]$ in a ball around ν^*. To obtain an equivalent expression more significant to numerical analysts, we take $\sigma_{\min}^{-1}[\mathfrak{D}F(\nu^*)]$ for β and, for γ, we take $\max \sigma_{\max}[\mathfrak{D}F(\nu)]$ in the neighborhood of ν^*. Thus, we obtain a bound closely related to the condition number of the Jacobian,

$$k\frac{\sigma_{\min}[\mathfrak{D}F(\nu^*)]}{\sigma_{\max}[\mathfrak{D}F(\nu)]}.$$

With this expression in mind consider, in turn, both formulations of the complementarity condition (1.12c, 1.13) as we approach the optimal solution. In the first case, say $F(X, Z) := Z - \mu X^{-1}$, we have

$$[\mathfrak{D}F(X, Z)](d_X, d_Z) = d_Z + \mu X^{-1} d_X X^{-1}.$$

As the optimal solution X^* is almost always rank deficient, the norm of X^{-1} can be arbitrarily large and we cannot bound $\sigma_{\max}$. This implies that the proven radius of quadratic convergence is exceedingly small.

On the other hand, for the expression that we claimed to be less nonlinear, namely $F(X, Z) = ZX - \mu I$, we can compute the derivative as

$$[\mathfrak{D}F(X, Z)](d_X, d_Z) = Zd_X + d_Z X,$$

and we bound $\sigma_{\max}$ by $\|Z\| + \|X\|$. Therefore, for problems where the Jacobian is of full rank (a condition, as we will later see, resulting from standard assumptions) and where we can therefore bound $\sigma_{\min}$, we obtain a nonzero radius of quadratic convergence.

The second reason to formulate the complementarity as (1.13), even more directly related to the condition number, is that, in the limit, an ill-conditioned system may prevent accurate solutions. This has been the bane of barrier methods and a reason of their disappearance in the sixties, before the current revival. While it is known that the log-barrier ill-conditioning does not affect interior-point solutions of standard linear programs, it seems clear that current state-of-the-art semidefinite programming codes are deeply affected by this ill-conditioning. Since accuracy of the solutions is the major focus of our work, we will return in more detail to the conditioning problem.

For future reference, after this transformation, the sequence of systems to solve for decreasing values of μ defining the central path becomes

$$\mathcal{A}^*(y) + Z = C, \tag{1.14a}$$

$$\mathcal{A}(X) = b, \tag{1.14b}$$

$$ZX = \mu I. \tag{1.14c}$$

1.5 Smoothing and Symmetrization

The transformation of $Z = \mu X^{-1}$ into $ZX = \mu I$ has one unfortunate consequence: The residual $(ZX - \mu I)$ is not symmetric unless X and Z commute and therefore a Newton step is not possible on the system (1.14) since it is overdetermined.

From the start, possibly influenced by the success of linear programming codes where this problem does not arise, practitioners have eliminated the problem by symmetrization. The AHO direction, for example, projects both sides of the complementarity equation onto the subspace of symmetric matrices with the aid of the operator

$$H(M) := \frac{1}{2}[M + M^t].$$

From a point, (X_k, y_k, Z_k), and a parameter $\mu > 0$ the optimality conditions for the parameterized family are symmetrized, linearized and a Newton system is solved for (d_X, d_y, d_Z). We call this set of equations the *Unscaled Symmetric System*,

$$\mathcal{A}^*(d_y) + d_Z = -(\mathcal{A}^*(y_y) + Z_k - C) =: \quad -F_d, \tag{1.15a}$$

$$\mathcal{A}(d_X) = \quad -(\mathcal{A}(X_k) - b) =: \quad -f_p, \tag{1.15b}$$

$$H(Z_k d_X + d_Z X_k) = \quad -H(Z_k X_k - \mu I) =: -H(F_c). \tag{1.15c}$$

The direction (d_X, d_y, d_Z) obtained from (1.15) is known as the AHO direction [4], experimentally one of the directions leading to the most accurate solutions [79] of problem (1.1,1.2). All the Monteiro-Zhang family of directions [60], which includes most directions extensively used and analyzed, can be obtained from scaling the cone [79] then solving (1.15): Consider an element of the automorphism group of $\mathbb{S}^n$ expressed by the non-singular matrix P and let

$$\widetilde{X} := PXP^t, \quad \widetilde{A}_i := P^{-t}A_iP^{-1}, \quad \widetilde{C} := P^{-t}CP^{-1}. \tag{1.16}$$

This can be viewed as working on the *scaled primal problem*

$$\min\left\{\left\langle \widetilde{C}, \widetilde{X}\right\rangle \mid \widetilde{\mathcal{A}}(\widetilde{X}) = b, \widetilde{X} \in \mathbb{S}^n_+\right\}, \tag{1.17}$$

to which corresponds the *scaled dual problem*

$$\max\left\{\langle b, \widetilde{y}\rangle \mid \widetilde{\mathcal{A}}^*\widetilde{(y)} + \widetilde{Z} = \widetilde{C}, \widetilde{Z} \in \mathbb{S}^n_+\right\}. \tag{1.18}$$

This dual can also be obtained from (1.2) by the transformation

$$\widetilde{y} := y, \qquad \widetilde{Z} := P^{-t}ZP^{-1}. \tag{1.19}$$

The symmetric direction for the family of parameterized programs in this transformed space is therefore given by

$$\widetilde{\mathcal{A}}^*(d_y) + d_Z = -(\widetilde{\mathcal{A}}^*(\widetilde{y}_k) + \widetilde{Z}_k - \widetilde{C}) =: \quad -\widetilde{F}_d, \tag{1.20a}$$

$$\widetilde{\mathcal{A}}(d_X) = \quad -(\widetilde{\mathcal{A}}(\widetilde{X}_k) - b) =: \quad -\widetilde{f}_p, \tag{1.20b}$$

$$H(\widetilde{Z}_k d_X + d_Z \widetilde{X}_k) = \quad -H(\widetilde{Z}_k \widetilde{X}_k - \mu I) =: -H(\widetilde{F}_c), \tag{1.20c}$$

where

$$\widetilde{F}_d = \widetilde{\mathcal{A}}^*(\widetilde{y}_k) + \widetilde{Z}_k - \widetilde{C} = P^{-t} F_d P^{-1}, \tag{1.21a}$$

$$\widetilde{f}_p = \quad \widetilde{\mathcal{A}}(\widetilde{X}_k) - b = \quad f_p, \tag{1.21b}$$

$$\widetilde{F}_c = \quad \widetilde{Z}_k \widetilde{X}_k - \mu I = P^{-t} F_c P^{-1}. \tag{1.21c}$$

In this sense, the symmetric system (1.15) is the basic direction-finding paradigm for the Monteiro-Zhang family. The popular path-following algorithms differ by the scaling matrix P. The first directions, historically, are listed in Table (1.1) along with the implementations using them. These directions were not discovered via the scaling approach but it provides a unifying view.

P	Direction	Solvers
I	AHO [4]	SDPPack,SDPA,SDPT3
$Z^{\frac{1}{2}}$	HKM [41, 48, 56]	CSDP,SDPA,SDPT3
$[X^{\frac{1}{2}}(X^{\frac{1}{2}}ZX^{\frac{1}{2}})^{-\frac{1}{2}}X^{\frac{1}{2}}]^{\frac{1}{2}}$	NT [62]	SDPA,SDPT3,SeDuMi

Table 1.1: Instances of Monteiro-Zhang scaling matrix P of equation (1.16).

CSDP	http://www.nmt.edu/~borchers/csdp.html [13, 14]
SDPA	http://www-neos.mcs.anl.gov/neos/solvers/SDP:SDPA [26, 28, 27]
SDPPACK	http://cs.nyu.edu/cs/faculty/overton/sdppack/sdppack.html [40]
SDPT3	http://www.math.cmu.edu/~reha/sdpt3.html [80]
SeDuMi	http://www2.unimaas.nl/~sturm/research.html [77]

Table 1.2: Major Semidefinite Solvers

1.6 Generic Symmetric Algorithm

From the development above we state Algorithm 1.6.1, a generic approach to solve primal-dual semidefinite pairs using the symmetric form of the optimality conditions. This is not meant as an implementation but rather as a birds-eye view, able to describe all currently popular search direction-based algorithms.

Algorithm 1.6.1 Generic Interior-Point for Monteiro-Zhang family

Given $\epsilon > 0$;	$\{$Tolerance$\}$
Given X, y, Z;	$\{$Must satisfy some condition$\}$
$\mu = \frac{\langle Z, X\rangle}{n}$;	$\{$Initial barrier parameter$\}$
while $\mu > \epsilon$ **do**	
Choose scaling P;	$\{$Possibly dependent on $X, Z\}$
Choose centrality $0 < \tau < 1$;	$\{$According to some condition)$\}$
$\mu \leftarrow \tau \frac{\langle Z, X\rangle}{n}$;	$\{$Update target$\}$
Solve (1.20);	$\{$Scaled AHO direction$\}$
Choose step length α;	$\{$To maintain positive definiteness$\}$
$X = X + d_X; y = y + d_y; Z = Z + d_Z$;	$\{$Update iterate$\}$
end while	

We have described path-following algorithms based on the Monteiro-Zhang family of directions from this admittedly high-level view to highlight the close kinship of all popular search directions in semidefinite programming. It may be worth noting that Monteiro and Zhang do not provide the only unifying view. There have been other successful attempts, notably Monteiro and Tsuchiya [58] and Kojima, Shindoh, Hara [48]. We chose to highlight Monteiro-Zhang because it includes all the important directions currently in use whether they are important for theoretical or practical reasons.

An obvious question arising from this approach concerns the properties that can be inferred from their expression [79]. Another is whether a different basic paradigm can yield another family of directions with their own properties, strengths and weaknesses. This is pertinent since it is still unclear which direction, among the Monteiro-Zhang family or elsewhere, is "best". Even the measure of efficiency is debatable since the algorithms with the lower polynomial bound on the number of iterations are often slower than the algorithms used in practice.

The work described in the following chapters is such a new paradigm. The main direction-finding system, which we introduce in the next chapter and call the Gauss-Newton direction, is to be viewed as the AHO direction, that is, as the unscaled basic approach, to which any of the scalings of the Monteiro-Zhang family can be applied. Work on such a scaling has already started [47] and a polytime algorithm has been demonstrated. We are concerned here with the unscaled approach, its own merits and comparison with AHO.

Chapter 2

Gauss-Newton Directions

We now proceed to describe the fundamental search directions we intend to use as the basis of our interior-point algorithms. Recall the parameterized family of programs whose solution set define the central path

$$(\mathrm{P}_\mu) \qquad \min\left\{\langle C, X\rangle - \mu \log\det(X) \;\middle|\; \mathcal{A}(X) = b, X \in \mathbb{S}^n_+\right\}, \tag{2.1}$$

$$(\mathrm{D}_\mu) \qquad \max\left\{\langle b, y\rangle + \mu \log\det(Z) \;\middle|\; \mathcal{A}^*(y) + Z = C, Z \in \mathbb{S}^n_+\right\}, \tag{2.2}$$

and their associated smoothed optimality conditions,

$$F_\mu := \begin{bmatrix} F_d \\ f_p \\ F_c \end{bmatrix} := \begin{bmatrix} \mathcal{A}^*(y) + Z - C \\ \mathcal{A}(X) - b \\ ZX - \mu I \end{bmatrix} = 0.$$

It is important to remark, once again, that F_μ maps $\mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$ to $\mathbb{S}^n \times \mathbb{R}^m \times \mathbb{M}^n$.

2.1 Over-Determined Systems

The goal of a path-following algorithm is to approximately follow the central path determined by $F_\mu(X, y, Z) = 0$. This is a nonlinear and over-determined system of equations. In a classical setting, it would generally be solved by a globally convergent minimization algorithm applied to the norm of F_μ. We proceed to describe this classical approach.

To simplify the notation, let

$$\mathbb{V} := \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n, \tag{2.3a}$$

$$v := (X, y, Z) \in \mathbb{V}. \tag{2.3b}$$

Consider the norm

$$\|F_\mu(v)\|^2 := \langle F_\mu(v), F_\mu(v)\rangle = \langle F_d(v), F_d(v)\rangle + \langle f_p(v), f_p(v)\rangle + \langle F_c(v), F_c(v)\rangle,$$

where each inner product is the appropriate one, the trace inner product for the matrices F_d and F_c, and the Euclidean inner product for the vector f_p. Observe that

$$F_\mu(v) = 0 \iff \|F_\mu(v)\|^2 =: \varphi(v) = 0.$$

A solution to $F_\mu(v) = 0$ is therefore a solution to $\min \varphi(v)$, a nonlinear least-squares problem. To derive an algorithm for the latter, it is usual to start from a linearization of F_μ at a given point v,

$$L_\mu(d) := F_\mu(v) + [\mathfrak{D}F_\mu(v)]d_v,$$

and then proceed to find the best solution of this linearization in the least-squares sense, that is, to find a solution $d_v := (d_X, d_y, d_Z)$ of the problem

$$\min\left\{\|[\mathfrak{D}F_\mu(v)]d_v + F_\mu(v)\| \;\middle|\; d_v \in \mathbb{V}\right\}. \tag{2.4}$$

We call the vector d_v implicitly defined by (2.4) the *Gauss-Newton direction*.

From the definition of the Gauss-Newton direction (2.4) the over-determined linear system we intend to solve in a least-squares sense is

$$[\mathfrak{D}F_\mu(v)]d_v = -F_\mu(v), \quad \text{where} \quad d_v := \begin{bmatrix} d_X \\ d_y \\ d_Z \end{bmatrix}, \quad \text{and} \quad F_\mu = \begin{bmatrix} F_d \\ f_p \\ F_c \end{bmatrix}. \tag{2.5}$$

We interchangeably use two operator formulations for this system,

$$\begin{aligned} \mathcal{A}^*(d_y) + d_Z &= -(\mathcal{A}^*(y) + Z - C) && (2.6a)\\ \mathcal{A}(d_X) &= -(\mathcal{A}(X) - b) && (2.6b)\\ \mathcal{Z}(d_X) + \mathcal{X}(d_Z) &= -(ZX - \mu I), && (2.6c)\end{aligned}$$

and

$$\begin{bmatrix} 0 & \mathcal{A}^* & \mathcal{I} \\ \mathcal{A} & 0 & 0 \\ \mathcal{Z} & 0 & \mathcal{X} \end{bmatrix} \begin{bmatrix} d_X \\ d_y \\ d_Z \end{bmatrix} = - \begin{bmatrix} F_d \\ f_p \\ F_c \end{bmatrix}, \tag{2.7}$$

where the operators $\mathcal{Z}$ and $\mathcal{X}$ are defined by

$$\mathcal{Z} : \mathbb{S}^n \to \mathbb{M}^n, \quad \mathcal{Z}(M) := ZM \quad \text{and} \quad \mathcal{X} : \mathbb{S}^n \to \mathbb{M}^n, \quad \mathcal{X}(M) := MX.$$

Using the now common notation for pseudo-inverses, we succinctly express the Gauss-Newton direction d_v as

$$(\textit{Gauss-Newton direction}) \qquad d_v = -[\mathfrak{D}F_\mu(v)]^\dagger F_\mu(v). \tag{2.8}$$

Where $A^\dagger$ is the Moore-Penrose inverse of A, the unique operator satisfying the four conditions

1. $AA^\dagger A = A$,

2. $A^\dagger A A^\dagger = A^\dagger$,

3. $(AA^\dagger)^t = AA^\dagger$,

4. $(A^\dagger A)^t = A^\dagger A$.

This is also called the $\{1,2,3,4\}$-inverse by Ben-Israel and Greville [10], still the authority on the subject.

The pseudo-inverse notation has the advantage of expressing not just a least-squares solution to (2.5), but, when the Jacobian is rank-deficient, expressing the solution of minimum norm, the solution to

$$\min\left\{\|d_\nu\| \;\middle|\; d_\nu \in \arg\min\left\{\left\|[\mathfrak{D}F_\mu(\nu)]d_\nu + F_\mu(\nu)\right\| \;\middle|\; d_\nu \in \mathbb{V}\right\}\right\}. \tag{2.9}$$

To prepare the way for an implementation, we also use an equivalent matrix formulation for the over-determined system,

$$[J_{gn}]\, d_\nu = \begin{bmatrix} 0 & A^t & I \\ A & 0 & 0 \\ (Z \oslash I) & 0 & (I \oslash X) \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = - \begin{bmatrix} f_d \\ f_p \\ f_c \end{bmatrix}, \tag{2.10}$$

where

$$\begin{aligned} d_x &:= \operatorname{svec}(d_X) \\ d_z &:= \operatorname{svec}(d_Z) \\ f_d &:= \operatorname{svec}(F_d) \\ f_c &:= \operatorname{avec}(F_c)\,. \end{aligned}$$

The operator $\operatorname{svec}(\cdot) : \mathbb{S}^n \to \mathbb{R}^{t(n)}$ multiplies the off-diagonal elements by $\sqrt{2}$ and then stacks, column by column, the upper triangle of a symmetric matrix into a vector of size equal to the *triangular number of* n,

$$t(n) := \frac{n(n+1)}{2}. \tag{2.11}$$

Its inverse operator is smat $(\cdot) : \mathbb{R}^{t(n)} \to \mathbb{S}^n$. For example, if $X \in \mathbb{S}^n, n = 3$,

$$\operatorname{svec}(X) = \begin{bmatrix} X_{11} & X_{12}\sqrt{2} & X_{22} & X_{13}\sqrt{2} & X_{23}\sqrt{2} & X_{33} \end{bmatrix}^t.$$

The $\sqrt{2}$ scaling ensures that we maintain the metric, $\langle X, Z\rangle = \langle \operatorname{svec}(X), \operatorname{svec}(Z)\rangle$. More formally, define the index function I_s and its inverse

$$I_s(i,j) \;:=\; t(j-1)+i, \tag{2.12a}$$
$$I_s^{-1}(k) \;:=\; \left(k - \frac{j^2-j}{2}, j\right), \quad \text{where} \quad j := \left\lceil \frac{-1+\sqrt{1+8k}}{2} \right\rceil, \tag{2.12b}$$

so that if $X \in \mathbb{S}^n$ and $\operatorname{svec}(X) = x$, then for any $1 \le i \le j \le n$, we have the component identities

$$\begin{aligned} X_{ij} &= x_{I_s(i,j)} \quad \text{and} \quad x_k = X_{I_s^{-1}(k)}, \text{ for } i = j; \\ X_{ij} &= \frac{1}{\sqrt{2}} x_{I_s(i,j)} \quad \text{and} \quad x_k = \sqrt{2} X_{I_s^{-1}(k)}, \text{ for } i \neq j. \end{aligned}$$

The corresponding operator, avec $(\cdot) : \mathbb{M}^n \to \mathbb{R}^{n^2}$, stacks the column of any matrix into a vector. We define the index function I_a and its inverse

$$I_a(i,j,n) \;:=\; n(j-1)+i, \tag{2.13a}$$
$$I_a^{-1}(k,n) \;:=\; \left(k - n\left\lfloor \frac{k-1}{n} \right\rfloor, \left\lfloor \frac{k-1}{n} \right\rfloor + 1\right), \tag{2.13b}$$

so that if $X \in \mathbb{M}^n$ and $\operatorname{avec}(X) = x$, then for any $1 \le i \le j \le n$, we have the component identities

$$X_{ij} = x_{I_a(i,j,n)}, \qquad x_k = X_{I_a^{-1}(k,n)}.$$

The binary operator $\oslash : \mathbb{S}^n \times \mathbb{S}^n \to \mathbb{M}^{n^2 \times t(n)}$, the *asymmetric Kronecker product*, is defined by the identity

$$\operatorname{avec}(AXB) = (A \oslash B)\operatorname{svec}(X).$$

The matrix $A \oslash B$ is of size $n^2 \times t(n)$ and we find each entry by using bases for the domain and co-domain. Let a basis for $\mathbb{S}^n$ be

$$E_{ij} := \begin{cases} \frac{1}{\sqrt{2}}(e_i e_j^t + e_j e_i^t), & i \neq j; \\ e_i e_j^t, & i = j; \end{cases}$$

where e_k is a vector in $\mathbb{R}^n$ with a 1 in position k and zeros elsewhere. To find entry (k, l) of $A \oslash B$, where $l = I_s(i, j)$, and $k = I_a(\bar{\imath}, \bar{\jmath}, n)$, we first consider the case $i \neq j$. By definition of $\oslash$,

$$\begin{aligned} e_k^t(A \oslash B)\mathrm{svec}\,(E_{ij}) &= e_k^t \mathrm{avec}\,(AE_{ij}B) \\ &= \frac{e_k^t}{\sqrt{2}} \mathrm{avec}\left(A(e_i e_j^t + e_j e_i^t)B\right) \\ &= \frac{e_k^t}{\sqrt{2}} \mathrm{avec}\left(Ae_i e_j^t B + Ae_j e_i^t B\right) \\ &= \frac{e_k^t}{\sqrt{2}} \mathrm{avec}\,(A_{:i}B_{j:} + A_{:j}B_{i:}) \\ &= \frac{1}{\sqrt{2}} [A_{:i}B_{j:} + A_{:j}B_{i:}]_{I_a^{-1}(k,n)}\,, \end{aligned}$$

where the notation $A_{:j}$ is meant to indicate column j of matrix A, and $A_{i:}$ is row i. With similar calculations for the case $i = j$, we obtain the kl component as

$$[A \oslash B]_{kl} = \begin{cases} \frac{1}{\sqrt{2}}(A_{\bar{\imath}i}B_{j\bar{\jmath}} + A_{\bar{\imath}j}B_{i\bar{\jmath}}), & i \neq j \\ A_{\bar{\imath}i}B_{j\bar{\jmath}}, & i = j. \end{cases}$$

We later need the following bounds.

Lemma 2.1.1 *For any matrix* $X \in \mathbb{S}^n$, $\|I \oslash X\| = \|X \oslash I\| = \sqrt{\frac{n+1}{2}}\|X\| \leq \sqrt{t(n)}\|X\|_2$.

Proof: The first equality is clear since the entries of $\|X \oslash I\|$ are permutations of the entries of $\|I \oslash X\|$. For the second equality, let

$$X = \sum_{i=1}^{t(n)} \alpha_i E_i$$

be a decomposition of X into an orthonormal basis of $\mathbb{S}^n$. Then

$$\begin{aligned}
\|X \oslash I\| &= \left\| \left(\sum_{i=1}^{t(n)} \alpha_i E_i \right) \oslash I \right\| \\
&= \left\| \sum_{i=1}^{t(n)} \alpha_i \left(E_i \oslash I \right) \right\| \\
&= \left(\left\langle \sum_{i=1}^{t(n)} \alpha_i \left(E_i \oslash I \right), \sum_{i=1}^{t(n)} \alpha_i \left(E_i \oslash I \right) \right\rangle \right)^{\frac{1}{2}} \\
&= \left(\sum_{i=1}^{t(n)} \langle \alpha_i \left(E_i \oslash I \right), \alpha_i \left(E_i \oslash I \right) \rangle \right)^{\frac{1}{2}} \\
&= \left(\sum_{i=1}^{t(n)} \alpha_i^2 \langle \left(E_i \oslash I \right), \left(E_i \oslash I \right) \rangle \right)^{\frac{1}{2}} \\
&= \left(\sum_{i=1}^{t(n)} \alpha_i^2 \left(\frac{n+1}{2} \right) \right)^{\frac{1}{2}} \\
&= \sqrt{\frac{n+1}{2}} \|X\|
\end{aligned}$$

where we used the the orthogonality of the matrices $\{E_i \oslash I\}$ and that $\langle E_i \oslash I, E_i \oslash I \rangle = \frac{n+1}{2}$. The last inequality is derived from a standard result, $\|X\| \le \sqrt{n}\|X\|_2$ ([43], page 313). □

Lemma 2.1.2 *For any matrix* $X \in \mathbb{S}^n$, $\|I \oslash X\|_2 = \|X \oslash I\|_2 \le \|X\|$.

Proof: Again the first equality is clear. For the inequality,

$$\begin{aligned}
\|X \oslash I\|_2 &= \max \left\{ \|(X \oslash I)v\| \mid \|v\| = 1 \right\} \\
&= \max \left\{ \|\text{avec} \left(X\text{smat} \left(v \right) \right) \| \mid \|v\| = 1 \right\} \\
&= \max \left\{ \|X\text{smat} \left(v \right) \| \mid \|v\| = 1 \right\} \\
&\le \max \left\{ \|X\| \|\text{smat} \left(v \right) \| \mid \|v\| = 1 \right\} \\
&= \|X\|.
\end{aligned}$$

The last equality is derived from $\|\mathrm{smat}\,(\nu)\,\| = \|\nu\|$. □

The matrix-vector formulation (2.10) of the Gauss-Newton system highlights the fact that the left-hand side Jacobian matrix J_{gn} operates on a vector space isomorphic to $\mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$.

If a feasible initial vector $\nu_0 = (X_0, y_0, Z_0)$ is provided, then feasible iterates based on a constrained least-squares problem may be considered. In contrast to (2.9), the defining problem for this direction is a constrained least-squares problem,

$$\min \left\{ \|[\mathfrak{D}F_c(\nu)](d_\nu) + F_c(\nu)\| \;\middle|\; \mathcal{A}^*(d_y) + d_Z = 0, \mathcal{A}(d_X) = 0, d_\nu \in \mathbb{V} \right\}. \tag{2.14}$$

We will call the solution to (2.14) the *Feasible Gauss-Newton direction.* Primal and dual feasibility are maintained and we find the constrained least-squares solution to the linearization of the smoothed complementarity equation.

If the system (2.7) is non-singular, an assumption we will justify shortly, we can find the Gauss-Newton direction by solving the *normal equations.* This is not what should be done numerically for accurate solutions since the condition number usually worsens, but it provides a mathematically useful expression. The normal equations are

$$[\mathfrak{D}F_\mu(\nu)]^*[\mathfrak{D}F_\mu(\nu)]d_\nu = -[\mathfrak{D}F_\mu(\nu)]^* F_\mu(\nu). \tag{2.15}$$

In equivalent operator notation,

$$\begin{aligned}
(\mathcal{A}^*\mathcal{A} + \mathcal{Z}^*\mathcal{Z})d_X + (\mathcal{Z}^*\mathcal{X})d_Z &= -(\mathcal{A}^* f_p + \mathcal{Z}^* F_c), && (2.16a)\\
(\mathcal{A}\mathcal{A}^*)d_y + \mathcal{A}d_Z &= -(\mathcal{A}F_d), && (2.16b)\\
(\mathcal{X}^*\mathcal{Z})d_X + \mathcal{A}^* d_y + (I + \mathcal{X}^*\mathcal{X})d_Z &= -(F_d + \mathcal{X}^* F_c), && (2.16c)
\end{aligned}$$

where the adjoint operators are

$$\mathcal{X}^* : \mathbb{M}^n \to \mathbb{S}^n, \quad \mathcal{X}^*(L) := \frac{1}{2}\left\{LX + X^t L^t\right\}, \tag{2.17}$$

$$\mathcal{Z}^* : \mathbb{M}^n \to \mathbb{S}^n, \quad \mathcal{Z}^*(L) := \frac{1}{2}\left\{ZL + L^t Z^t\right\}. \tag{2.18}$$

These definitions follow from

$$\langle \mathcal{X}(M), L\rangle = \text{trace}\,(MX)^t L = \text{trace}\, M^t LX = \frac{1}{2}\text{trace}\, M^t \left\{LX + XL^t\right\} = \langle M, \mathcal{X}^*(L)\rangle,$$

and similarly for $\mathcal{Z}$.

For later reference, we symbolically solve the normal equations to express some of the variables in terms of the others. First, in term of d_Z,

$$d_X = -(\mathcal{A}^*\mathcal{A} + \mathcal{Z}^*\mathcal{Z})^{-1}[\mathcal{A}^* f_p + Z^*(\mathcal{X} d_Z + F_c)], \tag{2.19a}$$

$$d_y = -\mathcal{A}^{*\dagger}(d_Z + F_d). \tag{2.19b}$$

Then in terms of d_X,

$$d_Z = (I + \mathcal{X}^*\mathcal{X} - \mathcal{A}^*\mathcal{A}^{*\dagger})^{-1}\mathcal{X}^*(\mathcal{X}F_d - \mathcal{Z}d_X - F_c) - F_d, \tag{2.20a}$$

$$d_y = \mathcal{A}^{*\dagger}(I + \mathcal{X}^*\mathcal{X} - \mathcal{A}^*\mathcal{A}^{*\dagger})^{-1}\mathcal{X}^*(\mathcal{Z}d_X + F_c - \mathcal{X}F_d). \tag{2.20b}$$

For the sake of completeness, we also express the complete symbolic solution to the normal equations

$$\begin{aligned} d_Z = {} & [I - (I + \mathcal{X}^*\mathcal{X} - \mathcal{A}^*\mathcal{A}^{*\dagger})\mathcal{X}^*\mathcal{Z}(\mathcal{A}^*\mathcal{A} + \mathcal{Z}^*\mathcal{Z})^{-1}\mathcal{Z}^*\mathcal{X}]^{-1} \\ & \left\{(I + \mathcal{X}^*\mathcal{X} - \mathcal{A}^*\mathcal{A}^{*\dagger})^{-1}\right. \\ & \left.[\mathcal{X}^*\mathcal{X}F_d - \mathcal{X}^*F_c + \mathcal{X}^*\mathcal{Z}(\mathcal{A}^*\mathcal{A} + \mathcal{Z}^*\mathcal{Z})^{-1}(\mathcal{A}^*f_p + \mathcal{Z}^*F_c)] - F_d\right\}, \end{aligned} \tag{2.21}$$

$$\begin{aligned} d_X = {} & -(\mathcal{A}^*\mathcal{A} + \mathcal{Z}^*\mathcal{Z})^{-1} \\ & \left\{\mathcal{A}^*f_p + \mathcal{Z}^*F_c\mathcal{Z}^*\mathcal{X}[I - (I + \mathcal{X}^*\mathcal{X} - \mathcal{A}^*\mathcal{A}^{*\dagger})\mathcal{X}^*\mathcal{Z}(\mathcal{A}^*\mathcal{A} + \mathcal{Z}^*\mathcal{Z})^{-1}\mathcal{Z}^*\mathcal{X}]^{-1}\right. \\ & \left\{(I + \mathcal{X}^*\mathcal{X} - \mathcal{A}^*\mathcal{A}^{*\dagger})^{-1}\right. \\ & \left.\left.[\mathcal{X}^*\mathcal{X}F_d - \mathcal{X}^*F_c + \mathcal{X}^*\mathcal{Z}(\mathcal{A}^*\mathcal{A} + \mathcal{Z}^*\mathcal{Z})^{-1}(\mathcal{A}^*f_p + \mathcal{Z}^*F_c)] - F_d\right\}\right\}, \end{aligned} \tag{2.22}$$

$$\begin{aligned} d_y &= -\mathcal{A}^{*\dagger}\Big\{[I-(I+\mathcal{X}^*\mathcal{X}-\mathcal{A}^*\mathcal{A}^{*\dagger})\mathcal{X}^*\mathcal{Z}(\mathcal{A}^*\mathcal{A}+\mathcal{Z}^*\mathcal{Z})^{-1}\mathcal{Z}^*\mathcal{X}]^{-1} \\ &\quad \Big\{(I+\mathcal{X}^*\mathcal{X}-\mathcal{A}^*\mathcal{A}^{*\dagger})^{-1} \\ &\quad [\mathcal{X}^*\mathcal{X}F_d-\mathcal{X}^*F_c+\mathcal{X}^*\mathcal{Z}(\mathcal{A}^*\mathcal{A}+\mathcal{Z}^*\mathcal{Z})^{-1}(\mathcal{A}^*f_p+\mathcal{Z}^*F_c)]-F_d\Big\}\Big\}. \end{aligned} \tag{2.23}$$

Using these expressions, it is possible to implement a Gauss-Newton based algorithm that is both fast and spares memory. It competes with the best current solvers in terms of efficiency. But our aim is to obtain as much accuracy as possible, a goal the normal equation approach cannot achieve.

2.2 Properties of the Directions

Before we use the Gauss-Newton directions in an algorithm, we consider the properties that motivate its use.

2.2.1 Well-Defined

To develop algorithms based on the Gauss-Newton directions, we need to provide conditions under which these directions are properly defined.

Lemma 2.2.1 *Let $A_{m\times n}$, $B_{p\times n}$ be matrices with $m \geq n$ and $p \leq n$. Also let the columns of P_B be a basis for the nullspace of B. If the matrix AP_B is of full rank then the optimal solution of*

$$\min\Big\{\|Ax-b\| \;\Big|\; Bx=0\Big\}$$

is $x = P_B\left\{(AP_B)^t(AP_B)\right\}^{-1}(AP_B)b = P_B(AP_B)^\dagger b$.

Proof: Let v^* be the optimal value of the above program and let $x = P_B y$. Then

$$\begin{aligned} v^* &= \min\{\|Ax-b\| \mid Bx=0\} \\ &= \min\{\|AP_By-b\|\} \\ &= \min\left\{y^t(AP_B)^t(AP_B)y-2b^t(AP_B)y+b^tb\right\}. \end{aligned}$$

Since AP_B is of full rank then $(AP_B)^t(AP_B)$ is positive definite, the objective function is therefore strictly convex and the optimization problem has a unique solution

$$y = \left\{(AP_B)^t(AP_B)\right\}^{-1}(AP_B)b.$$

The result follows by the transformation $x = P_B y$. $\square$

From Lemma 2.2.1 we obtain that the Gauss-Newton direction is well-defined for strictly feasible points under a weak assumption on the primal constraint, that $\mathcal{A}$ is surjective. This assumption, for theoretical purposes, is made without loss of generality since surjectivity of $\mathcal{A}$ is equivalent to the matrices $A_1, \ldots, A_m$ being linearly independent, a condition we enforce by pre-processing of the problem at the onset.

Lemma 2.2.2 *If $\mathcal{A}$ is surjective, the Gauss-Newton direction obtained from (2.9) exists and is unique for all $X \in \mathbb{S}^n_{++}$, $Z \in \mathbb{S}^n_{++}$. (The Jacobian is full-rank.)*

Proof: We show that $[\mathfrak{D}F_\mu(\nu)]$ is full-rank by considering its kernel. We express $[\mathfrak{D}F_\mu(\nu)]d_\nu = 0$ as

$$\mathcal{A}^*(d_y) + d_Z = 0, \tag{2.24a}$$

$$\mathcal{A}(d_X) = 0, \tag{2.24b}$$

$$Zd_X + d_Z X = 0. \tag{2.24c}$$

From (2.24a) we obtain $d_Z = -\mathcal{A}^*(d_y)$ and by (2.24c), $Zd_XX^{-1} = \mathcal{A}^*(d_y)$. Now

$$
\begin{aligned}
0 &= \langle 0, d_y \rangle \\
&= \langle \mathcal{A}(d_X), d_y \rangle \\
&= \langle \mathcal{A}^*(d_y), d_X \rangle \\
&= \left\langle Zd_XX^{-1}, d_X \right\rangle \\
&= \left\langle Z^{\frac{1}{2}}d_XX^{-\frac{1}{2}}, Z^{\frac{1}{2}}d_XX^{-\frac{1}{2}} \right\rangle \\
&= \|Z^{\frac{1}{2}}d_XX^{-\frac{1}{2}}\|^2.
\end{aligned}
$$

The last equation implies $d_X = 0$ since both X and Z are full rank. Substituting back into (2.24c) we get $d_Z = 0$. Finally, the surjectivity of $\mathcal{A}$ yields $d_y = 0$. The result follows by Lemma 2.2.1 with the identification $B = 0$ and $A = [\mathfrak{D}F(\nu)]$. □

Note that d_X and d_Z are always uniquely determined and d_y is uniquely determined if $\mathcal{A}$ is surjective. Failing this condition we may define the search direction as the best least-squares solution of (2.9) to regain uniqueness.

Note also that the result requires only positive definite X and Z. This is in contrast to the AHO direction, which may fail to exist, though sufficient conditions for existence are known. Monteiro and Zanjácomo [59], for example, show that the AHO direction is well-defined if $\|Z^{\frac{1}{2}}XZ^{\frac{1}{2}} - \mu I\| \leq \frac{\mu}{2}$; while Shida, Shindoh, and Kojima [73] show that $ZX + XZ \in \mathbb{S}^n_+$ is sufficient.

Corollary 2.2.3 *If $\mathcal{A}$ is surjective, the feasible Gauss-Newton direction obtained from (2.14) exists and is unique for all $X \in \mathbb{S}^n_{++}$, $Z \in \mathbb{S}^n_{++}$. (The projected Jacobian is full-rank.)*

Proof: Consider Lemma 2.2.1 and identify

$$
A \text{ with } \begin{bmatrix} \mathcal{Z} & 0 & \mathcal{X} \end{bmatrix}, \text{ and } B \text{ with } \begin{bmatrix} 0 & \mathcal{A}^* & \mathcal{I} \\ \mathcal{A} & 0 & 0 \end{bmatrix}.
$$

Let P_B be the projection onto the nullspace of B. We only need to show that AP_B is full rank. By way of contradiction, assume that $AP_By = 0$ with $y \neq 0$. Then $P_By \neq 0$ since P_B is full-rank and therefore

$$\begin{bmatrix} B \\ A \end{bmatrix} P_B y = 0,$$

contradicting Lemma 2.2.2. □

In addition to the existence of the direction when X and Z are positive definite, it seems important for accurate solutions that the non-singularity result holds in the limit. This is true of the AHO direction but not of NT or HKM whose Jacobians become increasingly ill-conditioned as we approach the optimal solution.

To guarantee this desired behavior, we need additional assumptions, on the optimal solution, of uniqueness and strict complementarity. This is not done without loss of generality; these assumptions will fail on some practical problems. Yet, with probability one, randomly generated programs exhibit the required condition as was shown by Alizadeh, Haeberly and Overton [5] (See also [65] for more generic properties of conic programs).

Lemma 2.2.4 *If $\mathcal{A}$ is surjective and the optimal primal-dual solution* (X^*, y^*, Z^*) *is unique and strictly complementary* $(\operatorname{rank}(X^*) + \operatorname{rank}(Z^*) = n)$, *then* $[\mathfrak{D}F_\mu(v)]$ *at* $\mu = 0$ *is non-singular.*

Proof: Since Z^* and X^* commute ($Z^*X^* = 0 = (Z^*X^*)^t = X^*Z^*$), they share an orthonormal matrix of eigenvectors Q such that $QZ^*Q^t = D_Z$ and $QX^*Q^t = D_X$ where D_Z, D_X are diagonal. We construct a permutation matrix P such that

$$PD_ZP^t = \begin{bmatrix} D_{Z1} & 0 & 0 \\ 0 & D_{Z22} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad PD_XP^t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & D_{X3} \end{bmatrix}.$$

Where D_{Z1} and D_{X3} are of same size. We assumed here that the rank of Z^* was at least as large as the rank of X^*. The case where the rank of X^* is larger is similar. Let

$$\widetilde{A_i} := QPA_iP^tQ^t,$$

where $\widetilde{\mathcal{A}}$ is the corresponding operator to obtain a system equivalent to (2.24), namely

$$\begin{aligned} \widetilde{\mathcal{A}}^*(d_y) + \widetilde{d_Z} &= 0 &\quad (2.25a)\\ \widetilde{\mathcal{A}}(\widetilde{d_X}) &= 0 &\quad (2.25b)\\ D_Z\widetilde{d_X} + \widetilde{d_Z}D_X &= 0. &\quad (2.25c) \end{aligned}$$

The respective solutions (d_X, d_y, d_Z) to (2.24) and $(\widetilde{d_X}, d_y, \widetilde{d_Z})$ to (2.25) are related by

$$\begin{aligned} \widetilde{d_X} &:= QPd_XP^tQ^t,\\ \widetilde{d_Z} &:= QPd_ZP^tQ^t. \end{aligned}$$

Consider an expansion of (2.25c),

$$\begin{aligned} 0 &= D_z\widetilde{d_X} + \widetilde{d_Z}D_X \\ &= \begin{bmatrix} D_{Z1} & 0 & 0 \\ 0 & D_{Z2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \widetilde{d_{X11}} & \widetilde{d_{X12}} & \widetilde{d_{X13}} \\ \widetilde{d_{X21}} & \widetilde{d_{X22}} & \widetilde{d_{X23}} \\ \widetilde{d_{X31}} & \widetilde{d_{X32}} & \widetilde{d_{X33}} \end{bmatrix} + \begin{bmatrix} \widetilde{d_{Z11}} & \widetilde{d_{Z12}} & \widetilde{d_{Z13}} \\ \widetilde{d_{Z21}} & \widetilde{d_{Z22}} & \widetilde{d_{Z23}} \\ \widetilde{d_{Z31}} & \widetilde{d_{Z32}} & \widetilde{d_{Z33}} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & D_{X3} \end{bmatrix} \\ &= \begin{bmatrix} D_{Z1}\widetilde{d_{X11}} & D_{Z1}\widetilde{d_{X12}} & D_{Z1}\widetilde{d_{X13}} + \widetilde{d_{Z13}}D_{X3} \\ D_{Z2}\widetilde{d_{X21}} & D_{Z2}\widetilde{d_{X22}} & D_{Z2}\widetilde{d_{X23}} + \widetilde{d_{Z23}}D_{X3} \\ 0 & 0 & \widetilde{d_{Z33}}D_{Xx} \end{bmatrix}. \end{aligned}$$

Therefore $D_{Z1}\widetilde{d_{X11}} = 0$, which implies $\widetilde{d_{X11}} = 0$ and similarly $\widetilde{d_{Z33}} = 0$, $\widetilde{d_{X12}} = 0$, $\widetilde{d_{X22}} = 0$. From the upper right block,

$$\begin{aligned} \widetilde{d_{X13}} &= -D_{Z1}^{-1}\widetilde{d_{Z13}}D_{X3}, \\ \widetilde{d_{X23}} &= -D_{Z2}^{-1}\widetilde{d_{Z23}}D_{X3}. \end{aligned}$$

From (2.25a) and (2.25b) we obtain orthogonality of the primal and dual steps, $\left\langle \widetilde{d}_Z, \widetilde{d}_X \right\rangle = 0$. Therefore

$$\begin{aligned}
0 &= \left\langle \widetilde{d}_Z, \widetilde{d}_X \right\rangle \\
&= \left\langle \begin{bmatrix} \widetilde{d}_{Z11} & \widetilde{d}_{Z12} & \widetilde{d}_{Z13} \\ \widetilde{d}_{Z21} & \widetilde{d}_{Z22} & \widetilde{d}_{Z23} \\ \widetilde{d}_{Z31} & \widetilde{d}_{Z32} & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & \widetilde{d}_{X13} \\ 0 & 0 & \widetilde{d}_{X23} \\ \widetilde{d}_{X31} & \widetilde{d}_{X32} & \widetilde{d}_{X32} \end{bmatrix} \right\rangle \\
&= \operatorname{trace}\left(\widetilde{d}_{Z13}\widetilde{d}_{X13} + \widetilde{d}_{Z23}\widetilde{d}_{X32} + \widetilde{d}_{Z31}\widetilde{d}_{X13} + \widetilde{d}_{Z32}\widetilde{d}_{X23}\right) \\
&= 2\operatorname{trace}\left(\widetilde{d}_{Z13}\widetilde{d}_{X13} + \widetilde{d}_{Z23}\widetilde{d}_{X23}\right) \\
&= -2\operatorname{trace}\left(\widetilde{d}_{Z13}D_{Z1}^{-1}\widetilde{d}_{Z13}D_{X3} + \widetilde{d}_{Z23}D_{Z2}^{-1}\widetilde{d}_{Z23}D_{X3}\right) \\
&= -2\left\{\operatorname{trace}\left(D_{X3}^{\frac{1}{2}}\widetilde{d}_{Z13}D_{Z1}^{-\frac{1}{2}}D_{Z1}^{-\frac{1}{2}}\widetilde{d}_{Z13}D_{X3}^{\frac{1}{2}} + \operatorname{trace}\left(D_{X3}^{\frac{1}{2}}\widetilde{d}_{Z23}D_{Z2}^{-\frac{1}{2}}D_{Z2}^{-\frac{1}{2}}\widetilde{d}_{Z23}D_{X3}^{\frac{1}{2}}\right)\right)\right\} \\
&= -2\left\{\|D_{Z1}^{-\frac{1}{2}}\widetilde{d}_{Z13}D_{X3}^{\frac{1}{2}}\|^2 + \|D_{X3}^{\frac{1}{2}}\widetilde{d}_{Z23}D_{Z2}^{-\frac{1}{2}}\|^2\right\}.
\end{aligned}$$

From the last equation we get $\widetilde{d}_{Z13} = 0$, $\widetilde{d}_{Z23} = 0$ and, similarly $\widetilde{d}_{X31} = 0$, $\widetilde{d}_{X32} = 0$. Finally, the structure of any solution to (2.25) is

$$\widetilde{d}_X = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \widetilde{d}_{X22} \end{bmatrix}, \quad \widetilde{d}_Z = \begin{bmatrix} \widetilde{d}_{Z11} & 0 & 0 \\ 0 & \widetilde{d}_{Z22} & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and therefore $\widetilde{d}_Z\widetilde{d}_X = 0$.

Assume that we have such a solution, $(\widetilde{d}_X, d_y, \widetilde{d}_Z)$. Then

$$(D_Z + \widetilde{d}_Z)(D_X + \widetilde{d}_X) = D_Z D_X + (D_Z\widetilde{d}_X + \widetilde{d}_Z D_X) + \widetilde{d}_Z\widetilde{d}_X = 0.$$

Then $(Z^* + P^tQ^t\widetilde{d}_Z QP, y + d_y, X^* + P^tQ^t\widetilde{d}_X QP)$ is also a solution to the primal-dual pair, assumed to be unique. The only solution to (2.25) and to the equivalent system (2.24) is therefore 0 and the projected Jacobian is full-rank. □

Note that the optimal solution must be unique and strictly complementary, and that $\mathcal{A}$ must be surjective for the above result to hold. If this fails, the Jacobian is rank-deficient. Consequently, if we intend to solve difficult problems accurately we need to consider the possibility of multiple optimal solutions and therefore of singular Jacobians. In practice, surjectivity of $\mathcal{A}$ is guaranteed by pre-processing, for example by doing a rank-revealing QR decomposition of $\mathcal{A}$. Moreover, if the implementation is meant to handle problems with multiple solutions, the obvious approach is to look for the best least-squares solution to the sub-problem, via a rank-revealing decomposition of the operator $[\mathfrak{D}F_\mu(\nu)]$. We will return to this issue when we discuss implementation.

The results of this section imply that the Gauss-Newton direction, similarly to AHO and in contrast to almost all other directions obtained from a symmetric scaled system [79], is well-defined as we approach and also at the optimal solution. This continuity property allows the implementation, if properly done, to obtain accurate solutions, especially in view of the distance to singularity of the Gauss-Newton system. Section 2.2.4 further explores this aspect of the linear system defining the Gauss-Newton direction.

2.2.2 Merit Function

The merit function we use, not only to derive the Gauss-Newton direction but also to gauge the progress of any algorithm is the squared norm of the infeasibility and of the complementarity,

$$\begin{aligned}
\varphi(\nu) &:= \frac{1}{2}\langle F_\mu(\nu), F_\mu(\nu)\rangle && (2.26a)\\
&= \frac{1}{2}\langle F_d, F_d\rangle + \frac{1}{2}\langle f_p, f_p\rangle + \frac{1}{2}\langle F_c, F_c\rangle && (2.26b)\\
&=: \varphi_d(\nu) + \varphi_p(\nu) + \varphi_c(\nu). && (2.26c)
\end{aligned}$$

Following our definition of derivatives, we can calculate the first and second derivatives of φ by expanding $\varphi(\nu + d_\nu)$ around ν,

$$\begin{aligned}
\varphi(\nu + d_\nu) &= \frac{1}{2}\|F(\nu + d_\nu)\|^2 \\
&= \frac{1}{2}\left\| F(\nu) + [\mathfrak{D}F(\nu)](d_\nu) + \frac{1}{2}[\mathfrak{D}^2F(\nu)](d_\nu, d_\nu) + o(\|d_\nu\|^2) \right\|^2 \\
&= \frac{1}{2}\Big\{ \langle F(\nu), F(\nu)\rangle \\
&\quad +2\langle F(\nu), [\mathfrak{D}F(\nu)](d_\nu)\rangle \\
&\quad +\langle F(\nu), [\mathfrak{D}F(\nu)](d)\rangle + \Big\langle [\mathfrak{D}^2F(\nu)](d_\nu, d_\nu), [\mathfrak{D}^2F(\nu)](d_\nu, d_\nu)\Big\rangle \\
&\quad +o(\|d_\nu\|^2)\Big\}.
\end{aligned}$$

From this expansion we obtain the following derivatives,

$$\begin{aligned}
[\mathfrak{D}\varphi(\nu)](d_\nu) &:= \langle F(\nu), [\mathfrak{D}F(\nu)](d_\nu)\rangle, \\
[\mathfrak{D}^2\varphi(\nu)](d_\nu, d_\nu) &:= \Big\langle F(\nu), [\mathfrak{D}^2F(\nu)](d_\nu, d_\nu)\Big\rangle + \langle [\mathfrak{D}F(\nu)](d_\nu), [\mathfrak{D}F(\nu)](d_\nu)\rangle.
\end{aligned}$$

We now specialize these expressions to each part of our merit function. First to the infeasibility measures, starting with primal infeasibility,

$$\begin{aligned}
\varphi_p(X) &:= \frac{1}{2}\langle f_p(X), f_p(X)\rangle \\
&= \frac{1}{2}\langle \mathcal{A}(X) - b, \mathcal{A}(X) - b\rangle, \\
[\mathfrak{D}\varphi_p(X)](d_X) &= \langle \mathcal{A}(X) - b, \mathcal{A}(d_X)\rangle, \\
[\mathfrak{D}^2\varphi_p(X)](d_X, d_X) &= \|\mathcal{A}(d_X)\|^2.
\end{aligned}$$

Then to dual infeasibility,

$$
\begin{aligned}
\varphi_d(y,Z) &:= \frac{1}{2}\langle F_d(y,Z), F_d(y,Z)\rangle \\
&= \frac{1}{2}\langle \mathcal{A}^*(y)+Z-C, \mathcal{A}^*(y)+Z-C\rangle, \\
[\mathfrak{D}\varphi_d(y,Z)](d_y,d_Z) &= \langle \mathcal{A}^*(y)+Z-C, \mathcal{A}^*(d_y)+d_Z\rangle, \\
[\mathfrak{D}^2\varphi_d(y,Z)]((d_y,d_Z),(d_y,d_Z)) &= \|\mathcal{A}^*(d_y)+d_Z\|^2.
\end{aligned}
$$

And to complementarity,

$$
\begin{aligned}
\varphi_c(X,Z) &:= \frac{1}{2}\langle F_c(X,Z), F_c(X,Z)\rangle \\
&= \frac{1}{2}\langle ZX-\mu I, ZX-\mu I\rangle, \\
[\mathfrak{D}\varphi_c(X,Z)](d_X,d_Z) &= \langle ZX-\mu I, Zd_X+d_ZX\rangle, \\
[\mathfrak{D}^2\varphi_c(X,Z)]((d_X,d_Z),(d_X,d_Z)) &= 2\langle F_c, d_Z d_X\rangle + \|Zd_X+d_ZX\|^2.
\end{aligned}
$$

2.2.3 Descent

We present here a classical result that is found, informally stated, in Dennis and Schnabel [20] but which we include here because our setting is more general and because it provides the original motivation for the use of the Gauss-Newton direction to solve semidefinite programs.

Lemma 2.2.5 *The Gauss-Newton direction* d_ν*, defined by (2.9) is a strict descent direction for the merit function* $\varphi(\nu) = \frac{1}{2}\langle F_\mu(\nu), F_\mu(\nu)\rangle$ *if and only if* $F_\mu(\nu)$ *is not perpendicular to the range of* $[\mathfrak{D}F_\mu(\nu)]$.

Proof: We compute the derivative of f in the direction of d_ν as

$$
\begin{aligned}
[\mathfrak{D}\varphi(\nu)]d_\nu &= \langle F_\mu(\nu), [\mathfrak{D}F_\mu(\nu)]d_\nu\rangle \\
&= \left\langle F_\mu(\nu), -[\mathfrak{D}F_\mu(\nu)][\mathfrak{D}F_\mu(\nu)]^\dagger F_\mu(\nu)\right\rangle.
\end{aligned}
$$

Now we observe that $[\mathfrak{D}F_\mu(\nu)][\mathfrak{D}F_\mu(\nu)]^\dagger$ is the orthogonal projection onto the range of $[\mathfrak{D}F_\mu(\nu)]$. It is therefore idempotent and we can write

$$\begin{aligned}
[\mathfrak{D}\varphi(\nu)]d_\nu &= -\Big\langle F_\mu(\nu), ([\mathfrak{D}F_\mu(\nu)][\mathfrak{D}F_\mu(\nu)]^\dagger)^*([\mathfrak{D}F_\mu(\nu)][\mathfrak{D}F_\mu(\nu)]^\dagger)F_\mu(\nu)\Big\rangle \\
&= -\Big\langle [\mathfrak{D}F_\mu(\nu)][\mathfrak{D}F_\mu(\nu)]^\dagger F_\mu(\nu), [\mathfrak{D}F_\mu(\nu)][\mathfrak{D}F_\mu(\nu)]^\dagger F_\mu(\nu)\Big\rangle \\
&= -\|[\mathfrak{D}F_\mu(\nu)][\mathfrak{D}F_\mu(\nu)]^\dagger F_\mu(\nu)\|^2 \\
&\le 0.
\end{aligned}$$

Therefore $[\mathfrak{D}\varphi(\nu)]d < 0$ if and only if $[\mathfrak{D}F_\mu(\nu)][\mathfrak{D}F_\mu(\nu)]^\dagger F_\mu(\nu) \neq 0$ and the result follows. □

In particular, Lemma 2.2.5 states that if $[\mathfrak{D}F_\mu(\nu)]$ is full rank then $[\mathfrak{D}\varphi(\nu)]d_\nu < 0$ and d_ν is a direction of strict descent. The feasible direction also enjoys a similar property.

Lemma 2.2.6 *The feasible Gauss-Newton direction* d_ν *defined by (2.14) is a descent direction for the merit function* $\varphi_c(\nu) = \frac{1}{2}\langle F_c, F_c\rangle$.

Proof: Using the notation and the result of Lemma 2.2.1, say $\mathcal{P}_B$ is the projection onto the nullspace of the operator corresponding to the feasibility $\mathcal{A}^*(d_y) + d_Z = 0, \mathcal{A}(d_X) = 0$. Then the feasible Gauss-Newton direction is expressed by

$$d_\nu = -\mathcal{P}_B\left([\mathfrak{D}F_c(\nu)]\mathcal{P}_B\right)^\dagger F_c(\nu). \tag{2.28}$$

Similarly to Lemma 2.2.5, we have

$$\begin{aligned}
[\mathfrak{D}\varphi_c(\nu)]d_\nu &= \langle F_c(\nu), [\mathfrak{D}F_c(\nu)]d_\nu\rangle \\
&= \Big\langle F_c(\nu), -[\mathfrak{D}F_c(\nu)]\mathcal{P}_B([\mathfrak{D}F_c(\nu)]\mathcal{P}_B)^\dagger F_c(\nu)\Big\rangle \\
&= -\|[\mathfrak{D}F_c(\nu)]\mathcal{P}_B([\mathfrak{D}F_c(\nu)]\mathcal{P}_B)^\dagger F_c(\nu)\|^2 \\
&\le 0.
\end{aligned}$$

Moreover, $[\mathfrak{D}f_c(\nu)]d_\nu < 0$ if and only if $\mathcal{P}_B([\mathfrak{D}F_c(\nu)]\mathcal{P}_B)^\dagger]F_c(\nu) \neq 0$. □

We expect the following to be a building tool for the global convergence analysis of any algorithm based on the Gauss-Newton direction.

Corollary 2.2.7 *If $\mathcal{A}$ is surjective, then both the Gauss-Newton direction defined by (2.9) and the feasible Gauss-Newton direction defined by (2.14) are directions of strict descent for all $X \in \mathbb{S}^n_{++}$, $Z \in \mathbb{S}^n_{++}$ for, respectively, $\frac{1}{2}\langle F_\mu(\nu), F_\mu(\nu)\rangle$ and $\frac{1}{2}\langle F_c(\nu), F_c(\nu)\rangle$.*

Proof: From lemmata 2.2.5 and 2.2.6 we have descent if the Jacobian is full-rank. From Lemmata 2.2.3, 2.2.3 and the hypotheses we have the required full-rank property. □

In summary, if $\mathcal{A}$ is surjective, and we may assume it is, the Gauss-Newton direction is a strict descent direction until stationarity of the merit function is attained.

2.2.4 Conditioning of the Jacobian

In this section we investigate the behavior of the singular values of the Gauss-Newton Jacobian, first, we compare them to the corresponding singular values of the AHO Jacobian to estimate the relative distance of both systems to singularity. Then we find expressions for their rate of change with respect to the barrier parameter μ.

Consider the following equivalent form of the over-determined system (2.6),

$$\mathcal{A}^*(d_y) + d_Z = -(\mathcal{A}^*(y) + Z - C), \tag{2.29a}$$

$$\mathcal{A}(d_X) = -(\mathcal{A}(X) - b), \tag{2.29b}$$

$$H(Zd_X + d_Z X) = -H(ZX - \mu I), \tag{2.29c}$$

$$K(Zd_X + d_Z X) = -K(ZX - \mu I), \tag{2.29d}$$

where

$$H(M) := \frac{1}{2}[M + M^t], \quad \text{(symmetric part);} \tag{2.30a}$$

$$K(M) := \frac{1}{2}[M - M^t], \quad \text{(skew-symmetric part).} \tag{2.30b}$$

Note that the first three equations (2.29a-2.29c) correspond to the symmetric (or AHO) system. Corresponding to the matrix formulation (2.10) we write

$$J_{gn} d = P \begin{bmatrix} J_{aho} \\ J_k \end{bmatrix} d = - \begin{bmatrix} F_s \\ F_k \end{bmatrix}, \tag{2.31}$$

for some permutation P of the rows, and where J_{gn} is the Jacobian of the Gauss-Newton system, J_{aho} is the Jacobian of the symmetric (AHO) system and J_k is the part of J_{gn} corresponding to the skew-symmetric component of the complementarity equation. To simplify the notation later, let

$$\bar{n} := t(n) + m + t(n), \tag{2.32}$$

$$\bar{m} := t(n) + m + n^2, \tag{2.33}$$

$$r := \bar{m} - \bar{n} = t(n) - t(n-1). \tag{2.34}$$

Note that J_{gn} is $(\bar{m} \times \bar{n})$, J_{aho} is $(\bar{n} \times \bar{n})$, and J_k is $(r \times \bar{n})$. We also use the following notation for the ordering of the singular values of a matrix $J_{m \times n}$, with $m \geq n$

$$\sigma_{\max}(J) := \sigma_1(J) \geq \sigma_2(J) \geq \ldots \geq \sigma_n(J) =: \sigma_{\min}(J).$$

From (2.31) and the above notation for singular values, we can write the following relation.

Lemma 2.2.8 *The singular values of* J_{gn} *and* J_{aho} *satisfy the following inequality for* $1 \leq k \leq \bar{n}$,

$$\sigma_k(J_{gn}) \geq \sigma_k(J_{aho}) \geq \sigma_{k+t(n-1)}(J_{gn}).$$

Proof: Follows directly from Corollary 3.1.3 of Horn and Johnson [44]. □

The principal implication of the result is that the Gauss-Newton Jacobian is no closer to singularity than the AHO Jacobian is since $\sigma_{\min}(J_{gn}) \geq \sigma_{\min}(J_{aho})$. We now consider the largest singular value and obtain an upper bound.

Lemma 2.2.9 *The largest singular value of the Gauss-Newton Jacobian is bounded. Specifically,* $\sigma_{\max}(J_{gn}) \leq \sqrt{\sigma^2_{\max}(J_{aho}) + \sigma^2_{\max}(J_k)}$.

Proof: With the above relations (2.31) between J_{gn}, P, J_{aho}, and J_k,

$$
\begin{aligned}
\sigma^2_{\max}(J_{gn}) &= \lambda_{\max}(J^t_{gn}J_{gn}) \\
&= \lambda_{\max}\left([(PJ_{aho})^t (PJ_k)^t] \begin{bmatrix} PJ_{aho} \\ PJ_k \end{bmatrix} \right) \\
&= \lambda_{\max}(J^t_{aho}J_{aho} + J^t_k J_k) \\
&= \|J^t_{aho}J_{aho} + J^t_k J_k\|_2 \\
&\leq \|J^t_{aho}J_{aho}\|_2 + \|J^t_k J_k\|_2 \\
&= \lambda_{\max}(J^t_{aho}J_{aho}) + \lambda_{\max}(J^t_k J_k) \\
&= \sigma^2_{\max}(J_{aho}) + \sigma^2_{\max}(J_k).
\end{aligned}
$$

Therefore $\sigma_{\max}(J_{gn}) \leq \sqrt{\sigma^2_{\max}(J_{aho}) + \sigma^2_{\max}(J_k)}$. $\square$

We can now investigate the condition number of the Gauss-Newton system. Say we have a sequence $v^{(k)} := (X^{(k)}, y^{(k)}, Z^{(k)}) \in \mathbb{S}^n_{++} \times \mathbb{R}^m \times \mathbb{S}^n_{++}$. Let $J^{(k)}$ be J_{gn} as defined in (2.10) with X, Z replaced with $X^{(k)}, Z^{(k)}$ and let J^* be the Jacobian at the optimal solution.

Assumptions 2.2.1 *We have an index set denoted by* k *and*

- $(X^{(k)}, y^{(k)}, Z^{(k)}) \longrightarrow (X^*, y^*, Z^*)$,
- (X^*, y^*, Z^*) *is the unique, strictly complementary optimal solution.*

We use the notation $v^{(k)}$ instead of v_μ to indicate a sequence not restricted to the central path. First, a simple technical result.

Lemma 2.2.10 *Under Assumptions 2.2.1,*

1. $J^{(k)} \to J^*$,
2. $\{\|J^{(k)}\|\}$ *is bounded.*

Proof: Result 2 follows directly from 1 and the fact that $\|J^*\|$ is bounded. We proceed to show 1. Since $X^{(k)} \to X^*$ and $Z^{(k)} \to Z^*$, for any $\epsilon > 0$, there is a $\bar{k}$ such that for any $k \geq \bar{k}$,

$$\|X^{(k)} - X^*\| \leq \frac{\epsilon}{2\sqrt{n}} \quad \text{and} \quad \|Z^{(k)} - Z^*\| \leq \frac{\epsilon}{2\sqrt{n}}.$$

With this choice of $k \geq \bar{k}$,

$$\begin{aligned}
\|J^{(k)} - J^*\| &= \left\| \begin{bmatrix} I \oslash (Z^{(k)} - Z^*) & 0 & (X^{(k)} - X^*) \oslash I \end{bmatrix} \right\| \\
&\leq \sqrt{n} \left\| \begin{bmatrix} Z^{(k)} - Z^* & X^{(k)} - X^* \end{bmatrix} \right\| \\
&\leq \sqrt{n}(\|Z^{(k)} - Z^*\| + \|X^{(k)} - X^*\|) \\
&\leq \epsilon.
\end{aligned}$$

And we obtain the required bound. □

Lemma 2.2.11 *Under Assumptions 2.2.1, The condition number of the Gauss-Newton Jacobian* $\kappa(J^{(k)}) := \frac{\sigma_{\max}(J^{(k)})}{\sigma_{\min}(J^{(k)})}$ *satisfies* $\kappa(J^{(k)}) \to \kappa(J^*) < \infty$.

By Lemma 2.2.2, the operator $J^{(k)}$ is of full rank which implies that the smallest singular value is bounded away from zero. By Lemma 2.2.10, the largest singular value is bounded away from infinity. □

By Lemmata 2.2.2 and 2.2.4, we know that the Gauss-Newton system is non-singular and has a bounded condition number as we approach the optimal solution. This behavior is shared with the AHO direction but not with NT or HKM, as any random example shows. The condition number of the NT and HKM systems, even on small, random problems, grows dramatically.

To see why the condition number might affect accuracy, we restate here the results of Gu [38]. Consider symmetric systems, under finite precision arithmetic and assume a backward-stable algorithm for the solution of the direction-finding system (1.20). The numerical solution $\widehat{d_\nu}$ to (1.20) is the exact solution to a nearby problem,

$$(J_s - \delta J_s)\widehat{d_\nu} = -(f + \delta f),$$

where J_s is the scaled Jacobian and where the perturbations vary with the choice of directions and the solution technique. In all cases, the computed solution $\widehat{d_\nu}$ and the exact solution d_ν may differ by

$$\frac{\|\widehat{d_\nu} - d_\nu\|}{\|d_\nu\|} \leq \frac{\kappa(J_s)}{1 - \kappa(J_s)\frac{\|\delta J_s\|}{\|J_s\|}} \left(\frac{\|\delta J_s\|}{\|J_s\|} + \frac{\|\delta f\|}{\|f\|} \right). \tag{2.35}$$

Therefore if $\|\delta J_s\| = \Omega(\sigma_{\min}(J_s))$, the computed direction may be completely different from the exact direction and the algorithm will stop making progress. A better condition number allows more accurate solutions. This is the accepted explanation for the fact that the AHO direction obtains much more accurate solutions than HKM and NT. For the Gauss-Newton direction, the perturbation analysis is slightly different and we return to it in chapter 4 when we discuss implementation but again, there is a dependence on $\kappa(J_{gn})$ and the better the condition number, the more accurate the solution.

Moreover the condition number of the Gauss-Newton system on most problems is smaller than the condition number of the AHO system. Informally, this is not surprising. Recall that, since both Jacobians have bounded norms, conditioning problems may occur only when the smallest singular value gets too small. Consider a problem where the smallest singular value is, in the limit, very small and recall that the difference between the two systems is that the skew-symmetric part of the complementarity equation is deleted in AHO. For any matrix A,

$$\sum \sigma_i^2(A) = \frac{1}{2} \sum \sigma_i^2(A + A^t) + \frac{1}{2} \sum \sigma_i^2(A - A^t).$$

Unless by some coincidence, the skew-symmetric part affects only the larger singular values, then the smallest singular value of the Gauss-Newton system is strictly larger than the AHO smallest singular value and the condition number is correspondingly smaller.

Now, we consider some instances where the Gauss-Newton direction coincides with other directions.

2.2.5 Coincidences

Recall that semidefinite programming can be viewed as a superset of linear programming. If the data involve only diagonal matrices (A_i, C), the primal-dual pair is an expression of a standard linear program. Therefore, given a point (X, y, Z) where the matrices are diagonal and restricted to be diagonal, one would hope that the Gauss-Newton direction coincides with the usual linear programming primal-dual direction. This was shown to be true for all of the Monteiro-Zhang family by Todd [79]. This is also the case for the Gauss-Newton direction.

Lemma 2.2.12 *Given constraint matrices* A_i *linearly independent and diagonal, objective function matrix* C *diagonal, and a current iterate* (X, y, Z) *with diagonal* $X \in \mathbb{S}^n_{++}$ *and* $Z \in \mathbb{S}^n_{++}$*, the Gauss-Newton direction* $d_v = (d_X, d_y, d_Z)$ *has* d_X *and* d_Z *diagonal. Moreover* $(\mathrm{diag}\,(d_X), d_y, \mathrm{diag}\,(d_Z))$ *solves the standard primal-dual Linear Programming system, namely*

$$\begin{aligned} A^t d_y + d_z &= -(A^t y + z - c), && (2.36a)\\ A d_x &= -(Ax - b), && (2.36b)\\ Z d_x + X d_z &= -(Zx - \mu e). && (2.36c) \end{aligned}$$

Proof: From the solution to (2.36) we construct $d_X = \mathrm{Diag}\,(d_x)$, $d_Z = \mathrm{Diag}\,(d_z)$, clearly a solution to the corresponding Gauss-Newton system (2.5) since the residuals are zero. Moreover, since the solution to the Gauss-Newton system is unique by Lemma 2.2.2, this choice of d_X, d_y, d_Z is the Gauss-Newton solution. Therefore the Gauss-Newton direction coincides with the standard primal-dual linear programming direction. $\square$

On the central path of semidefinite programs, the Gauss-Newton direction coincides with other well-known directions.

Lemma 2.2.13 *Suppose that* $\mathcal{A}$ *is surjective, that* X, y, Z *is on the central path with* $F_d = 0$*,* $f_p = 0$ *and* $ZX - \mu I = 0$*,* $\mu > 0$*. Suppose that the new target for the barrier parameter is* $\tau\mu$*, where* $0 < \tau < 1$*. Then the Gauss-Newton direction from (2.5) coincides with the HKM direction.*

Proof: Following [41, 48, 56] we express the HKM direction in its simplest form as the solution of (2.37) followed by symmetrization $d_X = \frac{1}{2}(d_X + d_X^t)$.

$$\begin{aligned} \mathcal{A}^*(d_y) + d_Z &= 0 && (2.37a)\\ \mathcal{A}(d_X) &= 0 && (2.37b)\\ Zd_X + d_Z X &= -(1-\tau)\mu I. && (2.37c) \end{aligned}$$

The first equation (2.37a) yields

$$d_Z = -\mathcal{A}^*(d_y). \tag{2.38}$$

This expression for d_Z implies that it is symmetric since the A_i are symmetric. We solve for d_X from the last equation (2.37c) to get

$$\begin{aligned} d_X &= -(1-\tau)\mu Z^{-1} - Z^{-1} d_Z X\\ &= -(1-\tau)X - \frac{1}{\mu} X d_Z X\\ &= (\tau - 1)X + \frac{1}{\mu} X \mathcal{A}^*(d_y) X. \end{aligned}$$

This, in turn, implies symmetry of d_X and the next step of an HKM approach, namely the symmetrization $d_X = (d_X + d_X^t)/2$, is not required. Therefore, the solution to (2.37) is a solution to the AHO system where equation (2.37c) is replaced by

$$\frac{1}{2}(Zd_X + d_Z X + d_X Z + X d_Z) = -(1-\tau)\mu I.$$

(This implies the known result that AHO and HKM coincide on the central path.) For the HKM direction to be equal to the Gauss-Newton direction, we have to check the additional condition that the skew-symmetric part is zero.

$$\begin{aligned} Zd_X - d_X Z + d_Z X - X d_Z &= \mu X^{-1}[(\tau-1)X + \frac{1}{\mu} X\mathcal{A}^*(d_y)X] \\ &\quad -[(\tau-1)X + \frac{1}{\mu} X\mathcal{A}^*(d_y)X]\mu X^{-1} \\ &\quad -\mathcal{A}^*(d_y)X + X\mathcal{A}^*(d_y) \\ &= 0. \end{aligned}$$

And therefore the solution to (2.37) is a valid solution to the Gauss-Newton system, which is known to be unique. □

Since it was shown by Todd [79] that most directions (including AHO, NT and HKM) coincide on the central path, from Lemma 2.2.13 we can now add the Gauss-Newton direction to this list. This is the only case where the Gauss-Newton direction coincides with other directions of the Monteiro-Zhang family. In general, the Gauss-Newton direction is different from all other directions.

2.2.6 Invariance

Todd [79] also introduced two concepts of scale-invariance with respect to the cone of positive definite matrices. This is different from the classical concept of scale-invariance with respect to the affine space defined by the constraint equations. A method for defining a search direction is *P-scale-invariant* if the direction at any iterate is the same as would result from scaling the problem and the iterate by an arbitrary non-singular P, using the method to determine the direction, and then scaling back.

In more detail, say that from an iterate (X, y, Z), and data A_i, C, b, a method finds a direction (d_X, d_y, d_Z). Also, given a scaling matrix P, the same method, applied to a problem from iterate $(PXP^t, y, P^{-t}ZP^{-1})$ and data $P^{-t}A_iP^{-1}, P^{-t}CP^{-1}$ finds direction $(\widetilde{d_X}, \widetilde{d_y}, \widetilde{d_Z})$, then the direction is P-scale-invariant if the directions agree, that is, if $(Pd_XP^t, d_y, P^{-t}d_ZP^{-1}) = (\widetilde{d_X}, \widetilde{d_y}, \widetilde{d_Z})$. It is *Q-scale-invariant* if the same relation holds when P is restricted to orthogonal matrices, $PP^t = I$. As this concept applies to the Gauss-Newton direction, we have the following result.

Lemma 2.2.14 *(Q-scale invariance) Let* (d_X, d_y, d_Z) *be the Gauss-Newton direction obtained at point* $(X, y, Z) \in \mathbb{S}^n_{++} \times \mathbb{R}^m \times \mathbb{S}^n_{++}$. *Consider the scaled primal-dual pair, (1.17-1.18) obtained from (1.16), (1.19) for some orthogonal* P. *Then the scaled vector* $(Pd_XP^t, y, P^{-t}d_ZP^{-1})$ *is the Gauss-Newton direction at the iterate* $(\widetilde{X}, \widetilde{y}, \widetilde{Z})$ *for the scaled problem.*

Proof: The Gauss-Newton direction may be computed from the Normal Equations since the Jacobian is of full rank by Lemma 2.2.2. The defining equations for the scaled problem therefore are

$$\begin{aligned}
(\widetilde{\mathcal{A}}^*\widetilde{\mathcal{A}} + \widetilde{\mathcal{Z}}^*\widetilde{\mathcal{Z}})d_X + (\widetilde{\mathcal{Z}}^*\widetilde{\mathcal{X}})d_Z &= -(\widetilde{\mathcal{A}}^*\widetilde{f}_p + \widetilde{\mathcal{Z}}^*\widetilde{F}_c), &(2.39a)\\
(\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}^*)d_y + \widetilde{\mathcal{A}}d_Z &= -(\widetilde{\mathcal{A}}\widetilde{F}_d), &(2.39b)\\
(\widetilde{\mathcal{X}}^*\widetilde{\mathcal{Z}})d_X + \widetilde{\mathcal{A}}^*d_y + (I + \widetilde{\mathcal{X}}^*\widetilde{\mathcal{X}})d_Z &= -(\widetilde{F}_d + \widetilde{\mathcal{X}}^*\widetilde{F}_c). &(2.39c)
\end{aligned}$$

In the following, $[\langle A_j, B\rangle]_j$ indicates the vector in $\mathbb{R}^m$ made from the inner products.

Substitute the scaled vector $(\widetilde{d_X}, \widetilde{d_y}, \widetilde{d_Z}) = (Pd_XP^t, y, P^{-t}ZP^{-1})$ into the left-hand side of (2.39a) to get

$$\begin{aligned}
&(\widetilde{\mathcal{A}}^*\widetilde{\mathcal{A}} + \widetilde{\mathcal{Z}}^*\widetilde{\mathcal{Z}})\widetilde{d_X} + (\widetilde{\mathcal{Z}}^*\widetilde{\mathcal{X}})\widetilde{d_Z}\\
&= \sum_{i=1}^m P^{-t}A_iP^{-1}[\langle P^{-t}A_iP^{-1}, P^{-t}d_XP^{-1}\rangle]\\
&+ \frac{1}{2}\{P^{-t}ZP^{-1}P^{-t}ZP^{-1}Pd_XP^t + P^{-t}ZP^{-1}P^{-t}ZP^{-1}Pd_XP^t\}\\
&+ \frac{1}{2}\{P^{-t}ZP^{-1}P^{-t}d_ZP^{-1}Pd_XP^t + PXP^tP^{-t}d_ZP^{-1}P^{-t}ZP^{-1}\}\\
&= P^{-t}\left\{\sum_{i=1}^m A_i\langle A_i, d_X\rangle\right\} + \frac{1}{2}\{ZZd_X + ZZd_X + Zd_X + Xd_ZZ\}P^{-1}\\
&= P^{-t}\{\mathcal{A}^*f_p + \mathcal{Z}^*F_c\}P^{-1} = P^{-t}\left\{\sum_{i=1}^m A_if_{pi} + \frac{1}{2}\{ZF_c + F_c^tZ\}\right\}P^{-1}\\
&= \sum_{i=1}^m P^{-t}A_iP^{-1}f_{pi} + \frac{1}{2}\{P^{-t}ZP^{-1}P^{-t}F_cP^{-1} + P^{-t}F_c^tP^{-1}P^{-t}ZP^{-1}\}\\
&= \widetilde{\mathcal{A}}^*\widetilde{f}_p + \widetilde{\mathcal{Z}}^*\widetilde{F}_c.
\end{aligned}$$

Therefore $(\widetilde{d_X}, \widetilde{d_y}, \widetilde{d_Z})$ satisfies (2.39a). Substitute $(\widetilde{d_X}, \widetilde{d_y}, \widetilde{d_Z})$ into the left-hand side of (2.39b) to get, for $1 \le j \le m$,

$$\begin{aligned}
\widetilde{\mathcal{A}}\widetilde{\mathcal{A}^*}\widetilde{d_y} + \widetilde{\mathcal{A}}\widetilde{d_Z} &= [\left\langle P^{-t}A_jP^{-1}, \sum_{i=1}^m P^{-t}A_iP^{-1}(d_y)_i \right\rangle]_j + [\langle P^{-t}A_jP^{-1}, P^{-t}d_ZP^{-1}\rangle]_j \\
&= [\left\langle A_j, \sum_{i=1}^m A_i(d_y)_i \right\rangle]_j + [\langle A_j, d_Z\rangle]_j = -[\langle A_j, F_{di}\rangle]_j \\
&= -[\langle P^{-t}A_jP^{-1}, P^{-t}F_{di}P^{-1}\rangle]_j = -\widetilde{\mathcal{A}}\widetilde{F_d}.
\end{aligned}$$

Therefore $(\widetilde{d_X}, \widetilde{d_y}, \widetilde{d_Z})$ satisfies (2.39b).

Finally, substitute $(\widetilde{d_X}, \widetilde{d_y}, \widetilde{d_Z})$ into the left-hand side of (2.39c) to get,

$$\begin{aligned}
&(\widetilde{\mathcal{X}}^*\widetilde{\mathcal{Z}})d_X + \widetilde{\mathcal{A}}^*d_y + (I + \widetilde{\mathcal{X}}^*\widetilde{\mathcal{X}})d_Z \\
&\quad = \frac{1}{2}\left\{P^{-t}ZP^{-1}Pd_XP^tPXP^t + PXP^tP^{-t}ZP^{-1}Pd_XP^t\right\} \\
&\quad + \sum_{i=1}^m P^{-t}A_iP^{-1}(d_y)_i + P^{-t}d_ZP^{-1} \\
&\quad + \frac{1}{2}\left\{P^{-t}d_ZP^{-1}PXP^tPXP^t + PXP^tPXP^tP^{-t}d_ZP^{-1}\right\} \\
&\quad = P^{-t}\left\{\frac{1}{2}\{Zd_XX + XZd_X\} + \sum_{i=1}^m A_i(d_y)_i + d_Z + \frac{1}{2}\{d_ZXX + XXd_Z\}\right\}P^{-1} \\
&\quad = -P^{-t}F_d + \mathcal{X}^*F_cP^{-1} = -P^{-t}\left\{F_d + \frac{1}{2}\{F_cX + XF_c\}\right\}P^{-1} \\
&\quad = -\left\{P^{-t}F_dP^{-1} + \frac{1}{2}\left\{P^{-t}F_cP^{-1}PXP^t + PXP^tP^{-t}F_c^tP^{-1}\right\}\right\} \\
&\quad = -\widetilde{F_d} + \widetilde{\mathcal{X}}^*\widetilde{F_c}.
\end{aligned}$$

Therefore $(\widetilde{d_X}, \widetilde{d_y}, \widetilde{d_Z})$ satisfies (2.39c) and we conclude that the Gauss-Newton direction is Q-scale invariant. □

While we are describing properties initially investigated by Todd, it is fair to remark that the Gauss-Newton direction is not P-scale invariant as any random example shows. Yet, on a related matter, one additional property of the Gauss-Newton direction is worth mentioning: it is invariant under affine transformation of the variables. This is the classical invariance with respect to the space.

Theorem 2.2.15 *The Gauss-Newton direction, at any point $v_c \in \mathbb{V}$ is invariant under affine transformation of the variables $w = \mathsf{H}(v) + h$ where $\mathsf{H} : \mathbb{V} \to \mathbb{V}$ is non-singular.*

Proof: The Gauss-Newton step in v-space, from current point v_c, is given by (2.8),

$$v_+ = v_c - [\mathfrak{D}_v \mathsf{F}_\mu(v_c)]^\dagger \mathsf{F}_\mu(v_c).$$

Using the affine scaling $w = \mathsf{H}(v) + h$, we define

$$\mathsf{G}(w) := \mathsf{F}_\mu(\mathsf{H}^{-1}(v - h)) = \mathsf{F}_\mu(v),$$

and obtain

$$[\mathfrak{D}_w \mathsf{G}(w)] = [\mathfrak{D}_w \mathsf{F}_\mu(\mathsf{H}^{-1}(w - h))] = [\mathfrak{D}_v \mathsf{F}_\mu(v)]\mathsf{H}^{-1}$$

The Gauss-Newton step, in w-space is

$$\begin{aligned} w_+ &= w_c - [\mathfrak{D}_w \mathsf{G}(w_c)]^\dagger \mathsf{G}(w_c) \\ &= w_c - \left\{[\mathfrak{D}_v \mathsf{F}_\mu(v_c)]\mathsf{H}^{-1}\right\}^\dagger \mathsf{G}(w_c). \end{aligned}$$

Since $[\mathfrak{D}_v \mathsf{F}_\mu(v_c)]$ is of full rank, $\left\{[\mathfrak{D}_v \mathsf{F}_\mu(v_c)]\mathsf{H}^{-1}\right\}^\dagger = \mathsf{H}[\mathfrak{D}_v \mathsf{F}(v_c)]^\dagger$ and

$$\begin{aligned} w_+ &= w_c - \mathsf{H}[\mathfrak{D}_v \mathsf{F}(v_c)]^\dagger \mathsf{G}(w_c) \\ &= \mathsf{H}(v_c) + h - \mathsf{H}[\mathfrak{D}_v \mathsf{F}(v_c)]^\dagger \mathsf{F}(v_c) \\ &= \mathsf{H}(v_+) + h. \end{aligned}$$

The Gauss-Newton step is therefore invariant under affine transformations of the space. $\square$

This last property is not shared by the AHO direction since, for example, H may map v to a point w where the AHO direction is not defined. Note also that this last invariance does not imply that scaling the rows of the operator will leave the steps unchanged. Scaling the feasibility rows by a large factor, for example, would favour feasibility over complementarity and would bring the Gauss-Newton iterates closer to the AHO iterates. This is not a useful goal; it hinders convergence in most cases. To find a scaling of the rows producing more accuracy by reducing the condition number or allowing faster convergence, on the other hand, can be useful and we are currently investigating that issue.

In this chapter we have studied some properties of the Gauss-Newton directions. We have seen that they differ from the symmetric directions of the Monteiro-Zhang family, especially in terms of the distance to singularity of the Jacobian matrix. This feature suggests that the directions should be used to obtain accurate solutions to semidefinite programs. We now develop such algorithms.

Chapter 3

Convergence

In this chapter we discuss convergence issues of algorithms based on the Gauss-Newton directions. First, from a classical stand-point, we consider global convergence via sufficient decrease of a merit function. The results of this approach do not lead to a proof of polynomial convergence. We include them because the algorithm described here is very close to what we do in practice, because the results require only weak conditions of the problem data and because we hope to eventually obtain a proof of polynomial convergence using this line of reasoning. We then briefly discuss asymptotic convergence. Finally, the major part of the chapter takes steps towards a polytime convergence proof for an infeasible interior-point algorithm based on the Gauss-Newton direction.

We recall the problem of interest, the semidefinite program pair

$$(\mathfrak{Primal}) \qquad \min\left\{\langle C, X\rangle \mid \mathcal{A}(X) = b, X \in \mathbb{S}^n_+\right\}, \tag{3.1}$$

and

$$(\mathfrak{Dual}) \qquad \max\left\{\langle b, y\rangle \mid \mathcal{A}^*(y) + Z = C, Z \in \mathbb{S}^n_+\right\}. \tag{3.2}$$

We also recall the merit function, the combined norm of the infeasibility and of the complementarity,

$$\varphi(\nu, \mu) := \frac{1}{2}\langle F_\mu(\nu), F_\mu(\nu)\rangle. \tag{3.3}$$

3.1 Classical Convergence

The algorithms we consider in this section fall into the framework of Algorithm 3.1.1: Starting from a possibly infeasible point, $(X^0, y^0, Z^0) = (I, 0, I)$, compute the Gauss-Newton direction and take a step in that direction, reduce the target parameter and iterate. We assume that the length of the step is dictated by a line search routine that finds the minimum of the merit function in the direction given without violating the semidefinite constraint. It may be possible to replace this exact line search with an approximate line search satisfying the Armijo-Goldstein or Wolfe conditions. The reduction in the barrier parameter ensures that the merit function is decreased at every iteration.

Algorithm 3.1.1 Generic Gauss-Newton based interior-point code

Given $\epsilon > 0$	{Tolerance}
$k := 0; X^{(k)} := I; Z^{(k)} := I; y^{(k)} := 0; \mu^{(k)} := \frac{\langle Z^{(k)}, X^{(k)}\rangle}{n} = 1;$	
while $\neg$Converged$(k, \nu^{(k)}, \mu^{(k)}, \epsilon)$ **do**	
$d^{(k)} = -[\mathfrak{D}F_\mu(\nu^{(k)})]^\dagger F_\mu(\nu^{(k)});$	{Gauss-Newton}
$\alpha^{(k)} :=$ LineSearch$(\nu^{(k)}, d^{(k)}, \mu^{(k)});$	{Decrease φ}
$\mu^{(k+1)} :=$ TargetSelect$(\nu^{(k)}, d^{(k)}, \alpha^{(k)}, \mu^{(k)});$	{See Algorithm 3.1.2}
$\nu^{(k+1)} := \nu^{(k)} + \alpha^{(k)} d^{(k)};$	{New iterate}
$k := k + 1;$	
end while	

The numerical convergence criteria should be based on the merit function, the iterates and on the central path target. Following standard practice ([30], Section 8.2.3.2) we assume that the user provides a value ϵ indicating the desired accuracy. The convergence test ensures

- $\varphi(\nu^{(k-1)}, \mu^{(k-1)}) - \varphi(\nu^{(k)}, \mu^{(k)}) < \epsilon\left(1 + \left|\varphi(\nu^{(k)}, \mu^{(k)})\right|\right),$
- $\left\|\nu^{(k-1)} - \nu^{(k)}\right\| < \sqrt{\epsilon}\left(1 + \left\|\nu^{(k)}\right\|\right),$
- $\mu^{(k)} < \epsilon.$

The first two conditions imply convergence of the merit function values and convergence of the iterates. The third condition implies that we have reduced the complementarity to the required tolerance. Note that there is no need to scale the merit function before comparing to ϵ since we are solving a least-squares problem with zero residual.

Assumptions 3.1.1 *Throughout this section, the following conditions are assumed to hold:*

- *The operator $\mathcal{A}$ is surjective.*
- *There is a point $\nu^* \in \mathbb{V}$ such that $\varphi(\nu^*, 0) = 0$.*

The first condition, for theoretical purposes, is made without loss of generality since we can in principle eliminate redundant constraints before attempting to solve the optimization problem. The second condition implies that the primal-dual pair has an optimal solution with no duality gap. Note that we do not insist that the optimal solution be strictly complementary or even unique. Nor do we require a Slater point.

The first requirement for global convergence is that the search direction be a descent direction for the merit function. This was settled by Lemma 2.2.5. We need now to quantify the decrease in the merit function. To that end we need a few technical results.

Lemma 3.1.1 *The gradient of the merit function φ is Lipschitz continuous.*

Proof: Since φ is at least twice continuously differentiable we obtain a Lipschitz constant L satisfying

$$\|\nabla\varphi(\nu_1, \mu_1) - \nabla\varphi(\nu_2, \mu_2)\| \leq L\,\|(\nu_1, \mu_1) - (\nu_2, \mu_2)\|$$

by taking the norm of the second derivative in the appropriate interval,

$$L := \sup\Big\{ \left\|\nabla^2\varphi(\xi)\right\| \;\Big|\; \xi \in [(\nu_1, \mu_1), (\nu_2, \mu_2)] \Big\}, \tag{3.4}$$

where the derivatives are defined in Section 2.2.2. □

Using the Lipschitz constant, we express the decrease as a function of the direction.

Lemma 3.1.2 *Assume that the maximum feasible step in the Gauss-Newton direction at iterate* (ν, μ) *is* d*, then*

$$\min_{\alpha \in [0,1]} \varphi(\nu + \alpha d, \mu) \leq \varphi(\nu, \mu) + \delta,$$

where $\delta < 0$ *is given by*

$$\delta = \begin{cases} \frac{1}{2}[\mathfrak{D}\varphi(\nu, \mu)]d, & \textit{if } [\mathfrak{D}\varphi(\nu, \mu)]d + L\|d\|^2 < 0; \\ -\frac{|[\mathfrak{D}\varphi(\nu,\mu)]d|^2}{2L\|d\|^2} & , \textit{otherwise}; \end{cases} \tag{3.5}$$

where L *is defined as in (3.4).*

Proof: Since d is a descent direction by Lemma 2.2.5,

$$\varphi(\nu + \alpha d, \mu) \leq \varphi(\nu, \mu) + \alpha[\mathfrak{D}\varphi(\nu, \mu)]d + \frac{1}{2}\alpha^2 L\|d\|^2. \tag{3.6}$$

Minimize the right hand-side of (3.6) with respect to $\alpha \in [0, 1]$, a quadratic form. There are two possible solutions:

- Case $[\mathfrak{D}\varphi(\nu, \mu)]d + L\|d\|^2 \geq 0$, and the minimum occurs at the boundary, $\alpha^* = 1$.
- Case $[\mathfrak{D}\varphi(\nu, \mu)]d + L\|d\|^2 < 0$, and the minimum occurs at a stationary point, when $[\mathfrak{D}\varphi(\nu, \mu)]d + \alpha L\|d\|^2 = 0$, or
$$\alpha^* = \frac{[\mathfrak{D}\varphi(\nu, \mu)]d}{L\|d\|^2}.$$

Substitute α^* back and minimize the left hand-side of (3.6). $\square$

At this point we have a decrease of the merit function from $\varphi(\nu, \mu)$ to $\varphi(\nu + d_\nu, \mu)$. The next step of Algorithm 3.1.1 reduces the target parameter by some fraction. To maintain part of the decrease we just obtained, we need to bound the reduction in the barrier parameter.

Lemma 3.1.3 *Given any* $\epsilon > 0$ *, if* τ *is chosen according to*

$$\tau \geq \begin{cases} 0, & \textit{if } (\mu \operatorname{trace}(ZX - \mu I) + \frac{1}{2}\mu^2 n) \leq \epsilon; \\ \frac{\operatorname{trace}(ZX-\mu I)}{\mu n} + 1 - \frac{\sqrt{\operatorname{trace}(ZX-\mu I)^2 - 2\epsilon n}}{\mu n}, & \textit{otherwise}; \end{cases}$$

then

$$\varphi(\nu, \tau\mu) \le \varphi(\nu, \mu) + \epsilon.$$

Proof: From the definition of φ,

$$\begin{aligned}
\varphi(\nu, \tau\mu) &= \frac{1}{2}\left\{\|\mathcal{A}(X) - b\|^2 + \|\mathcal{A}^*(y) + Z - C\|^2 + \|ZX - \tau\mu I\|^2\right\} \\
&= \frac{1}{2}\left\{\|f_p(\nu)\|^2 + \|F_d(\nu)\|^2 + \|F_c(\nu) + (1-\tau)\mu I\|^2\right\} \\
&= \frac{1}{2}\left\{\|f_p(\nu)\|^2 + \|F_d(\nu)\|^2 + \|F_c(\nu)\|^2\right\} \\
&\quad + \frac{1}{2}\left\{2\langle F_c(\nu), (1-\tau)\mu I\rangle + (1-\tau)^2\mu^2 n\right\} \\
&= \varphi(\nu, \mu) + \frac{1}{2}\left\{2\langle F_c(\nu), (1-\tau)\mu I\rangle + (1-\tau)^2\mu^2 n\right\}.
\end{aligned}$$

We need $\epsilon \ge f(\tau) := (1-\tau)\text{trace}\,(ZX - \mu I) + \frac{1}{2}(1-\tau)^2\mu^2 n$. Note that f is a strictly convex quadratic function and that $f(1) = 0$. Therefore, by the intermediate value theorem, there are two solutions to $f(\tau) = \epsilon$, one on the interval $\tau < 1$ and the other on the other side. We distinguish two cases since we are interested only in values of $\tau \in [0, 1]$.

- Case $\mu\text{trace}\,(ZX - \mu I) + \frac{1}{2}\mu^2 n \le \epsilon$. Then any value $\tau \ge 0$ will suffice.

- Case $\mu\text{trace}\,(ZX - \mu I) + \frac{1}{2}\mu^2 n > \epsilon$. Then we solve $f(\tau) = \epsilon$,

$$\begin{aligned}
(1-\tau) &= \frac{-\mu\text{trace}\,(ZX - \mu I) \pm \sqrt{(\mu\text{trace}\,(ZX - \mu I)^2 - 2\mu^2 n\epsilon}}{\mu^2 n} \\
&= -\frac{\text{trace}\,(ZX - \mu I)}{\mu n} \pm \frac{\sqrt{(\text{trace}\,(ZX - \mu I))^2 - 2n\epsilon}}{\mu n}.
\end{aligned}$$

Since we are interested only in the larger of the two zeroes, we need

$$1 - \tau \le -\frac{\text{trace}\,(ZX - \mu I)}{\mu n} + \frac{(\mu\text{trace}\,(ZX - \mu I))^2 - 2n\epsilon}{\mu n}.$$

The combination of both cases yields the claimed result. $\square$

From the analysis above we obtain a strict decrease at every step of the Algorithm.

Corollary 3.1.4 *Between step* k *and* $k+1$ *of Algorithm 3.1.1, the decrease of the merit function satisfies*

$$\varphi(\nu^{(k+1)}, \mu^{(k+1)}) \le \varphi(\nu^{(k)}, \mu^{(k)}) + \gamma,$$

for some $\gamma < 0$.

Proof: From Lemma 3.1.2 we obtain

$$\varphi(\nu^{(k+1)}, \mu^{(k)}) \le \varphi(\nu^{(k)}, \mu^{(k)}) + \delta,$$

for some $\delta < 0$. We choose τ to keep a fraction of this decrease, say one half of the decrease, to obtain

$$\varphi(\nu^{(k+1)}, \tau\mu^{(k)}) \le \varphi(\nu^{(k)}\mu^{(k)}) + \gamma,$$

where $\gamma = \frac{1}{2}\delta$ by choosing τ according to Lemma 3.1.3, identifying ϵ with $-\frac{1}{2}\delta$. $\square$

We are now in a position to describe the *TargetSelect* routine of the Algorithm 3.1.1. This is done in Algorithm 3.1.2. Here we compute the real decrease in the merit function instead of the estimated decrease.

Algorithm 3.1.2 Barrier parameter update

function $\mu^{(k+1)} = \text{TargetSelect}\left(\nu^{(k)}, d^{(k)}, \alpha^{(k)}, \mu^{(k)}\right)$;
$\epsilon := \frac{1}{2}\left(\varphi\left(\nu^{(k)} + \alpha^{(k)} d^{(k)}, \mu^{(k)}\right) - \varphi\left(\nu^{(k)}, \mu^{(k)}\right)\right)$; {Keep half of decrease}
Choose τ according to Lemma 3.1.3
$\mu^{(k+1)} := \tau\mu^{(k)}$; {New target}

The next step in obtaining a global convergence result from this line of reasoning would require obtaining a lower bound for δ defined in (3.5) either as a constant term or as a fraction of the current value of the current merit function. This surely involves restricting the iterates to some neighborhood of the central path but our work has yet to produce the required bound. This is why we approach polytime convergence from a different point of view in the next section.

3.2 Polytime Convergence

In contrast to the approach above, the path taken here does not produce a practical algorithm. We include it because the main result highlights a dependence of the convergence rate on the conditioning of the Jacobian and also because of the questions it raises.

This is the first attempt at a proof of polytime convergence of which we are aware for the Gauss-Newton direction. Although an algorithm based on a projected and scaled Gauss-Newton direction was demonstrated in [47]. The approach is not usual. The iterates are not explicitly maintained feasible, nor even positive definite; rather, we maintain the weaker condition that the Jacobian of the optimality conditions is of full rank. Moreover, our measure of distance to the central path combines feasibility and complementarity. The main result appears in Theorem 3.2.13.

A strictly feasible or so-called interior point $v_0 = (X_0, y_0, Z_0)$ is such that

$$X_0 \in \mathbb{S}^n_{++}, \tag{3.7a}$$

$$Z_0 \in \mathbb{S}^n_{++}, \tag{3.7b}$$

$$\mathcal{A}(X_0) = b, \tag{3.7c}$$

$$\mathcal{A}^*(y_0) + Z_0 = C. \tag{3.7d}$$

Assumptions 3.2.1 *Throughout this section, the following conditions are assumed to hold:*

- *There is a point v^0 satisfying condition 3.7.*
- *The operator $\mathcal{A}$ is surjective.*
- *The optimal solution to the primal-dual pair (3.1-3.2) is unique and satisfies strict complementarity (i.e. $Z + X \in \mathbb{S}^n_{++}$).*

Under Assumptions 3.2.1, for every $\mu > 0$, there is a unique solution in $\mathbb{S}^n_{++} \times \mathbb{R}^m \times \mathbb{S}^n_{++}$ to $F_\mu(X, y, Z) = 0$, which we denote by (X_μ, y_μ, Z_μ). This set of solutions is called the *central path*. The limit point of the central path corresponding to $\mu \to 0$ is the solution of the semidefinite pair

(3.1-3.2). At the start of the algorithm, we need a point on the central path or close to it. This point may be obtained via a self-dual embedding of the program. We will not pursue this further. The interested reader may consult [67, 68, 18].

To simplify the statements of the algorithm and of the following results we define

$$F_\mu(X, y, Z) := \begin{bmatrix} \mathcal{A}^*(y) + Z - C \\ \mathcal{A}(X) - b \\ ZX - \mu I \end{bmatrix}, \qquad \text{(central path defining function)} \tag{3.8a}$$

$$F_{\tau\mu}(X, y, Z) := \begin{bmatrix} \mathcal{A}^*(y) + Z - C \\ \mathcal{A}(X) - b \\ ZX - \tau\mu I \end{bmatrix}, 0 < \tau < 1. \qquad \text{(merit vector function)} \tag{3.8b}$$

The algorithm described in this section approximately follows the central path by attempting to solve $F_\mu(X, y, Z) = 0$ for decreasing values of μ. This is common to all path-following algorithms. The novelty of the approach described here is to treat this approximation subproblem as a nonlinear equation and to apply classical tools.

One difference from standard practice resulting from this point of view is the relation between the iterates and the barrier parameter: The scalar μ is not updated using the iterates as is usually the case ($\mu = \tau\frac{\langle Z,X\rangle}{n}$), but rather it is reduced by a factor $\tau < 1$ at every step ($\mu \leftarrow \tau\mu$). In consequence, the initial point (X_0, y_0, Z_0) depends on μ_0, rather than the reverse. Another important difference is that no attempt is made to dampen the step to maintain the iterates within the cone of positive definite matrices. The algorithm maintains only the weaker full rank condition on the Jacobian. The cone constraint is satisfied if we start close to an interior point because every iterate remains within the radius of convergence of the target path point.

To simplify the expressions throughout, we define, for any subscript ξ,

$$v_\xi := (X_\xi, y_\xi, Z_\xi), \qquad d_v := (d_X, d_y, d_Z).$$

We also define canonical central path points ν_μ and $\nu_{\tau\mu}$ such that

$$F_\mu(\nu_\mu) = 0, \qquad F_{\tau\mu}(\nu_{\tau\mu}) = 0.$$

Algorithm 3.2.1 Gauss-Newton infeasible short-step

Given $\mu_0 > 0$;	{Initial barrier parameter}
Given $\epsilon > 0$;	{Merit function tolerance}
Find $X_0; y_0; Z_0$;	{Must satisfy (3.30)}
$X = X_0; y = y_0; Z = Z_0$;	{Initial iterate}
$\mu = \mu_0$;	{Initial barrier parameter}
Choose $0 < \tau < 1$;	{Chosen according to (3.27)}
while $\max\{\tau\mu, \|F_{\tau\mu}(\nu)\|\} > \epsilon$ **do**	
$d_\nu = -[\mathfrak{D}F_{\tau\mu}(\nu)]^\dagger F_{\tau\mu}(\nu;)$	{Gauss-Newton direction}
$X = X + d_X; y = y + d_y; Z = Z + d_Z$;	{Update iterate}
$\mu := \tau\mu$;	{Update target}
end while	

3.2.1 Merit Function and Central Path

This section describes some relations between the value of our chosen merit function $\|F_{\tau\mu}\|$ and the distance of the iterate to the central path. Note that we do not assume that the iterates are primal or dual feasible. Our measure of distance to the central path combines estimates of both infeasibility and complementarity. The section also describes the progress of the Gauss-Newton direction in minimizing $\|F_{\tau\mu}\|$. The results are of a technical nature and used as building blocks of the next section.

We begin this section with a well known result about approximations of inverses, often referred to as the Banach Lemma. For a proof see [45].

Lemma 3.2.1 *Suppose* $M \in \mathbb{M}^n$ *and* $\|M\| < 1$. *Then* $I - M$ *is non-singular and*

$$\|(I - M)^{-1}\| \leq \frac{1}{1 - \|M\|}.$$

$\square$

Since the Gauss-Newton direction is obtained from an over-determined system of equations, pseudo-inverses allow succinct expressions of the solution. Namely, the least squares solution to $[\mathfrak{D}F_{\tau\mu}(\nu)]d_\nu = -F_{\tau\mu}(\nu)$ is $d_\nu = -[\mathfrak{D}F_{\tau\mu}(\nu)]^\dagger F_{\tau\mu}(\nu)$, where $(\cdot)^\dagger$ indicates the Moore-Penrose inverse.

To generalize to Gauss-Newton some results well-known about Newton's method [46] we require a bound on the norm of the pseudo-inverse.

Lemma 3.2.2 *Suppose that* $A \in \mathbb{R}^{m\times n}$ *and* $B \in \mathbb{R}^{n\times m}$*, where* $m \geq n$*; and assume that* BA *is non-singular. Then*

$$\|A^\dagger\| \leq \|(BA)^{-1}B\|.$$

Proof: Define the singular value decompositions $A = U_A\Sigma_A V_A^t$ and $B = U_B\Sigma_B V_B^t$. Then

$$\begin{aligned}
\|(BA)^{-1}B\| &= \left\| \left[U_B\Sigma_B V_B^t U_A\Sigma_A V_A^t\right]^{-1} U_B\Sigma_B V_B^t \right\| \\
&= \left\| \left[U_B \left[\overline{\Sigma}_B 0\right] V_B^t U_A \begin{bmatrix} \overline{\Sigma}_A \\ 0 \end{bmatrix} V_A^t \right]^{-1} U_B \left[\overline{\Sigma}_B 0\right] V_B^t \right\| \\
&= \left\| \left[U_B \left[\overline{\Sigma}_B 0\right] \begin{bmatrix} Q_1 & Q_2 \\ Q_3 & Q_4 \end{bmatrix} \begin{bmatrix} \overline{\Sigma}_A \\ 0 \end{bmatrix} V_A^t \right]^{-1} U_B \left[\overline{\Sigma}_B 0\right] V_B^t \right\| \\
&= \left\| V_A \overline{\Sigma}_A^{-1} Q_1^{-1} \overline{\Sigma}_B^{-1} U_B^t U_B \overline{\Sigma}_B \right\| \\
&= \left\| \overline{\Sigma}_A^{-1} Q_1^{-1} \overline{\Sigma}_B^{-1} \overline{\Sigma}_B \right\| \\
&= \left\| \overline{\Sigma}_A^{-1} Q_1^{-1} \right\|.
\end{aligned}$$

Since $Q^tQ = I$, we have $Q_1^tQ_1 + Q_3^tQ_3 = I$ and therefore $I \succeq Q_1^tQ_1$. This implies that all the singular values of Q_1 are at most 1; and all the singular values of Q_1^{-1} are at least 1. Therefore

$$\|\overline{\Sigma}_A^{-1} Q_1^{-1}\| \geq \|\overline{\Sigma}_A^{-1}\| = \|A^\dagger\|,$$

which is the required bound on the norm of the Moore-Penrose inverse. □

From Lemma 3.2.1 and Lemma 3.2.2, we obtain the following result about approximation of pseudo-inverses.

Lemma 3.2.3 *Suppose that* $\overline{A}$ *is an approximation to the pseudo-inverse of* A *in the sense that* $\|I - \overline{A}A\| < 1$. *Then*

$$\|A^\dagger\| \le \frac{\|\overline{A}\|}{1 - \|I - \overline{A}A\|}.$$

Proof: Consider that $\|I - \overline{A}A\| < 1$ is the required condition of Lemma 3.2.1. Therefore we write

$$\|A^\dagger\| \le \|(\overline{A}A)^{-1}\overline{A}\| \le \|(\overline{A}A)^{-1}\|\|\overline{A}\| \le \frac{\|\overline{A}\|}{1 - \|I - \overline{A}A\|},$$

where the first inequality is obtained from Lemma 3.2.2. □

Essentially from this bound on the norm of approximate pseudo-inverses we establish a relation between the distance to the central path of an iterate (X, y, Z) and the current value of our merit function $\|F_{\tau\mu}(X, y, Z)\|$. To simplify the result we first establish Lipschitz continuity of the first derivative.

Lemma 3.2.4 *The operator* $[\mathfrak{D}F_{\tau\mu}(\nu)]$ *is Lipschitz continuous with constant 1 with respect to* ν..

Proof: From the definition of $[\mathfrak{D}F_{\tau\mu}(\nu)]$, we obtain,

$$\|[\mathfrak{D}F_{\tau\mu}(\nu + d_\nu)] - [\mathfrak{D}F_{\tau\mu}(\nu)]\|^2 = \left\| \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ d\mathcal{Z} & 0 & d\mathcal{X} \end{bmatrix} \right\|^2.$$

Direct calculations, with $d_\nu = (d_X, d_y, d_Z)$, yield

$$\begin{aligned} \|[\mathfrak{D}F_{\tau\mu}(\nu + d_\nu)] - [\mathfrak{D}F_{\tau\mu}(\nu)]\|^2 &= \max\left\{\|d_Z\nu_x + \nu_z d_X\|^2 \mid \|(\nu_x, \nu_z)\|^2 = 1\right\} \\ &\le \max\left\{\|(d_Z, d_X)\|^2 \|(\nu_x, \nu_z)\|^2 \mid \|(\nu_x, \nu_z)\|^2 = 1\right\} \\ &\le \|(d_Z, d_X)\|^2 \\ &\le \|d_\nu\|^2. \end{aligned}$$

Hence a constant of 1 will suffice. □

Lemma 3.2.5 *Under Assumptions 3.2.1, there is a* $\delta > 0$ *so that for all* ν *such that* $\|\nu - \nu_{\tau\mu}\| < \delta$,

$$\|[\mathfrak{D}F_{\tau\mu}(\nu)]\| \leq 2\,\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\|\,, \tag{3.9a}$$

$$\left\|[\mathfrak{D}F_{\tau\mu}(\nu)]^{\dagger}\right\| \leq 2\left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\right\|, \tag{3.9b}$$

$$\frac{\|\nu - \nu_{\tau\mu}\|}{2\,\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\|} \leq \|F_{\tau\mu}(\nu)\|\,, \tag{3.9c}$$

$$\|F_{\tau\mu}(\nu)\| \leq 2\,\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\|\,\|\nu - \nu_{\tau\mu}\|\,. \tag{3.9d}$$

Moreover, we can choose any δ *satisfying*

$$\delta < \frac{\sigma_{\min}}{2}, \tag{3.10}$$

where $\sigma_{\min}$ *denotes the smallest singular value of* $[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]$.

Proof: Since $[\mathfrak{D}F_{\tau\mu}(\nu)]$ is Lipschitz continuous with constant 1, using the reverse triangle inequality we get

$$\|[\mathfrak{D}F_{\tau\mu}(\nu)]\| - \|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\| \leq \|[\mathfrak{D}F_{\tau\mu}(\nu)] - [\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\| \leq \|\nu - \nu_{\tau\mu}\|$$

Therefore

$$\|[\mathfrak{D}F_{\tau\mu}(\nu)]\| \leq \|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\| + \|\nu - \nu_{\tau\mu}\|\,.$$

Take δ small enough so that

$$\delta < \|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\| \tag{3.11}$$

to obtain (3.9a). For the second result (3.9b), take δ small enough so that

$$\delta < \frac{1}{2\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\|}, \tag{3.12}$$

which implies $\|\nu - \nu_{\tau\mu}\| \leq \frac{1}{2\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\|}$. Now we write

$$\begin{aligned}
\left\|I - [\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}[\mathfrak{D}F_{\tau\mu}(\nu)]\right\| &= \left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\left([\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})] - [\mathfrak{D}F_{\tau\mu}(\nu)]\right)\right\| \\
&\leq \left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\right\| \|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})] - [\mathfrak{D}F_{\tau\mu}(\nu)]\| \\
&\leq \left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\right\| \|\nu_{\tau\mu} - \nu\| \\
&\leq \frac{\left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\right\|}{2\left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\right\|}.
\end{aligned}$$

From the last inequality we get

$$\left\|I - [\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}[\mathfrak{D}F_{\tau\mu}(\nu)]\right\| \leq \frac{1}{2}. \tag{3.13}$$

Then, from Lemma 3.2.3 with the identification $A = [\mathfrak{D}F_{\tau\mu}(\nu)]$ and $\overline{A} = [\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}$, and from (3.13) we obtain

$$\left\|[\mathfrak{D}F_{\tau\mu}(\nu)]^{\dagger}\right\| \leq \frac{[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}}{1 - \|I - [\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}F_{\tau\mu}(\nu)\|} \leq 2\left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\right\|,$$

our second required inequality. For the third inequality (3.9c), we use the Fundamental theorem of calculus to express

$$[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}F_{\tau\mu}(\nu) = [\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\int_0^1 [\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu} + t(\nu - \nu_{\tau\mu}))](\nu - \nu_{\tau\mu})\,dt.$$

Take norms on both sides to get

$$\begin{aligned}
&\left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}F_{\tau\mu}(\nu)\right\| \\
&= \left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\int_0^1[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu}+t(\nu-\nu_{\tau\mu}))](\nu-\nu_{\tau\mu})\,dt\right\| \\
&= \left\|(\nu-\nu_{\tau\mu})-\int_0^1[I-[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu}+t(\nu-\nu_{\tau\mu}))](\nu-\nu_{\tau\mu})\,dt\right\| \\
&\geq \|\nu-\nu_{\tau\mu}\| - \int_0^1\left\|I-[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu}+t(\nu-\nu_{\tau\mu}))]\right\|\|\nu-\nu_{\tau\mu}\|\,dt \\
&\geq \|\nu-\nu_{\tau\mu}\| - \frac{1}{2}\|\nu-\nu_{\tau\mu}\| \\
&= \frac{1}{2}\|\nu-\nu_{\tau\mu}\|,
\end{aligned}$$

Where the last inequality follows from (3.13). Therefore

$$\frac{\|\nu-\nu_{\tau\mu}\|}{2} \leq \left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}F_{\tau\mu}(\nu)\right\| \leq \left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\right\|\|F_{\tau\mu}(\nu)\|.$$

The fourth inequality (3.9d) is obtained similarly. We use the assumption $F_{\tau\mu}(\nu_{\tau\mu}) = 0$ and the bound (3.9a) to get

$$\begin{aligned}
\|F_{\tau\mu}(\nu)\| &= \left\|F_{\tau\mu}(\nu_{\tau\mu})+\int_0^1[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu}+t(\nu-\nu_{\tau\mu}))](\nu-\nu_{\tau\mu})\,dt\right\| \\
&\leq \int_0^1\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu}+t(\nu-\nu_{\tau\mu}))]\|\,\|\nu-\nu_{\tau\mu}\|\,dt \\
&\leq \int_0^1 2\,\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\|\,\|\nu-\nu_{\tau\mu}\|\,dt \\
&= 2\,\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\|\,\|\nu-\nu_{\tau\mu}\|.
\end{aligned}$$

Now we need to restrict δ using (3.11) and (3.12). Take

$$\delta = \min\left\{\frac{1}{2\,\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^{\dagger}\|}, \|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\|\right\} = \frac{\sigma_{\min}}{2}$$

to complete the result. □

Corollary 3.2.6 *Under Assumptions 3.2.1, let $\delta > 0$ be small enough to satisfy the conclusions of Lemma 3.2.5. Then for all ν, ν_c such that $\|\nu - \nu_{\tau\mu}\| < \delta$, $\|\nu_c - \nu_{\tau\mu}\| < \delta$,*

$$\frac{\|(\nu - \nu_{\tau\mu})\|}{4\kappa\,\|(\nu_c - \nu_{\tau\mu})\|} \leq \frac{\|F_{\tau\mu}(\nu)\|}{\|F_{\tau\mu}(\nu_c)\|} \leq \frac{4\kappa\,\|(\nu - \nu_{\tau\mu})\|}{\|(\nu_c - \nu_{\tau\mu})\|},$$

where $\kappa = \|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\|\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^\dagger\|$ is the condition number of the Jacobian matrix at the central path point.

Proof: By Lemma 3.2.5, inequalities (3.9d) and (3.9c),

$$\|F_{\tau\mu}(\nu)\| \leq 2\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\|\|\nu - \nu_{\tau\mu}\|, \qquad \text{and} \qquad \|F_{\tau\mu}(\nu_c)\| \geq 2\frac{\|\nu_c - \nu_{\tau\mu}\|}{\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^\dagger\|}.$$

Therefore

$$\frac{\|F_{\tau\mu}(\nu)\|}{\|F_{\tau\mu}(\nu_c)\|} \leq \frac{2\,\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]\|\,\|(\nu - \nu_{\tau\mu})\|}{\frac{\|(\nu_c - \nu_{\tau\mu})\|}{2\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^\dagger\|}} = 4\kappa\frac{\|(\nu - \nu_{\tau\mu})\|}{\|(\nu_c - \nu_{\tau\mu})\|}.$$

The other inequality is similar. □

Corollary 3.2.7 *Under Assumptions 3.2.1, let $\delta > 0$ be small enough to satisfy the conclusions of Lemma 3.2.5. Then for all ν such that $\|\nu - \nu_{\tau\mu}\| < \delta$, the Jacobian $[\mathfrak{D}F_{\tau\mu}(\nu)]$ is of full column rank.*

Proof: From (3.9b), we see that the smallest nonzero singular value of $[\mathfrak{D}F_{\tau\mu}(\nu)]$ is bounded below on the entire neighborhood about $\nu_{\tau\mu}$. Therefore, no nonzero singular value approaches 0. □

From these relations between the central path and our merit function, we obtain a radius of quadratic convergence to a point on the central path as well as a decrease of the merit function.

Theorem 3.2.8 *Let $\sigma_{\min}$ be the smallest singular value of $[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]$. Under Assumptions 3.2.1 there is a $\delta > 0$ such that for all ν_c such that $\|\nu_c - \nu_{\tau\mu}\| < \delta$, the Gauss-Newton step*

$$\nu_+ = \nu_c - [\mathfrak{D}F_{\tau\mu}(\nu_c)]^\dagger F_{\tau\mu}(\nu_c)$$

is well-defined and

$$\|\nu_+ - \nu_{\tau\mu}\| \le \frac{1}{\sigma_{\min}} \|\nu_c - \nu_{\tau\mu}\|^2 .$$

Moreover, we can choose any $\delta < \frac{\sigma_{\min}}{2}$. *Hence the Gauss-Newton iteration converges quadratically to the central path.*

Proof: Let δ be small enough so that the hypothesis of Lemma 3.2.5 holds, i.e. $\delta < \frac{\sigma_{\min}}{2}$. First we express the error on the iterate both before and after the step, then by the fundamental Theorem of calculus and the fact that $[\mathfrak{D}F_{\tau\mu}(\nu_c)]$ is of full column rank (and hence that $[\mathfrak{D}F_{\tau\mu}(\nu_c)]^\dagger[\mathfrak{D}F_{\tau\mu}(\nu_c)] = I$),

$$\begin{aligned}
(\nu_+ - \nu_{\tau\mu}) &= (\nu_c - \nu_{\tau\mu}) - [\mathfrak{D}F_{\tau\mu}(\nu_c)]^\dagger F_{\tau\mu}(\nu_c) \\
&= [\mathfrak{D}F_{\tau\mu}(\nu_c)]^\dagger \int_0^1 ([\mathfrak{D}F_{\tau\mu}(\nu_c)] - [\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu} + t(\nu_c - \nu_{\tau\mu}))])(\nu_c - \nu_{\tau\mu})dt.
\end{aligned}$$

Take norms on both sides and use the Lipschitz continuity of $[\mathfrak{D}F_{\tau\mu}(\nu)]$ to get

$$\|\nu_+ - \nu_{\tau\mu}\| \le \frac{1}{2} \left\|[\mathfrak{D}F_{\tau\mu}(\nu_c)]^\dagger\right\| \|\nu_c - \nu_{\tau\mu}\|^2 .$$

Now use Lemma 3.2.5, inequality (3.9b) to get

$$\|\nu_+ - \nu_{\tau\mu}\| \le \left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^\dagger\right\| \|\nu_c - \nu_{\tau\mu}\|^2 ,$$

the required reduction of the error. □

The next result relates the reduction in the error to the reduction in the merit function.

Corollary 3.2.9 *Let* $\sigma_{\min}$ *and* $\sigma_{\max}$ *be, respectively, the smallest and largest singular value of* $[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]$. *Under Assumptions 3.2.1 there is a* $\delta > 0$ *where for all* ν_c *such that* $\|\nu_c - \nu_{\tau\mu}\| < \delta$, *the next iterate,* $\nu_+ = \nu_c - [\mathfrak{D}F_{\tau\mu}(\nu_c)]^\dagger F_{\tau\mu}(\nu_c)$, *satisfies*

$$\|F_{\tau\mu}(\nu_+)\| \le \frac{1}{2} \|F_{\tau\mu}(\nu_c)\| .$$

Moreover, we can choose any δ such that

$$\delta < \frac{\sigma_{\min}^2}{8\sigma_{\max}}. \tag{3.14}$$

Proof: Consider the inequality (3.9d) at the point v_+ to obtain

$$\|F_{\tau\mu}(v_+)\| \leq 2\sigma_{\max} \|v_+ - v_{\tau\mu}\|.$$

Now assume that δ satisfies the condition of Theorem 3.2.8 and apply the result as well as inequality (3.9c) at the point v_c to get

$$\begin{aligned} \|F_{\tau\mu}(v_+)\| &\leq 2\frac{\sigma_{\max}}{\sigma_{\min}} \|v_c - v_{\tau\mu}\|^2, \\ &\leq 2\frac{\sigma_{\max}}{\sigma_{\min}} \|v_c - v_{\tau\mu}\| \frac{2}{\sigma_{\min}} \|F_{\tau\mu}(v_c)\| \\ &= 4\frac{\sigma_{\max}}{\sigma_{\min}} \|F_{\tau\mu}(v_c)\| \|v_c - v_{\tau\mu}\|. \end{aligned}$$

Therefore we need $\|(v_c - v_{\tau\mu})\| < \delta$, with δ as defined in (3.14), to obtain the required decrease. $\square$

3.2.2 Smallest Singular Value

The behavior of the smallest singular value, because of its appearance in every bound, is of concern to us. We depart from the main goal of the section to explore this behavior. On the central path $v_\mu = (X_\mu, y_\mu, Z_\mu)$, the singular value of interest is defined by

$$\sigma_{\min}^2 := \min\left\{ \|[\mathfrak{D}F(v_\mu)]s\|^2 \mid \|s\|^2 = 1 \right\}, \tag{3.15}$$

where $s = (S_x, s_y, S_z) \in \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$. We are interested in the rate of change of this singular value as μ changes. We therefore define the perturbed problem

$$\sigma_*^2(\epsilon) := \min\left\{ f(s, \epsilon) \mid \|s\|^2 = 1 \right\}, \tag{3.16}$$

where

$$f(s,\epsilon) := \|\mathcal{A}^*(s_y) + S_z\|^2 + \|\mathcal{A}(S_x)\|^2 + \|Z_\mu S_x + S_z X_\mu\|^2 . \tag{3.17}$$

The perturbation is implicit: X_μ, y_μ, Z_μ change if $\mu \to \mu + \epsilon$. By a result of Fiacco [23] (Corollary 3.4.2), the change in the optimal value of (3.16) is given by

$$[\mathfrak{D}_\epsilon \sigma_*^2(\epsilon)] = [\mathfrak{D}_\epsilon f(s,\epsilon)]. \tag{3.18}$$

This implies, in our case,

$$[\mathfrak{D}_\epsilon \sigma_*^2(\epsilon)] = 2\langle [\mathfrak{D}_\epsilon Z(0)] S_x + S_z [\mathfrak{D}_\epsilon X(0)], Z_\mu S_x + S_z X_\mu \rangle. \tag{3.19}$$

We need an expression for the derivatives $[\mathfrak{D}_\epsilon Z(0)], [\mathfrak{D}_\epsilon X(0)]$ which we obtain from the Implicit Function Theorem cited here with the required generality from [86] (Theorem 12.4.1 and following Corollary 1)

Theorem 3.2.10 *Consider a vector function* $F : X \times Y \to Z$ *defined on a ball* $\Omega_r := \left\{ x \mid \|x - x^0\| \le r, \|y - y^0\| \le r \right\}$ *and satisfying*

- $F(x^0, y^0) = 0$*;*
- $[\mathfrak{D}_y F(x,y)]$ *exists on* Ω_r *and is continuous in both* x *and* y*;*
- $F(x,y)$ *is continuous on* Ω_r*;*
- $[\mathfrak{D}_y F(x^0, y^0)]^{-1}$ *exists in* $Z \to Y$*;*
- $[\mathfrak{D}_x F(x,y)]$ *exists on* Ω_r *and is continuous at* (x^0, y^0).

Then there exists positive numbers r_0, r_1 *and a continuous map* $G : X \to Y$ *on* $\|x - x^0\| \le r_0 \le r$ *satisfying*

- $\|G(x) - y^0\| \le r_1 \le r$*;*
- $G(x^0) = y^0$*;*

- $F(x, G(x)) = 0$;*and*
- $[\mathfrak{D}G(x^0)] = -[\mathfrak{D}_y F(x^0, y^0)]^{-1}[\mathfrak{D}_x F(x^0, y^0)]$.

We will identify x with ϵ and y with $v = (X, y, Z)$ in the function

$$F(X, y, Z, \epsilon) := \begin{bmatrix} \mathcal{A}^*(y) + Z - C \\ \mathcal{A}(X) - b \\ ZX - (\mu + \epsilon)I \end{bmatrix}$$

to obtain that there is a $G : \mathbb{R} \to \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n$ with

- $G(0) = (X_\mu, y_\mu, Z_\mu)$;
- $F(G(\epsilon), \epsilon) = 0$;
- $[\mathfrak{D}G(0)] = -[\mathfrak{D}_v F(x_\mu, y_\mu, Z_\mu, 0)]^{-1}[\mathfrak{D}_\epsilon F(x_\mu, y_\mu, Z_\mu, 0)]$.

We have the derivatives

$$[\mathfrak{D}_v F(X_\mu, y_\mu, Z_\mu, 0)] = \begin{bmatrix} 0 & \mathcal{A}^* & \mathcal{I} \\ \mathcal{A} & 0 & 0 \\ \mathcal{Z}_\mu & 0 & \mathcal{X}_\mu \end{bmatrix}, \quad \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n \to \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n,$$

and

$$[\mathfrak{D}_\epsilon F(X_\mu, y_\mu, Z_\mu, 0)] = \begin{bmatrix} 0 \\ 0 \\ -\mathcal{I} \end{bmatrix}, \quad \mathbb{R} \to \mathbb{S}^n \times \mathbb{R}^m \times \mathbb{S}^n.$$

We therefore obtain the required implicitly defined derivative as

$$[\mathfrak{D}G(0)] := \begin{bmatrix} [\mathfrak{D}_\epsilon X(0)] \\ [\mathfrak{D}_\epsilon y(0)] \\ [\mathfrak{D}_\epsilon Z(0)] \end{bmatrix} = \begin{bmatrix} 0 & \mathcal{A}^* & \mathcal{I} \\ \mathcal{A} & 0 & 0 \\ \mathcal{Z}_\mu & 0 & \mathcal{X}_\mu \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \mathcal{I} \end{bmatrix}.$$

Note that because of the structure of the last operator $[\mathfrak{D}_\epsilon F(X_\mu, y_\mu, Z_\mu, 0)]$, we only need the top and bottom right blocks of the inverse. Alternatively we only need to solve

$$\begin{aligned} \mathcal{A}^*([\mathfrak{D}_\epsilon y(0)]) + [\mathfrak{D}_\epsilon Z(0)] &= 0 \\ \mathcal{A}([\mathfrak{D}_\epsilon X(0)]) &= 0 \\ \mathcal{Z}_\mu([\mathfrak{D}_\epsilon X(0)]) + \mathcal{X}_\mu([\mathfrak{D}_\epsilon Z(0)]) &= \mathcal{I}. \end{aligned}$$

The solution is

$$[\mathfrak{D}_\epsilon X(0)] = Z_\mu^{-1}(I - \mathcal{A}^*[\mathcal{A}Z_\mu^{-1}\mathcal{A}^*]^{-1}\mathcal{A}Z_\mu^{-1}), \tag{3.20a}$$

$$[\mathfrak{D}_\epsilon y(0)] = -[\mathcal{A}Z_\mu^{-1}\mathcal{A}^*]^{-1}\mathcal{A}Z_\mu^{-1}X^{-1}, \tag{3.20b}$$

$$[\mathfrak{D}_\epsilon Z(0)] = -\mathcal{A}^*[\mathcal{A}Z_\mu^{-1}\mathcal{A}^*]^{-1}\mathcal{A}\mu^{-1}. \tag{3.20c}$$

From (3.19) and the derivatives (3.20) above, we obtain

$$\begin{aligned} &[\mathfrak{D}_\epsilon \sigma^*(0)] = \\ &\quad 2\big\langle -\mathcal{A}^*[\mathcal{A}Z_\mu^{-1}\mathcal{A}^*]^{-1}\mathcal{A}\mu^{-1}S_x + S_z Z_\mu^{-1}(I - \mathcal{A}^*[\mathcal{A}Z_\mu^{-1}\mathcal{A}^*]^{-1}\mathcal{A}Z_\mu^{-1}), Z_\mu S_x + S_z X_\mu\big\rangle. \end{aligned} \tag{3.21}$$

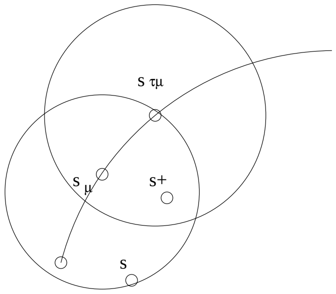
It would be interesting to investigate further this derivative, to find out where it is positive, negative and zero. We only have tentative numerical results that seem to indicate that the smallest singular value is a pseudo-convex function but a complete theoretical description is yet to come.

3.2.3 Convergence of the Algorithm

At this point we have established all the necessary relations between our merit function and the distance between an iterate and the central path. We now describe the convergence of Algorithm 3.2.1. For reference, we repeat the definitions of the two canonical points ν_μ and $\nu_{\tau\mu}$ on the central path. These points satisfy

$$F_\mu(\nu_\mu) = 0, \qquad F_{\tau\mu}(\nu_{\tau\mu}) = 0. \tag{3.22}$$

The general idea of the algorithm is that, from a iterate ν_k , "close enough" to ν_μ, within the radius of quadratic convergence, we choose a target on the central path $\nu_{\tau\mu}$ in such a way that the next iterate ν_{k+1}, obtained from the Gauss-Newton direction, is now "close enough" to $\nu_{\tau\mu}$ for the process to be repeated. The key point is not that the convergence is quadratic, since we never let the process run to convergence, but rather that the iterates remain close to the central path and that we can estimate the distance from an iterate to its target.



The proof is in three parts. First we estimate the distance between two points on the central paths in terms of the required radius of convergence.

Lemma 3.2.11 *Let* $\sigma_{\min}$ *and* $\sigma_{\max}$ *be, respectively, the smallest and largest singular values of* $[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]$.

1. *If we choose* $0 < \tau < 1$ *such that*

$$1 - \tau \le \frac{\sigma_{\min}^2}{8\sqrt{n\mu}}, \tag{3.23}$$

then

$$\|\nu_\mu - \nu_{\tau\mu}\| \le \frac{1}{2}\left(\frac{\sigma_{\min}}{2}\right), \tag{3.24}$$

which implies ν_μ *is within half of the radius of quadratic convergence of* $\nu_{\tau\mu}$.

2. If we choose $0 < \tau < 1$ *such that*

$$1 - \tau \leq \frac{\sigma_{\min}^3}{32\sqrt{n}\mu\sigma_{\max}}, \tag{3.25}$$

then

$$\|\nu_\mu - \nu_{\tau\mu}\| \leq \frac{1}{2}\left(\frac{\sigma_{\min}^2}{8\sigma_{\max}}\right). \tag{3.26}$$

In this case ν_μ *is within half of the radius of guaranteed constant decrease of the merit function in (3.14) in Corollary 3.2.9.*

Proof: First note that a straightforward calculation based on the definition of ν_μ (3.22) yields

$$\|F_{\tau\mu}(\nu_\mu)\| = \sqrt{n}(1-\tau)\mu.$$

By Lemma 3.2.5, inequality (3.9d)

$$\begin{aligned} \|\nu_\mu - \nu_{\tau\mu}\| &\leq 2\left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^\dagger\right\| \|F_{\tau\mu}(\nu_\mu)\| \\ &= 2\left\|[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]^\dagger\right\| (1-\tau)\sqrt{n}\mu. \end{aligned}$$

Let τ satisfy (3.25) to get

$$\|\nu_\mu - \nu_{\tau\mu}\| \leq \frac{\sigma_{\min}}{4},$$

which, by Theorem 3.2.8, yields one half of the quadratic radius of convergence. The proof of part 2 of the lemma is similar. □

We now estimate the distance to the new target after a Gauss-Newton step.

Lemma 3.2.12 *Let* $\sigma_{\min}$ *and* $\sigma_{\max}$ *be, respectively, the smallest and largest singular values of*

$[\mathfrak{D}F_{\tau\mu}(\nu_{\tau\mu})]$. *Suppose that the point* ν_c *is well-centered in the sense that*

$$\|\nu_\mu - \nu_c\| \quad \leq \quad \min\left\{\frac{\sigma_{\min}}{4}, \frac{\sigma_{\min}}{16\sigma_{\max}}\right\},$$

and we choose τ *to satisfy*

$$0 < \tau < 1, \quad 1 - \tau \leq \min\left\{\frac{\sigma_{\min}^2}{8\sqrt{n}\mu}, \frac{\sigma_{\min}^3}{32\sqrt{n}\mu\sigma_{\max}}\right\}, \tag{3.27}$$

as in Lemma 3.2.11. Then, after one Gauss-Newton step, the new point ν_+ *will be within half the radius of convergence of* $\nu_{\tau\mu}$, *i.e.*

$$\|\nu_{\tau\mu} - \nu_+\| \quad \leq \quad \frac{\sigma_{\min}}{4}. \tag{3.28}$$

Moreover, the merit function is reduced

$$\|F_{\tau\mu}(\nu_+)\| \leq \frac{1}{2}\|F_{\tau\mu}(\nu_c)\|. \tag{3.29}$$

Proof:

By hypothesis and by Lemma 3.2.11,

$$\|\nu_c - \nu_\mu\| \leq \frac{\sigma_{\min}}{4}, \quad \|\nu_\mu - \nu_{\tau\mu}\| \leq \frac{\sigma_{\min}}{4}.$$

Therefore

$$\begin{aligned} \|\nu_c - \nu_{\tau\mu}\| \quad &\leq \quad \|\nu_c - \nu_\mu + \nu_\mu - \nu_{\tau\mu}\| \\ &\leq \quad \|\nu_c - \nu_\mu\| + \|\nu_\mu - \nu_{\tau\mu}\| \\ &\leq \quad \frac{\sigma_{\min}}{2}, \end{aligned}$$

which is within the radius of quadratic convergence of $\nu_{\tau\mu}$. After one Gauss-Newton step, by

Theorem 3.2.8, we get

$$\begin{aligned}
\|\nu_+ - \nu_{\tau\mu}\| &\leq \frac{1}{\sigma_{\min}} \|\nu_c - \nu_{\tau\mu}\|^2 \\
&\leq \frac{1}{\sigma_{\min}} (\frac{\sigma_{\min}}{2})^2 \\
&= \frac{\sigma_{\min}}{4}.
\end{aligned}$$

Therefore the new point is within half the radius of convergence of $\nu_{\tau\mu}$ and the procedure is repeated.

The constant reduction of the merit function follows from Corollary 3.2.9. □

We now present the main result of this chapter, the polynomial convergence proof (dependent on the smallest singular value) of Algorithm 3.2.1. The dependence on the smallest singular value is both interesting since we should expect convergence to depend on such a parameter, yet it is also somewhat unsatisfying since we cannot estimate this value while the algorithm is executing. We could have formulated the proof in terms of the smallest singular value of the current central path target but here, the dependence is on the smallest singular values over all central path points.

Theorem 3.2.13 *Suppose that we are given an initial barrier parameter estimate* $\mu_0 > 0$*, positive tolerance* $\epsilon > 0$ *and* $Z_0, X_0 \in \mathbb{S}^n_{++}$ *such that* $\nu_0 = (X_0, y_0, Z_0)$ *is a well-centered starting point:* ν_0 *is within half the quadratic convergence radius of* ν_{μ_0} *in Theorem 3.2.8*

$$\|\nu_{\mu_0} - \nu_0\| \leq \frac{1}{2}\left(\frac{\sigma_{\min}}{2}\right), \tag{3.30}$$

and ν_0 *is within half the radius for guaranteed constant decrease of the merit function given in Corollary 3.2.9*

$$\|\nu_{\mu_0} - \nu_0\| \leq \frac{1}{2}\left(\frac{\sigma^2_{\min}}{8\sigma_{\max}}\right),$$

where $0 < \sigma_{\min}$ *(respectively* $\sigma_{\max}$*) is smaller than the smallest (respectively larger than the largest) singular value of* $F'_{\omega\mu_0}(\nu_{\omega\mu_0})$*, for all* $\frac{\epsilon}{\mu_0} < \omega < 1$*. We also choose* $\tau > \frac{1}{2}$ *and satisfying (3.27) (for* $\mu = \epsilon$*) in Lemma 3.2.12. Then the Algorithm 3.2.1 converges to* $\bar{\nu}$*, which is* ϵ*-optimal in the* *following sense*

$$\tau^k \mu_0 \le \epsilon, \quad \left\| F_{\tau^k \mu_0}(\bar{\nu}) \right\| \le \epsilon, \quad \left\| \bar{\nu} - \nu_{\tau^k \mu_0} \right\| \le 2\sigma_{\min}\epsilon,$$

in

$$\mathcal{O}\left(\max\left\{ \frac{\log \frac{\epsilon}{\mu_0}}{\log \tau}, \log\left(\frac{\|F_{\tau\mu}(\nu_0)\| + (\frac{1-\tau}{2\tau-1})\mu_0\sqrt{n}}{\epsilon} \right) \right\} \right) \tag{3.31}$$

iterations.

Proof: By Lemma 3.2.5,

$$\|\nu_k - \nu_{\tau\mu_0}\| \le 2 \left\| [\mathfrak{D} F_{\tau\mu_0}(\nu_{\tau\mu_0})]^\dagger \right\| \|F_{\tau\mu_0}(\nu_k)\| .$$

which results in the desired bound on $\left\| \nu_k - \nu_{\tau^k \mu_0} \right\|$, if $\left\| F_{\tau^k \mu_0}(\nu_k) \right\| \le \epsilon$. From the constant decrease guarantee we get (we add and subtract the multiple of the identity in the third term in the norm)

$$\begin{aligned}
\left\| F_{\tau^k \mu_0}(\nu_k) \right\| &\le \frac{1}{2} \left\| F_{\tau^k \mu_0}(\nu_{k-1}) \right\| \\
&\le \frac{1}{2} \left\| F_{\tau^{k-1} \mu_0}(\nu_{k-1}) \right\| + \frac{1}{2}\tau^{k-1}(1-\tau)\mu_0\sqrt{n} \\
&\le \frac{1}{2^2} \left\| F_{\tau^{k-2} \mu_0}(\nu_{k-2}) \right\| + \frac{\mu_0\sqrt{n}}{2^2}\left\{ \tau^{k-2}(1-\tau) + 2\tau^{k-1}(1-\tau) \right\} \\
&\le \frac{1}{2^k} \|F_{\mu_0}(\nu_0)\| + (1-\tau)\mu_0\sqrt{n}\left(\frac{1}{2^k} + \frac{\tau}{2^{k-1}} + \frac{\tau^2}{2^{k-2}} + \ldots + \frac{\tau^{k-1}}{2} \right)
\end{aligned}$$

We can assume $\tau > \frac{1}{2}$ since that represents the worst case behavior. Then

$$\begin{aligned}
\left\|F_{\tau^k\mu_0}(\nu_k)\right\| &\leq \frac{1}{2^k}\|F_{\mu_0}(\nu_0)\| \\
&\quad +(1-\tau)\mu_0\sqrt{n}\tau^k\left(\frac{\tau^{-k}}{2^k}+\frac{\tau^{1-k}}{2^{k-1}}+\frac{\tau^{2-k}}{2^{k-2}}+\ldots+\frac{\tau^{-1}}{2}\right) \\
&= \frac{1}{2^k}\|F_{\tau\mu}(\nu_0)\| \\
&\quad +(1-\tau)\mu_0\sqrt{n}\tau^k\left(\left(\frac{1}{2\tau}\right)^k+\left(\frac{1}{2\tau}\right)^{k-1}+\left(\frac{1}{2\tau}\right)^{k-2}+\ldots+\frac{1}{2\tau}\right) \\
&= \frac{1}{2^k}\|F_{\tau\mu}(\nu_0)\|+(1-\tau)\mu_0\sqrt{n}\tau^k\left(\frac{1-(\frac{1}{2\tau})^k}{2\tau-1}\right) \\
&\leq \tau^k\left\{\|F_{\tau\mu}(\nu_0)\|+(1-\tau)\mu_0\sqrt{n}\left(\frac{1}{2\tau-1}\right)\right\},
\end{aligned}$$

where the last inequality follows for $\tau > \frac{1}{2}$. Therefore, we obtain $\left\|F_{\tau^k\mu_0}(\nu_k)\right\| \leq \epsilon$ by choosing

$$k \geq \left\lceil \frac{\log(\epsilon)-\log\left(\|F_{\mu_0}(\nu_0)\|+\frac{1-\tau}{2\tau-1}\mu_0\sqrt{n}\right)}{\log(\tau)} \right\rceil.$$

This guarantees that we are close to the central path. We also need the barrier parameter to be close to zero, i.e. $\mu_0\tau^k \leq \epsilon$. This is equivalent to

$$k \geq \frac{\log\frac{\epsilon}{\mu_0}}{\log\tau},$$

which yields the required bound on the number of iterations. $\square$

Note that, while the iterates are likely to remain within the positive definite cone during most of the progress of the algorithm, since we do not enforce this condition, there is the possibility that, at termination, the cone constraint is violated by ϵ. This departs from standard practice where the cone constraint is always satisfied, but is within the spirit of an ϵ-optimal solution: The cone constraint and the affine constraints are treated in a similar manner and will be satisfied to the same tolerance.

One question of interest at this point is whether we can pre-condition the problem to raise the smallest singular value to an arbitrary value before solving. This would complete the convergence proof but, more importantly, would turn Algorithm 3.2.1 into something of practical value.

3.3 Asymptotic Convergence

We are also able to specialize a standard result pertaining to the asymptotic convergence rate of the Gauss-Newton method. This describe the convergence to the solution when the barrier parameter μ is 0.

Theorem 3.3.1 *Assume a primal-dual pair with a unique, strictly complementary, optimal solution, denoted by* ν^**. Then for each* $c \in (1, \infty)$*, there is an* $\epsilon > 0$ *such that, from* $\nu^{(0)}$ *satisfying* $\left\|\nu^{(0)} - \nu^*\right\| < \epsilon$*, the sequence generated by the Gauss-Newton method converges to* ν^* *and obeys*

$$\left\|\nu^{(k+1)} - \nu^*\right\| \leq \frac{c\alpha}{2\sigma_{\min}^2} \left\|\nu^{(k)} - \nu^*\right\|^2. \tag{3.32}$$

where $\|J(\nu)\| \leq \alpha$ *in the* ϵ*-neighborhood of* ν^* *and where* $\sigma_{\min}$ *represents the smallest singular value of* $J(\nu^*)$*.*

Proof: The proof relies on [20], Theorem 10.2.1. We give here only the details pertaining to our case. First by Lemma 3.2.4, we have a Lipschitz constant of 1. By continuity of $[\mathfrak{D}F(\nu)]$, the Jacobian is bounded in a region around ν^* and we obtain the required α. Also, by the assumptions and Lemma 2.2.4 the smallest singular value is bounded away from zero. From these we obtain (3.32) and convergence. □

This last result suggests that, whatever upper-level algorithm we use, if we use the Gauss-Newton as the search direction, then as soon as we estimate $\sigma_{\min}$, we can compute the radius of superlinear convergence and, once within it, we can set the barrier parameter to zero, ignore the cone constraint and let the algorithm converge to the optimal solution.

3.4 Towards a Long-Step Algorithm

The Gauss-Newton direction for solving semidefinite programs was introduced in [49] without a proof of convergence but with experimental results that warranted more research. Then, in [47], a scaled version of the direction was used in an algorithm shown to be polynomially convergent. The algorithm and the convergence proof presented in this chapter are new in that the direction is used without any scaling and the algorithm never explicitly forces the iterates to remain within the positive definite cone.

The dependence on the smallest singular value of the Jacobian for choosing τ, though unsurprising in the context, should be relaxed to some other, more easily estimated function of the data. But the ultimate goal of this avenue of research is to establish convergence of a practical infeasible algorithm using long steps, that is, not restricted to a narrow neighborhood of the central path. This is still the object of investigation.

The merit function quantifies an absolute distance of the iterates to feasibility and to complementarity. Moreover, it is a simple matter to favour, for example, primal feasibility or even a subset of the constraints, by weighting the corresponding norm of the merit function. This leads to a weighted least-squares problem and such preprocessing can be done if the application suggests such an approach.

On the other hand, it may be that a relative measure of infeasibility and complementarity is more appropriate. In this case we could preprocess the problem data to ensure that the norms of matrices A_i and of C are of the same order of magnitude. This could be done if one suspects that they have been scaled inappropriately and is done in some implementations.

None of these transformations of the merit function profoundly affect what we have done in this chapter. They could be accommodated in the same framework. The appropriate scaling probably depends on whether one is interested in theoretical convergence or good practical behavior.

Chapter 4

Implementation and Experiments

We now consider the heart of all algorithms based on the Gauss-Newton direction, the numerical solution of the system (2.10), which we repeat here for convenience

$$J d_v = \begin{bmatrix} 0 & A^t & I \\ A & 0 & 0 \\ (Z \oslash I) & 0 & (I \oslash X) \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = - \begin{bmatrix} f_d \\ f_p \\ f_c \end{bmatrix}. \tag{4.1}$$

The operator J is the matrix representation of the Jacobian. It has dimension $\bar{m} \times \bar{n}$, where the row dimension is $\bar{m} := t(n) + m + n^2$, the column dimension is $\bar{n} := t(n) + m + t(n)$ and where $t(n) := n(n+1)/2$. These last parameters are m, the number of constraints in the primal problem and n, the dimension of the matrices in the primal space.

4.1 Accuracy and Stability

Where the Gauss-Newton approach best demonstrates its strength is when the problems to solve are small, dense and require accurate solutions. In those cases, short of a rank-revealing factorization, one of the best practical method for least-squares is a QR factorization with column pivoting

[35],

$$Q^t J P = R.$$

The orthogonal matrix Q is the product of Householder reflections and the permutation matrix P is chosen so that for each reflection, we permute to pivot on the column of largest norm. After the factorization we also obtain bounds on the smallest and largest singular values of J [22], cited in [11],

$$|R_{11}| \le \sigma_{\max}(J) \le \sqrt{n}|R_{11}|, \qquad |R_{\bar{n}\bar{n}}| \le \sigma_{\min}(J) \le 2^{1-\bar{n}}|R_{\bar{n}\bar{n}}|.$$

In practice, the lower bound on $\sigma_{\min}$ is much better than the theoretical bound and it is usual to use $R_{\bar{n}\bar{n}}$ as an approximation.

If, at a certain stage (say r) of the factorization, we detect that the pivot element gets too small (smaller than some tolerance), we conclude that J is rank deficient and we have the numerical rank, r. We then find the Gauss-Newton step using this factorization:

$$h = Q^t(-f), \qquad R\widetilde{d_\nu} = h, \qquad d_\nu = P\widetilde{d_\nu}.$$

Of course, we do not actually form Q. The Householder reflections are kept in factored form in the space allocated to J and applied to the right hand-side as they are computed.

If the resulting step d_ν does not lead to an accurate solution of the semi-normal equation, we assume that we terminated the factorization too late and the numerical rank is smaller than r. We drop one more column of R and re-compute the step. This, admittedly heuristic approach, does not cost much since the larger cost is in the factorization which we do only once. The full procedure is described in Algorithm 4.1.1. It requires $2\bar{n}(\bar{m} - \bar{n}/3)$ flops for the QR factorization, the most costly subroutine, and $\mathcal{O}(\bar{n}^2)$ for all other operations (triangular solve and various matrix-vector multiplications).

To quantify the accuracy of the algorithm, we first assume that the construction of J and of f does not introduce errors of order worse than machine-epsilon since it involves only matrix-vector

Algorithm 4.1.1 Dense solver for Gauss-Newton direction

Given $\epsilon > 0$; {Tolerance}
Given $J; f$; {Current Jacobian and right-hand side}
$r := \bar{n}$; {Assume full-rank}
$[P, Q, R, r] := qr(J, r)$; {Factor and return rank}
$\overline{h} = -J^t f$;
$h := -Q'f$; {Apply Householder to right-hand side}
repeat
  Solve $R_{1:r,1:r}\widetilde{d_\nu} = h_{1:r}$; {Solve non-singular block}
  $z := zeros(1 : \bar{n} - r)$; {Fill for zero singular values}
  $d_\nu := P \begin{bmatrix} \widetilde{d_\nu} \\ z \end{bmatrix}$; {Permute back}
  $r := r - 1$; {Decrease numerical rank}
until $(\|R^t R d_\nu - \overline{h}\| < \epsilon)$

products. Throughout this section we will denote machine-epsilon by

$$\varepsilon \approx 2 \times 10^{-16},$$

where the approximate value is valid for all tests we report here.

Factorization via Householder transformations is backward-stable because multiplication by orthogonal matrices is backward-stable and we know the resulting decomposition to satisfy

$$QR = A + \delta J, \quad \text{where} \quad \|\delta J\| = \mathcal{O}(\varepsilon)\|J\|.$$

Using QR decomposition by Householder reflections and column pivoting, we can be more precise. We know from [50] that the computed solution $\widehat{d_\nu}$ to (4.1) is the exact solution to a nearby least-squares problem, namely

$$\min\left\{\|(J + \delta J)d_\nu + (f + \delta f)\| \;\middle|\; d_\nu \in \mathbb{V}\right\},$$

where

$$\|\delta J\| \leq c\varepsilon n^{1/2}\|J\|, \qquad \|\delta f\| \leq c\varepsilon\|f\|, \qquad c = (6\bar{m} - 3\bar{n} + 41)\bar{n}. \tag{4.2}$$

To quantify the error on a step, following [42], we can express the relative error in the solution as

$$\frac{\|\widehat{d_\nu} - d_\nu\|}{\|d_\nu\|} \leq \frac{\varepsilon\kappa(J)}{1-\varepsilon\kappa(J)}\left(2 + (\kappa(J)+1)\frac{\|Jd_\nu + f\|}{\|J\|\|d_\nu\|}\right). \tag{4.3}$$

Perhaps more telling is the bound obtained by Demmel [19],

$$\frac{\|\widehat{d_\nu} - d_\nu\|}{\|d_\nu\|} \leq \epsilon\left\{\frac{2\kappa(J)}{\cos\theta} + \kappa^2(J)\tan\theta\right\} + \mathcal{O}(\epsilon^2),$$

where the bracketed term can be viewed as the condition number for the least-squares problem and where

$$\sin\theta = \frac{\|J\nu + f\|}{\|f\|}, \qquad \epsilon = \max\left\{\frac{\|\delta J\|}{\|J\|}, \frac{\|\delta f\|}{\|f\|}\right\} \leq \mathcal{O}(\varepsilon).$$

The angle θ is between the residual and the right-hand side, while ϵ is a relative measure of the perturbation in the involved quantities.

Note that the dependence on $\kappa^2(J)$ is not a concern in our case, for the residual tends to zero as we approach the optimal solution and therefore $\tan\theta$ also tends to zero. The direction could be inaccurate far from optimality, but that is where inaccurate solutions are not problematic. The direction becomes more accurate as we need it to be.

To compare this result with what is obtained for symmetric directions, we state an important theorem of Gu [38], the first, to our knowledge, to consider the floating-point accuracy of semidefinite solvers using symmetric directions. He obtains, for the AHO direction,

$$\frac{\|\widehat{d_\nu} - d_\nu\|}{\|d_\nu\|} \leq \frac{\kappa(J_{aho})}{1-\kappa(J_{aho})\frac{\|\delta J_{aho}\|}{\|J_{aho}\|}}\left(\frac{\|\delta J_{aho}\|}{\|J_{aho}\|} + \frac{\|\delta f\|}{\|f\|}\right), \tag{4.4}$$

where the error satisfies

$$\|\delta J_{aho}\| = \mathcal{O}\left(\varepsilon\|J_{aho}\|\right) + \mathcal{O}\left(\varepsilon\left(\|\mathcal{E}\| + \|\mathcal{F}\|\right)\|\bar{\mathcal{E}}^{-1}\|\right). \tag{4.5}$$

The $\mathcal{E}, \mathcal{F}$ terms are defined using the symmetric Kronecker product,

$$(\mathsf{G} \oslash_s \mathsf{K})\mathrm{svec}\,(\mathsf{H}) := \frac{1}{2}\mathrm{svec}\left(\mathsf{KHG}^t + \mathsf{GHK}^t\right),$$

as

$$\bar{\mathcal{E}} = \mathcal{E} + \mathcal{O}(\varepsilon\|Z\|), \quad \mathcal{E} = Z \oslash_s I, \quad \mathcal{F} = X \oslash_s I.$$

The structure of the bounds (4.4,4.3) is similar except that the error term δJ_{aho} is potentially much larger than δJ; contrast (4.2) and (4.5). Moreover, since $\kappa(J)$ for Gauss-Newton is usually smaller than $\kappa(J_{aho})$, we conclude that the bound on the error is no larger for Gauss-Newton than for AHO.

Table 4.1 shows the ratio of both condition numbers for well-conditioned random problems. Each entry in the table is the worst case of random problems with the number of constraints varying from $\lfloor t(n)/2 \rfloor$ to $t(n) - 1$.

n	2	4	8	16	32
$\kappa(J_{aho})/\kappa(J)$	2	79	562	2314	131686

Table 4.1: Comparaison of condition numbers of the AHO and GN systems.

The Gauss-Newton direction should not run into any problems until $\varepsilon\kappa(J) = \Omega(1)$ which will happen, if at all, closer to the optimal solution than for AHO. This is confirmed by numerical experimentation. Of course, this argument describes the accuracy possible in the computation of the step and not the accuracy of the solution of the optimization problem. An analysis of the accuracy of the whole iterative process described in algorithm 4.1.1 remains to be done. But an accurate step computation is one element explaining the accuracy of the solutions exhibited throughout this chapter.

4.1.1 Well-Conditioned Problems

For the first set of experiments, exemplified by Table 4.2, we have used a simple random problem generator. The problems are all well-conditioned and have strictly complementary unique optimal solutions. In all cases, Algorithm 4.1.1 was able to solve so that infeasibility measure and the complementarity gap are smaller than 10^{-13}. The number of iterations is only weakly dependent on the size of the problems. We first display the progress of one solution, in some detail, to relate the decrease in the infeasibility and in the complementarity gap. Not surprisingly, infeasibility decreases faster than complementarity since the corresponding equations are linear.

Iter	$\|F_d\|$	$\|f_p\|$	$\langle Z, X\rangle/n$
1	+8.900e+01	+4.175e+01	+8.749e-01
2	+5.433e+01	+3.545e+01	+8.580e-01
3	+1.503e+01	+1.759e+01	+6.269e-01
4	+3.066e+00	+3.465e-03	+3.841e-01
5	+2.131e-02	+4.408e-04	+1.278e-01
6	+5.573e-03	+8.302e-05	+2.873e-02
7	+1.042e-04	+3.153e-06	+1.367e-02
8	+1.611e-05	+2.913e-07	+7.862e-04
9	+7.275e-08	+1.620e-09	+8.680e-05
10	+4.473e-13	+2.041e-14	+1.812e-06
11	+1.017e-14	+2.800e-15	+3.784e-08
12	+1.030e-14	+3.426e-15	+7.902e-10
13	+1.318e-14	+5.184e-15	+1.650e-11
14	+1.138e-14	+4.835e-15	+4.964e-14

Table 4.2: One instance of well-conditioned problem. $n = 15, m = 30$.

The more interesting aspect of the accuracy of the Gauss-Newton direction is exemplified in Table 4.4 where we contrast the infeasibility $\max\{\|F_d\|, \|F_c\|\}$ and the complementarity gap $\langle Z, X\rangle/n$ obtained from various directions. Larger numbers are better. Table 4.4 results were obtained by averaging the outcomes of 100 random instances of each type of problems generated by the SDPT3[1] version 2.1 [78] random problem generator (normally distributed data from the MATLAB command `randn`). The implementation was SDPT3, to which we added the Gauss-Newton direction, and which we instrumented to let the algorithm run until no more progress was possible. The test problems are of four different classes described in Table 4.3 where B is the weighted adjacency matrix of a graph and where the indices i, j, in the case of Lovász θ function, loop on the vertices corresponding to each edge of a graph.

We note that the Gauss-Newton direction was, in every case, capable of a more accurate solution than all other directions, even the direction GT, developed specifically for that purpose [81]. We include symmetric directions other than AHO to highlight the empirical result first noticed by Todd, Toh and Tütüncü [78], and often exhibited afterwards, that among symmetric directions, AHO is the more accurate. From this point onward, we will mostly restrict our

[1] `http://www.math.cmu.edu/~reha/sdpt3.html`

Random :	$\min\left\{\langle C, X\rangle \mid \mathcal{A}(X) = b, X \in \mathbb{S}^n_+\right\}$
Norm min. :	$\min\left\{\|B_0 + \sum_{i=1}^m x_i B_i\| \mid x \in \mathbb{R}^m\right\}$
Maxcut :	$\min\left\{\langle B - \mathrm{Diag}\, Be, X\rangle \mid \mathrm{Diag}\,(X) = e/4, X \in \mathbb{S}^n_+\right\}$
Lovász θ :	$\min\left\{\langle C, X\rangle \mid \langle I, X\rangle = 1, e_i e_j^t + e_j e_i^t = 0, X \in \mathbb{S}^n_+\right\}$

Table 4.3: SDPT3 test problems.

comparisons to AHO but the reader should keep in mind that other directions would, in general, do worse.

	$-\log_{10}\max\{\|F_d\|, \|f_p\|\}$					$-\log_{10}\langle Z, X\rangle/n$				
	AHO	HKM	NT	GT	GN	AHO	HKM	NT	GT	GN
random	14.7	9.1	10.6	9.5	13.3	12.0	10.7	5.5	12.8	14.4
norm min.	15.0	9.9	10.9	15.3	14.8	13.8	12.7	10.1	14.4	14.9
Maxcut	15.7	9.6	10.6	14.8	15.8	14.4	12.4	9.3	14.4	15.5
Lovász θ	14.7	9.2	9.2	13.9	14.8	14.2	13.7	12.8	14.1	15.3

Table 4.4: Solutions of SDPT3 test problems. Average of one hundred random instances.

4.1.2 Ill-Conditioned Problems

For the second series of test, we generate ill-conditioned problems in the following manner: First we create an orthogonal matrix Q, then from a chosen rank $1 < r < n$, we generate positive diagonal matrices D_x, D_z of dimension $r \times r$ and $(n-r)\times(n-r)$ respectively from which we obtain

$$X^* = Q \begin{bmatrix} D_x & 0 \\ 0 & 0 \end{bmatrix} Q^t, \qquad Z^* = Q \begin{bmatrix} 0 & 0 \\ 0 & D_z \end{bmatrix} Q^t.$$

We also generate a random y^* and

$$A_k = Q^t \begin{bmatrix} U_k & L_k^t \\ L_k & V_k \end{bmatrix} Q,$$

for random U_k, V_k, L_k where $\|L_k\|_2 \approx 10^{-10}$. Then we form $b = \mathcal{A}\mathrm{svec}(X^*)$, and $C = Z^* + \mathrm{smatrix}(\mathcal{A}^t y^*)$. From [6], we know that this procedure will, in general, create instances with ill-conditioned Jacobians at the solution. We report on 50 random instances for each of the various dimensions and ranks and average the results in Table 4.5. The dimensions are chosen so that the reader can compare these numbers with those in [38]. The Infeasibility column corresponds to the average of $-\log_{10} \max\{\|F_d\|, \|f_p\|\}$ and the Gap column corresponds to the average of $-\log_{10} \langle Z, X\rangle / n$. We notice that for the ill-conditioned problems of Table 4.5, the Gauss-Newton

			AHO			GN		
r	n	m	iter.	Infeas.	Gap	iter.	Infeas.	Gap
3	10	9	18	14.2	15.1	13	14.3	15.4
6	20	24	22	12.1	14.6	17	13.8	16.3

Table 4.5: Solutions of ill-conditioned problems. Average of fifty random instances.

direction was in all cases more accurate than AHO. Moreover, the number of iterations to attain this accuracy was less than AHO. This is explained by the marginal progress that AHO does in the last iterations while the progress of the Gauss-Newton direction is not affected by this kind of ill-conditioning.

4.1.3 When Slater's Constraint Qualification Fails

Even if we assumed that the problems, up to now, had strictly feasible points, there are classes of practical problems where this assumption fails. Recently Gruber and Rendl[37] developed a robust algorithm specifically designed for these problems. Since the requirements for the Gauss-Newton direction do not include strict interior points, we attempted the same problems. The first example problem is

$$\max\left\{\langle C, X\rangle \;\middle|\; \mathrm{diag}\,(X) = e, \langle J, X\rangle = \alpha, X \in \mathbb{S}^n_+\right\}, \tag{4.6}$$

where e is the all-ones vector and J is the all-ones matrix. For positive values of the parameter α, the problem has strictly feasible primal points but this interior region shrinks to the empty set as α is reduces to zero. The first set of experiments reported by Gruber and Rendl[37] uses $\alpha = 10^{-7}$.

n	m	iter	$\|F_d\|$	$\|f_p\|$	$\langle Z, X\rangle$
10	11	24	9.296152e-13	5.594466e-11	5.842689e-07
20	21	25	2.260052e-12	4.882512e-15	7.906645e-07
30	31	22	4.211995e-12	9.367561e-15	2.190712e-06
40	41	23	1.816565e-12	7.334760e-15	2.301071e-06

Table 4.6: Problem (4.6) with $\alpha = 10^{-7}$ and accuracy set to 10^{-5}.

We have done the same experiment and report the result in Table 4.6. Each column, after the first two indicating the dimension, represents the worst case of each of twenty random experiment for the corresponding entry. We have set the required accuracy at 10^{-5} as they did. The important point to note is that their number of iterations is never less that 114 (see Table 2 of [37]). Our result of less than 25 iterations compares very favourably and illustrates a strength of the Gauss-Newton approach: since feasibility is not given precedence over complementarity, the near-absence of feasible points inside the cone is of no consequence. We had to make no modifications to the implementation for these or any other problems.

The second experiment is generated for the same problem but with $\alpha = 0$. In this case, there is no strictly interior primal point. We report the results in Table 4.7. To contrast with Gruber and Rendl's result, the reader needs to be aware that their algorithm averaged 115 iterations(see Table 3 in [37]).

n	m	iter	$\|F_d\|$	$\|f_p\|$	$\langle Z, X\rangle$
10	11	23	2.377587e-12	4.076899e-08	9.096929e-06
20	21	22	2.479800e-12	1.516398e-08	1.050569e-06
30	31	25	5.153522e-12	5.729514e-08	1.416319e-06

Table 4.7: Problem (4.6) with $\alpha = 0$ and accuracy set to 10^{-5}.

Even more telling is that while Gruber and Rendl needed to relax the accuracy requirements to solve the problems without interior points, and even then the algorithm failed two instances, the Gauss-Newton algorithm can reach any required level of accuracy for these problems. We ran the experiment a third time, with $\alpha = 0$, requesting increased accuracy and report the results in Table 4.8.

n	m	iter	$\|F_d\|$	$\|f_p\|$	$\langle Z, X\rangle$
50	51	32	3.467449e-12	6.469784e-15	-1.233752e-14

Table 4.8: Problem (4.6) with $\alpha = 0$ and increased accuracy.

Gruber and Rendl then moved on to problems where both primal and dual feasible sets fail to have interior points:

$$\max\left\{\langle C, X\rangle \mid \left\langle v_1 v_1^t, X\right\rangle = 0, \left\langle v_i v_i^t, X\right\rangle = 1, 2 \le i \le m, X \in \mathbb{S}_+^n\right\}, \tag{4.7}$$

where the vectors v_i, with $1 \le i \le i$ are chosen randomly but orthogonal and $C = \sum_{i=1}^{i=n-1} \alpha_i v_i v_i^t$, for some random positive vector α.

n	m	iter	$\|F_d\|$	$\|f_p\|$	$\|\langle Z, X\rangle\|$
10	9	7	8.015765e-11	8.009760e-11	5.080324e-06
20	19	8	9.000070e-11	9.001461e-11	1.743560e-06
30	28	8	1.272965e-10	9.000636e-11	2.615344e-06
40	15	5	2.593612e-09	9.999999e-11	1.651607e-07
40	30	5	2.642280e-09	1.000001e-10	3.821364e-07
40	39	5	2.231644e-09	9.999997e-11	5.396936e-07
50	49	5	2.928152e-09	9.999996e-11	6.840107e-07

Table 4.9: Problem (4.7) with accuracy set to 10^{-5}.

We report the result in Table 4.9. We never needed more than 8 iterations while Gruber and Rendl ([37], Table 4) report an average of 52 iterations to attain the same accuracy.

4.1.4 DIMACS Challenge Problems

More recently, because of the DIMACS Challenge[2], a new set of test problems surfaced that proved very difficult to solve for all current implementations. Among them is a series of H^∞ control problems of low dimension and small feasible region, and whose Jacobians are rank deficient at the optimal solution. Our algorithm was not designed for this type of problems, yet it performed surprisingly well. Table 4.10 contrasts our results with the best result available at this time taken

[2] http://dimacs.rutgers.edu/Challenges/Seventh/

from benchmarks run by Hans Mittelmann (http://plato.la.asu.edu/errors.html).

Problem	Error	SDPT3	SeDuMi	GN
hinf12	$\|\|b - \mathcal{A}(X)\|\|$	.22e-7	.22e-12	.63e-04
	$\max\{0, -\lambda(X)\}$	0	0	.14e-16
	$\|\|\mathcal{A}^*(y) + Z - C\|\|$	.14e-7	0	.24e-9
	$\max\{0, -\lambda(Z)\}$	0	0	0
hinf13	$\|\|b - \mathcal{A}(X)\|\|$	.16e-3	.21e-3	.42e-08
	$\max\{0, -\lambda(X)\}$	0	0	0
	$\|\|\mathcal{A}^*(y) + Z - C\|\|$	.53e-11	0	.37e-05
	$\max\{0, -\lambda(X)\}$	0	.96e-2	.23e-10

Table 4.10: H^∞ control problems.

Even though our implementation did not do as well as SeDuMi in this case, it is worth noting that these outstanding results of SeDuMi are the product of a number of years of tuning the implementation to handle ill-posed problems. We have only a research-level implementation with no special handling of such difficult cases. That we can produce these approximate solutions attests to the robustness of the Gauss-Newton direction.

4.2 Sources of Sparsity

There are three sources of zeroes in the solution of the semidefinite program pair. The first and simplest to handle arises directly from the domain of the primal variables. For simplicity of exposition, until now we assumed that the primal domain was $\mathbb{S}^n_+$, the cone of semidefinite matrices of order n. Yet, for a number of applications the domain actually is a Cartesian product of semidefinite cones, $\mathbb{S}^{n_1}_+ \times \ldots \times \mathbb{S}^{n_k}_+$, where $n_1 + \ldots + n_k = n$. A typical example comes from H^∞-control where the problem to solve has the form

$$\min\left\{\sum_{j=1}^{k} \langle C_j, X_j\rangle \mid \sum_{j=1}^{k} \langle A_{ij}, X_j\rangle = b_i, 1 \le i \le m, X_j \in \mathbb{S}^{n_j}_+, 1 \le j \le k\right\}.$$

After transformation of the problem into our standard formulation, the resulting primal variable (an embedding of $\mathbb{S}^{n_1} \times \ldots \times \mathbb{S}^{n_k}$ into $\mathbb{S}^n, n = n_1 + \ldots + n_k$ via $X = X_1 \oplus \ldots \oplus X_k$) has a block diagonal structure.

The domain might also be a product of semidefinite cones, Lorentz cones and nonnegative orthants constraints, all of which can be embedded in a semidefinite cone. We have already seen, for example, that a standard linear program requiring $x \in \mathbb{R}^n_+$ can be solved via $X \in \mathbb{S}^n_+$ and X diagonal. If we use a projection P_x that only extracts the diagonal elements, then the resulting system solved at each iteration is exactly the size one expects for a linear program, namely $(n+m+n) \times (n+m+n)$.

In the case of a Lorentz cone of order n (denoted $\mathbb{L}^n_+$),

$$\mathbb{L}^n_+ := \left\{ x \in \mathbb{R}^{n+1} \mid x_0 \geq \sqrt{x_1^2 + \ldots + x_n^2} \right\},$$

the embedding is

$$x \in \mathbb{L}^n_+ \iff \mathrm{Arrow}(x) := \begin{bmatrix} x_0 & x_1 & x_2 & \ldots & x_n \\ x_1 & x_0 & 0 & \ldots & 0 \\ x_2 & 0 & x_0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ x_n & 0 & 0 & \ldots & x_0 \end{bmatrix} \in \mathbb{S}^{n+1}_+.$$

The operator corresponding to Diag in the standard linear programming case is Arrow in the case of the Lorentz cone. Again, most entries are zeros.

To handle this first source of zeros we will use two related projections

$$P_x : \mathbb{R}^{t(n)} \to \mathbb{R}^{nz(x)}, \quad E_x : \mathbb{R}^{n^2} \to \mathbb{R}^{nz(zx)} \tag{4.8}$$

where $nz(x)$ is the number of nonzero elements of $x = \mathrm{svec}\,(X)$, the upper triangular part of X and $nz(zx)$ is the number of nonzeros of ZX considered as a non-symmetric matrix. (The projection corresponding to P_x in $\mathbb{S}^n$-space will be denoted $\mathcal{P}_x$.) These projections are constructed at the start of the algorithm, from the structure of the domain of the primal variables. For example, say the domain is $\mathbb{S}^2_+ \times \mathbb{L}^2_+ \times \mathbb{R}_+$ and

$$
\begin{aligned}
X &= \begin{bmatrix} 1 & 2 & 0 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 5 & 7 & 0 \\ 0 & 0 & 5 & 6 & 0 & 0 \\ 0 & 0 & 7 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 \end{bmatrix}, \\
\operatorname{svec}(X) &= [1,2,3,0,0,4,0,0,5,6,0,0,7,0,8,0,0,0,0,0,9]^t, \\
P_x(\operatorname{svec}(X)) &= [1,2,3,4,5,6,7,8,9]^t, \\
E_x(\operatorname{avec}(X)) &= [1,2,3,4,5,7,5,6,0,7,0,8,9]^t.
\end{aligned}
$$

The projected primal feasibility equation and corresponding equation of (4.1) are now

$$\mathcal{A}\mathcal{P}_x^*(\mathcal{P}_x d_X) = b, \qquad AP_x^t P_x d_x = -f_p.$$

We have obtained a system where the number of variables accurately represents the original problem. The embedding into a larger semidefinite cone has not cost us anything in terms of memory.

The second source of zeros arises from the sparsity pattern of the matrices A_i and C and the dual feasibility equation, $\sum_{i=1}^m A_i y_i + Z = C$. If all A_i and C are sparse or more precisely, if the union of the sparsity patterns of these matrices is sparse then Z must also be sparse. This is especially true in the case of relaxation of combinatorial problems. A typical example, that also happens to be one of the best relaxations for the Maxcut problem [8] yields a pattern as in Figure 4.1 where the nonzeros cover only 25% of the matrix Z.

We will handle these zeros with the same type of projection, detected at the start of the algorithm by considering the union of the sparsity patterns of all A_i and of C. We want P_z to be

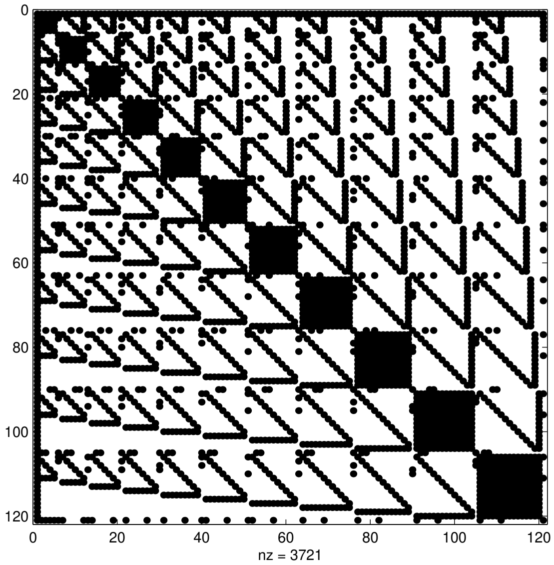


Figure 4.1: Maxcut relaxation dual variable Z sparsity structure.

a zero-one matrix and have the maximum number of zeros subject to

$$P_z : \mathbb{R}^{t(n)} \to \mathbb{R}^{nz(z)}, \qquad P_z \operatorname{svec}(H) = \operatorname{svec}(H), H = C, A_i, 1 \leq i \leq m. \tag{4.9}$$

With this constraint, $nz(z)$ indicates the number nonzero elements of $z = \operatorname{svec}(Z)$, the upper triangular part of Z. (The corresponding projection in $\mathbb{S}^n$-space will be denoted $\mathcal{P}_z$.) The projected dual feasibility equation corresponding equation of (4.1) are

$$\mathcal{P}_z \mathcal{A}^*(y) + \mathcal{P}_z Z = \mathcal{P}_z C, \qquad P_z A^t y + P_z d_z = -P_z f_d.$$

With this projection, we eliminate some columns from the Gauss-Newton system (corresponding to the zero components of Z) and we eliminate some rows (corresponding to constraints on these zeros).

We note that P_z projects onto a subspace of the co-domain of P_x and therefore z has at least as many zeros as x. This implies that we can project the complementarity equation using E_x and obtain

$$E_x(Z \oslash I)P_x d_x + E_x(I \oslash X)P_z d_z = -E_x f_c.$$

This eliminates some more rows from the system. For example, if the original system is a standard linear program, the resulting Jacobian is of size $(n+m+n) \times (n+m+n)$, as one would expect for a linear program.

We try to exploit the sparsity of z and of the A_i as much as possible. This is the subject of the next section. Before we end this section, we mention the last source of zeros in the problem, the asymmetric Kronecker products $Z \oslash I$ and $I \oslash X$. The resulting matrices have at most n^3 nonzeros (if X and Z are dense) while being of dimension $n^2 \times t(n)$. This is illustrated in Figure 4.2

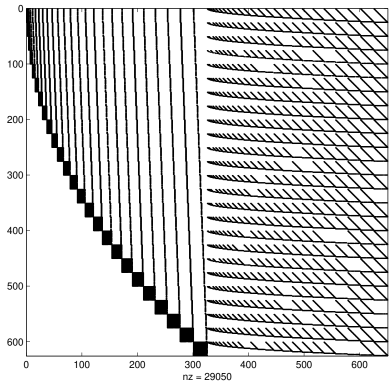


Figure 4.2: Sparsity structure of $[Z \oslash I, I \oslash X]$ $(n = 25)$.

It would also be possible to handle this sparsity via projections if we were to solve the system by a QR factorization of J. But since we intend to solve the system in two steps, for d_z first, then for d_x, d_y, the nonzero pattern gets more complex. We therefore opted for a sparse matrix structure, using compressed columns to which we will apply permutations to minimize fill during the factorization phase.

After the projections, the system (4.1) becomes

$$J_p d_v = \begin{bmatrix} 0 & P_z A^t & I \\ A P_x^t & 0 & 0 \\ E_x(Z \oslash I)P_x^t & 0 & E_x(I \oslash X)P_z^t \end{bmatrix} \begin{bmatrix} P_x d_x \\ d_y \\ P_z d_z \end{bmatrix} = - \begin{bmatrix} P_z f_d \\ f_p \\ E_x f_c \end{bmatrix}, \tag{4.10}$$

where J_p is of size $(nz(z)+m+nz(zx)) \times (nz(x)+m+nz(z))$, and we are solving for $P_x d_x, d_y, P_z d_z$. To visually contrast the original Jacobian and the reduced Jacobian under the effect of the projections, see Figure 4.3

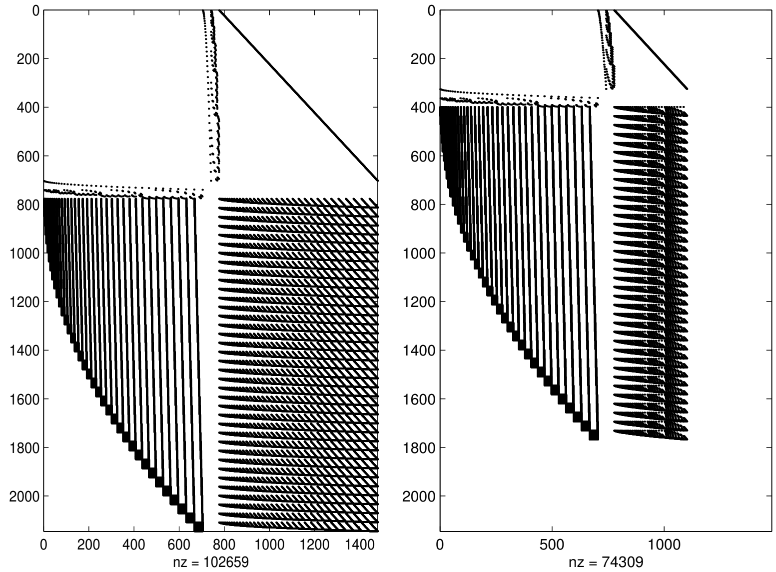


Figure 4.3: Sparsity structure of full and of reduced Jacobian (Maxcut instance).

4.3 Separability for Sparsity

Consider in (4.10) that the first two columns of J, especially if A is sparse, will be very sparse. If there is a way to first solve for d_z, taking advantage of this sparsity, then back solve for d_y, d_x, it might prove advantageous to do so. We first describe how to separate in more general terms.

Consider $K \in \mathbb{R}^{h \times k}$, $L \in \mathbb{R}^{h \times l}$, $h \in \mathbb{R}^h$ where $h > k + l$ and the optimization problem

$$\min \left\{ \|Lz + Kw + h\| \mid w \in \mathbb{R}^k, z \in \mathbb{R}^l \right\}. \tag{4.11}$$

Lemma 4.3.1 *The least-squares solution to (4.11) can be expressed by*

1. $w^* = -[(I - LL^\dagger)K]^\dagger h$,

2. $z^* = -L^\dagger(h + Kw^*)$.

Proof: Say w^* is the $\mathbb{R}^k$ part of the optimal solution. Then

$$\min \left\{ \|Lz + Kw + h\| \mid w \in \mathbb{R}^k, z \in \mathbb{R}^l \right\} = \min \left\{ \|Lz + \bar{h}\| \mid z \in \mathbb{R}^l \right\},$$

where $\bar{h} = Kw^* + h$. The solution of which is given by 2. Substituting back we get

$$\min \left\{ \|Lz + Kw + h\| \mid w \in \mathbb{R}^k, z \in \mathbb{R}^l \right\} = \min \left\{ \|(I - LL^\dagger)(Kw + h)\| \mid w \in \mathbb{R}^k \right\},$$

the solution of which is given by 1. □

This approach is inspired by [33, 34] where partial separability was used to isolate variables involved in linear blocks from those involved in nonlinear blocks. The motivation, in our case, is different: some of the variables are subjected to dense operators, others to sparse operators. We are trying to isolate them and treat them differently.

Since, for the class of problems we have in mind, Z and A are sparse, and correspondingly d_Z is sparse, we make the following identification between the columns of J_p from (4.10) and Lemma 4.3.1, defining new symbols to simplify the notation:

$$[L \mid K] := \left[\begin{array}{cc|c} 0 & P_z A^t & I \\ A P_x^t & 0 & 0 \\ E_x(Z \oslash I)P_x^t & 0 & E_x(I \oslash X)P_z^t \end{array}\right] =: \left[\begin{array}{cc|c} 0 & \widetilde{A}_z^t & I \\ \widetilde{A}_x & 0 & 0 \\ \widetilde{Z} & 0 & \widetilde{X} \end{array}\right]. \tag{4.12}$$

We also define

$$\widetilde{d_x} := P_x d_x, \quad \widetilde{d_z} := P_z d_z, \quad \widetilde{f_d} := P_z f_d, \quad \widetilde{f_c} := E_x f_c, \tag{4.13}$$

and $\widetilde{d_y} := d_y, \quad \widetilde{f_p} := f_p$.

4.3.1 Solution via Pseudo-Inverse

For the sake of completeness, we give an expression for the pseudo-inverses and the symbolic solution of the system. We need $L^\dagger$ and $I - LL^\dagger$.

Lemma 4.3.2 *For* L *defined as in (4.12),*

$$L^\dagger := \begin{bmatrix} 0 & (\widetilde{A}_x^t\widetilde{A}_x + \widetilde{Z}^t\widetilde{Z})^{-1}\widetilde{A}_x^t & (\widetilde{A}_x^t\widetilde{A}_x + \widetilde{Z}^t\widetilde{Z})^{-1}\widetilde{Z}^t \\ (\widetilde{A}_z^t)^\dagger & 0 & 0 \end{bmatrix}.$$

Proof: Since the Jacobian is of full column rank, then L is also full column rank. To satisfy the Moore-Penrose equations, we need only to show $L^\dagger L = I$ and $LL^\dagger = (LL^\dagger)^t$. Both equations are readily verified by simple matrix multiplication. □

To solve (4.10) using the technique of Lemma 4.3.1 we need to solve in a least-squares sense a system of size $(nz(z) + m + nz(zx)) \times nz(z)$, thin and built from sparse operators,

$$\begin{bmatrix} (I - \widetilde{A}_z^t(\widetilde{A}_z^t)^\dagger) \\ \widetilde{A}_x(\widetilde{A}_x^t\widetilde{A}_x + \widetilde{Z}^t\widetilde{Z})^{-1}\widetilde{Z}^t \\ (I - \widetilde{Z}(\widetilde{A}_x^t\widetilde{A}_x + \widetilde{Z}^t\widetilde{Z})^{-1}\widetilde{Z}^t)\widetilde{X} \end{bmatrix} \widetilde{d_z} = - \begin{bmatrix} (I - \widetilde{A}_z^t(\widetilde{A}_z^t)^\dagger)\widetilde{f_d} \\ \widetilde{f_p} + \widetilde{A}_x(\widetilde{A}_x^t\widetilde{A}_x + \widetilde{Z}^t\widetilde{Z})^{-1}(\widetilde{Z}^t\widetilde{f_c} - \widetilde{A}_x^t\widetilde{f_p}) \\ \widetilde{f_c} - \widetilde{Z}(\widetilde{A}_x^t\widetilde{A}_x + \widetilde{Z}^t\widetilde{Z})^{-1}(\widetilde{Z}^t\widetilde{f_c} - \widetilde{A}_x^t\widetilde{f_p}) \end{bmatrix}. \tag{4.14}$$

We solve (4.14) by a QR factorization via column-oriented Householder reflections. Before the factorization, we need to solve

$$(\widetilde{A}_x^t\widetilde{A}_x + \widetilde{Z}^t\widetilde{Z})[U \mid V] = [\widetilde{Z}^t \mid \widetilde{A}_x^t], \tag{4.15}$$

a very sparse system, to obtain U and V needed to get

$$\begin{aligned}
\widetilde{A}_x(\widetilde{A}_x^t\widetilde{A}_x+\widetilde{Z}^t\widetilde{Z})^{-1}\widetilde{Z}^t &= \widetilde{A}_x U, & \widetilde{Z}(\widetilde{A}_x^t\widetilde{A}_x+\widetilde{Z}^t\widetilde{Z})^{-1}\widetilde{Z}^t\widetilde{X} &= \widetilde{Z}U\widetilde{X},\\
\widetilde{A}_x(\widetilde{A}_x^t\widetilde{A}_x+\widetilde{Z}^t\widetilde{Z})^{-1}\widetilde{Z}^t\widetilde{f}_c &= \widetilde{A}_x U\widetilde{f}_c, & \widetilde{A}_x(\widetilde{A}_x^t\widetilde{A}_x+\widetilde{Z}^t\widetilde{Z})^{-1}\widetilde{A}_x^t\widetilde{f}_p &= \widetilde{A}_x V\widetilde{f}_p,\\
\widetilde{Z}(\widetilde{A}_x^t\widetilde{A}_x+\widetilde{Z}^t\widetilde{Z})^{-1}\widetilde{Z}^t\widetilde{f}_c &= \widetilde{Z}U\widetilde{f}_c, & \widetilde{Z}(\widetilde{A}_x^t\widetilde{A}_x+\widetilde{Z}^t\widetilde{Z})^{-1}\widetilde{A}_x^t\widetilde{f}_p &= \widetilde{Z}V\widetilde{f}_p.
\end{aligned}$$

Once we have $\widetilde{d_z}$, we can obtain $\widetilde{d_x}, \widetilde{d_y}$ via

$$\begin{aligned}
\widetilde{d_y} &= -(\widetilde{A}_z^t)^\dagger(\widetilde{d_z}+\widetilde{f}_d),\\
\widetilde{d_x} &= -(\widetilde{A}_x^t\widetilde{A}_x+\widetilde{Z}^t\widetilde{Z})^{-1}(\widetilde{A}_x^t\widetilde{f}_p+\widetilde{Z}^t(\widetilde{X}\widetilde{d_z}+\widetilde{f}_c)) = -V\widetilde{f}_p - U(\widetilde{X}\widetilde{d_z}+\widetilde{f}_c).
\end{aligned}$$

This approach has not proved to be very accurate for hard problems, probably, in part, because of the construction of $\widetilde{A}\widetilde{A}^t+\widetilde{Z}\widetilde{Z}^t$, which is akin to the normal equations. Nevertheless, for well-conditioned, sparse problem, it can be very fast. The factorization of (4.15), can conceivable be done by an downdating/updating procedure, making the code faster but we have not investigated this factorization.

For a production version of our code, we might use this fast approach for the initial iterations and then, if memory permits, use the approach described in the next section to obtain the required accuracy.

4.3.2 Solution via Householder Reflections

There is a numerically better approach for the solution of (4.10) using the technique of Lemma 4.3.1. Note that we require $L^\dagger$ as well as $I - LL^\dagger$. Since L is of full column rank, we could use the identity

$$L^\dagger = (L^tL)^{-1}L^t.$$

But this would unnecessarily worsen the condition number in the case where the columns of L are nearly dependent. A better approach is to compute $L = Q_LR_L$ to obtain

$$L^\dagger = (L^tL)^{-1}L^t = (R_L^tQ_L^tQ_LR_L)^{-1}R_L^tQ_L^t = (R_L^tR_L)^{-1}R_L^tQ_L^t = R_L^{-1}R_L^{-t}R_L^tQ_L^t = R_L^{-1}Q_L^t,$$

and therefore

$$LL^\dagger = Q_L R_L R_L^{-1} Q_L^t = Q_L Q_L^t.$$

The numerous cancellations are the key to the accuracy of the result.

The second requirement is to efficiently compute $(I - LL^\dagger)K = (I - Q_L Q_L^t)K$. Say that $Q_L = P_1 P_2 \cdots P_l$ where the P_i are Householder matrices $P_i = I - \beta v_i v_i^t$. Then

$$\begin{aligned} I - Q_L Q_L^t &= I - \{P_1 P_2 \cdots P_{l-1} P_l P_l P_{l-1} \cdots P_1\} \\ &= I - \{(I - \beta_1 v_1 v_1^t) \cdots (I - \beta_l v_l v_l^t)(I - \beta_l v_l v_l^t) \cdots (I - \beta_1 v_1 v_1^t)\}. \end{aligned}$$

The reflections P_i are stored and applied in factored form without ever being formed, in the standard manner,

$$(I - \beta v v^t)K = K - \beta v (K^t v)^t$$

and the application of P_i is later denoted by *ApplyHouse* in algorithm descriptions. The overall procedure of the sparse solver for the Gauss-Newton direction is given by Algorithm 4.3.1.

The computational cost of this algorithm, as indicated below its description, can be bounded by the cost of the two QR factorizations. The bound is an overestimate of the flop count since it does not take into account the sparsity of the Jacobian, but only the sparsity of the X and Z matrices. On the other hand it does not account for the memory accesses incurred by this approach.

Since L is very sparse but K, and more so $(I - Q_L Q_L^t)K$ is denser, we treat these two parts separately, with the first using a sparse matrix structure, and the second a dense matrix structure which we call $\overline{K}$ in the algorithmic description 4.3.1.

There is one more advantage of this technique. Since the sparsity pattern of L is the same at every iteration, we can spend some time at the beginning to find an adequate fill-minimizing ordering.

Of course, this particular handling of sparsity via a two-step QR factorization would defeat the purpose of finding accurate solutions if stability of the usual QR algorithm was lost. Following

Algorithm 4.3.1 Sparse solver for Gauss-Newton direction

Given $f; J \equiv [LK]; \overline{K} := K; \overline{f} := f;$ $\{$Current right-hand side and Jacobian$\}$
Construct L;
Factor $Q_L R_L = L$; $\{Q_L = P_1 \cdots P_l\}$
Save and discard R_L;
for $i = 1, \ldots, l$ **do**
  $\overline{K} :=$ ApplyHouse$(P_i, \overline{K})$;
end for $\{\overline{K} = Q_L^t K\}$
for $i = l, \ldots, 1$ **do**
  $\overline{K} :=$ ApplyHouse$(P_i, \overline{K})$;
end for $\{\overline{K} = QQ_L^t K\}$
$\overline{K} := \overline{K} - K$; $\{\overline{K} = (I - Q_L Q_L^t)K\}$
Discard Q_L;
Factor $\overline{Q}_K \overline{R}_K = \overline{K}$; $\{\overline{Q}_K = P_{l+1} \cdots P_{l+k}\}$
for $i = l+1, \ldots, l+k$ **do**
  $\overline{f} :=$ ApplyHouse$(P_i, \overline{f})$;
end for $\{\overline{f} = Q_K^t f\}$
Solve $\overline{R}_K d_z = \overline{f}$;
$f := f + K d_z$;
for $i = 1, \ldots, l$ **do**
  $f :=$ ApplyHouse(P_i, f);
end for $\{f = Q_L^t (f + K d_z)\}$
Discard $\overline{R}_K, \overline{Q}_K$; Retrieve R_L;
Solve $R_L \begin{bmatrix} d_y \\ d_x \end{bmatrix} = -f$;

$$\begin{aligned}
\text{QR cost} &= 2\left[nz(x) + m + nz(z)\right]^2 \left[nz(z) + m + nz(zx) - (nz(x) + m + nz(z)/3)\right] \\
&= 2\left[nz(x) + m + nz(z)\right]^2 \left[\frac{2}{3}(nz(z) + m) - \frac{1}{3}nz(x) + nz(zx)\right] \\
\text{Total cost} &= \mathcal{O}\big(\left[nz(x) + m + nz(z)\right]^2 \left[nz(zx) + m\right]\big)
\end{aligned}$$

the pioneering work of Gu [38] we can show that this is not the case. The critical element is our methods of computing $(I-LL^{\dagger})K$. This is not an orthogonal matrix, yet there is a way to compute the product that maintains the accuracy of a product of orthogonal matrices instead of a general matrix product.

Lemma 4.3.3 *The calculations of* $(I - LL^{\dagger})K$ *in Algorithm 4.3.1, under the usual floating-point model of arithmetic, satisfies*

$$\mathbf{fl}((I - LL^{\dagger})K) = (I - LL^{\dagger})(K + \Delta),$$

where $\|\Delta\| = \mathcal{O}(\varepsilon)\|K\|$.

Proof: We are computing this matrix product via the identity

$$\begin{aligned}
(I - LL^{\dagger})K &= (I - Q_L Q_L^t)K \\
&= (I - \{P_1 P_2 \cdots P_{l-1} P_l P_l P_{l-1} \cdots P_1\})K \\
&= I - \{(I - \beta_1 v_1 v_1^t) \cdots (I - \beta_l v_l v_l^t) \cdots (I - \beta_1 v_1 v_1^t)\}K.
\end{aligned}$$

where the P_i are Householder reflections. If we apply the sequence of reflections by the usual $P_i K = (I - \beta_i v_i v_i^t)K = K - \beta_i v_i (v_i^t K)$, then it is known (Lemma 3.1 and Theorem 3.5 of [19]) that the sequence of products satisfies $\mathbf{fl}(P_l \ldots P_i K) = P_l \ldots P_i(K + E)$, where $\|E\| \leq l\varepsilon\|K\|$. Since we are simply applying twice the number of reflections, we obtain the result. □

Lemma 4.3.4 *(Backward stability.) The solution* $\widehat{d_v}$ *obtained by algorithm 4.3.1 satisfies*

$$\min\{|(J + \delta J)d_v + (f + \delta f)|\}$$

where $\|\delta J\| = \mathcal{O}(\varepsilon)\|J\|$, *and* $\|\delta f\| = \mathcal{O}(\varepsilon)\|f\|$.

Proof: This is a consequence of Lemma (4.3.3) and the standard proof of backward stability of least-squares solution via QR factorization (See, for example [42], Theorem 19.3). □

4.4 Solution via Givens Rotations

Long after we had implemented the previous approaches, it occurred to us that a row-by-row factorization might be as effective in solving equation (4.10), especially after a re-ordering of the columns to produce the equivalent system

$$\begin{bmatrix} I & P_z A^t & 0 \\ 0 & 0 & AP_x^t \\ E_x(I \oslash X)P_z^t & 0 & E_x(Z \oslash I)P_x^t \end{bmatrix} \begin{bmatrix} P_z d_z \\ d_y \\ P_x d_x \end{bmatrix} = - \begin{bmatrix} P_z f_d \\ f_p \\ E_x f_c \end{bmatrix}. \tag{4.16}$$

A factorization via Givens rotation is usually twice as expensive as via Householder reflections but the particular structure of the matrix in equation (4.16) suggest that we do not need to start the factorization until the first row corresponding to the operators $\mathcal{Z}$ and $\mathcal{X}$, a substantial saving. Moreover we need not construct the whole operator before beginning the factorization. As described in [29], we can form the operator on-demand, row by row and never have to store more than one row. An added advantage is that we need not store the rotations, only the result of the factorization.

We implemented this procedure and the preliminary results were indistinguishable in term of accuracy from our previous implementation. But the most appealing aspect of this approach is the potential for parallelism. The same sparsity pattern repeats itself at every n rows of the operator because of the Kronecker products. This implies that up to a number of rows could be processed at the same time since their rotations are independent. As a simple example consider the product of 2×2 matrices,

$$K = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad X = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}, \quad K \otimes X = \begin{bmatrix} x_{11} & x_{12} & 0 & 0 \\ x_{21} & x_{22} & 0 & 0 \\ 0 & 0 & 2x_{11} & 2x_{12} \\ 0 & 0 & 2x_{21} & 2x_{22} \end{bmatrix}.$$

It is clear that we could rotate the second and fourth in parallel on different processors. A related but more sophisticated idea was developed and implemented by Chu and George [16, 17] for dense matrices: On p processors, for a matrix with m rows, they allocate m/p consecutive rows to each processors. Independently and in parallel, these horizontal blocks are reduced to triangular form by Givens rotations. During a second phase, where all the interprocessor synchronization cost is incurred, the blocks are further reduced to obtain a triangular matrix. The scheme is meant to reduce both the synchronization cost between processors and the idle time. Such a scheme could easily be specialized to matrices formed by Kronecker products. We have not yet completed such a parallel implementation but it seems that the factorization of operators built from Kronecker products would benefit greatly from their parallel structure.

4.5 Benefits

In summary, we have three different implementations of an algorithm aimed at accurate solutions of semidefinite programs. The first, for small and dense problems, obtains very accurate solutions to all problems in a wide range, from well-conditioned to problems without Slater points. The second implementation, for larger sparse data, decomposes the problem, at each inner iteration, into systems of order of the nonzeros in their corresponding variables. The third implementation, still under development, tries to leverage the structure on Kronecker products on parallel architectures. In all cases, the inner routine is backward stable and all implementations start from possibly infeasible points and are therefore practical. On the negative side, the algorithms are more costly to run than usual symmetric direction algorithms, an unsurprising tradeoff, considering their robustness.

Chapter 5

Sequential Quadratic Programming

Until now we have used classical tools of nonlinear programming to develop and analyse a modern problem in linear optimization. In a reversal of roles, we now attempt to use semidefinite programming as the subproblem solver in a nonlinear optimization toolbox. This should be viewed as an application of semidefinite programming.

A proven approach for the unconstrained minimization of a function $f : \mathbb{R}^n \to \mathbb{R}$ is to build and solve a quadratic model at a local estimate $x^{(k)}$, Newton's method. In this chapter we propose a direct extension of this modeling approach to constrained minimization: A local quadratic model of both the objective function and the constraints is built; since this model is too hard to solve, it is relaxed using the Lagrangean dual, which is then solved by semidefinite programming techniques. The key idea in this approach is to use the latest technique of cone linear programming to obtain a better model than is usual in SQP methods and the key ingredient is the equivalence between the Lagrangean and semidefinite relaxations.

As illustration of how semidefinite programs is used to good effect, recall the well-known Rayleigh-Ritz quotient to obtain the smallest eigenvalue of a symmetric matrix A. An equivalent formulation yields (for example, see [43])

$$\lambda_1(A) = \min\left\{x^t A x \mid x^t x = 1\right\}. \tag{5.1}$$

One approach to prove this result involves Lagrange multipliers: the optimal x must be a stationary point of the Lagrangean $L(x, \lambda) = x^t A x - \lambda(x^t x - 1)$. This shows that the optimal x is an eigenvector; and substitution into the objective function shows that the corresponding eigenvalue is the smallest. But now, consider instead, $x^t A x = \operatorname{trace}(x^t A x) = \operatorname{trace}(A x x^t)$ and let $X := x x^t$. We write the program (5.1) as

$$\min\left\{\langle A, X\rangle \mid \langle I, X\rangle = 1, X \succeq 0, X = x x^t\right\},$$

where $\langle A, B\rangle = \operatorname{trace}(A^t B)$, the trace inner product; $A \succeq 0$ (resp. $A \succ 0$) denotes positive semidefiniteness (resp. positive definiteness); and $A \succeq B$ denotes $A - B \succeq 0$. The symmetric matrix space $\mathbb{S}^n$ is equipped with the Löwner partial order.

Note that the rank one constraint ($X = x x^t$) is redundant because we have only one constraint [64]. We therefore drop it and construct the dual to obtain

$$\max\left\{\lambda \mid \lambda I \preceq A, \lambda \in \mathbb{R}\right\},$$

which obviously has $\lambda_1(A)$ as optimal value. Since the dual has a strictly interior point, the primal attains the same value and we get the Rayleigh-Ritz result. In this manner we use semidefinite programming to construct and solve Lagrangean relaxations.

In this chapter, we wish to illustrate some of the strengths, both theoretical and practical, of considering semidefinite relaxations of quadratic programs as the tool of choice for solving Lagrangean relaxations that arise from quadratic models of general nonlinear programs.

5.1 The Simplest Case

Moving up in complexity we consider the unconstrained problem

$$\min\left\{f(x) \mid x \in \mathbb{R}^n\right\}.$$

When possible, the method of choice for this problem is Newton's method, which minimizes a quadratic model of the objective function. To ensure a solution (or convexity) of the model, Newton's method is often implemented within a Trust-Region, or Restricted-Step approach. This very efficient variation proceeds from an initial estimate of the solution; develops a second-order model of the objective function deemed valid in a region around the estimate; and finally solves the model (the trust-region subproblem)

$$\min\left\{q_0(d) := d^tQd + 2b^td \mid q_1(d) := d^td \le \delta^2, d \in \mathbb{R}^n\right\}. \tag{5.2}$$

The model is constructed from $Q = \nabla^2 f(x^{(k)})$ (or an approximation of the Hessian), $b = \nabla f(x^{(k)})$ and the parameter δ represents the radius where the model is deemed valid. The trust-region may be scaled or even arise from a non-convex quadratic. A solution d is then used as the step to the next estimate $x^{(k+1)} = x^{(k)} + d$.

One of the interesting properties of (5.2), first shown in Stern and Wolkowicz [75] using semi-definite programming, is that even though generally non-convex, the problem exhibits no duality gaps. The Lagrangean dual of (5.2) is written as

$$\max\left\{-b^t(Q+\lambda I)^\dagger b - \lambda\delta^2 \mid Q + \lambda I \succ 0, \lambda \ge 0\right\}, \tag{5.3}$$

a nonlinear, concave semidefinite program, where $(\cdot)^\dagger$ is the Moore-Penrose generalized inverse. In addition, the Lagrangean dual has been shown [69] equivalent to the following linear semidefinite program,

$$\max\left\{(\delta^2+1)\lambda - t \mid \begin{bmatrix} t & b^t \\ b & Q \end{bmatrix} \succeq \lambda I, t \in \mathbb{R}, \lambda \ge 0\right\}. \tag{5.4}$$

We take the dual of the above linear semidefinite program (5.4) to get a semidefinite program equivalent to (5.2),

$$\min \left\{ \langle P_0, Y \rangle \mid \langle E_0, Y \rangle = 1, \langle P_I, Y \rangle \leq \delta^2, Y \succeq 0 \right\}. \tag{5.5}$$

The programs are equivalent in the sense that the optimal values are equal and that the optimal solution to (5.2) can be extracted from the optimal solution to (5.5). The variable in this latter program, Y, belongs to the cone of symmetric positive semidefinite matrices of dimension $(n+1) \times (n+1)$. Also,

$$P_0 = \begin{bmatrix} 0 & b^t \\ b & Q \end{bmatrix}, P_I = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}, E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{0} \end{bmatrix}.$$

The reader will note that (5.5) may be obtained as we did for the for the Rayleigh-Ritz program, by homogenization of (5.2), transformation to matrix space and then by dropping the rank one constraint. (We abuse the term homogenization to mean elimination of the linear terms from a quadratic function.) We will do this in detail for a more general program later on.

This pair of linear primal-dual semidefinite programs (5.5, 5.4) have strict interior points. Therefore the optimal values are equal; moreover, they are attained. Finally, part of the first column of the primal semidefinite solution, the matrix Y, is feasible for (5.2). And, possibly with an additional displacement chosen in the nullspace of the Lagrangean, this first column yields the same objective value for (5.2) as its dual optimal. By this procedure, usually known as *lifting* of (5.2) to the cone of semidefinite matrices and projecting back (by the first column), we see that there are no duality gaps for (5.2). This is made precise in [75].

Theorem 5.1.1 *The optimal solution to (5.2) and to its Lagrangean dual problem (5.3) are attained and the corresponding objective values are equal.*

The interesting aspect of this theorem is that the Lagrangean dual is shown equivalent to a semidefinite program and its optimal value is deduced from this latter program. Therefore, interior-point algorithms, as developed in the previous chapters, may be used to solve (5.2), even if the objective function and the feasible set are non-convex. The result has been extended to upper and lower bounded trust-region subproblems but, interestingly, not to a finite number of constraints. With as few as two constraints, a duality gap may appear [89, 90].

5.2 Multiple Trust-Regions

Consider now a quadratic objective function constrained by multiple quadratics,

$$\min\left\{x^t Q_0 x + 2b_0^t x - a_0 \mid x^t Q_k x + 2b_k^t x - a_k \leq 0, 1 \leq k \leq m, x \in \mathbb{R}^n\right\}. \tag{5.6}$$

As soon as two or more trust-regions are considered, the necessary and sufficient conditions that hold for one trust region may no longer be sufficient for (5.6). This is reflected in the duality gap exhibited by some instances of multiple trust-region programs.

To directly derive the relaxations, we introduce the vector $y = (x_0\ x)^t$. We then require $x_0^2 = 1$ or, in terms of the new variable, $y^t E_0 y = 1$, to get a homogeneous program equivalent to (5.6),

$$\min\left\{y^t P_0 y \mid y^t E_0 y = 1, y^t P_k y \leq 0, 1 \leq k \leq m, y \in \mathbb{R}^{n+1}\right\}, \tag{5.7}$$

where

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{0} \end{bmatrix} \text{ and } P_k = \begin{bmatrix} -a_k & b_k^t \\ b_k & Q_k \end{bmatrix}, 0 \leq k \leq m.$$

The homogenization simplifies the notation and opens the way to the semidefinite relaxation. We rewrite (5.7) using matrix variables,

$$\min\left\{\langle Y, P_0\rangle \mid \langle Y, E_0\rangle = 1, \langle Y, P_k\rangle \leq 0, 1 \leq k \leq m, Y = yy^t\right\}.$$

The rank-one constraint is relaxed to a semidefinite constraint; a procedure that yields the Lagrangean relaxation. After some rearrangement of terms, the Lagrangean dual of (5.7) reads

$$\max\left\{\min\left\{y^t(P_0 + \sum_{k=1}^{m} \lambda_k P_k + \lambda_0 E_0)y - \lambda_0 \mid y \in \mathbb{R}^{n+1}\right\} \mid \lambda \in \mathbb{R} \times \mathbb{R}_+^m\right\}.$$

For the inner minimization to be bounded we must now have

$$P_0 + \sum_{k=1}^{m} \lambda_k P_k + \lambda_0 E_0 \succeq 0, \quad \text{which implies} \quad Q_0 + \sum_{k=1}^{m} \lambda_k Q_k \succeq 0.$$

This, by the way, is where the duality gap arises. The standard necessary optimality conditions for (5.6) do not require the Hessian of the Lagrangean to be semidefinite. But the Lagrangean dual program we are deriving here requires the same Hessian to be semidefinite. We therefore cannot expect the primal variables corresponding to an optimal dual solution to be, in general, optimal for (5.6).

To complete the derivation, we note that the minimum over y will be attained at $y = 0$ from which we get the dual program

$$\max\left\{-\lambda_0 \mid P_0 + \lambda_0 E_0 + \sum_{k=1}^{m} \lambda_k P_k \succeq 0, \lambda \in \mathbb{R} \times \mathbb{R}^m_+\right\}. \tag{5.8}$$

We have now justified the claim of equivalence of the Lagrangean and semidefinite relaxations since dropping the rank-one condition on the homogenized primal (5.2) or taking the semidefinite dual of (5.8) will result in the following, which we will therefore simply refer to as the relaxation of (5.6),

$$\min\left\{\langle P_0, Y\rangle \mid \langle E_0, Y\rangle = 1, \langle P_k, Y\rangle \leq 0, 1 \leq k \leq m, Y \succeq 0\right\}. \tag{5.9}$$

This resulting semidefinite relaxation of (5.7) is equivalent to the one considered in the literature [74, 15, 66].

The optimal value of the relaxation provides a lower bound for (5.6). We now need an approximation for the optimum x. Feasibility properties of the first column of the semidefinite relaxation were first shown by Fujie and Kojima [25] for an equivalent problem with linear objective function. For an alternate view of this result, see [1], from which we extract the next results. Consider the feasible set of the nonlinear program (5.6),

$$\hat{F} := \{x \in \mathbb{R}^n \mid x^t Q_k x + 2b_k^t x - a_k \leq 0, 1 \leq k \leq m\};$$

the feasible set of the semidefinite program (5.9),

$$\tilde{F} := \{Y \succeq 0 \mid \langle E_0, Y\rangle = 1, \langle P_k, Y\rangle \leq 0, 1 \leq k \leq m\};$$

and the projector map,

$$P_R \colon \mathbb{S}^n \to \mathbb{R}^n, \quad P_R(Y) = P_R\left(\begin{bmatrix} a & x^t \\ x & X \end{bmatrix}\right) = x.$$

Theorem 5.2.1 *Suppose that Y is a feasible solution of (5.9). The projected vector, $x = P_R(Y)$, is then feasible for all convex constraints of (5.6).*

Proof: Since we are concerned only with convex constraints, we may consider only those where $Q_k \succeq 0$ and compute

$$\begin{aligned} x^tQ_kx + 2b_k^tx - a_k - \langle P_k, Y\rangle &= x^tQ_kx - \langle Q_k, X\rangle \\ &= -\big\langle Q_k, X - xx^t\big\rangle. \end{aligned}$$

Since $Y \succeq 0$ implies $X - xx^t \succeq 0$, we obtain

$$\begin{aligned} x^tQ_kx + 2b_k^tx - a_k &= \langle P_k, Y\rangle - \big\langle Q_k, X - xx^t\big\rangle \\ &\leq \langle P_k, Y\rangle \\ &\leq 0. \end{aligned}$$

And therefore x is feasible for all convex constraints of (5.6). □

This feasibility of the first column is interesting to consider in more detail. First, in the case of a problem where the quadratic constraints are convex (but maybe the objective is not) there is an obvious way to improve this first column solution when it is not optimal.

An optimal pair Y,λ to the semidefinite relaxation, if Y is not rank one, will in general map to a vector x for which complementarity fails but improving the objective value while remaining feasible is then easy.

Lemma 5.2.2 *Consider an instance of (5.6) with convex constraints. If the semidefinite primal optimal solution* Y *is not rank one, let* $\tilde{x} = P_R(Y)$, *(part of the first column of* Y*). Then there is a* $\bar{x}$ *chosen in* $\mathcal{N}(Q_0 + \sum \lambda_k Q_k)$, *the nullspace of the Lagrangean, such that* $x = \tilde{x} + \bar{x}$, *is feasible and will improve the primal objective value of (5.6).*

The idea is to choose a displacement along the nullspace of the Lagrangean until one or more slack constraints is satisfied with equality. The value of the objective function is lowered since

$$0 = \bar{x}^t(Q_0 + \sum \lambda_k Q_k)\bar{x} \geq \bar{x}^t Q_0 \bar{x}$$

and therefore $(\tilde{x} + \bar{x})^t Q_0 (\tilde{x} + \bar{x}) \leq \tilde{x}^t Q_0 \tilde{x}$. □

Consider now a more general case where the constraints may not be convex. Note that Theorem 5.2.1 implies that the projected first column x is feasible for any nonnegative combination of constraints,

$$\sum_{k=1}^{m} \lambda_k (x^t Q_k x + 2b_k^t x - a_k) \leq 0, \quad \lambda \geq 0, \tag{5.10}$$

which results in a convex function. Thus we obtain feasible points for convex combinations of constraints of (5.6) as in (5.10) from feasible points Y of the relaxation (5.9), even when these are not rank one. Therefore the relaxation provides a convex approximation to the feasible set $\hat{F}$. However, it actually provides a better approximation than this would initially lead us to believe.

Let us define a *valid inequality* for (5.6) as

$$\sum_{k=1}^{m} \lambda_k (x^t Q_k x + 2b_k^t x - a_k) \leq 0, \quad \text{where} \quad Q_0 + \sum_{k=1}^{m} \lambda_k Q_k \succeq 0, \quad \lambda \geq 0. \tag{5.11}$$

These inequalities, an infinite number of them, are not, in general, convex. (Simply consider (5.2) where the objective is strictly convex while the constraint is not.) However, they provide geometric insight into the semidefinite relaxation. The set of vectors satisfying all valid inequalities,

$$\left\{x \;\middle|\; \sum_{k=1}^{m} \lambda_k(x^tQ_kx + 2b_k^tx - a_k) \le 0, \quad Q_0 + \sum_{k=1}^{m} \lambda_k Q_k \succeq 0, \lambda \ge 0\right\}$$

establishes a relation between the set of projected columns of semidefinite solutions and some intersection of the original constraints.

We now use the geometric descriptions sketched above to provide an approximate solution to (5.6) from the optimum of (5.9). We use the first column of the optimum Y but then we use the properties of the valid inequalities (5.11) to improve this column by moving onto a boundary of a valid inequality.

In the general case of a non-convex feasible region, we obtain a step, which, unlike Lemma 5.2.2, *attains* complementary slackness, though not necessarily feasibility. Again, the value of the objective function is improved. This additional step is a generalization of an idea introduced by Moré and Sorensen [61] to solve (5.2) and there is an explicit expression for the step as there is for (5.2), given here in Lemma (5.2.3). We give the technical construction of the step in the following lemma and its value in Corollary 5.2.4.

Lemma 5.2.3 *Suppose that* λ *and* $Y = \begin{bmatrix} 1 & x^t \\ x & X \end{bmatrix}$ *are feasible for (5.9) and its dual. Let*

$$y := \begin{bmatrix} 1 \\ x \end{bmatrix}, \qquad Z := P_0 + \lambda_0 E_0 + \sum_{k \in \mathcal{I}} \lambda_k P_k, \quad \mathcal{I} = \{1, \dots m\}$$

and suppose that they satisfy $ZY = 0$.

Let the matrix Y *be factored as*

$$Y = TT^t,$$

where T *is* $(n+1) \times r$ *and full column rank* $r \ge 2$. *Let the matrix* S *be* $r \times (r-1)$ *and full column rank with* $\mathcal{R}(S) = \mathcal{N}(T_{1,:})$, *i.e. with range space given by the orthogonal complement to the first* *row of* T*. Define*

$$R := TS,\ \bar{P} := \sum_{k\in\mathcal{I}} \lambda_k P_k,\ \ c := y^t \bar{P} y,$$

$$K := R^t \left(\bar{P} y y^t \bar{P} - c\bar{P}\right) R.$$

Choose v *such that*

$$Rv \neq 0 \text{ and } v^t K v \geq 0, \tag{5.12}$$

and define

$$a := v^t R^t \bar{P} R v,\ \ b := 2 v^t R^t \bar{P} y.$$

Then, for z *defined as follows, we have*

$$TSv =: \begin{bmatrix} 0 \\ z \end{bmatrix} \neq 0, \tag{5.13}$$

and

$$b^2 - 4ac \geq 0. \tag{5.14}$$

Moreover, if we define

$$\alpha := \begin{cases} \left(-b \pm \sqrt{b^2 - 4ac}\right)/(2a) & \text{if } a \neq 0 \\ \frac{-c}{b} & \text{if } a = 0, \end{cases}$$

and

$$w := y + \alpha \begin{bmatrix} 0 \\ z \end{bmatrix},$$

then

$$w^t \bar{P} w = 0,\ \ \text{and } Zw = 0. \tag{5.15}$$

Proof: That (5.13) holds and $Zw = 0$ follows directly from construction of R and the assumption of complementary slackness, $ZY = 0$. Note that $ZY = ZTT^t = 0$ implies $ZT = 0$. We still need to show the equality of the quadratic form in (5.15). Now

$$w^t\bar{P}w = y^t\bar{P}y + \alpha 2v^tR^t\bar{P}y + \alpha^2 v^tR^t\bar{P}Rv.$$

(We assume that a w exists to make this quadratic 0. This may be seen from the ordinary trust-region subproblem with the given λ defining the single constraint.) The discriminant for this quadratic in α is defined in (5.14), where

$$b^2 - 4ac = 4v^tKv.$$

Therefore, the discriminant is nonnegative, and the quadratic has a real solution α as given by the standard formula. □

We now make explicit the value of the previous lemma in finding an approximate solution to (5.6).

Corollary 5.2.4 *Suppose that* Y,Z,λ,w *are defined as in Lemma 5.2.3 above. Then the Lagrangean dual bound is attained by* w *as well as complementary slackness.*

Proof: That complementarity is attained is seen directly from the second equation of (5.15). And from both equations we obtain

$$0 = w^tZw - w^t\bar{P}w = w^t(P_0 - \lambda_0 E_0)w.$$

Therefore, $w^tP_0w = q_0(x + \alpha z) = -\lambda_0$, the dual Lagrangean bound. □

5.3 Approximations of Nonlinear Programs

We assume the reader is familiar with *Sequential Quadratic Programming*, denoted *SQP*. We recall only the main features and refer the reader to [76] and [12] for details. The usual justifications for the application of *SQP* to the nonlinear program

$$\min \left\{ f_0(x) \mid f_i(x) = 0, 1 \le i \le m, x \in \mathbb{R}^n \right\},$$

stem from applying Newton's method to obtain stationarity of its Lagrangean $\mathscr{L}(x, \lambda) := f_0(x) + \sum \lambda_i f_i(x)$,

$$\begin{aligned} \nabla f_0(x^*) + \sum \nabla f_i(x^*)\lambda_i^* &= 0, \\ f(x^*) &= 0. \end{aligned}$$

We will sometimes use the notation $f(x) := [f_1(x) \dots f_m(x)]^t$ and for the matrix of all gradients, $[\nabla f_1(x) \dots \nabla f_m(x)]$, we will use $f'(x)$. An iterative attempt at the non-linear system above by Newton's method with some simplification involving $d = x^{(k+1)} - x^{(k)}$ and $\delta_\lambda = \lambda^{(k+1)} - \lambda^{(k)}$, will produce the first-order Newton step,

$$\begin{bmatrix} \nabla^2 \mathscr{L}(x^{(k)}, \lambda^{(k)}) & f'(x^{(k)}) \\ f'(x^{(k)})^t & 0 \end{bmatrix} \begin{bmatrix} d \\ \lambda^{(k+1)} \end{bmatrix} = \begin{bmatrix} -\nabla f_0(x^{(k)}) \\ -f(x^{(k)}) \end{bmatrix}.$$

This system produces a direction d and a new vector of Lagrangean multiplier estimates $\lambda^{(k+1)}$. The key justification for *SQP* is that the system of equations (5.3) may be derived as the first-order necessary conditions of the quadratic program

$$\begin{aligned} \min \quad & f_0(x^{(k)}) + \nabla f_0(x^{(k)})^t d + \tfrac{1}{2} d^t \nabla^2 \mathscr{L}(x^{(k)}, \lambda^{(k)}) d \\ \text{s.t.} \quad & f_i(x^{(k)}) + \nabla f_i(x^{(k)})^t d = 0, \qquad 1 \le i \le m. \end{aligned} \tag{5.16}$$

Stationarity of the Lagrangean of (5.16) yields the first line of (5.3), and feasibility yields the second line. This is why *SQP* is viewed as an extension of Newton's method to constrained optimization.

It is now standard procedure to extend the derivation seen above to the inequality constrained program.

$$\min\left\{f_0(x) \;\middle|\; f_i(x) \leq 0, 1 \leq i \leq m, x \in \mathbb{R}^n\right\}, \tag{5.17}$$

and obtain the subproblem,

$$\begin{aligned} \min \quad & f_0(x^{(k)}) + \nabla f_0(x^{(k)})^t d + \tfrac{1}{2} d^t \nabla^2 \mathscr{L}(x^{(k)}, \lambda^{(k)}) d \\ \text{s.t.} \quad & f_i(x^{(k)}) + \nabla f_i(x^{(k)})^t d \leq 0, \qquad 1 \leq i \leq m, \end{aligned} \tag{5.18}$$

In summary, from the Taylor first-order expansion of $\mathscr{L}(x, \lambda)$, we obtain the standard *SQP* subproblem, which approximates the objective function to second order yet approximates the constraints only to first order. Consider now a second-order Taylor expansion of $\mathscr{L}(x, \lambda)$,

$$\begin{bmatrix} \sum \lambda_i \nabla f_i(x^{(k)}) + \nabla^2 \mathscr{L}(x^{(k)}, \lambda^{(k)}) d + H_3(\delta_x, \delta_\lambda) \\ f'(x^{(k)}) d + \frac{1}{2} d^t f''(x^{(k)}) d \end{bmatrix} = \begin{bmatrix} -\nabla f_0(x^{(k)}) \\ -f(x^{(k)}) \end{bmatrix},$$

where we have grouped the third-order derivatives under the name H_3 because we intend to neglect them. Consider also replacing $\nabla^2 \mathscr{L}(x^{(k)}, \lambda^{(k)})$ by $\nabla^2 \mathscr{L}(x^{(k)}, \lambda^{(k+1)})$. We then obtain an approximation of the necessary optimality conditions which sits between a first and a second-order expansion and is obtained by solving

$$\begin{aligned} \min \quad & f_0(x^{(k)}) + \nabla f_0(x^{(k)})^t d + \tfrac{1}{2} d^t \nabla^2 f_0(x^{(k)}) d \\ \text{s.t.} \quad & f_i(x^{(k)}) + \nabla f_i(x^{(k)})^t d + \tfrac{1}{2} d^t \nabla^2 f_i(x^{(k)}) d \leq 0, \quad 1 \leq i \leq m \\ & d^t d \leq \delta^2, \end{aligned} \tag{5.19}$$

without the additional trust-region, which is added to ensure a bounded solution.

Such a straightforward subproblem has often been considered, but has, just as often, been discarded as unsolvable. One notable exception is an algorithm by Maany [54] developed, interestingly enough, because the standard *SQP* approach failed on the highly nonlinear orbital trajectory problems they were studying [21]. Because (5.19) is a closer approximation to the original problem (5.17) than the quadratic program, we expect it to be a better subproblem to solve in a sequential programming approach and, in fact we have the following,

Lemma 5.3.1 *Assume that* $x^{(k)}$ *is feasible for (5.17). If the (5.19) subproblem is solved by* $d = 0$ *with multipliers* λ*, then the pair of vectors* $x^{(k)}$ *and* λ *satisfies the first-order conditions and second-order conditions of (5.17). Conversely, if* $x^{(k)}$ *and* λ *satisfy the first and second-order necessary conditions of (5.17), then the pair of vectors* $d = 0$*,* λ *satisfy the first and second-order conditions of (5.19).*

This implies that the (5.6) subproblem does better than the (5.16) subproblem since they both solve the first-order conditions but only the former guarantees second-order optimality conditions. This is expected of a trust-region approach.

Is also does better by providing second-order multiplier estimates in the sense that the multipliers $\lambda^{(k+1)}$ obtained from (5.19) satisfy

$$\min\left\{\|\nabla f_0(x^{(k)}) + \nabla^2 \mathscr{L}(x^{(k)}, \eta)d + \sum \eta_i \nabla f_i(x^{(k)})\|_2^2 \mid \eta \in \mathbb{R}^m\right\}.$$

If we are close to the solution we therefore obtain, directly from the solution of the subproblem, not only a good search direction in primal space, but better multiplier estimates than provided by the standard (5.16) subproblem. (For more details on second-order multiplier estimates, see [30].)

5.4 Quadratically Constrained Programming

Note that, for simplicity, we assume that our constraints are nonlinear. Linear constraints have to be treated differently, essentially squared [66]. Equivalently, linear constraints may be eliminated or mapped to a linear constraint in matrix space.

Homogenization of (5.19), obtained by adding a component d_0 to the vector d, together with the constraint $d_0^2 = 1$, yield the semidefinite relaxation,

$$\min\left\{\langle P_0, Y\rangle \mid \langle E_0, Y\rangle = 1, \langle P_i, Y\rangle \leq 0, 1 \leq i \leq m, \langle P_I, Y\rangle \leq \delta^2\right\}, \tag{5.20}$$

where

$$P_i = \begin{bmatrix} -a_i & \nabla f_i(x^{(k)})^t & 0 \\ \nabla f_i(x^{(k)}) & \nabla^2 f_i(x^{(k)}) & 0 \\ 0 & 0 & 0 \end{bmatrix},\ a_i = -2f_i(x^{(k)}),\quad 0 \le i \le m,$$

and where E_0 and P_I have their usual definitions,

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, P_I = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix},$$

and $Y \succeq 0$.

But this relaxation may be infeasible if the current estimate is too far from the feasible region. To overcome this difficulty in *SQP*, Vardi [84] suggested a heuristic shift of the linear constraints. We do a related shift of our second-order constraints by allowing the additional component d_0 to take values between zero and one. That is, we change $d_0^2 = 1$ to $d_0^2 \le 1$. This additional relaxation allows for a feasible subproblem. Of course we would want d_0 to be as close to 1 as possible and examination of the subproblem shows that it automatically tries to make d_0 "large". We need no heuristic to choose a Vardi-type parameter.

The dual program to (5.20) is then

$$\max\left\{-\lambda_0 \mid P_0 + \lambda_0 E_0 + \sum_{i=1}^{m} \lambda_i P_i + \lambda_I P_I \succeq 0, \lambda \in \mathbb{R} \times \mathbb{R}_+^m\right\}. \tag{5.21}$$

Solving the primal-dual pair (5.20),(5.21), in the case of gap-free (5.17), is enough since, as we have seen, the first column is optimal for the quadratic approximation. But, in general, we need an appropriate merit function to ensure sufficient decrease at each step and guarantee global convergence of the algorithm, whether we use a line search or a trust-region strategy.

After solving the (5.6) subproblem for a direction $d \neq 0$, the next iterate is obtained by $x^{(k+1)} = x^{(k)} + d$. This new point serves for the expansion of a new problem by second-order polynomials and we iterate until the subproblem yields $d = 0$. As with any trust-region based algorithm, we adjust the trust-region radius according to the ratio of predicted improvement to actual improvement. At the end, we have a solution satisfying both first and second-order conditions of (5.17). Somewhat more formally, Algorithm 5.4.1 describes the approach.

Algorithm 5.4.1 Sequential Quadratically Constrained Programming

Given $f_i, \nabla f_i, \nabla^2 f_i, x^{(0)}$ {Functions and derivatives}
Given ϵ {Steplength tolerance}
$k := 0$ {Iteration count}
repeat
  $Y \in \text{argmin}\{\langle P_0, Y\rangle : \langle P_i, Y\rangle \leq 0, \langle E_0, Y\rangle = 1, Y \succeq 0\}$; {Solve semidefinite pair}
  $\lambda^{(k+1)} \in \text{argmax}\{-\lambda_0 : P_0 + \sum \lambda P_i + \lambda_0 E_0 \succeq 0, \lambda \in \mathbb{R} \times \mathbb{R}^m_+\}$;
  $d := P_R(Y)$; {Project down by first column}
  $x^{(k+1)} := x^{(k)} + d$; {New point}
  $r^k := \frac{\varphi(x^{(k)}) - \varphi(x^{(k+1)})}{q_0(x^{(k)}) - q_0(x^{(k+1)})}$ {Decrease ratio}
  if $(r^k < \frac{1}{4})$ **then**
    $\delta = \delta/4$ {Bad model, shrink trust-region}
  else
    if $(r^k > \frac{3}{4})$ and $\|x^{(k+1)} - x^{(k)}\| = \delta$ **then**
      $\delta = 2\delta$ {Good model, expand trust-region}
    end if
  end if
  $k := k + 1$ {Bump iteration}
until $(\|d\| \leq \epsilon)$ {Attained optimality}
Find maximal $d \in \mathcal{N}(\nabla^2 \mathscr{L})$ such that $f(x^{(k)} + d) \leq 0$
$x^{(k)} := x^{(k)} + d$; {Nullspace move}
return$(x^{(k)}, \lambda^{(k)})$ {Primal iterate and multiplier}

If the (5.19) subproblem is convex, or more generally, if it is an instance without duality gaps, then solving the semidefinite relaxation, which may be done efficiently, will be enough since the primal semidefinite solution will be rank one. We will have a pair of primal-dual vectors satisfying the sufficient conditions for optimality of (5.6).

This takes care of the convex case and of many non-convex cases. In other cases, we move along the nullspace of the Lagrangean until we hit one of the constraints. This nullspace-restricted step improves the objective value even if it does not lead to an optimal solution.

Example 5.4.1 *Illustrative comparison of SQP and SQ^{2}P.*

$$\min\left\{-x_1 - x_2 \mid x_1^3 - x_2 \le 0,\ x_1^3 + x_2^2 - 1 \le 0\right\}$$

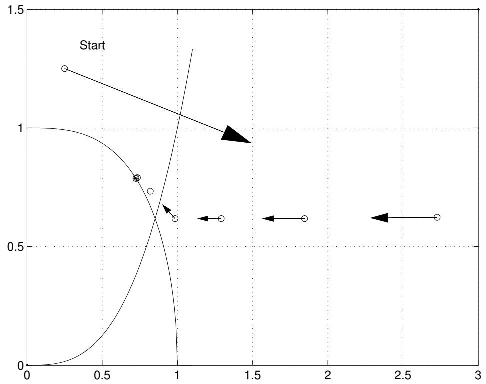


Figure 5.1: Iterations of SQP on Example 5.4.1, from initial point $(\frac{1}{4}\ \frac{5}{4})^t$. *As the first iteration demonstrates, the direction given by the QP subproblem can be poor.*

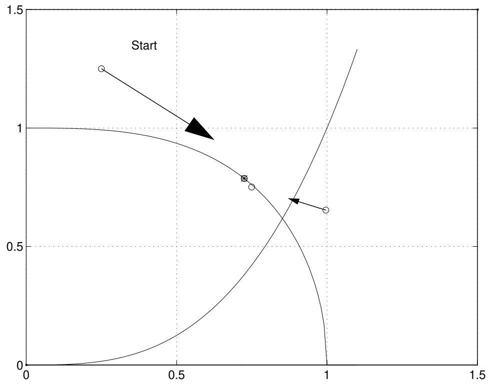


Figure 5.2: Iterations of SQ^{2}P on the same example. The horizontal scale is changed to highlight the value of the direction provided by the semidefinite subproblem.

5.5 Conclusion

Efficient approaches to unconstrained optimization based on Newton's method all involve local quadratic models of the objective function. Yet for constrained optimization, the extension of Newton's method, *SQP*, uses linear approximations. Some second-order information is included in the model, but in an aggregate form.

In this chapter we have outlined an approach that deals more closely with the true quadratic model of the problem at hand. One of the key features is the relationship between the Lagrangean and Semidefinite relaxations which leads to what we have called the *SQ^{2}P* algorithm for general nonlinear programs.

This algorithm builds second-order approximations of both the objective function and the constraints and then solves the Lagrangean relaxation of this quadratic model via semidefinite programming. The approach provides a stronger relaxation than the standard quadratic program used in *SQP* methods; at every step it provides better multiplier estimates; it handles potential infeasibility of the subproblem in a straight-forward manner; finally, it aims at solutions satisfying both first and second-order optimality conditions. Many implementation issues still need to be resolved but the recent advances in numerical solutions of large semidefinite programs encourage further study.

As a final note, we should make clear that it may turn out that the semidefinite relaxation is not exactly the right one to use for efficiency reasons. But the point remains that we are nowadays in a position to do better than the linear relaxations so popular during the seventies and eighties. Because we had at our disposals good linear programming solvers, the world seemed linear or, at least, mathematical models tried to make it so. We now have good solvers for quadratic programs, either via semidefinite relaxations or, possibly via some second-order cone relaxation, and we should make full use of these new tools.

Chapter 6

Future Directions of the Gauss-Newton Direction

Starting from the optimality conditions of the log barrier problem associated with the standard linear semidefinite program, we used the classical tool box of applied mathematicians and numerical analysts to obtain a family of search directions and interior-point algorithms. This approach led us to consider the least-squares solution of the linearized and smoothed optimality conditions. We obtained a solution, which we call the Gauss-Newton search direction, that is different from search directions previously considered.

The Gauss-Newton direction was shown to be well-defined and to guarantee descent of a valid merit function; computed on the prototypical cases of a standard linear program and on the central path of a semidefinite program, it coincides with the major primal-dual search directions; under usual assumptions, the defining system of equations is full rank and is at least as far from singularity as the best practical directions known until now; finally it is invariant under both affine transformation of the space and orthogonal transformation of the underlying cone.

From this direction we have exhibited an accurate algorithm for solving semidefinite programs with an implementation aimed at small dense problems and another aimed at large sparse problems.

Our approach might be construed as a step backward since the subproblem we are solving at every step is larger than the subproblem solved by all other algorithms. Yet the weaknesses of current implementations based on other directions, their failure to accurately solve even well-conditioned problems to machine precision, for example, should convince the reader that the last word has not been written from a practitioner's point of view and that our approach is valid even if more costly in its current implementation.

The main objective of our work was to develop a robust algorithm, one that would reliably and accurately solve any problem in a wide family. Along the way, we tried with limited success to develop a proof of polynomial convergence of the algorithm. Within our scope, the experimental results and the bounds of Chapter 4 should convince the reader that we have attained our objective. We restricted ourselves to feasible problems and therefore one obvious avenue of research is to explore the behavior of the direction on infeasible problems.

At the onset of our research, after the introduction of the Gauss-Newton direction, we believed that the next interesting question was to apply the scalings of the Monteiro-Zhang family to the the iterates and explore the characteristics of the resulting scaled Gauss-Newton directions. This might still be interesting and was done for the Nesterov-Todd scaling [47] but much more interesting questions arise when we consider that all the Monteiro-Zhang family can be obtained by applying projections to the Gauss-Newton system.

For a simple example, as we described i equation (2.31), the AHO system can be obtained by projecting away the skew-symmetric part of the Gauss-Newton Jacobian and the corresponding right-hand side,

$$\mathsf{J}_{aho} = \mathsf{P}\mathsf{J}_{gn}, \quad \mathsf{f}_{aho} = \mathsf{P}\mathsf{f}_{gn}, \qquad \text{where} \quad \mathsf{P} = [\mathrm{I} \ \ 0]\,,$$

and where the identity and the zero are of appropriate dimension. The other directions in the family trade the identity for another matrix that depends on the current iterate.

Because this operator is applied before solving $\mathsf{J}\mathsf{d} = -\mathsf{f}$, in a mind attuned to the needs of numerical linear algebra, it is suggestive of a pre-conditioner, but a pre-conditioner chosen for all the wrong reasons, namely to create a square system, oblivious to the increase in the condition number. Are we not emulating the sixties where every least-squares problem was transformed via the normal equation to a square system?

If we view the relation between the Gauss-Newton and every other direction as the result of some pre-conditioning, the interesting questions obviously become: What are the right pre-conditioners for a given family of problems? How can we obtain more accuracy, simpler systems, sparser systems? Should we not apply an iterative scheme until we get close to the optimal solution? For theoretical reasons, we also might be interested in finding justifications for the pre-conditioners that yield the AHO or NT directions. It might even be possible to find some optimization problems for which those pre-conditioners are the solutions.

Also on the computational side, we intend to re-implement our algorithm using the latest parallel numerical linear algebra techniques. Since the principal operator is built from Kronecker products, an inherently parallel structure, we expect major gains from this area of research. On a more prosaic note, we intend to develop interfaces to the major databases of problems, transforming a research project into a useful tool for the optimization community.

Bibliography

[1] A. Alfakih, S. Kruk, and H. Wolkowicz. A note on geometry of semidefinite relaxations. Technical Report in progress, University of Waterloo, Waterloo, Canada, 1999.

[2] Farid Alizadeh. *Combinatorial optimization with interior point methods and semidefinite matrices*. PhD thesis, University of Minnesota, 1991.

[3] Farid Alizadeh. Optimization over positive semi-definite cone; interior-point methods and combinatorial applications. In P.M. Pardalos, editor, *Advances in Optimization and Parallel Computing*, pages 1–25. North–Holland, 1992.

[4] Farid Alizadeh, Jean-Pierre A. Haeberly, and Michael L. Overton. A new primal-dual interior-point method for semidefinite programming. In J.G. Lewis, editor, *Proceedings of the Fifth SIAM Conference on Applied Linear Algebra*, pages 113–117. SIAM, 1994.

[5] Farid Alizadeh, Jean-Pierre A. Haeberly, and Michael L. Overton. Complementarity and nondegeneracy in semidefinite programming. *Math. Programming*, 77(2, Ser. B):111–128, 1997. Semidefinite programming.

[6] Farid Alizadeh, Jean-Pierre A. Haeberly, and Michael L. Overton. Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. *SIAM J. Optim.*, 8(3):746–768 (electronic), 1998.

[7] Farid Alizadeh and Stefan Schmieta. *Handbook of Semidefinite Programming: theory, algorithms and applications*, chapter 8 (Symmetric Cones, Potential Reduction Methods and

Word-by-Word Extensions). Number 27 in International series in operations research & management science. Kluwer Academic Publishers, 101 Phillip Drive, Assinippe Park, Norwell MA 02061, 2000.

[8] Miguel Anjos and Henry Wolkowicz. A tight semidefinite relaxation of the cut polytope. Technical Report Research Report, CORR 00-19, University of Waterloo, Waterloo, Ontario, 2000. 24 pages.

[9] Richard Bellman and Ky Fan. On systems of linear inequalities in Hermitian matrix variables. In *Proc. Sympos. Pure Math., Vol. VII*, pages 1–11. Amer. Math. Soc., Providence, R.I., 1963.

[10] Adi Ben-Israel and Thomas N. E. Greville. *Generalized inverses: theory and applications.* Robert E. Krieger Publishing Co. Inc., Huntington, N.Y., 1980. Corrected reprint of the 1974 original.

[11] Åke Björck. *Numerical methods for least squares problems.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[12] Paul T. Boggs and Jon W. Tolle. Sequential quadratic programming. In *Acta numerica, 1995*, pages 1–51. Cambridge Univ. Press, Cambridge, 1995.

[13] Brian Borchers. Csdp, 2.3 user's guide. *Optimization Methods and Software*, 11(1):597–611, 1999.

[14] Brian Borchers. Csdp, a c library for semidefinite programming. *Optimization Methods and Software*, 11(1):514–623, 1999.

[15] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[16] Eleanor Chu and Alan George. QR factorization of a dense matrix on a shared-memory multiprocessor. *Parallel Comput.*, 11(1):55–71, 1989.

[17] Eleanor Chu and Alan George. QR factorization of a dense matrix on a hypercube multiprocessor. *SIAM J. Sci. Statist. Comput.*, 11(5):990–1029, 1990.

[18] E. de Klerk, C. Roos, and T. Terlaky. Initialization in semidefinite programming via a self-dual skew-symmetric embedding. *Oper. Res. Lett.*, 20(5):213–221, 1997.

[19] James W. Demmel. *Applied numerical linear algebra.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[20] J. E. Dennis, Jr. and Robert B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. Corrected reprint of the 1983 original.

[21] L.C.W. Dixon, S.E. Hersom, and Z.A. Maany. Initial experience obtained solving the low thrust satellite trajectory optimisation problem. Technical Report T.R. 152, The Hatfield Polytechnic Numerical Optimization Centre, 1984.

[22] D. K. Faddeev, V. N. Kublanovskaya, and V. N. Faddeeva. Sur les systèmes linéaires algébriques de matrices rectangulaires et mal-conditionnées. In *Programmation en Mathématiques Numériques (Actes Colloq. Internat. C.N.R.S. No. 165, Besançon, 1966)*, pages 161–170. Éditions Centre Nat. Recherche Sci., Paris, 1968.

[23] Anthony V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming.* Academic Press Inc., Orlando, Fla., 1983.

[24] Anthony V. Fiacco and Garth P. McCormick. *Nonlinear programming.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 1990. Sequential unconstrained minimization techniques.

[25] Tetsuya Fujie and Masakazu Kojima. Semidefinite programming relaxation for nonconvex quadratic programs. *J. Global Optim.*, 10(4):367–380, 1997.

[26] Katsuki Fujisawa. The software of the primal-dual interior-point method for semidefinite pro-

gramming SDPA (semidefinite programming algorithm). *Systems Control Inform.*, 44(2):51–58, 2000.

[27] Katsuki Fujisawa, Mituhiro Fukuda, Masakazu Kojima, and Kazuhide Nakata. Numerical evaluation of SDPA (semidefinite programming algorithm). In *High performance optimization*, pages 267–301. Kluwer Acad. Publ., Dordrecht, 2000.

[28] Katsuki Fujisawa, Masakazu Kojima, and Kazuhide Nakata. The interior-point method software SDPA (semidefinite programming algorithm) for semidefinite programming problems. *Sūrikaisekikenkyūsho Kōkyūroku*, (1114):149–159, 1999. Continuous and discrete mathematics for optimization (Kyoto, 1999).

[29] Alan George and Michael T. Heath. Solution of sparse linear least squares problems using Givens rotations. *Linear Algebra Appl.*, 34:69–83, 1980.

[30] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical optimization*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1981.

[31] Michel Goemans and Franz Rendl. *Handbook of Semidefinite Programming: theory, algorithms and applications*, chapter 12 (Combinatorial optimization). Number 27 in International series in operations research & management science. Kluwer Academic Publishers, 101 Phillip Drive, Assinippe Park, Norwell MA 02061, 2000.

[32] M.X. Goemans and D.P. Williamson. .878-approximation algorithms for MAX CUT and MAX 2SAT. In *ACM Symposium on Theory of Computing (STOC)*, 1994.

[33] G. H. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM J. Numer. Anal.*, 10:413–432, 1973. Collection of articles dedicated to the memory of George E. Forsythe.

[34] G. H. Golub and V. Pereyra. Differentiation of pseudoinverses, separable nonlinear least square problems and other tales. pages 303–324. Publ. Math. Res. Center Univ. Wisconsin, No. 32, 1976.

[35] Gene H. Golub. Numerical methods for solving linear least squares problems. *Apl. Mat.*, 10:213–216, 1965.

[36] Alexander Graham. *Kronecker products and matrix calculus: with applications.* Ellis Horwood Ltd., Chichester, 1981. Ellis Horwood Series in Mathematics and its Applications.

[37] G. Gruber and F. Rendl. Semidefinite programs whithout feasible interior points. Technical report, University of Klagenfurt-Research Group Operations Research, 1999.

[38] Ming Gu. Primal-dual interior-point methods for semidefinite programming in finite precision. *SIAM J. Optim.*, 10(2):462–502 (electronic), 2000.

[39] Osman Güler. Barrier functions in interior point methods. *Math. Oper. Res.*, 21(4):860–885, 1996.

[40] Jean-Pierre A. Haeberly, Madhu V. Nayakkankuppam, and Michael L. Overton. Mixed semidefinite-quadratic-linear programs. In *Advances in linear matrix inequality methods in control*, pages xviii, 41–55. SIAM, Philadelphia, PA, 2000.

[41] Christoph Helmberg, Franz Rendl, Robert J. Vanderbei, and Henry Wolkowicz. An interior-point method for semidefinite programming. *SIAM J. Optim.*, 6(2):342–361, 1996.

[42] Nicholas J. Higham. *Accuracy and stability of numerical algorithms.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[43] Roger A. Horn and Charles R. Johnson. *Matrix analysis.* Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original.

[44] Roger A. Horn and Charles R. Johnson. *Topics in matrix analysis.* Cambridge University Press, Cambridge, 1994. Corrected reprint of the 1991 original.

[45] L. V. Kantorovich and G. P. Akilov. *Functional analysis.* Pergamon Press, Oxford, second edition, 1982. Translated from the Russian by Howard L. Silcock.

[46] C. T. Kelley. *Iterative methods for optimization.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[47] E. De Klerk, J. Peng, C. Roos, and T. Terlaky. A scaled Gauss-Newton primal-dual search direction for semidefinite programming. Technical report, Delft University of Technology, Faculty of Technical Mathematics and Informatics, Delft, The Netherlands, 1999.

[48] Masakazu Kojima, Susumu Shindoh, and Shinji Hara. Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices. *SIAM J. Optim.*, 7(1):86–125, 1997.

[49] S. Kruk, M. Muramatsu, F. Rendl, R.J. Vanderbei, and H. Wolkowicz. The Gauss-Newton direction in linear and semidefinite programming. Technical Report CORR 98-16, University of Waterloo, Waterloo, Canada, 1998. To appear in Optimization Methods and Software.

[50] Charles L. Lawson and Richard J. Hanson. *Solving least squares problems.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. Revised reprint of the 1974 original.

[51] A. S. Lewis. Derivatives of spectral functions. *Math. Oper. Res.*, 21(3):576–588, 1996.

[52] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1(2):166–190, 1991.

[53] László Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25(1):1–7, 1979.

[54] Z.A. Maany. A new algorithm for highly curved constrained optimization. Technical Report T.R. 161, The Hatfield Polytechnic Numerical Optimization Centre, 1985.

[55] Jan R. Magnus and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics.* John Wiley & Sons Ltd., Chichester, 1999. Revised reprint of the 1988 original.

[56] Renato D. C. Monteiro. Primal-dual path-following algorithms for semidefinite programming. *SIAM J. Optim.*, 7(3):663–678, 1997.

[57] Renato D. C. Monteiro and M. J. Todd. *Handbook of Semidefinite Programming: theory, algorithms and applications*, chapter 10 (Path-Following Methods). Number 27 in International series in operations research & management science. Kluwer Academic Publishers, 101 Phillip Drive, Assinippe Park, Norwell MA 02061, 2000.

[58] Renato D. C. Monteiro and Takashi Tsuchiya. Polynomial convergence of a new family of primal-dual algorithms for semidefinite programming. *SIAM J. Optim.*, 9(3):551–577 (electronic), 1999.

[59] Renato D. C. Monteiro and Paulo R. Zanjácomo. A note on the existence of the Alizadeh-Haeberly-Overton direction for semidefinite programming. *Math. Programming*, 78(3, Ser. A):393–396, 1997.

[60] Renato D. C. Monteiro and Yin Zhang. A unified analysis for a class of long-step primal-dual path-following interior-point algorithms for semidefinite programming. *Math. Programming*, 81(3, Ser. A):281–299, 1998.

[61] Jorge J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM J. Sci. Statist. Comput.*, 4(3):553–572, 1983.

[62] Yu. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Res.*, 22(1):1–42, 1997.

[63] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[64] Gábor Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Math. Oper. Res.*, 23(2):339–358, 1998.

[65] Gábor Pataki and Levent Tunçel. On the generic properties of convex optimization problems in conic form. Technical Report CORR-97-16, University of Waterloo, 1997. To appear in Mathematical Programming A.

[66] S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for $(0,1)$-quadratic programming. *J. Global Optim.*, 7(1):51–73, 1995.

[67] F.A. Potra and R. Sheng. A superlinearly convergent primal-dual infeasible-interior-point algorithm for semidefinite programming. Technical Report Reports on Computational Mathematics, 78, University of Iowa, Iowa City, IA, 1995.

[68] Florian A. Potra and Rongqin Sheng. On homogeneous interior-point algorithms for semidefinite programming. *Optim. Methods Softw.*, 9(1-3):161–184, 1998.

[69] Franz Rendl and Henry Wolkowicz. A semidefinite framework for trust region subproblems with applications to large scale minimization. *Math. Programming*, 77(2, Ser. B):273–299, 1997. Semidefinite programming.

[70] R. Tyrrell Rockafellar. *Conjugate duality and optimization*. Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1974. Lectures given at the Johns Hopkins University, Baltimore, Md., June, 1973, Conference Board of the Mathematical Sciences Regional Conference Series in Applied Mathematics, No. 16.

[71] R. Tyrrell Rockafellar. *Convex analysis*. Princeton University Press, Princeton, NJ, 1997. Reprint of the 1970 original, Princeton Paperbacks.

[72] H. S. Sendov and A. S. Lewis. Twice differentiable spectral functions. Technical report, University of Waterloo, 02 2000. To appear in SIAM J. Matrix Analysis.

[73] Masayuki Shida, Susumu Shindoh, and Masakazu Kojima. Existence and uniqueness of search directions in interior-point algorithms for the SDP and the monotone SDLCP. *SIAM J. Optim.*, 8(2):387–396 (electronic), 1998.

[74] N. Z. Shor. Quadratic optimization problems. *Izv. Akad. Nauk SSSR Tekhn. Kibernet.*, 1987(1):128–139, 222, 1987.

[75] Ronald J. Stern and Henry Wolkowicz. Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM J. Optim.*, 5(2):286–313, 1995.

[76] J. Stoer. Principles of sequential quadratic programming methods for solving nonlinear programs. In *Computational mathematical programming (Bad Windsheim, 1984)*, pages 165–207. Springer, Berlin, 1985.

[77] Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11/12(1-4):625–653, 1999. Interior point methods.

[78] M. J. Todd, K. C. Toh, and R. H. Tütüncü. On the Nesterov-Todd direction in semidefinite programming. *SIAM J. Optim.*, 8(3):769–796 (electronic), 1998.

[79] M.J. Todd. A study of search directions in primal-dual interior-point methods for semidefinite programming. *Optim. Methods Softw*, 11&12:1–46, 1999.

[80] K. C. Toh, M. J. Todd, and R. H. Tütüncü. SDPT3—a MATLAB software package for semidefinite programming, version 1.3. *Optim. Methods Softw.*, 11/12(1-4):545–581, 1999. Interior point methods.

[81] K.C. Toh. Search directions for primal-dual interior-point methods in semidefinite programming. Technical report, Dept. of Mathematics, National University of Singapore, 1998.

[82] Levent Tunçel. *Handbook of Semidefinite Programming: theory, algorithms and applications*, chapter 9 (Potential Reduction and Primal-Dual Methods). Number 27 in International series in operations research & management science. Kluwer Academic Publishers, 101 Phillip Drive, Assinippe Park, Norwell MA 02061, 2000.

[83] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.

[84] Avi Vardi. A trust region algorithm for equality constrained minimization: convergence properties and implementation. *SIAM J. Numer. Anal.*, 22(3):575–591, 1985.

[85] Henry Wolkowicz, Romesh Saigal, and Lieven. Vandenberghe, editors. *Handbook of Semidefinite Programming: theory, algorithms and applications.* Number 27 in International series

in operations research & management science. Kluwer Academic Publishers, 101 Phillip Drive, Assinippe Park, Norwell MA 02061, 2000.

[86] Arthur Wouk. *A course of applied functional analysis.* Wiley-Interscience [John Wiley & Sons], New York, 1979. Pure and Applied Mathematics.

[87] V.A. Yakubovich. The solution of certain matrix inequalities in automatic control theory. *Soviet Math. Dokl.*, 3:620–623, 1962. In Russian, 1961.

[88] V.A. Yakubovich. Solution of certain matrix inequalities encountered in nonlinear control theory. *Soviet Math. Dokl.*, 5:652–656, 1964.

[89] Y. Yuan. On a subproblem of trust region algorithms for constrained optimization. *Math. Programming*, 47(1 (Ser. A)):53–63, 1990.

[90] Ya Xiang Yuan. A dual algorithm for minimizing a quadratic function with two quadratic constraints. *J. Comput. Math.*, 9(4):348–359, 1991.