ALGORITHMS FOR UNSYMMETRIC CONE OPTIMIZATION AND AN IMPLEMENTATION FOR PROBLEMS WITH THE EXPONENTIAL CONE

A DISSERTATION
SUBMITTED TO THE INSTITUTE FOR
COMPUTATIONAL AND MATHEMATICAL
ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Santiago Akle Serrano
March 2015

This dissertation is online at: http://purl.stanford.edu/sn367tt9726

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Michael Saunders, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Yinyu Ye, Co-Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Margot Gerritsen

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Symmetric cone optimization subsumes linear optimization, second-order cone optimization, and semidefinite optimization. It is of interest to extend the algorithmic developments of symmetric cone optimization into the realm of unsymmetric cones. We analyze the theoretical properties of some algorithms for unsymmetric cone problems. We show that they achieve excellent worst-case iteration bounds while not necessarily being practical to implement.

Using lessons from this analysis and inspired by the Mehrotra predictor-corrector algorithm, we extend the homogeneous implementation ECOS to handle problems modeled with Cartesian products of the positive orthant, second-order cones, and the exponential cone, and we empirically validate its efficiency.

Acknowledgements

To my family, my friends, and my teachers.

I thank Michael Saunders for his patience, encouragement and mentorship. Yinyu Ye for his guidance and for introducing me to this area of research. Margot Gerritsen for being a great inspiration and source of moral support for all ICME students.

I'm very grateful to my parents Ana and Luis for lovingly encouraging me to follow my desire for scientific enquiry, and to Sebastian for being an ally in this exploration. To Therese for her love and companionship, for all the support, encouragement, and patience that made this work possible.

I also want to thank my friends in ICME, my good friends in Mexico, Victor, my teachers at ITAM, and the people of Mexico for the CONACYT funds.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Whenever we use a convex optimization algorithm, we wish to be certain of the computational cost it will incur, the number of iterations it will take, the precision it will achieve, and how these metrics will change as it solves larger problems. The study of polynomial-time interior-point algorithms for convex optimization has yielded some answers to these questions. For example, it is known that when the convex minimization algorithm is written in the form

$$\text{minimize } c^T x \text{ subject to } Ax = b, \quad x \in \mathcal{X},$$

where $x \in \mathbb{R}^n$ is the decision variable, c forms a linear objective, $A \in \mathbb{R}^{m \times n}$ has *full row rank*, $\mathcal{X}$ is a convex subset of $\mathbb{R}^n$ with non-empty interior, and when there exists a cheaply computable strongly nondegenerate self-concordant barrier f for the set $\mathcal{X}$, then an algorithm exists with a guaranteed polynomial bound on the number of iterations [43, 55, 48]. Defining an appropriate barrier function for an arbitrary set $\mathcal{X}$ is in practice very difficult. However, when $\mathcal{X}$ is a Cartesian product formed from certain *proper cones*, such a barrier is known and particularly good algorithms can be defined. These algorithms are the object of study of interior-point polynomial-time conic programming.

A cone $\mathcal{K}$ is a set closed under conic combinations (for any $x_1, \ldots x_n \in \mathcal{K}$, and any $0 \leq \alpha_1, \ldots, \alpha_n$ we have that $\sum \alpha_i x_i \in \mathcal{K}$). Proper cones are topologically closed sets that do not contain straight lines and have non-empty interiors. The set of vectors normal to the supporting hyperplanes of a proper cone $\mathcal{K}$ forms a second proper cone called the dual cone $\mathcal{K}^\star$. The existence of the dual cone implies a classification of cones into those that are identical to their dual, called self-dual cones, and those that are not equal to their dual, called non-self-dual. A second classification of cones divides them into homogeneous and non-homogeneous cones. The set of homogeneous and self-dual cones is the set of symmetric cones. It is known that all symmetric cones are Cartesian products of five basic symmetric cones [24].

The dual of a conic problem is again a conic problem with the dual cone as conic constraint. Symmetric conic programs are those for which the cone is symmetric. These include linear programming (where the cone is the positive

orthant), second-order cone programming (where the cone is the Lorentz cone), and semidefinite programming (where the cone is the set of positive semidefinite matrices). Symmetric cone problems can also be defined from Cartesian products of these previously listed cones.

The homogeneous self-dual embeddings from linear programming, which solve the primal and dual problem simultaneously and can detect infeasible and unbounded problems, also generalize to conic programming. However, joining the primal and dual problems into one will double the number of variables. Since each iteration of an interior-point method solves a linear system of size proportional to the number of variables, doubling them can increase the computational cost by up to a factor of eight. However, techniques exist to define search directions from smaller more manageable systems. As it turns out, when the cones are symmetric, a type of self-concordant barrier called self-scaled exists. And for any ordered pair of points $x, s \in \mathcal{K}$ in the symmetric cone (endowed with a self-scaled barrier) a Nesterov-Todd scaling point can be defined. These Nesterov-Todd points are used to define smaller Newton-like systems and cheaper search directions [44].

Whenever the cones are not symmetric, Nesterov-Todd scaling points do not exist and other strategies must be used [42, 40]. Yinu Ye and Anders Skajaa [51] suggest using an alternative set of search directions, defined by the solution of a small system, that do not use the Nesterov-Todd points. In that work the authors analyze the behavior of these directions for the simplified homogeneous embedding and define a predictor-corrector algorithm that has good theoretical guarantees as well as good empirical behavior. In this work we study these directions further and show that they can be used to define path-following algorithms and also to define an algorithm akin to a potential reduction algorithm, both for the full homogeneous embedding. These alternative directions are guaranteed to work only when the iterates are in a small region of the feasible set defined in terms of a measure of distance to the central path. We show that a predictor-corrector algorithm with the same theoretical guarantees as the one defined by Ye and Skajaa can be defined for a different measure of distance to the central path for the simplified homogeneous embedding.

It is of particular interest to develop a conic optimization code that can solve problems modeled from Cartesian products of the positive orthant, second-order cones, and the exponential cone. The first two of these are symmetric cones while the exponential cone is unsymmetric. For the symmetric cones, exceptionally good barriers are known. For the exponential cone a barrier that satisfies all requirements is defined in the work of Chares [12].

For conic programming problems one can define a merit function (also called a potential function) that diverges toward ∞ as the iterates approach a suboptimal boundary of the feasible set, and diverges to $-\infty$ as the iterates approach the solution of the problem. This merit function is used for the theoretical analysis of the complexity, and in the implementation of the algorithms as a way to measure progress and select step-lengths. However, this merit function is defined in terms of a *conjugate pair* of barriers for the primal and dual cones. As it turns out, given a barrier for the primal cone, it might not be simple to

define a computable conjugate barrier for the dual cone. In this work we define the conjugate pair for the Chares [12] barrier of the exponential cone and show that it is cheap to compute and so are its gradient and Hessian.

The Mehrotra predictor-corrector algorithm is one of the most successful primal-dual interior-point methods for symmetric cones. This algorithm uses linear combinations of predictor directions (tangent to the central path) and corrector directions (toward the central path) while dynamically defining the coefficients of the linear combination. In this work we extend the implementation of an open-source implementation of the Mehrotra predictor-corrector algorithm called ECOS [16] to support problems modeled with Cartesian products of the positive orthant, the Lorentz cone, and the exponential cone. Even though the heuristics we use prevent us from proving the complexity bounds for the final form of our algorithm, we are able to show that we achieve good empirical behavior.

The area of polynomial-time interior-point algorithms for conic programming has been active for decades and thoroughly explored. The discovery of polynomial-time algorithms for linear programming [29, 47, 22, 30, 35, 20], their extension to conic programming [21, 27, 39, 44, 2, 3, 37, 44], and the definition of primal-dual and homogeneous versions of conic programming problems [33, 52, 6, 28, 1, 32] has yielded robust, efficient, and precise methods that have become essential tools for science and engineering. The extension of these methods to new cones will yield valuable algorithms for diverse applications.

This work is structured as follows. Chapter 2 covers the basic definitions of cones and some essential facts from convex analysis. Chapter 3 defines conic programming problems, defines the dual conic problem, covers the theory used to detect infeasible and unbounded problems, and defines the full and simplified homogeneous embedding. Chapter 4 describes self-concordant functions and barriers and visits some implications of the definition. Chapter 5 describes algorithms that achieve state-of-the-art polynomial bounds for unsymmetric conic programming but are not necessarily practical to implement because they solve linear systems that are too large to be practical. In this chapter we also describe how Nesterov-Todd scaling points are used to define algorithms that are more practical and solve linear systems of more moderate size. Chapter 6 studies search directions analogous to those defined by Ye and Skajaa but used in the context of the full homogeneous embedding. In this chapter, path-following and potential-reduction-like algorithms are defined. Chapter 7 defines a predictor-corrector algorithm for the simplified homogeneous embedding but using an alternative measure of centrality. Chapter 8 defines the conjugate barrier for the Chares [12] barrier for the exponential cone. Chapter 9 describes how to transform several important types of convex problems into exponential cone problems. Chapter 10 describes ECOS, our extension, and the numerical experiments to validate its behavior. Finally in Chapter 11 we state our conclusions and present future avenues for exploration.

Chapter 2

Preliminaries

2.1 Proper cones, their duals, and theorems of the alternative

A cone $\mathcal{K}$ is a subset of the Euclidean space with the property that for all vectors $x \in \mathcal{K}$ and all nonnegative scalars $\alpha \geq 0$ the scaled vector $\alpha x \in \mathcal{K}$. A convex cone is a cone that is also a convex set. When a cone is convex, any weighted sum of its elements with arbitrary positive scalars is contained in the cone. These weighted sums are called *conic combinations*. The converse is also true:if all conic combinations of $\mathcal{K}$ belong to $\mathcal{K}$ then the set is a convex cone.

Lemma 2.1.1. *If $\{x_1, \ldots, x_n\} \subset \mathcal{K}$ and $\alpha_1, \ldots, \alpha_n$ are positive scalars, then $\sum \alpha_i x_i \in \mathcal{K}$. And conversely if all $\sum \alpha_i x_i \in \mathcal{K}$ then $\mathcal{K}$ is a convex cone.*

Proof. Because $\mathcal{K}$ is convex, the *convex* combination

$$z = \sum_i \frac{\alpha_i}{\sum_j \alpha_j} x_i \in \mathcal{K},$$

and since $\mathcal{K}$ is a cone,

$$(\sum_j \alpha_j) z = \sum_i (\alpha_i x_i) \in \mathcal{K}.$$

So any conic combination of elements in the cone is also in the cone.

To show the converse, assume that all conic combinations belong to $\mathcal{K}$; then for any $x \in \mathcal{K}$ and any $\alpha \geq 0$ the product αx is a conic combination and therefore $\mathcal{K}$ is a cone. Finally, if $\sum_i \alpha_i = 1$ then the convex combination $z = \sum_i \alpha_i x_i$ is itself a conic combination and therefore $z \in \mathcal{K}$, which implies $\mathcal{K}$ is convex. □

For the purposes of conic programming, we require the cones to be *proper*. Such cones have characteristics necessary for the existence of barriers for their interior. Specifically, proper cones are topologically closed, have non-empty

interiors, and contain no straight lines. Cones with no straight lines are called *pointed*. Containing no straight lines is equivalent to the statement: $x \in \mathcal{K}$ implies $-x \notin \mathcal{K}$. For example, the positive orthant is a pointed cone while a half-space is not pointed.

Because proper cones are convex, they have a dual representation as the intersection of all half-spaces that contain them. The union of all normals to these half spaces forms another cone called the dual cone, denoted $\mathcal{K}^\star$. More precisely, let $\mathcal{K}$ be a proper cone and let $\langle\ ,\ \rangle$ be an inner-product.

Definition 2.1.1. The set $\mathcal{K}^\star = \{s \mid 0 \leq \ \langle s, x\rangle \ \forall\ x \in \mathcal{K}\}$ is called the *dual cone* of $\mathcal{K}$.

The following results about dual cones are presented without proof. For a detailed analysis see [11] or [43].

Lemma 2.1.2. *1. If $\mathcal{K}$ is a cone then $\mathcal{K}^\star = \{s \mid 0 \leq \ \langle s, x\rangle \ \forall x \in \mathcal{K}\}$ is a closed cone.*

2. If $\bar{K}$ denotes set closure, then $\bar{\mathcal{K}} = (\mathcal{K}^\star)^\star$. Hence if $\mathcal{K}$ is closed, then $\mathcal{K} = (\mathcal{K}^\star)^\star$.

3. If $\mathcal{K}$ is a proper cone, then so is $\mathcal{K}^\star$.

The definition of *dual cone* depends on the choice of inner product. For a given cone, two different inner products yield different dual cones. For most of this work the selected inner product is the Euclidean dot product; however, all examples of semidefinite programming are more natural when the inner product between two matrices X and S is defined as $\langle X, S\rangle = \mathbf{tr}(X^T S)$, where $\mathbf{tr}$ denotes trace. Semidefinite programming does not play an important role in this work except for a few examples, so this exception should not create much confusion.

The definition of dual cone yields a classification of cones into those that are identical to their dual, called self-dual, and those that differ from their dual, called non-self-dual. Symmetric cones are a subset of self-dual cones that are also homogeneous. Homogeneous cones are defined by the following property. For each pair x, s of elements in the cone $\mathcal{K}$ there exists a linear mapping A such that $Ax = s$ and (also) the image of the cone under the map is again the cone. The set of symmetric cones has been completely characterized [44, 24], yielding five different elementary cones from which all symmetric cones are constructed by Cartesian products.

We will restrict ourselves to the real Euclidean space. Of all elementary symmetric cones only two are subsets of the real Euclidean space: the Lorentz cone (also called the second-order cone) and the cone of positive definite matrices (semidefinite cone). The positive orthant of $\mathbb{R}^n$ is algebraically equivalent to the Cartesian product of n semidefinite cones of size 1. However it makes sense to talk about the positive orthant as one cone and not the product of trivial semidefinite cones.

Conic optimization problems defined in terms of the nonnegative orthant are called linear optimization problems, those defined in terms of the Lorentz cone

are called second-order cone programs (SOCPs), and those defined in terms of the positive semidefinite cone are called semidefinite programs (SDP). The exponential cone (a non-self-dual cone) can transform some problems with exponentials in the objective function and constraints into the conic programming formalism. These problems include entropy functions, geometric programming problems, logistic regression, and others.

Definition 2.1.2. The exponential cone $\mathcal{K}_e \subseteq \mathbb{R}^3$ is the 3-dimensional cone

$$\mathcal{K}_e = \mathbf{cl}\left\{(x,y,z) \mid z > 0,\ \exp\left(\frac{x}{z}\right) \leq \frac{y}{z}\right\},$$

where **cl** denotes the closure of the set.

The exponential cone is also the union

$$\mathcal{K}_e = \left\{(x,y,z) \mid z > 0,\ \exp\left(\frac{x}{z}\right) \leq \frac{y}{z}\right\} \cup \{(x,y,z) \mid x \leq 0,\ y \geq 0,\ z = 0\}.$$

The dual for this cone is the set

$$\mathcal{K}_e{}^\star = \mathbf{cl}\left\{(u,v,w) \mid u < 0,\ \exp\left(\frac{w}{u}\right) \leq -\frac{ev}{u}\right\},$$

where $e = \exp(1)$. The dual cone is also the union

$$\mathcal{K}_e{}^\star = \left\{(u,v,w) \mid u < 0,\ \exp\left(\frac{w}{u}\right) \leq -\frac{ev}{u},\right\} \cup \{(u,v,w) \mid u = 0\ v \geq 0,\ w \geq 0\}.$$

For a proof of the duality of the pair $\mathcal{K}_e$ and $\mathcal{K}_e{}^\star$ and the representation of the closure see [12, Section 4.3]. The exponential cone and its dual are an example of a non-self-dual cone pair.

We now show that the nonnegative orthant $\mathbb{R}^n_+$, the Lorentz cone $\mathcal{L}_n$, and the cone of positive semidefinite matrices S^n_+ are self-dual.

Lemma 2.1.3. *Let $\mathbb{R}^n_+ \subseteq \mathbb{R}^n$ be the nonnegative orthant (vectors of nonnegative entries in $\mathbb{R}^n$). Then $(\mathbb{R}^n_+)^\star$ is again the nonnegative orthant.*

Proof. If $s \in \mathbb{R}^n$ has some negative entry, say $s_i < 0$, then $x = e_i$ satisfies $s^T x < 0$, and s can't be in $(\mathbb{R}^n_+)^\star$. Therefore $(\mathbb{R}^n_+)^\star \subseteq \mathbb{R}^n_+$. On the other hand if $s \in \mathbb{R}^n_+$, all weighted sums $\sum_i s_i x_i$ with nonnegative coefficients x_i will yield positive values and therefore $s^T x \geq 0$, so $\mathbb{R}^n_+ \subseteq (\mathbb{R}^n_+)^\star$. □

The second example of a self-dual cone is the set of positive definite matrices S^n_+ interpreted as a subset of all symmetric matrices S^n of size n. To resolve the apparent conflict in definition between subsets of $\mathbb{R}^n$ and subsets of S^n, observe that the space of symmetric matrices S^n is in fact $\mathbb{R}^{n(n+1)/2}$, where one linear isomorphism from $s \in \mathbb{R}^{n(n+1)/2}$ to $S \in S^n$ is simply filling the upper triangular part of S column-wise and then completing the lower triangular part minus the diagonal for symmetry. Then, the inner product $\langle x, s\rangle$ between members of $\mathbb{R}^{n(n+1)/2}$ can be defined as $\mathbf{tr}\,(S^T X)$. Then by definition the cone dual to S^n_+ is

$$\mathcal{K}^\star = \left\{S \mid \mathbf{tr}\,(S^T X) > 0\ \forall\ X \in S^+_n\right\}.$$

Lemma 2.1.4. *The cone $\mathcal{K}^\star$ dual to S_+^n is again S_+^n.*

Proof. Assume that $S \notin S_+^n$. Then S is a symmetric indefinite matrix with eigensystem $S = V\Lambda V^T$, where at least one eigenvalue λ_i is negative. The positive semidefinite matrix $X = -\lambda_i V_i V_i^T$ will be such that $\text{trace}\,(S^T X) = -\lambda_i^2 < 0$ and therefore $S \notin \mathcal{K}^\star$, so $\mathcal{K}^\star \subseteq S_+^n$.

On the other hand, if $S \notin \mathcal{K}^\star$ then there exists some $X \in S_+^n$ for which $\text{trace}\,(S^T X) < 0$. If $X^{1/2}$ is defined as the matrix $X^{1/2} = V\Lambda^{1/2}V^T$, where $\Lambda^{1/2}$ is the diagonal matrix with $\sqrt{\lambda_i}$ for $\lambda_i > 0$ and zero otherwise, then $X = X^{1/2}X^{1/2^T}$. Because of the property

$$\mathbf{tr}(S^T X) = \mathbf{tr}(X^{1/2^T} S^T X^{1/2}),$$

the equivalence $\mathbf{tr}\,(S^T X) = \sum x_i^T S^T x_i$ holds (here x_i are the columns of $X^{1/2}$). Then $\mathbf{tr}\,(S^T X) < 0 \iff \sum x_i^T S^T x_i < 0 \iff x_i^T S^T x_i < 0$ for some i, and therefore $S \notin S_+^n$. □

The third and last example of a self-dual cone is the *Lorentz* cone or second-order cone, defined by the inequality

$$\mathcal{L}_{n+1} = \left\{(x_0, x) \mid x_0 > 0,\ x_0^2 > \|x\|_2^2\right\}. \tag{2.1}$$

Lemma 2.1.5. *The cone $\mathcal{L}^\star$ dual to $\mathcal{L}$ is again $\mathcal{L}$.*

Proof. Define $\mathcal{L}^\star = \left\{(u, u_0) \mid u^T x + u_0 x_0 \geq 0\right\}$ and assume $u \notin \mathcal{L}$, so that $\|u\| > u_0$. For the choice $(x, x_0) = (-u, \|u\|) \in \mathcal{L}$ the bound $x^T u + x_0 u_0 = -\|u\|^2 + \|u\| u_0 < 0$ holds. Therefore $(u, u_0) \notin \mathcal{L}^\star$, which implies that $\mathcal{L}^\star \subseteq \mathcal{L}$.

Now assume that $(u, u_0) \notin \mathcal{L}^\star$ but $(u, u_0) \in \mathcal{L}$. Then for some $(x, x_0) \in \mathcal{L}$, $u^T x + u_0 x_0 < 0$. However since $(x, x_0) \in \mathcal{L}$ and $(u, u_0) \in \mathcal{L}$, we have $0 \leq -\|u\|\|x\| + u_0 x_0 \leq u^T x + u_0 x_0$, whereas both inequalities imply that $0 \leq -\|u\|\|x\| + u_0 x_0 \leq u^T x + u_0 x_0 < 0$, which is contradictory. Therefore no such (u, u_0) exists and $\mathcal{L} = \mathcal{L}^\star$.

□

Cartesian products of cones are again cones and Cartesian products of proper cones are proper cones themselves. A Cartesian product of two cones $\mathcal{K}_1$ and $\mathcal{K}_2$ is denoted by $\mathcal{K}_1 \times \mathcal{K}_2$ and defined as the set of all $x = (x_1, x_2) \in \mathbb{R}^{n_1+n_2}$ with $x_1 \in \mathcal{K}_1$ and $x_2 \in \mathcal{K}_2$. Since the product of two cones is again a cone, this definition can be extended to define a product of any number of cones. As expected, the dual of a product of cones $\mathcal{K} = \mathcal{K}_1 \times \cdots \times \mathcal{K}_p$ is the product of the corresponding dual cones $\mathcal{K}^\star = \mathcal{K}_1^\star \times \cdots \times \mathcal{K}_p^\star$. So, whenever all the constituent cones are self-dual, the resulting product cone is itself self-dual.

2.1.1 Recession directions

A recession direction $d \in \mathbb{R}^n$ of a closed set C is a vector for which the half line that points in the recession direction and starts at any $x \in C$ is entirely

contained in C. Recession directions of closed convex sets form a convex cone, and closed convex cones coincide with their set of recession directions.

The following lemma will be useful in the sequel when we define certificates of unboundedness and infeasibility of conic programming problems. Denote by $0^+(C)$ [9] the set of all recession directions of a convex set C.

Lemma 2.1.6. *If $\mathcal{K}$ is a closed convex cone, then $0^+(\mathcal{K}) = \mathcal{K}$.*

Proof. Suppose $x \in \mathcal{K}$ and $d \in \mathcal{K}$. Then for any $\alpha \geq 0$ the conic combination $x+\alpha d$ is in $\mathcal{K}$, and therefore $d \in 0^+(\mathcal{K})$ and $\mathcal{K} \subseteq 0^+(\mathcal{K})$. Conversely if $d \in 0^+(\mathcal{K})$, for all $\alpha \geq 0$, we have $0 + \alpha d = \alpha d \in \mathcal{K}$ and therefore $d \in \mathcal{K}$ and $0^+(\mathcal{K}) \subset \mathcal{K}$. Here we used that 0 is a member of any closed cone. □

Any non-empty set contains at least $0 \in 0^+(C)$ in its recession directions; however, if a set closed, convex, and *unbounded* then $0^+(C)$ contains more directions. We state without proof the following result that relates unbounded convex sets and recession directions.

Lemma 2.1.7. *A non-empty closed convex set C is unbounded iff $0^+(C) \neq 0$.*

The proof for this lemma can be found in [49, Thm 8.4].

2.1.2 Theorem of the alternative

Minkowski's separating hyperplane theorem (2.1.8) is an essential tool in convex analysis that we use to prove a theorem of the alternative for conic programming problems.

Theorem 2.1.8. *Let $B \subseteq \mathbb{R}^n$ and $C \subseteq \mathbb{R}^n$ be convex subsets of $\mathbb{R}^n$ with disjoint intersection. Then there exists a vector $y \neq 0$ and a scalar β such that*

$$b^T y \leq \beta \leq c^T y,$$

for all $b \in B$ and $c \in C$.

Proof. See [49, Theorem 11.3]. □

In the sequel, the following theorem of the alternative helps show that Slater's constraint qualification implies strong duality.

Theorem 2.1.9. *Suppose $A \in \mathbb{R}^{m \times n}$, and let $\mathcal{K}$ be a closed convex cone. Either there exists $x \in \operatorname{int} \mathcal{K}$ such that $Ax = 0$ or there exists y such that $A^T y \in \mathcal{K}^\star$.*

Proof. Since A is linear, the image of $\mathcal{K}$ under A (denoted $A[\mathcal{K}]$) is convex. From the separation theorem (2.1.8), if $\operatorname{int} A[\mathcal{K}]$ does not contain 0 then there exists a separating hyperplane defined by y with $0 \leq \beta \leq y^T A^T x$ for all $x \in \mathcal{K}$. However, if $0 \in \operatorname{int} A[\mathcal{K}]$ then it is easy to see that no such hyperplane exists. It is also evident that if the separating hyperplane exists, then $\beta = 0$, for if $x \in \operatorname{int} \mathcal{K}$, the sequence $\frac{1}{n} x \in \operatorname{int} \mathcal{K}$ satisfies $\frac{1}{n} x^T y \to 0$ for any y.

This in turn implies that either $Ax = 0$ has a solution with $x \in \operatorname{int} \mathcal{K}$ or there exists a nonzero $y^T A x \geq 0$ for all $x \in \mathcal{K}$; in other words, $A^T y \in \mathcal{K}^\star$. □

Chapter 3

Conic Programming

3.1 Conic programming problems

In this section we loosely follow the exposition of Renegar [48] and Nesterov and Ye [45]. We formalize the definition of a conic problem, state the definition of the dual problem, and cite some useful results on weak and strong duality. For a more complete treatment of conic programming and conic duality we refer to [48, 8, 11, 34]. For a more general treatment of duality in the context of convex programming see [48]. We begin with the definition of a conic programming problem.

A cone problem is a convex optimization problem where the objective function is linear in the decision variables and the constraints are formed by the intersection of an affine set and a cone. More precisely: if $\mathcal{K}$ is a proper cone then a cone problem is a problem of the form

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \; & c^T x \\ \text{subject to } & Ax = b, \\ & x \in \mathcal{K}, \end{aligned} \tag{PC}$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

If $\mathcal{F}$ denotes the *feasible set* of problem (PC), then $\mathcal{F}$ is formed by the intersection of the affine space $\{x : Ax = b\}$ with the cone $\mathcal{K}$. The relative interior of $\mathcal{F}$ (denoted $\operatorname{rint} \mathcal{F}$) is the intersection of the affine space $\{x : Ax = b\}$ with $\operatorname{int} \mathcal{K}$. If the feasible set is empty then the problem is called *infeasible*; if the feasible set is non-empty then the problem is *feasible*; if $\operatorname{rint} \mathcal{F}$ is not empty the problem is *strictly feasible*; and finally if the problem is feasible but the objective is unbounded below, the problem is called *unbounded*.

We make the simplifying assumption that A has full row rank and therefore A^T defines an injective map. This assumption does not restrict the theoretical applicability of the methods. For any problem one can always remove redundant

rows from A to form a smaller system $(\hat{A}, \hat{b})$ with full row rank and the same solution set $\{x : \hat{A}x = \hat{b}\} = \{x : Ax = b\}$.

3.2 Conic duality

Weak duality refers to the observation that the dual objective values are lower bounds for the primal objective values. This is true for all convex optimization problems, but because of the special structure of conic programming problems we can derive an explicit form of the dual function and show that the dual problem is also a conic optimization problem.

The Lagrangian for problem (PC) is

$$\mathcal{L}(x, y, s) = c^T x + y^T(b - Ax) - s^T x,$$

with $x \in \mathcal{K}$ and $s \in \mathcal{K}^\star$. For all feasible x, $y \in \mathbb{R}^m$ and $s \in \mathcal{K}^\star$, the inequality

$$\mathcal{L}(x, y, s) = c^T x - s^T x \leq c^T x \tag{3.1}$$

holds. Therefore the *dual function* defined by

$$f^\star(y, s) = \inf_x \{\mathcal{L}(x, y, s)\}$$

will satisfy

$$f^\star(y, s) = \inf_x \{\mathcal{L}(x, y, s)\} \leq \inf_x \{\mathcal{L}(x, y, s), x \in \mathcal{F}\} \leq \inf_x \left\{c^T x,\ x \in \mathcal{F}\right\} = p^\star,$$

where $p^\star$ is the primal optimal value. This implies

$$f^\star(y, s) \leq p^\star \leq c^T x, \tag{3.2}$$

which in words means that any dual objective value will be a lower bound for all primal objective values.

Observe that if $c - A^T y - s = 0$ then $\mathcal{L}(x, y, s) = b^T y$. However if $c - A^T y - s \neq 0$, then $\Delta x = -c + A^T y + s$ satisfies $\mathcal{L}(\alpha \Delta x, y, s) = b^T y - \alpha \|\Delta x\|_2^2$ and as $\alpha \to \infty$, $\mathcal{L}(\alpha \Delta x, y, s) \to -\infty$. In this case $\mathcal{L}(x, y, s)$ is unbounded below and we can conclude that

$$f^\star(y, s) = \begin{cases} b^T y & c - A^T y - s = 0, \\ -\infty & \text{otherwise.} \end{cases}$$

The dual problem in *standard form* incorporates constraints for the region where $f^\star(y, s)$ is finite and has the form

$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{\text{maximize}}\ b^T y \\ &\text{subject to } A^T y + s = c, \\ &\qquad s \in \mathcal{K}^\star. \end{aligned} \tag{DC}$$

A consequence of weak duality is that, when the primal is unbounded it admits no lower bound, and therefore there can be no feasible dual point. On the other hand, unboundedness of the dual implies that the dual objective values admit no upper bound and there can be no feasible primal point. Conversely if there is a feasible primal point the dual must be bounded, and if there is a dual feasible point the primal must be bounded.

It is important to understand not only when problems are solvable but when the solution set is bounded, for this has algorithmic consequences. We can show that whenever there exists a *strictly* feasible primal point, the optimal set of the dual problem is bounded, and conversely when there exists a *strictly* feasible dual point, the optimal set of the primal problem is bounded. To show this we need the following lemma, which shows that a problem is unbounded if and only if there exists a recession direction along which the objective is reduced.

Lemma 3.2.1. *The primal problem is unbounded iff there exists a recession direction Δx for the feasible set $\mathcal{F}$ such that $c^T \Delta x < 0$.*

Proof. If such a direction exists then the problem is unbounded, for if x is a feasible point, the half-line $x + \alpha \Delta x$ for $\alpha > 0$ is feasible and $c^T(x + \alpha \Delta x) = c^T x + \alpha(c^T \Delta x) \to -\infty$ as $\alpha \to \infty$.

To show the converse, assume that the primal problem is unbounded. Choose an arbitrary point $x_0 \in \mathcal{F}$ and for every $k \in \mathbf{N}$ form the set $C_k = \mathcal{F} \cap \{c^T x \leq -k\}$ and the set $\hat{C}_k = \{\frac{x - x_0}{\|x - x_0\|} \; x \in C_k\}$. The sets $\hat{C}_k$ are nonempty bounded and closed and the sequence $\hat{C}_{k+1} \subseteq \hat{C}_k$ is monotonically decreasing. The Cantor intersection theorem states that there exists a d such that $d \in \cap_{i=0}^{\infty} \hat{C}_k$. Therefore there exists a sequence of $\beta_k > 0$ such that $x + \beta_k d \in C_k \subseteq \mathcal{F}$.

The sequence β_k admits no upper bound, for $-k > c^T(x_0 + \beta_k d) > c^T x_0 - \beta_k \|d\| \|c\|$ implies that $\beta_k > \frac{k - c^T x_0}{\|d\| \|c\|}$ and therefore $x_0 + \beta d \in \mathcal{F}$ for all $\beta > 0$. This establishes that d is a recession direction. Finally if $c^T x_0 = p_0$ and $c^T d \geq 0$ then $c^T(x_0 + \beta_k d) \geq p_0$ for all β_k reaching a contradiction. This implies that d is a recession direction with $c^T d < 0$. □

With this result it is simple to prove that if primal *strictly* feasible points exist then the dual problem is bounded, and that if dual *strictly* feasible points exist then the primal is bounded.

Lemma 3.2.2. *If the dual problem is strictly feasible, then no recession direction Δx with $c^T \Delta x \leq 0$ exists and the dual is bounded. If furthermore the primal problem is feasible, then the solution exists and the dual solution set is bounded.*

Proof. Assume there exists a strictly feasible dual $A^T \hat{y} + \hat{s} = c$ and that there is a recession direction for the primal feasible set Δx such that $c^T \Delta x \leq 0$. Since $A \Delta x = 0$, we have $\hat{y}^T A \Delta x = c^T \Delta x - \hat{s}^T \Delta x = 0$ and therefore $c^T \Delta x = \hat{s}^T \Delta x > 0$, which contradicts the existence of the recession direction. The strict inequality is due to the strict feasibility of $\hat{s}$. This result also implies that if the solution set exists, it must be bounded, otherwise there must exist a recession

direction for the feasible set where $(x^\star + \alpha\Delta x)^T c = p^\star$ for all $\alpha \geq 0$ with $\Delta x^T c = 0$. □

3.2.1 Strong duality

Denote by $p^\star = c^T x^\star$ (the primal objective value at the solution), and by $d^\star = b^T y^\star$ (the dual objective value at a solution of the dual problem). A primal-dual pair is said to satisfy *strong duality* if the equality $d^\star = p^\star$ holds. Strong duality has several consequences, one of them being the condition that at the solution ${s^\star}^T x^\star = 0$.

Observe that

$$d^\star = f^\star(y^\star, s^\star) \leq \mathcal{L}(x^\star, y^\star, s^\star) = c^T x^\star - {y^\star}^T(b - Ax^\star) - {s^\star}^T x^\star = p^\star - {s^\star}^T x^\star \leq p^\star = d^\star,$$

so that

$$p^\star - {s^\star}^T x^\star = p^\star,$$

and ${s^\star}^T x^\star = 0$. The condition ${s^\star}^T x^\star = 0$ is called *complementarity* and as we will now show, is a sufficient condition for a primal dual feasible point to be optimal.

Lemma 3.2.3. *If (x, y, s) is primal and dual feasible and if $x^T s = 0$, then (x, y, s) is primal and dual optimal.*

Proof. Assume x, y, s are primal and dual feasible, and that $x^T s = 0$.

$$x^T(A^T y + s) = x^T c \tag{3.3}$$

$$\implies b^T y = x^T c \tag{3.4}$$

$$\implies f^\star(y, s) = f(x). \tag{3.5}$$

Therefore $f(x) = p^\star$ and $f^\star(y, s) = d^\star$. □

3.2.2 Conditions for strong duality

For linear optimization problems strong duality always holds, but for conic programming the picture is more complicated. However, if there exist strictly feasible primal and dual points, then strong duality holds at all solutions.

Theorem 3.2.4. *If there exists a* strictly *feasible primal-dual point, then the primal and dual are solvable and strong duality will hold at all optimal pairs $x^\star, s^\star$.*

Proof. Since the dual problem is strictly feasible, Lemma 3.2.2 implies that the primal is bounded. However, the primal problem is feasible by assumption and therefore the primal is solvable. The same argument with the role of primal and dual reversed shows that the dual is also solvable. Furthermore, from Lemma 3.2.2 it follows that the solution sets of the primal and dual problem are bounded.

To show that strong duality holds we construct the somewhat artificial system

$$\tilde{A} = \begin{pmatrix} A & -b \\ -c^T & p^\star - 1 \end{pmatrix},$$

where $p^\star$ is the primal optimal value, and let $\tilde{\mathcal{K}}$ be the cone $\tilde{\mathcal{K}} = \mathcal{K} \times R^+ \times R^+$. By the theorem of the alternative (2.1.9) one of the two following statements must be true:

1. There exists $x \in \operatorname{int} \mathcal{K}$, $\tau > 0, \kappa > 0$ such that

$$\begin{pmatrix} A & -b & \\ -c^T & p^\star & -1 \end{pmatrix} \begin{pmatrix} x \\ \tau \\ \kappa \end{pmatrix} = 0.$$

2. There exists $\begin{pmatrix} -y \\ -\eta \end{pmatrix}$ such that

$$\begin{pmatrix} A^T & -c \\ -b^T & p^\star \\ & -1 \end{pmatrix} \begin{pmatrix} -y \\ -\eta \end{pmatrix} \in \mathcal{K}^\star \times R^+ \times R^+.$$

The first case never holds because if it did, we would have $A\frac{x}{\tau} = b$, $p^\star > c^T\frac{x}{\tau}$ and $\frac{x}{\tau}$ is a primal feasible point with a lower objective value than $p^\star$. Therefore, a solution $\begin{pmatrix} -y^T, & -\eta \end{pmatrix}^T$ must exist for the second system. Observe that $\eta \neq 0$ for if $\eta = 0$ then $A^T y + s = 0$ for some $s \in \mathcal{K}^\star$ and $b^T y \geq 0$, so (y, s) is a recession direction with $b^T y \geq 0$, which contradicts the strict feasibility of the primal. Finally observe that the equations $A^T\frac{y}{\eta} + \frac{s}{\eta} = c$, $\frac{s}{\eta} \in \mathcal{K}^\star$, and $b^T\frac{y}{\eta} \geq p^\star$ hold. Therefore $\frac{y}{\eta}, \frac{s}{\eta}$ is a feasible dual point and weak duality implies that $b^T\frac{y}{\eta} = p^\star$. This establishes the existence of a dual feasible point that achieves the primal optimal value. Therefore the dual optimal value $d^\star$ has to be equal to the primal optimal value $p^\star$. □

3.2.3 Certificates of infeasibility and unboundedness

Assume that there exists a dual direction with

$$b^T \Delta y > 0,\ A^T \Delta y + \Delta s = 0,\ \Delta s \in \mathcal{K}^\star. \tag{CI}$$

If (y, s) is a dual feasible point, we know from (2.1.6) that the point $(y + \alpha\Delta y, s + \alpha\Delta s)$ is feasible. Also along this direction the dual objective

$$f^\star(y + \alpha\Delta y, s + \alpha\Delta s) = -b^T(y + \alpha\Delta y)$$

can be increased arbitrarily. This implies that the dual problem is unbounded, and from weak duality we can conclude that the primal is infeasible.

On the other hand, if there exists a direction Δx that satisfies

$$A\Delta x = 0,\ c^T\Delta x < 0,\ x \in \mathcal{K}, \tag{CU}$$

then $x+\alpha\Delta x$ is feasible for all $\alpha > 0$ and along Δx the objective can be decreased arbitrarily. This implies the primal is unbounded and the dual infeasible.

A certificate of unboundedness is a direction Δx that satisfies (CU) and a certificate of infeasibility is a direction $\Delta y, \Delta s$ that satisfies (CI).

3.3 Self-dual problems

A problem P with dual D is *self-dual* if P is identical to D except maybe for a simple permutation of the variables. We now derive the construction of a self dual problem to set the stage for the discussion of the self-dual embedding and its variations.

Let (P) be a general conic problem

$$\begin{aligned}
\text{minimize} \quad & c_1^T x_1 + c_2^T x_2, && (3.6)\\
\text{subject to} \quad & A_{11}x_1 + A_{12}x_2 - b_1 \in \mathcal{K}_1, && (3.7)\\
& A_{21}x_1 + A_{22}x_2 - b_2 = 0, && (3.8)\\
& \text{and } x_1 \in \mathcal{K}_2. && \text{(P)}
\end{aligned}$$

The Lagrangian for this problem is given by

$$\begin{aligned}
\mathcal{L}(x_1, x_2, z_1, z_2, z_3) = {} & c_1^T x_1 + c_2^T x_2,\\
& - z_1(A_{11}x_1 + A_{12}x_2 - b_1)\\
& - z_2(A_{21}x_1 + A_{22}x_2 - b_2)\\
& - z_3^T x_1,
\end{aligned}$$

where $z_1 \in \mathcal{K}_1^\star$ and $z_3 \in \mathcal{K}_2^\star$. The Lagrangian is bounded below if

$$\nabla_{x_1}\mathcal{L} = c_1 - A_{11}^T z_1 - A_{21}^T z_2 - z_3 = 0, \text{ and} \tag{3.9}$$

$$\nabla_{x_2}\mathcal{L} = c_2 - A_{12}^T z_1 - A_{22}^T z_2 = 0. \tag{3.10}$$

Therefore the dual problem (D) has the form

$$\begin{aligned}
\text{minimize} \quad & -b_1^T z_1 - b_2^T z_2,\\
\text{subject to} \quad & -A_{11}^T z_1 - A_{21}^T z_2 + c_1 \in \mathcal{K}_2^\star,\\
& -A_{12}^T z_1 - A_{22}^T z_2 + c_2 = 0,\\
& \text{and } z_1 \in \mathcal{K}_1^\star.
\end{aligned} \tag{D}$$

If we require that the matrices $A_{11} = -A_{11}^T$, $A_{22} = -A_{22}^T$, $A_{12} = -A_{21}^T$, that the vectors $c_1 = -b_1$ and $c_2 = -b_2$, and that the cones $\mathcal{K}_1 = \mathcal{K}_2^\star$ and $\mathcal{K}_2 = \mathcal{K}_1^\star$,

then (P) = (D). Therefore a conic programming problem that can be written as

$$\begin{aligned} &\text{minimize} && c_1^T x_1 + c_2^T x_2 \\ &\text{subject to} && \begin{pmatrix} A_1 & A_2 \\ -A_2^T & A_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} s_1 \\ 0 \end{pmatrix} = \begin{pmatrix} -c_1 \\ -c_2 \end{pmatrix}, \\ & && x_1 \in \mathcal{K} \text{ and } s_1 \in \mathcal{K}^\star, \end{aligned} \tag{SD}$$

with A_1 and A_3 skew-symmetric, is self-dual.

When a self-dual problem has a strictly feasible point, it automatically has a strictly feasible dual point. Then, any self-dual problem with a strict interior is solvable, with a bounded solution set and strong duality holding at its solutions. By the symmetry of the objective functions, whenever strong duality holds the optimal value is zero.

Lemma 3.3.1. *If P is self-dual and strong duality holds, the optimal value $p^\star = 0$.*

Proof. Observe that for any feasible point x we have $-c^T x < d^\star = p^\star < c^T x$, and at the optimal $x^\star$ we have $-c^T x^\star = c^T x^\star$, and therefore $c^T x^\star = 0$. □

3.4 Homogeneous embedding and certificates of infeasibility

The self-dual embedding is a construction initially defined for linear programming [56, 53], and then extended to general conic programming [46, 15, 33, 45]. The self-dual embedding extends a conic programming problem into a larger self-dual problem for which an initial strictly feasible point is known. This larger primal-dual problem is always solvable and strong duality holds at its solution. When this larger problem is solved it can yield one of three things: a solution for the original problem, a certificate that proves the problem is not solvable, or information that the problem is badly formed and strong duality does not hold at the solution and neither a certificate of optimality nor a certificate of infeasibility or unboundedness can be found.

3.4.1 The self-dual embedding

To construct the self-dual embedding, the primal and dual problem are combined and three artificial variables τ, κ and θ are added. Additionally, two non-negativity constrains $\tau > 0$ and $\kappa > 0$ are imposed. Given an initial point $z_0 = (y_0, x_0, \tau_0, s_0, \kappa_0, \theta_0)$ with $x_0 \in \operatorname{int} \mathcal{K}$, $s_0 \in \operatorname{int} \mathcal{K}^\star$, $\tau_0 > 0$, $\kappa_0 > 0$ and $\theta_0 = 1$, the initial complementarity is defined as $\mu_0 = \frac{x_0^T s_0 + \tau_0 \kappa_0}{\nu + 1}$ where for now ν is some positive constant.

We also define the primal and dual residuals

$$p_r = -Ax_0 + \tau_0 b, \; d_r = A^T y_0 + s_0 - \tau_0$$

and denote by g_r the residual

$$g_r = c^T x_0 - b^T y_0 + \kappa_0.$$

Let G be the matrix

$$G = \begin{pmatrix} & A & -b & p_r \\ -A^T & & c & d_r \\ b^T & -c^T & & g_r \\ -p_r^T & -d_r^T & -g_r^T & \end{pmatrix} \tag{3.11}$$

and define the problem

$$\begin{aligned} &\text{minimize } \mu_0(\nu+1)\theta \\ &\text{subject to } G\begin{pmatrix} y \\ x \\ \tau \\ \theta \end{pmatrix} - \begin{pmatrix} 0 \\ s \\ \kappa \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\mu_0(\nu+1) \end{pmatrix} \\ &\qquad x \in \mathcal{K},\ s \in \mathcal{K}^\star,\ 0 \le \tau,\ 0 \le \kappa. \end{aligned} \tag{HSD}$$

Lemma 3.4.1. *Problem* (HSD) *is self dual.*

Proof. A simple permutation of the variables allows us to write (HSD) as

$$\begin{aligned} &\text{minimize } \mu_0(\nu+1)\theta \\ &\text{subject to } \begin{pmatrix} & c & -A^T & d_r \\ -c^T & & b^T & g_r \\ A & -b & & p_r \\ -d_r^T & -g_r^T & -p_r^T & \end{pmatrix} \begin{pmatrix} x \\ \tau \\ y \\ \theta \end{pmatrix} - \begin{pmatrix} s \\ \kappa \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\mu_0(\nu+1) \end{pmatrix}, \\ &(x,\tau) \in \mathcal{K} \times \mathbb{R}^+, (s,\kappa) \in \mathcal{K}^\star \times \mathbb{R}^+ m \end{aligned} \tag{3.12}$$

which is clearly of the form of (SD). □

Lemma 3.4.2. *For any feasible point, $x^T s + \tau\kappa = \mu_0(\nu+1)\theta$.*

Proof. Since the matrix G in (3.11) is skew-symmetric, we have

$$\big(y, x, \tau, \theta\big)^T G \big(y, x, \tau, \theta\big) - s^T x - \tau\kappa = -\mu_0(\nu+1)\theta,$$

so that

$$s^T x + \tau\kappa = \mu_0(\nu+1)\theta. \tag{3.13}$$

□

Lemma 3.4.3. *The point z_0 is primal and dual strictly feasible. Therefore* (HSD) *is solvable and strong duality holds.*

Proof. By assumption the elements x_0, τ_0, s_0 and κ_0 are strictly feasible with respect to their cones; therefore we only need to show that z_0 is feasible with respect to the linear constraints.

Observe that, by the definition of the residuals, the first three linear constraints are trivially satisfied. Therefore we can write

$$\begin{pmatrix} & A & -b \\ -A^T & & c \\ b^T & -c^T & \end{pmatrix} \begin{pmatrix} y_0 \\ x_0 \\ s_0 \end{pmatrix} + \begin{pmatrix} p_r \\ d_r \\ g_r \end{pmatrix} - \begin{pmatrix} 0 \\ s_0 \\ \kappa_0 \end{pmatrix} = 0. \tag{3.14}$$

Since the above matrix is skew-symmetric, by multiplying the above by $\begin{pmatrix} y_0 & x_0 & s_0 \end{pmatrix}$ on the left, we conclude that

$$\begin{pmatrix} y_0 \\ x_0 \\ s_0 \end{pmatrix}^T \begin{pmatrix} p_r \\ d_r \\ g_r \end{pmatrix} - x_0^T s_0 - \tau_0 \kappa_0 = 0, \tag{3.15}$$

which is equivalent to

$$\begin{pmatrix} -p_r \\ -d_r \\ -g_r \\ 0 \end{pmatrix}^T \begin{pmatrix} y_0 \\ x_0 \\ s_0 \\ 1 \end{pmatrix} = -\mu_0(\nu + 1)$$

and the last linear constraint of (HSD) holds. □

Because (HSD) is self-dual and because it is strictly feasible, strong duality holds and at the solution the optimal value is zero. Therefore $\theta^\star = 0$ and at the solution the equations

$$\begin{pmatrix} & A & -b \\ -A^T & & c \\ b^T & -c^T & \end{pmatrix} \begin{pmatrix} y^\star \\ x^\star \\ \tau^\star \end{pmatrix} - \begin{pmatrix} 0 \\ s^\star \\ k^\star \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.16a}$$

$$(s^\star)^T (x^\star) + \tau^\star \kappa^\star = 0 \tag{3.16b}$$

hold (here we used (3.13) for (3.16b)).

From a solution of (HSD), we can (in most cases) construct either a solution for both (PC) and (DC) or a certificate of either infeasibility or unboundedness. Observe that if we find a solution with $\tau^\star > 0$ we can form the triplet

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{s} \end{pmatrix} = \begin{pmatrix} x^\star/\tau^\star \\ y^\star/\tau^\star \\ s^\star/\tau^\star \end{pmatrix},$$

which is primal and dual feasible and for which $\hat{x}^T \hat{s} = 0$ and is therefore optimal.

On the other hand, if there is a solution with $\kappa^\star > 0$, from constraint (3.16a) we conclude that $b^T y^\star - c^T x^\star > 0$.

From the rest of the constraints of (HSD), the equations

$$Ax^\star = 0,\ A^T y^\star + s^\star = 0,\ x^\star \in \mathcal{K},\ s^\star \in \mathcal{K}^\star$$

hold. Therefore whenever $c^T x^\star < 0$, the vector $x^\star$ is a certificate of unboundedness, and whenever $-b^T y^\star < 0$, the pair of vectors $(y^\star, s^\star)$ is a certificate of infeasibility.

3.4.2 A note on ill-formed problems

Since by construction the homogeneous embedding has a *strictly* primal feasible point and by self-duality this is also a dual *strictly* feasible point, it is solvable and by Theorem 3.2.4 strong duality holds at the solution.

So what happens if for the original primal-dual pair strong duality does not hold? We can discard a few possibilities. At the solution, $\tau = 0$, for otherwise $x/\tau, y/\tau, s/\tau$ is a solution for which $\frac{x}{\tau}^T \frac{s}{\tau} = 0$, which is contradictory.

Neither $c^T x < 0$ nor $b^T y > 0$ can be true, for otherwise the problem would be infeasible or unbounded. This implies that $-c^T x + b^T y - \kappa = 0$ and $\kappa = 0$. Therefore, if a problem is feasible but strong duality does not hold at the solution, then both $\tau = 0$ and $\kappa = 0$.

3.4.3 The simplified homogeneous embedding

The simplified homogeneous embedding [53] removes the variable θ from the self-dual embedding and must be solved by a method that reduces the infeasibility of the linear constraints at the same time as it proceeds towards optimality. The simplified homogeneous self-dual embedding is the problem of finding a nonzero feasible point for the equations

$$Ax = \tau b, \tag{3.17}$$
$$A^T y + s = \tau c, \tag{3.18}$$
$$c^T x - b^T y + \kappa = 0, \tag{3.19}$$
$$x \in \mathcal{K}, s \in \mathcal{K}^\star, \tag{3.20}$$
$$\tau, \kappa \geq 0. \tag{sHSD}$$

The following lemma is important for understanding some properties of the feasible points of problem (3.19).

Lemma 3.4.4. *For any solution of the feasibility problem* (sHSD)*, the complementarity relations* ${x^\star}^T s^\star$ *and* $\tau^\star \kappa^\star$ *hold.*

Proof. Let $\tilde{G}$ be the skew-symmetric matrix

$$\tilde{G} = \begin{pmatrix} & A & -b \\ -A^T & & c \\ b^T & -c^T & \end{pmatrix}. \tag{3.21}$$

Observe that the feasibility problem can be written as

$$\tilde{G}\begin{pmatrix} y \\ x \\ \tau \end{pmatrix} - \begin{pmatrix} 0 \\ s \\ \kappa \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{3.22}$$

$$x \in \mathcal{K},\ s \in \mathcal{K}^\star, \tag{3.23}$$

$$\tau,\ \kappa \geq 0. \tag{3.24}$$

Therefore, for any feasible point,

$$\begin{pmatrix} y \\ x \\ \tau \end{pmatrix}^T \begin{pmatrix} & A & -b \\ -A^T & & c \\ b^T & -c^T & \end{pmatrix} \begin{pmatrix} y \\ x \\ \tau \end{pmatrix} - \begin{pmatrix} y \\ x \\ \tau \end{pmatrix}^T \begin{pmatrix} 0 \\ s \\ \kappa \end{pmatrix} = 0$$

and by the skew symmetry of $\tilde{G}$ we can conclude that

$$\begin{pmatrix} y \\ x \\ \tau \end{pmatrix}^T \begin{pmatrix} 0 \\ s \\ \kappa \end{pmatrix} = 0.$$

Since both products $x^T s$ and $\tau\kappa$ are positive, we have $x^T s = 0$ and $\tau\kappa = 0$. □

Since the equations for the simplified homogeneous embedding are identical to the optimality conditions of the homogeneous embedding (3.16), we can extract solutions for the original problem and certificates of infeasibility or unboundedness in exactly the same manner as for the homogeneous self-dual embedding.

3.4.4 Interior solutions

It is entirely possible that a well-formed problem embedded in either of the homogeneous embeddings contains a solution where both $\tau = 0$ and $\kappa = 0$. If this were to happen for a well-formed original (pre embedding) problem, we would not be able to draw any conclusion. To take advantage of the homogeneous embeddings, the optimization problem has to be solved with a method that finds interior solutions when they exist. As it turns out, interior-point methods are ideally suited for this task [25].

Chapter 4

Interior point theory

4.1 Self-concordant functions

Nesterov and Nemirovski showed that whenever there exists a computable strongly nondegenerate self-concordant barrier for the feasible set of a convex problem in standard form, then a polynomial-time algorithm exits. Nesterov and Nemirovski also showed [43] that every cone admits a strongly nondegenerate self-concordant barrier, the so-called universal barrier. This would seem to imply that conic programming is of polynomial complexity. However it is not the case because conic programs formulated with the copositive cone have been shown to generalize some NP-complete problems [18]. This apparent contradiction does not show that P=NP, but rather that a barrier function computable in polynomial time does not exist for the copositive cone (unless P=NP).

The existence of nondegenerate self-concordant barriers is responsible for the polynomial-time behavior of interior-point methods for conic programming problems. This is because self-concordant functions are very similar to their quadratic approximation, making Newton's method very efficient.

We now state two equivalent definitions of self-concordant functions and derive some bounds for the function value. These bounds allow us to analyze the behavior of Newton's method on them. The results from this section will be essential for the analysis of the computational complexity of primal-dual interior-point methods in the following sections.

Let $f(x)$ be a twice-differentiable, strictly convex function with open domain $D_f \subseteq \mathbb{R}^n$, and denote by $g(x)$ its gradient and by $H(x)$ its Hessian. Denote by $\|y\|_x$ the norm of y induced by the Hessian of f, i.e. $\|y\|_x = \sqrt{y^T H(x) y}$.

Denote by $B_x(x, r) \subseteq \mathbb{R}^n$ the open ball about x with radius r in the norm induced by $H(x)$ i.e. $B_x(x, r) = \{y \mid \|y - x\|_x < r\}$. This set is of particular importance and is called the *Dikin* ellipsoid.

Definition 4.1.1. A convex function $f : D_f \subseteq \mathbb{R}^n \to \mathbb{R}$ is said to be *(strongly nondegenerate)* self-concordant if for all $x \in D_f$,

$$\mathcal{B}_x(x, 1) \subset D_f, \tag{4.1}$$

and for any $y \in B_x(x,1)$ and all $v \in \mathbb{R}^n \setminus \{0\}$,

$$1 - \|y-x\|_x \leq \frac{\|v\|_y}{\|v\|_x} \leq \frac{1}{1-\|y-x\|_x}. \tag{4.2}$$

This definition differs from that initially made by Nesterov and Nemirovski in [43]; however, Renegar [48] shows it to be equivalent for all important purposes.

The following *local* upper bound is valid only within the *Dikin* ellipsoid of radius one.

Theorem 4.1.1. *Suppose $x \in \operatorname{int} D_f$ and $\|y-x\|_x \leq 1$. Then*

$$f(y) \leq f(x) + g(x)^T(y-x) + \omega_\star(\|x-y\|_{H(x)}),$$

with $\omega_\star(t) = -t - \log(1-t)$.

Proof. From the fundamental theorem of calculus the equality

$$f(y) = f(x) + g(x)^T(y-x) + \int_0^1 \int_0^\tau \|y-x\|^2_{x+\eta(y-x)}\, d\eta\, d\tau$$

holds. Using the upper bound from (4.2) we get the bound

$$f(y) \leq f(x) + g(x)^T(y-x) + \int_0^1 \int_0^\tau \frac{\|y-x\|_x^2}{(1-\eta\|y-x\|_{H(x)})^2} d\eta\, d\tau,$$

and integrating yields

$$f(y) \leq f(x) + g(x)^T(y-x) - \|y-x\|_{H(x)} - \log\left(1-\|y-x\|_{H(x)}\right).$$

□

Lemma 4.1.2. *For any $x \in D_f$ and $y \in \mathcal{B}_x(x,1)$,*

$$\left| \|y-x\|^2_{x+\tau(y-x)} - \|y-x\|_x^2 \right| \leq \|y-x\|_x^2 \left(\frac{1}{(1-\|y-x\|_x)^2} - 1 \right).$$

Proof. To simplify notation denote $\beta = \|y-x\|_x$ and $\beta_\tau = \|y-x\|_{x+\tau(y-x)}$. If $\beta_\tau \geq \beta$ then from the upper bound of (4.2) we conclude that

$$\left|\beta_\tau^2 - \beta^2\right| \leq \frac{\beta^2}{(1-\tau\beta)^2} - \beta^2 = \beta^2 \left(\frac{1}{(1-\tau\beta)^2} - 1 \right).$$

If on the other hand $\beta_\tau < \beta$, then from the lower bound of (4.2) and $0 < 1-\tau\beta < 1$ we conclude that

$$\begin{aligned} \left|\beta_\tau^2 - \beta^2\right| &= \beta^2 - \beta_\tau^2 \\ &\leq \beta^2 \left(1 - (1-\tau\beta)^2\right) \\ &\leq \beta^2 \left(\frac{1}{(1-\tau\beta)^2} \right) \left(1 - (1-\tau\beta)^2\right) \\ &= \beta^2 \left(\frac{1}{(1-\tau\beta)^2} - 1 \right). \end{aligned} \tag{4.3}$$

□

The controlled variation of the Hessian implies a useful bound on the error of a quadratic approximation within the Dikin ellipsoid. Using the previous lemma we can prove the following bound.

Theorem 4.1.3. *Let* $q_x(y) = f(x) + g(x)^T(y-x) + \frac{1}{2}\|y-x\|^2_{H(x)}$. *For all* $y \in B_x(x,1)$

$$|f(y) - q_x(y)| \leq \frac{\|y-x\|_x^3}{3\left(1-\|y-x\|_x\right)}.$$

Proof. Let $\phi(\tau) = f(x+\tau(y-x))$ be a univariate functional. From the fundamental theorem of calculus we can write

$$\phi(1) = \phi(0) + \phi'(0) + \frac{1}{2}\phi''(0) + \int_0^1 \int_0^\tau \phi''(\eta) - \phi''(0) d\eta \; d\tau.$$

Therefore

$$\left|\phi(1) - \phi(0) - \phi'(0) - \frac{1}{2}\phi''(0)\right| \leq \int_0^1 \int_0^\tau |\phi''(\eta) - \phi''(0)| \, d\eta \; d\tau$$

Using Lemma 4.1.2 we derive the inequality

$$|\phi''(\eta) - \phi''(0)| = \left|\|y-x\|^2_{x+\eta(y-x} - \|y-x\|_x^2\right| \leq \beta^2 \left(\frac{1}{(1-\eta\beta)^2} - 1\right),$$

where β is as defined in (4.1.2). Integrating yields the bound

$$\int_0^1 \int_0^\tau \beta^2 \left(\frac{1}{(1-\eta\beta)^2} - 1\right) d\eta = \beta^3 \int_0^1 \frac{\tau^2}{(1-\tau\beta)} d\tau \leq \frac{\beta^3}{(1-\beta)} \int_0^1 \tau^2 d\tau = \frac{\beta^3}{3\left(1-\tau\beta\right)},$$

which implies the desired result

$$\left|\phi(1) - \phi(0) - \phi'(0) - \frac{1}{2}\phi''(0)\right| \leq \frac{\beta^3}{3\left(1-\tau\beta\right)}.$$

□

We derive a *global* lower bound for the value of $f(y)$ in terms of $f(x)$, its gradient $g(x)$, and the distance $\|x-y\|_x$. We follow the argument in [41] for which we introduce the original definition of self-concordant functions. We omit the proof of equivalence between definitions and refer to [48] for details.

Suppose $f : \mathbb{R}^n \to \mathbb{R}$, with $f \in C^3$ a strictly-convex function, choose $x \in D_f$ and an arbitrary direction $v \in \mathbb{R}^n$. Define the univariate function $\phi(t) = f(x+tv)$.

Definition 4.1.2. $f(x)$ is a self-concordant function iff there exists a constant $M_f \geq 0$ such that

$$|\phi'''(t)| \leq M_f \phi''(t)^{\frac{3}{2}},$$

for any $x \in D_f$ and any v.

Any self-concordant function can be scaled so that $M_f = 2$. Observe that if $\bar{\phi}(t) = C\phi(t)$ then $|\bar{\phi}'''(t)| \leq M_f \frac{1}{\sqrt{C}} \bar{\phi}''(t)^{\frac{3}{2}}$, so that $\bar{f}(x) = (M_f/2)^2 f(x)$ will be self-concordant with constant $M_{\bar{f}} = 2$.

Lemma 4.1.4. *Let f be a self-concordant function with $M_f = 2$ and let ϕ be defined as above. Then the function $\psi(t) = \frac{1}{\sqrt{\phi''(t)}}$ satisfies $|\psi'| \leq 1$.*

Proof. Clearly

$$\psi'(t) = \frac{-\phi'''(t)}{2\sqrt{\phi''(t)}^3},$$

and from Definition 4.1.2 we conclude that $|\psi'(t)| \leq 1$. □

Lemma 4.1.5. *For any pair $x, y \in D_f$,*

$$\|x - y\|_y \geq \frac{\|x - y\|_x}{1 + \|x - y\|_x}.$$

Proof. Define ψ as in Lemma 4.1.4 with $v = y - x$. Then $\psi(0) = \frac{1}{\|y-x\|_x}$ and $\psi(1) = \frac{1}{\|y-x\|_y}$. Now observe that

$$\psi(1) = \psi(0) + \int_0^1 \psi'(\tau) d\tau \leq \psi(0) + 1, \tag{4.4}$$

so that

$$\frac{1}{\|y - x\|_y} \leq 1 + \frac{1}{\|y - x\|_x}.$$

□

We are now in a position to prove the lower bound.

Theorem 4.1.6. *If $x \in D_f$, then for any $y \in D_f$*

$$f(x) + g(x)^T(y - x) + \omega\left(\|y - x\|_x\right) \leq f(y),$$

with $\omega(t) = t - \log(1 + t)$.

Proof. From the fundamental theorem of calculus,

$$f(y) = f(x) + g(x)^T(y - x) + \int_0^1 \int_0^\tau \|y - x\|^2_{x+\eta(y-x)} \, d\eta \, d\tau.$$

Using the lower bound from Lemma 4.1.5 we get the bound

$$f(y) \geq f(x) + g(x)^T(y - x) + \int_0^1 \int_0^\tau \frac{\|y - x\|_x^2}{(1 + \eta \|y - x\|_x)^2} d\eta \, d\tau,$$

and integrating yields

$$f(y) \geq f(x) + g(x)^T(y - x) + \|y - x\|_{H(x)} - \log\left(1 + \|y - x\|_x\right).$$

□

The global lower bound in Theorem 4.1.6 and the local upper bound of Theorem 4.1.1 are written in terms of the scalar functions $\omega(t) = t - \log(1+t)$ and $\omega^\star(t) = -u - \log(1-u)$. The following lemma from [41] relates the two.

Lemma 4.1.7. *For any $t > 0$, $\omega(t) = \max_{0 \le z \le 1} \{zt - \omega^\star(z)\}$ and for any $0 \le \tau < 1$, $\omega^\star(\tau) = \max_{0 \le z} \{z\tau - \omega(z)\}$.*

Proof. It is simple to see that the minimizing z for any $\tau > 0$ is given by $z = \frac{\tau}{1-\tau}$. Substituting into $\tau z^\star - \omega(z^\star)$ yields the desired expression for $\omega^\star$. The second equality is deduced by a similar reasoning. □

Lemma 4.1.8. *If $\|g(y) - g(x)\|_{H^{-1}(y)} < 1$ then*

$$f(y) - g(x)^T(y-x) - f(x) \le \omega^\star\left(\|g(y) - g(x)\|_{H^{-1}(y)}\right).$$

Proof. Let $\phi(z) = f(z) - g(x)^T z$. Since it is the sum of a self-concordant function and a linear function, $\phi(z)$ is self-concordant and admits the global lower bound

$$\phi(z) \ge \phi(y) + \nabla\phi(y)^T(y-z) + \omega(\|y-z\|_y). \tag{4.5}$$

Since $\nabla\phi(x) = g(x) - g(x) = 0$, x is its minimizer and therefore minimizing both sides yields the bound

$$\begin{aligned}
\phi(x) = \min_{z \in D_f}\{\phi(z)\} &\ge \min_{z \in D_f}\left\{\phi(y) - \nabla\phi(y)(y-z) + \omega(\|y-z\|_y)\right\} \\
&\ge \min_{z \in D_f}\left\{\phi(y) - \|\nabla\phi(y)\|_{H^{-1}(y)}\|(y-z)\|_y + \omega(\|y-z\|_y)\right\} \\
&= \phi(y) - \omega^\star\left(\|\nabla\phi(y)\|_{H^{-1}(y)}\right),
\end{aligned} \tag{4.6}$$

where for the last equality we used (4.5). The desired bound follows. □

4.2 Newton's method on self-concordant functions

From the bounds of the previous section we gain some insight on the behavior of Newton's method for self-concordant functions. We are also able to understand the regime of quadratic convergence and derive some results used to bound the computational complexity of primal-dual methods for conic programming problems. We only state the results that are necessary for the analysis of the following sections. For a more thorough description see [11, 43, 48].

We start by bounding the reduction attainable along the Newton direction in terms of the *Newton decrement*.

Definition 4.2.1. Let f be a nondegenerate self-concordant function, H be its Hessian, and g its gradient. The *Newton direction* Δx is the solution to the system $H(x)\Delta x = -g(x)$. The *Newton decrement* is the value $\lambda = \|\Delta x\|_x$.

The process of updating the iterate x with the rule $x^+ = x + \frac{1}{1+\lambda}\Delta x$ is called the *damped Newton method* and it is a useful theoretical tool to understand the behavior of Newton's method. Using the bound from Theorem (4.1.1) we can show that along the Newton direction we can reduce the function value by at least $\lambda - \log(1+\lambda)$.

Theorem 4.2.1. *For the Newton direction Δx, there exists $\alpha > 0$ such that $x + \alpha\Delta x$ is feasible and*

$$f(x + \alpha\Delta x) \leq f(x) - \lambda + \log(1+\lambda).$$

Proof. Choose $\alpha = \frac{1}{1+\lambda}$, so that $\|x + \alpha\Delta x - x\|_x = \frac{\lambda}{1+\lambda} < 1$ and $x + \alpha\Delta x \in \mathcal{B}_x(x, 1) \in \mathcal{K}$. For the second statement observe that (4.1.1) implies that

$$\begin{aligned} f(x + \alpha\Delta x) &\leq f(x) + \alpha g(x)^T(\Delta x) - \alpha \|\Delta x\|_x - \log(1 - \alpha \|\Delta x\|_x) \\ &= f(x) - \alpha\lambda^2 - \alpha\lambda - \log(1 - \alpha\lambda) \\ &= f(x) - \lambda + \log(1+\lambda). \end{aligned} \tag{4.7}$$

□

Newton's method on self-concordant functions allows us to detect when a minimizer exists. The following lemma shows that if $\lambda < 1$ then f has a minimizer. A corollary is that if f is unbounded, then the Newton decrement is bounded below and a guaranteed decrement can be achieved in every step.

Lemma 4.2.2. *Let f be a self-concordant function. If $\lambda(x) < 1$ then f has a minimizer $x^\star$.*

Proof. We follow the proof from [41]. The strategy is to bound a level set of f, for if the level set $L_f(f(x)) = \{y \mid f(y) \leq f(x)\}$ is bounded then f must have a minimizer.

From the global lower bound of (4.1.6), for all $y \in D_f$

$$\begin{aligned} & f(y) \geq f(x) + g(x)^T(y - x) + w(\|y - x\|_x), \\ \implies & f(y) \geq f(x) - \lambda \|y - x\|_x + w(\|y - x\|_x). \end{aligned}$$

Therefore if $y \in L_f(f(x))$,

$$\begin{aligned} f(x) \geq f(y) &\geq f(x) + g(x)^T(y - x) + w(\|y - x\|_x) \\ \implies f(x) &\geq f(x) - \lambda \|y - x\|_x + w(\|y - x\|_x) \end{aligned} \tag{4.8}$$

and

$$1 \geq \lambda \geq \frac{1}{\|y - x\|_x} w\left(\|y - x\|_x\right) = 1 - \frac{1}{\|y - x\|_x} \log\left(1 + \|y - x\|_x\right).$$

The function $1 - \frac{1}{t}\log(1+t)$ is monotonically increasing and asymptotically approaches 1 as the argument tends to infinity. Therefore for any $\lambda < 1$ there exists a value $\bar{t}$ such that $\|y - x\|_x \leq \bar{t}$ for all $y \in L_f(f(x))$ and the level set is bounded. □

When λ is small enough, in particular $\lambda < 1$, there exists a minimizer and also the difference between the function value and the optimal value is small, in the following sense.

Lemma 4.2.3. *Let $\lambda(x) < 1$. Then $f(x) - f(x^\star) \leq \omega^\star(\lambda)$.*

Proof. It suffices to let $x = x^\star$ and $y = x$ in Lemma 4.1.8, for then

$$f(x) - f(x^\star) \leq \omega^\star\left(\|g(x)\|_{H(x)^{-1}}\right) = \omega^\star(\lambda).$$

□

4.3 Barrier functions

Interior point methods remove the conic constraints from the problem formulation by incorporating them into the objective. This is done by adding a barrier function to guarantee that iterates that reduce the objective value will stay within the barrier function's domain. The prototypical example is the logarithmic barrier function for the positive orthant: $f(x) = -\sum \log(x_i)$, used in linear programming. Aside from 'blowing up' as an iterate approaches the boundary of the positive orthant, this barrier is also self-concordant and ν-logarithmically homogeneous with $\nu = n$, where the latter means that any scaling of the variables by $\tau > 0$ will result in a change in function value of exactly $-\nu \log(\tau)$ in the following way:

$$f(\tau x) = -\nu \log(\tau) + f(x).$$

For linear programming, ν equals the dimensionality of the cone n. This does not hold for all cones and barriers; for example, the barrier for the semidefinite cone $f(X) = -\log(|X|)$ for matrices of size $n \times n$ is also logarithmically homogeneous and self-concordant. In this case $f(\tau X) = -n \log(\tau) + f(X)$ but symmetric matrices form a subspace of dimension $n(n+1)/2$. Another example is the self-concordant 2-logarithmically homogeneous barrier for the Lorentz cone $f(x, x_0) = -\log\left(x_0^2 - \|x\|_2^2\right)$. In this case the dimension of the cone does not change the value of ν, and this barrier has parameter $\nu = 2$ for all dimensions.

The coefficient ν is called the complexity of the barrier and as Nesterov and Nemirovski have shown, complexity bounds for interior point methods must be derived in terms of this number. Between two different barriers for the same cone, the barrier with the smaller complexity parameter will yield an algorithm with a lower worst-case complexity.

In [43, Theorem 2.3.6] Nesterov and Nemirovski show that n is the minimum parameter for any ν-logarithmically homogeneous barrier for the positive orhtant and therefore this barrier is optimal for that geometry. They also show that in general no useful barrier can have a parameter smaller than 1 [43, Corollary 2.3.3].

Definition 4.3.1. Let f be a nondegenerate self-concordant function with domain $\mathcal{K}$. If $f(\tau x) = -\nu \log(\tau) + f(x)$ for all $\tau > 0$, then f is a ν-logarithmically homogeneous barrier for $\mathcal{K}$.

Logarithmic homogeneity yields some useful algebraic properties that we now list.

Lemma 4.3.1. *If f is a ν-logarithmically homogeneous barrier function, then the following hold:*

1. $g(\tau x) = \frac{1}{\tau} g(x)$
2. $H(\tau x) = \frac{1}{\tau^2} H(x)$
3. $g(x) = -H(x)x$
4. $g(x)^T x = -\nu$
5. $x^T H(x) x = \nu$
6. $g(x) H^{-1}(x) g(x) = \nu$

Proof. 1. Differentiating $f(\tau x) = -\nu \log(\tau) + f(x)$ with respect to x yields the first equality.

2. Differentiating the first property with respect to x yields the second equality.

3. From property 1 it follows that
$H(x)x = \lim_{\alpha \to 0} \left(\frac{g(x+\alpha x) - g(x)}{\alpha} \right) = \lim_{\alpha \to 0} \left(\frac{-1}{1+\alpha} \right) g(x) = -g(x).$

4. Using property 1 again it follows that
$g(x)^T x = \lim_{\alpha \to 0} \left(\frac{f(x+\alpha x) - f(x)}{\alpha} \right) = \lim_{\alpha \to 0} \left(-\nu \frac{\log(1+\alpha)}{\alpha} \right) = -\nu.$

5. This follows trivially from 3 and 4.

6. This also follows trivially from 3 and 4.

□

Barriers for product cones are constructed from barriers for their constituent cones in the following way.

Lemma 4.3.2. *If $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2$ and f_1 and f_2 are self-concordant logarithmically-homogeneous barriers with parameters ν_1 and ν_2 then $f_1(x_1) + f_2(x_2)$ is a self-concordant logarithmically-homogeneous barrier for $\mathcal{K}$ with parameter $\nu_1 + \nu_2$.*

This fact is trivial and we leave it without proof.

The dual barrier function

The most efficient primal-dual interior-point methods for conic programming are defined in terms of barriers for the primal and dual cones that satisfy a particular property called *conjugacy*. If f is a proper convex function, then

$$f^\star(s) = -\inf_{x \in D_f} \left\{ f(x) + x^T s \right\}$$

is its convex conjugate. Conjugation of continuous convex functions is a symmetric operation in the sense that a continuous convex function is its *biconjugate* $f^{\star\star}(x) = f(x)$.

We wish to note that the definition we use of a Legendre transform is consistent with that used by Renegar in [48] but different from the one used by Nesterov and Nemirovski [43], where $\hat{f}(s)$ of $f(x)$ is defined as $\hat{f}(s) = \sup_{x \in D_f} \left\{ -f(x) + x^T s \right\}$. It is trivial to show that $\hat{f}(-s) = f^\star(s)$. However, the domain for $f^\star$ is the cone dual to $\mathcal{K}$, while the domain of $\hat{f}(s)$ is its anti-dual, namely $-\mathcal{K}^\star$. The difference between definitions is merely cosmetic and Renegar's definition is more useful for our purposes.

The conjugate pair of a function f is in some sense its dual. This characteristic will be important for deducing properties of the central path. We now state some of the properties of conjugate pairs and specializations of these for ν-logarithmically homogeneous barrier functions. We refer to [8, Lecture 3] and [43] for more details on the Legendre transformation.

Definition 4.3.2. Let $f(x)$ be a closed convex function with domain D_f and denote by

$$f^\star(s) = -\inf_{x \in D_f} \left\{ f(x) + x^T s \right\}$$

the function conjugate to $f(x)$.

Theorem 4.3.3. *Fenchel-Moreau The conjugation of convex and continuous functions is a symmetric operation in the sense that* $(f^\star)^\star = f$.

Since $-f^\star(s)$ is the infimum over a family of linear functions (in s), it is a concave function and $f^\star(s)$ is convex. However, when f is a strongly nondegenerate self-concordant ν-logarithmically homogeneous barrier for the proper cone $\mathcal{K}$ a stronger result holds, and in fact the function $f^\star$ will have the same properties and the same parameter ν, but will be a barrier for $\mathcal{K}^\star$.

Theorem 4.3.4. *If f is a ν-logarithmically homogeneous self-concordant barrier for the proper cone $\mathcal{K}$ then $f^\star$ is a ν-logarithmically homogeneous self-concordant barrier for the cone $\mathcal{K}^\star$.*

Proof. See [43, Theorem 2.4.4]. □

Finally we list the following properties that hold for conjugate pairs of ν-logarithmically homogeneous barriers.

Lemma 4.3.5. *Let f be a barrier for the cone $\mathcal{K}$ with parameter ν, and let $f^\star$ be the conjugate to f. Let g and $g^\star$ denote the gradient of f and $f^\star$ respectively, and let H and $H^\star$ denote the Hessians of f and $f^\star$ respectively.*

1. $g^\star(s) = \arg\min_{x \in D_f} \{f(x) + x^T s\}$
2. $g^\star(-g(x)) = x$
3. $H^\star(-g(x)) = H^{-1}(x)$
4. $f(-g^\star(x)) = -f^\star(x) - \nu$

Proof. See [48, Theorem 3.3.4]. □

4.4 Self-dual cones and self-scaled barriers

In [44] Nesterov and Todd analyze the specializations of interior-point methods to self-dual cones and define the concept of self-scaled barrier. These barriers, which exist only for self-dual cones, generalize some useful properties of the logarithmic barrier for the positive orthant. The definitions from this section have important algorithmic implications, one of them being that the search directions used are computed from smaller linear systems than when the cone is not self-dual and the barriers self-scaled.

Definition 4.4.1. A ν-logarithmically homogeneous self-concordant barrier f for a cone $\mathcal{K}$ that satisfies

$$H(y)x \in \operatorname{int} \mathcal{K}$$

and

$$f^\star(H(y)x) = f(x) - 2f(y) - \nu$$

for all y and $x \in \operatorname{int} \mathcal{K}$ is a self-scaled barrier.

In particular it can be shown that for every ordered pair $x, s \in \mathcal{K} \times \mathcal{K}$ there exists a *scaling point*, which is an element $w \in \operatorname{int} \mathcal{K}$ such that $H(w)x = s$ and $H(w)g(s) = g(x)$ [44, Theorem 3.2].

Self-concordant self-scaled barriers admit a stronger upper bound than self-concordant functions. The former can only be bounded within the Dikin ellipsoid. The latter can be bounded up to a constant of the distance to the boundary of the cone. This measure of distance is defined as follows [44, Section 4]: For a direction Δx and a point $x \in \operatorname{int} \mathcal{K}$ the function $\sigma_x(\Delta x)$ is defined as

$$\sigma_x(\Delta x) = \frac{1}{\sup\{\alpha \mid x - \alpha \Delta x \in \mathcal{K}\}}.$$

That is, the reciprocal to the largest step length that keeps the line from x along $-\Delta x$ in the cone. When the largest step size is very small then $\sigma_x(\Delta x)$ takes very large values, and conversely when the step size is large then $\sigma_x(\Delta x)$ takes small values.

With this measure of distance the bound, [44, Theorem 4.2] gives

$$f(x+\alpha\Delta x) \leq f(x)+\alpha g(x)^T\Delta x+\frac{\|\Delta x\|_x^2}{\sigma_x(-\Delta x)^2}\left(-\alpha\sigma_x(-\Delta x)-\ln(1-\alpha\sigma_x(-\Delta x)\right).$$

Observe that for this case the upper bound tends to infinity as $\alpha \to \frac{1}{\sigma_x(-\Delta x)}$ and not as $\alpha\|\Delta x\|_x \to 1$, which is the case for the self-concordant bound of Theorem 4.1.1. This together with the inequality $\sigma_x(-\Delta x) \leq \|\Delta x\|_x$ implies that the bound is valid in a larger set.

Chapter 5

Homogeneous primal-dual interior-point algorithms for general conic programming

In this chapter we describe the primal-dual potential reduction algorithms for general convex programming with the homogeneous embedding and prove worst-case iterations bounds. We also describe how Nesterov-Todd scaling points are used to define algorithms with the same worst-case iteration bounds while solving smaller linear systems at every iteration.

We introduce the concept of the barrier problem for the homogeneous embedding and the idea of the central path. We show that because the central path is the set of minimizers for the family of all barrier problems it can be defined as a set of nonlinear equations. We use the special structure of the self-dual problem to show that these nonlinear equations have a particularly simple form. We introduce a measure of centrality: a function that is positive on the feasible set and takes the value zero on the central path and prove some basic facts about it. We also introduce the concept of a potential function: A merit function that can be used to find the solution of a conic programming problem, and will be essential for proving the worst-case iteration bounds for the potential reduction algorithm.

The original variants of these algorithms were developed by Nesterov and described in [40]. Versions of these algorithms using the homogeneous embeddings are also described by Nesterov and Ye [45] and by Strum et al. [33]. The versions we describe here are variations on these algorithms.

This section deals with the homogeneous embeddings. Because the variables τ and κ make the notation cumbersome, we use the following definitions: the symbol $z = (y, x, \tau, s, \theta)$ represents the concatenation of all variables, so a sequence of iterates can be succinctly written as $\{z_k\}$. We redefine the symbol x to represent the concatenation of the primal variables $x = (x, \tau)$, while the

symbol $s = (s, \kappa)$ represents the concatenation of the dual variables. With this notation the linear constraints of the homogeneous embedding can be written as

$$G \begin{pmatrix} y \\ x \\ \theta \end{pmatrix} - \begin{pmatrix} 0 \\ s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\mu_0 \nu \end{pmatrix}, \tag{5.1}$$

where G is the skew symmetric matrix defined in (3.11).

We refer to the feasible set defined by (5.1) as L, so an iterate z is be linear feasible iff $z \in L$. Whenever we write $f(x)$ we mean the barrier of the cone for the new definition of x: $f(x) = \hat{f}(x) - \log(\tau)$, where $\hat{f}$ is the barrier for the cone $\mathcal{K}$. Similarly $f^\star(s) = \hat{f}^\star(s) - \log(\kappa)$, where $\hat{f}^\star$ is the barrier for $\mathcal{K}^\star$. We override the symbol $\mathcal{K}$ to denote the new cone $\mathcal{K} \times \mathbb{R}^+$. We also override the symbol $\mathcal{K}^\star$ to denote $\mathcal{K}^\star \times \mathbb{R}^+$ and the definition of ν to correspond with the new definition of the barrier, $\nu + 1$.

For future reference, the minimization problem for the homogeneous embedding problem is

$$\begin{aligned} &\text{minimize } \mu_0 \nu \theta \\ &\text{subject to } G \begin{pmatrix} y \\ x \\ \theta \end{pmatrix} - \begin{pmatrix} 0 \\ s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\mu_0 \nu \end{pmatrix} \\ &\qquad x \in \mathcal{K},\ s \in \mathcal{K}^\star. \end{aligned} \tag{HSD}$$

5.1 The barrier problems and the central path

The barrier problem arises from replacing the conic constraints by the barrier functions in order to define a linearly constrained convex optimization problem:

$$\begin{aligned} &\text{minimize } \frac{1}{\mu} \mu_0 \nu \theta + f(x) + f^\star(s) \\ &G \begin{pmatrix} y \\ x \\ \theta \end{pmatrix} - \begin{pmatrix} 0 \\ s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\mu_0 \nu \end{pmatrix}. \end{aligned} \tag{PD$_\mu$}$$

The barrier problem is parametrized by the *barrier parameter* $\mu > 0$ and weighs the relative importance of the barriers with respect to the objective. For each value of μ a unique minimizer $y(\mu), x(\mu), s(\mu), \theta(\mu)$ exists and at the minimizer $\theta(\mu) = \frac{\mu}{\mu_0}$ holds. Because each point in the central path is feasible, and as $\mu \to 0$, $\theta \to 0$ continuously, in the limit a solution with $\theta = 0$, $s \in \mathcal{K}^\star$ and $x \in \mathcal{K}$ is reached. Because the homogeneous embedding problem is self-dual, the solution has objective value 0 and this limiting point is the solution for the embedding problem.

Contrary to some sources in the literature we incorporate the barrier parameter into the linear objective and not into the barriers. This is consistent with the treatment in [48, 43, 45] and simplifies the arguments about potential

functions. Since we are interested in the solution of problems in the homogeneous embedding we present the particular instance of the barrier problem for the homogeneous embedding only.

We begin by showing that for any $\mu > 0$ the barrier problem is solvable.

Theorem 5.1.1. *Problem* (PD_μ) *is solvable.*

Proof. The Lagrangian for (PD_μ) is given by

$$\mathbf{L}(y,x,\theta,s,\lambda_y,\lambda_x,\lambda_\theta) = \frac{1}{\mu}\mu_0\nu\theta + f(x) + f^\star(s) - \begin{pmatrix}\lambda_y\\ \lambda_x\\ \lambda_\theta\end{pmatrix}^T \left(G\begin{pmatrix}y\\ x\\ \theta\end{pmatrix} - \begin{pmatrix}0\\ s\\ 0\end{pmatrix} + \begin{pmatrix}0\\ 0\\ \mu_0\nu\end{pmatrix}\right).$$

Since the Lagrangian is convex on the primal variables and tends to infinity as they approach the boundaries of the cones, any minimizer is in the interior of the cones. In that region the Lagrangian is differentiable and therefore it has a minimizer (which we denote $y^\star, x^\star, \theta^\star, s^\star$) iff the optimality equations

$$\begin{gathered} G^T\begin{pmatrix}-\lambda_y\\ -\lambda_x\\ -\lambda_\theta\end{pmatrix} + \begin{pmatrix}0\\ g(x^\star)\\ \frac{1}{\mu}\mu_0\nu\end{pmatrix} = \begin{pmatrix}0\\ 0\\ 0\end{pmatrix}, \\ \lambda_x + g^\star(s^\star) = 0 \end{gathered} \tag{5.2}$$

hold. When the Lagrangian has a finite minimizer the barrier problem is solvable. Therefore all that remains to do is show that there is a feasible point for (5.2), for then the Lagrangian has a finite minimizer.

Let y, x, θ, s be a primal feasible point, let $\hat{s} = -g(\frac{1}{\mu}x)$ and let $\hat{x} = -g^\star(\frac{1}{\mu}s)$. From ($\mathrm{PD}_\mu$) the relation

$$G^T\begin{pmatrix}-\frac{1}{\mu}y\\ -\frac{1}{\mu}x\\ -\frac{1}{\mu}\theta\end{pmatrix} + \begin{pmatrix}0\\ -\frac{1}{\mu}s\\ \frac{1}{\mu}\mu_0\nu\end{pmatrix} = \begin{pmatrix}0\\ 0\\ 0\end{pmatrix} \tag{5.3}$$

holds. From the properties of the gradients of conjugate pairs, $g(\hat{x}) = -\frac{1}{\mu}s$ and $g^\star(\hat{s}) = -\frac{1}{\mu}x$, and therefore the relations

$$\begin{gathered} G^T\begin{pmatrix}-\frac{1}{\mu}y\\ -\frac{1}{\mu}x\\ -\frac{1}{\mu}\theta\end{pmatrix} + \begin{pmatrix}0\\ g(\hat{x})\\ \frac{1}{\mu}\mu_0\nu\end{pmatrix} = \begin{pmatrix}0\\ 0\\ 0\end{pmatrix} \\ \lambda_x = \frac{1}{\mu}x = -g^\star(\hat{s}) \end{gathered} \tag{5.4}$$

hold. Observe that here $\hat{x}$ and $\hat{s}$ take the role of $x^\star$ and $s^\star$ in (5.2) and $\frac{1}{\mu}y$, $\frac{1}{\mu}x$, $\frac{1}{\mu}\theta$ take the role of $\lambda_y, \lambda_x, \lambda_\theta$. Therefore the point $\frac{1}{\mu}y$, $\frac{1}{\mu}x$, $\frac{1}{\mu}\theta, \hat{x}, \hat{s}$ is a feasible point for (5.2). □

Lemma 5.1.2. *For any $\mu > 0$ the minimizer of* (PD_μ) *is unique.*

Proof. Since by definition f and $f^\star$ are strictly convex, $x(\mu)$ and $s(\mu)$ are unique. Lemma 3.4.2 implies that $\theta(\mu)$ is also unique. Finally the uniqueness of $y(\mu)$ follows from the first linear equation $-A^T y + d_r\theta + \tau c = 0$ and the full row rank assumption for A. □

From the uniqueness of the minimizer for each value of μ we can define a map $\mu \to z(\mu) = (y(\mu), x(\mu), \tau(\mu), s(\mu), \kappa(\mu), \theta(\mu))$. It is not hard to show that the objective function in the barrier problem varies continuously with μ. From this observation it is simple to show that the map $\mu \to z(\mu)$ is continuous. We can therefore interpret the set of all minimizers $z(\mu)$ as a path in the variable space. This object is called the *central path*.

5.1.1 An alternative characterization of the central path

The central path is also the set of points $z \in L$ that satisfy the nonlinear equation $s + \mu g(x) = 0$. This alternative representation of the central path is the subject of Theorem 5.1.4 for which we first show the following lemma.

Lemma 5.1.3. *Let $z^\star$ be a minimizer of* (PD_μ)*. Then the minimizer of*

$$\begin{aligned} &\text{minimize} \quad \mu_0\nu\theta + f(x) + f^\star(s) \\ &G\begin{pmatrix} y \\ x \\ \theta \end{pmatrix} - \begin{pmatrix} 0 \\ s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\frac{1}{\mu}\mu_0\nu \end{pmatrix} \end{aligned} \tag{5.5}$$

is given by $\frac{1}{\mu}z^\star$.

Proof. It is simple to show that for any $z \in L$, the point $\frac{1}{\mu}z$ is feasible in (5.5). Furthermore if ψ denotes the objective function of (PD_μ), and $\hat{\psi}$ the objective function of (5.5), then $\psi(z) = \hat{\psi}(\frac{1}{\mu}z)$. For observe that

$$\psi(z) = \frac{1}{\mu}\mu_0\nu\theta + f(x) + f^\star(s) = \mu_0\nu\frac{\theta}{\mu} + f(\frac{1}{\mu}x) + f^\star(\frac{1}{\mu}s) - 2\nu\log(\mu) = \hat{\psi}(\frac{1}{\mu}z).$$

This implies that if $z^\star$ minimizes (PD_μ) then $\frac{1}{\mu}z^\star$ minimizes (5.5). □

Theorem 5.1.4. *The minimizer of* (PD_μ) *is uniquely defined by the equations*

$$\mu g(x) + s = 0 \tag{5.6a}$$

$$G\begin{pmatrix} y \\ x \\ \theta \end{pmatrix} - \begin{pmatrix} 0 \\ s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\mu_0\nu \end{pmatrix}. \tag{5.6b}$$

Proof. Since the problem (PD_μ) is bounded and feasible, it is solvable and therefore there exist Lagrange multipliers $\lambda_y, \lambda_x,\ \lambda_\theta$ for which the first-order optimality conditions

$$G \begin{pmatrix} y \\ x \\ \theta \end{pmatrix} - \begin{pmatrix} 0 \\ s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\mu_0\nu \end{pmatrix}, \tag{5.7a}$$

$$G^T \begin{pmatrix} -\lambda_y \\ -\lambda_x \\ -\lambda_\theta \end{pmatrix} + \begin{pmatrix} 0 \\ g(x^\star) \\ \frac{1}{\mu}\mu_0\nu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{5.7b}$$

$$\lambda_x + g^\star(s) = 0 \tag{5.7c}$$

hold. Using the skew symmetry of G and the properties of the gradients of conjugate pairs of functions, and defining $\lambda_s = -g(x)$, we can write (5.7) as

$$\begin{gathered} G^T \begin{pmatrix} -y \\ -x \\ -\theta \end{pmatrix} + \begin{pmatrix} 0 \\ g(\lambda_x) \\ \mu_0\nu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \\ G \begin{pmatrix} \lambda_y \\ \lambda_x \\ \lambda_\theta \end{pmatrix} - \begin{pmatrix} 0 \\ \lambda_s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\frac{1}{\mu}\mu_0\nu \end{pmatrix}, \\ x + g^\star(\lambda_s) = 0. \end{gathered} \tag{5.8}$$

Equations (5.8) are the optimality conditions for the minimization problem (5.5) of Lemma (5.1.3) and therefore $\lambda_y, \lambda_x, \lambda_s, \lambda_\theta$ solve (5.5). Using Lemma 5.1.3 we can conclude that if $z^\star$ minimizes (PD_μ) then $\frac{1}{\mu}z^\star$ minimizes (5.5) and the Lagrange multipliers of (5.7) are equal to $\frac{1}{\mu}z^\star$. Using (5.7c) we conclude that $x^\star = -\mu g^\star(s^\star)$, that $s^\star = \mu g(x^\star)$, and that at the minimizer (5.6) holds.

To show the converse, assume that at z equations (5.6) hold; then (5.6a) can be written as

$$G^T \begin{pmatrix} -\frac{1}{\mu}y \\ -\frac{1}{\mu}x \\ -\frac{1}{\mu}\theta \end{pmatrix} + \begin{pmatrix} 0 \\ g(x) \\ \frac{1}{\mu}\mu_0\nu \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

and (5.6b) as

$$\frac{1}{\mu}x + g^\star(s) = 0.$$

Then, z together with $\lambda_x = \frac{1}{\mu}x, y = \frac{1}{\mu}\lambda_y, \theta = \frac{1}{\mu}\lambda_\theta$ satisfy (5.7) and therefore z minimizes (PD_μ). □

5.2 Potential reduction algorithms for conic programming problems

Since the point $x(\mu), y(\mu), s(\mu)$ on the central path is the minimizer of

$$\begin{aligned} &\text{minimize } \frac{\mu_0}{\mu}\nu\theta + f(x) + f^\star(s) \\ &G\begin{pmatrix} y \\ x \\ \theta \end{pmatrix} - \begin{pmatrix} 0 \\ \underline{s} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\mu_0\nu \end{pmatrix}, \end{aligned} \tag{PD$_\mu$}$$

and this problem is convex with self-concordant objective, Newton's method is efficient at finding points close to the central path.

This suggests the following strategy: For a fixed μ_k use Newton's method to approximately minimize (PD$_\mu$) in order to find a point z_k close to the central path. Then, reduce μ_k to μ_{k+1} and use Newton's method starting from z_k to compute a new iterate z_{k+1} that approximately minimizes (PD$_\mu$), and so on. This scheme forms a sequence that tracks the central path to the solution of the conic programming problem.

The question of how to choose μ at each iteration remains. Potential reduction methods set $\mu = \frac{x^T s}{\rho}$ at every iteration, where $\rho > \nu$ is a constant chosen appropriately. At iteration k they solve for the Newton direction of the barrier problem with $\mu = \frac{x_k^T s_k}{\rho}$ and choose a step size by doing a linesearch to reduce a merit function (the potential function). Before we introduce the potential function we argue why $\rho > \nu$ is necessary.

Lemma 5.2.1. *For any $\mu > 0$ the point on the central path satisfies $\mu = \frac{x^T s}{\nu}$.*

Proof. Since at the central path $s + \mu g(x) = 0$ and f is a ν-logarithmically homogeneous barrier, claim (4.3.1) gives $x^T s = \mu x^T g(x) = \mu\nu$. □

If the iterate is on the central path and $\mu = \frac{x_k^T s_k}{\nu}$ is chosen, then the barrier problem will be at its minimizer and the Newton direction will have length zero. This would cause the method to stall. A choice of $\rho > \nu$ implies that the barrier problem is never fully solved and that at each iteration the value of $x^T s$ is reduced. This in turn implies that $\theta \to 0$. A choice of $\rho < \nu$ is contradictory, for this choice would result in the Newton direction for a barrier problem with a larger μ instead of a smaller one.

5.2.1 Newton direction for the barrier problem

The Newton direction for the barrier problem has the form

$$\begin{pmatrix} \nabla^2 F(z) & B^T \\ B & \end{pmatrix}\begin{pmatrix} \Delta z \\ \lambda_z \end{pmatrix} = \begin{pmatrix} -\nabla F(z) \\ 0 \end{pmatrix}, \tag{5.9}$$

where B is the matrix that encodes the linear equality constraints in (PD_μ):

$$B = \begin{pmatrix} & A & -b & p_r & & \\ -A^T & & c & d_r & -I & \\ b^T & -c^T & & g_r & & -1 \\ -p_r^T & -d_r^T & -g_r^T & & & \end{pmatrix}, \tag{5.10}$$

and the function $F(z)$ is the objective of the barrier problem, namely the self-concordant convex function

$$F(z) = \frac{\mu_0}{\mu}\nu\theta + f(x) + f^\star(s).$$

For future reference it is useful to expand (5.9) into the following systems of equations. The first

$$\begin{pmatrix} 0 \\ H(x)\Delta x \\ 0 \end{pmatrix} + G^T \begin{pmatrix} \bar{\lambda_y} \\ \bar{\lambda_x} \\ \bar{\lambda_\theta} \end{pmatrix} = \begin{pmatrix} 0 \\ -g^\star(s) \\ -\frac{\mu_0}{\mu}\nu \end{pmatrix} \tag{5.11}$$

corresponds to the primal barrier, the second

$$H^\star(s)\Delta s - \bar{\lambda_x} = -g^\star(s) \tag{5.12}$$

corresponds to the dual barrier, and the third

$$G \begin{pmatrix} \Delta y \\ \Delta x \\ \Delta\theta \end{pmatrix} - \begin{pmatrix} 0 \\ \Delta s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{5.13}$$

enforces linear feasibility.

5.2.2 The potential function

Potential functions are a useful tool for analyzing conic programming algorithms. With them it can be shown that a potential reduction primal-dual conic programming algorithm achieves a precision of ε in $\mathcal{O}(\sqrt{\nu}\log(1/\varepsilon))$ iterations. This is the state of the art complexity bound for general conic programming.

The usefulness of potential functions is not limited to theoretical aspects; potential reduction algorithms have proven to be robust and computationally efficient. Their merit lies in the fact that potential functions define a very principled way to choose a step length, so that the next iterate will achieve a sufficient reduction in the complementarity while staying centered enough.

The potential function we use in this work was first presented for linear programming by Ye [54] and then generalized to conic programming by Nesterov [43]. For a more detailed explanation of potential reduction in the context of general conic programming see [40].

We now introduce the potential function Ψ and the *functional proximity* measure Ω, and we visit some of their properties and those of a modified Newton

method as applied to the reduction of Ψ. This will lay the foundation for the presentation of the standard computational complexity results on potential reduction methods.

Define

$$\Psi(x,s) = \rho\log(x^T s) + f(x) + f^\star(s) - \nu\log(\nu) + \nu, \tag{5.14}$$

where $\rho > \nu$ is a scalar, $f(x)$ is the barrier for the primal cone, and $f^\star(s)$ is the *conjugate* barrier for the dual cone.

Observe that if $z_k \in L$ is a sequence that approaches a sub-optimal limit on the boundary of the cones, the barrier term will tend to infinity and the complementarity term $\rho\log(x^T s)$ will be bounded below (otherwise the complementarity tends to zero, contradicting the sub-optimality of the limit), and therefore Ψ will tend to infinity. On the other hand if the iterate approaches an optimal point, the term $\rho\log(x^T s)$ will tend to $-\infty$, dominating the effect of the barriers, and Ψ will tend to $-\infty$. Potential reduction algorithms work by following Ψ to $-\infty$ to find a solution to the problem.

The functional proximity measure $\Omega(x,s) : \mathcal{K} \times \mathcal{K}^\star \to \mathbb{R}$ defined by

$$\Omega(x,s) = \nu\log(x^T s) + f(x) + f^\star(s) - \nu\log(\nu) + \nu \tag{5.15}$$

is a useful way to evaluate the distance from a point to the central path. The function Ω is positive in the feasible set and $\Omega(x,s) = 0$ iff the argument is on the central path.

Lemma 5.2.2. *The function $\Omega(x,s) \geq 0$ and $\Omega(x,s) = 0$ iff $s + \mu g(x) = 0$ with $\mu = \frac{x^T s}{\nu}$.*

Proof. Let $\mu = \frac{x^T s}{\nu}$. From the definition of the conjugate function we have that

$$\begin{aligned} -f^\star\left(\frac{s}{\mu}\right) &= \inf_x \left\{ \frac{x^T s}{\mu} + f(x) \right\} \\ &\leq \frac{x^T s}{\mu} + f(x) \end{aligned}$$

and then

$$\begin{aligned} 0 &\leq f^\star(\frac{s}{\mu}) + \frac{x^T s}{\mu} + f(x) \\ 0 &\leq \nu\log(\mu) + f^\star(s) + f(x) + \nu \\ 0 &\leq \nu\log(x^T s) + f^\star(s) + f(x) - \nu\log(\nu) + \nu \\ 0 &\leq \Omega(x,s). \end{aligned}$$

On the other hand if $s + \mu g(x) = 0$ then

$$\begin{aligned} \Omega(x,s) &= \Omega(x, -\mu g(x)) \\ &= \nu\log(\mu\nu) + f^\star(-\mu g(x)) + f(x) - \nu\log(\nu) + \nu \\ &= \nu\log(\mu) + \nu\log(\nu) - \nu\log(\mu) - f(x) - \nu + f(x) - \nu\log(\nu) + \nu \\ &= 0. \end{aligned}$$

For the converse, if $\Omega(x,s) = \nu\log(x^Ts) + f^\star(s) + f(x) - \nu\log(\nu) + \nu = 0$ then the properties of ν-logarithmically homogeneous barriers imply $-f^\star(\frac{s}{\mu}) = f(x) + \frac{x^Ts}{\mu} \geq \inf_{\hat{x}}\left\{f(\hat{x}) + \frac{\hat{x}^Ts}{\mu}\right\} = f^\star\left(\frac{s}{\mu}\right)$ so $\hat{x} = x$ minimizes $f(\hat{x}) + \frac{\hat{x}^Ts}{\mu}$, which in turn implies that $g(x) + \frac{s}{\mu} = 0$. □

Now we can establish some results about the function Ψ in (5.14).

Lemma 5.2.3. *The function Ψ is unbounded below in the feasible set.*

Proof. Using (5.14) and (5.15) write $\Psi(x,s) = (\rho-\nu)\log(x^Ts) + \Omega(x,s)$, therefore at the central path we then have $\Psi(x(\mu),s(\mu)) = (\rho-\nu)\log(\mu\nu)$ and therefore $\Psi \to -\infty$ as $\mu \to 0$. □

A converse result also holds because Ψ induces an upper bound on the complementarity. Thus, reducing Ψ to $-\infty$ implies that x^Ts will tend to zero.

Lemma 5.2.4. *If $\Psi(x,s) \leq (\rho-\nu)\log(\varepsilon)$ for some feasible x, s then $x^Ts \leq \varepsilon$.*

Proof. Since $\Psi(x,s) = (\rho-\nu)\log(x^Ts) + \Omega(x,s)$, the bound $(\rho-\nu)\log(x^Ts) \leq \Psi(x,s)$ holds. Therefore $\Psi(x,s) \leq (\rho-\nu)\log(\varepsilon)$ implies that $(\rho-\nu)\log(x^Ts) \leq (\rho-\nu)\log(\varepsilon)$, which in turn implies $x^Ts \leq \varepsilon$. □

The following is a rephrasing of the previous result that is useful to analyze the computational complexity of the potential reduction algorithms.

Lemma 5.2.5. *Any algorithm that produces a sequence of feasible iterates $\{x_k, s_k\}$ such that $\Psi(x_{k+1}, s_{k+1}) < \Psi(x_k, s_k) - \delta$ for some $\delta > 0$, will converge to an ε accurate iterate in $\mathcal{O}\left((\rho-\nu)\log(\frac{1}{\varepsilon})\right)$ iterations.*

Proof. Let x_0, s_0 be some starting iterate and denote $\Psi_0 = \Psi(x_0, s_0)$. Using the bound $(\rho-\nu)\log(x^Ts) \leq (\rho-\nu)\log(x^Ts) + \Omega(x,s) = \Psi(x,s)$, we get that

$$(\rho-\nu)\log(x_k^Ts_k) \leq \Psi_0 - \delta k,$$

which holds iff

$$\frac{\Psi(x_0,s_0)}{\delta} + \frac{\rho-\nu}{\delta}\log(1/\varepsilon) \leq k.$$

□

5.3 A primal-dual potential reduction algorithm

The following is a potential reduction algorithm with a backtracking linesearch. The search direction for the potential reduction algorithm is the solution to the system of linear equations

$$\begin{pmatrix} 0 \\ H(x)\Delta x \\ 0 \end{pmatrix} + G^T\begin{pmatrix}\lambda_y \\ \lambda_x \\ \lambda_\theta\end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{\rho}{x^Ts}s - g(x) \\ 0 \end{pmatrix}, \tag{5.16}$$

together with

$$H^\star(s)\Delta s - \lambda_x = -\frac{\rho}{x^T s}x - g^\star(s) \tag{5.17}$$

and

$$G\begin{pmatrix}\Delta y\\ \Delta x\\ \Delta\theta\end{pmatrix} - \begin{pmatrix}0\\ \Delta s\\ 0\end{pmatrix} = \begin{pmatrix}0\\ 0\\ 0\end{pmatrix}. \tag{5.18}$$

The above equations can be written in the form

$$\begin{pmatrix}\nabla^2 F(z) & B^T\\ B & 0\end{pmatrix}\begin{pmatrix}\Delta z\\ \lambda_z\end{pmatrix} = \begin{pmatrix}-\nabla_z\Psi(z)\\ 0\end{pmatrix}, \tag{5.19}$$

where B is the matrix that encodes all the linear constraints:

$$B = \begin{pmatrix} & A & -b & p_r & & \\ -A^T & & c & d_r & -I & \\ b^T & -c^T & & g_r & & -1\\ -p_r^T & -d_r^T & -g_r^T & & & \end{pmatrix}. \tag{5.20}$$

And the function $F(z)$ is the objective of the barrier problem, namely the self-concordant convex function

$$F(z) = \frac{\mu_0}{\mu}\nu\theta + f(x) + f^\star(s).$$

The matrix $\nabla^2 F(z)$ is the Hessian of $F(z)$, and $-\nabla_w\Psi$ is the negative gradient of the potential function.

It is clear that this is not the Newton direction for the potential function Ψ, for the Newton direction is the solution to

$$\begin{pmatrix}\nabla^2\Psi(z) & B^T\\ B & 0\end{pmatrix}\begin{pmatrix}\Delta z\\ \lambda_z\end{pmatrix} = \begin{pmatrix}-\nabla_z\Psi(z)\\ 0\end{pmatrix}.$$

However, it is easy to see that the potential reduction direction Δz coincides with the Newton direction for the barrier problem (PD_μ) with barrier parameter $\mu = x^T s/\rho$.

Lemma 5.3.1. *The potential reduction direction is the Newton direction for the barrier problem with parameter* $\mu = \frac{x^T s}{\rho}$.

Proof. Let $\lambda_y, \lambda_x, \lambda_\theta$ be the Lagrange multipliers in (5.16) and (5.17). If we choose the new Lagrange multipliers

$$\begin{pmatrix}\bar{\lambda_y}\\ \bar{\lambda_x}\\ \bar{\lambda_\theta}\end{pmatrix} = \begin{pmatrix}\lambda_y\\ \lambda_x\\ \lambda_\theta\end{pmatrix} - \frac{\rho}{x^T s}\begin{pmatrix}y\\ x\\ \theta\end{pmatrix},$$

then

$$H^\star(s)\Delta s - \bar{\lambda_x} = -g^\star(s),$$

and

$$G^T \begin{pmatrix} \bar{\lambda_y} \\ \bar{\lambda_x} \\ \bar{\lambda_\theta} \end{pmatrix} = G^T \begin{pmatrix} \lambda_y \\ \lambda_x \\ \lambda_\theta \end{pmatrix} + \frac{\rho}{x^T s} G \begin{pmatrix} y \\ x \\ \theta \end{pmatrix} = G^T \begin{pmatrix} \lambda_y \\ \lambda_x \\ \lambda_\theta \end{pmatrix} + \frac{\rho}{x^T s} \begin{pmatrix} 0 \\ s \\ -\mu_0 \nu \end{pmatrix},$$

where in the second equality we used the skew-symmetry of G and in the third the feasibility of w. Taken together the above imply that

$$\begin{pmatrix} 0 \\ H(x)\Delta x \\ 0 \end{pmatrix} + G^T \begin{pmatrix} \bar{\lambda_y} \\ \bar{\lambda_x} \\ \bar{\lambda_\theta} \end{pmatrix} = \begin{pmatrix} 0 \\ -g^\star(s) \\ -\frac{\rho}{x^T s}\mu_0 \nu \end{pmatrix},$$

$$H^\star(s)\Delta s - \bar{\lambda_x} = -g^\star(s),$$

and

$$G \begin{pmatrix} \Delta y \\ \Delta x \\ \Delta \theta \end{pmatrix} - \begin{pmatrix} 0 \\ \Delta s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

which define the Newton direction for the barrier problem (5.11)–(5.13). □

Algorithm 1 Potential reduction algorithm for conic programming

Given w_0, $\eta \in (0, \frac{1}{2})$ and $\beta \in (0, 1)$
while $x^T s > \varepsilon$ **do**
  Solve for the search direction Δw
  Set $\lambda \leftarrow \left(\|\Delta x\|_x^2 + \|\Delta s\|_s^2 \right)^{\frac{1}{2}}$
  while $\Psi(w_k + \alpha \Delta w) > \Psi(w_k) + \eta \alpha \lambda^2$ **do** ▷ Backtracking Linesearch
    $\alpha = \beta \alpha$
  end while
  $w_{k+1} \leftarrow w_k + \alpha \Delta w$
  $k \leftarrow k + 1$
end while

We now show the worst-case iteration bounds for the primal-dual potential reduction of Algorithm 1. We need to show that Ψ admits the upper bound

$$\Psi(z + \alpha \Delta z) \leq \Psi(z) - \alpha(\lambda^2 + \lambda) - \log(1 - \alpha\lambda),$$

where in this case $\lambda = \left(\|\Delta x\|_x^2 + \|\Delta s\|_s \right)^{1/2}$, that the backtracking linesearch selects a step size $\alpha \geq \frac{\beta}{1+\lambda}$, and that the decrease in $\Psi(z)$ along Δz is at least $\eta\beta\frac{\lambda^2}{1+\lambda}$. Finally we use the bounds derived in Chapter 4 to show that the choice $\rho = \nu + \sqrt{\nu}$ implies that λ is bounded below by $\lambda > 0.5$, which implies a worst-case computational complexity bound for Algorithm 1 of $\mathcal{O}\left(\sqrt{\nu}\log(\frac{1}{\varepsilon})\right)$.

Claim 5.3.2. *Along the potential reduction directions the function Ψ admits the upper bound $\Psi(x + \alpha\Delta x, s + \alpha\Delta s) \leq \Psi(x, s) - \alpha(\lambda^2 + \lambda) - \log(1 - \alpha\lambda)$.*

Proof. From (5.19), it is clear that $\Delta z^T \nabla^2 F(z) \Delta z = -\Delta z \nabla \Psi(z)$, which implies

$$\|\Delta x\|_x^2 + \|\Delta s\|_s^2 = -\frac{\rho}{x^T s}\left(s^T \Delta x + x^T \Delta s\right) - g(x)^T \Delta x - g^\star(s)^T \Delta s. \tag{5.21}$$

Since

$$\begin{aligned}\Psi(x+\alpha\Delta x, s+\alpha\Delta s) - \Psi(x,s) = \rho \log\left(\frac{x^T s + \alpha\left(s^T\Delta x + x^T \Delta s\right)}{x^T s}\right) \\ + f(x+\alpha\Delta x) - f(x) + f^\star(s+\alpha\Delta s) - f^\star(s),\end{aligned} \tag{5.22}$$

and $f(x) + f^\star(s)$ is a self-concordant function, we have

$$\begin{aligned}\Psi(x+\alpha\Delta x, s+\alpha\Delta s) - \Psi(x,s) \leq \rho \log\left(\frac{x^T s + \alpha\left(s^T\Delta x + x^T \Delta s\right)}{x^T s}\right) \\ + \alpha\left(g(x)^T\Delta x + g^\star(s)^T \Delta s\right) - \alpha\lambda - \log\left(1-\alpha\lambda\right).\end{aligned} \tag{5.23}$$

Using the bound $\log(1+x) \leq x$, and using equality (5.21) we get the bound

$$\begin{aligned}\Psi(x+\alpha\Delta x, s+\alpha\Delta s) - \Psi(x,s) &\leq \frac{\rho}{x^T s}\left(\alpha\left(s^T\Delta x + x^T\Delta s\right)\right) \\ -\frac{\rho}{x^T s}\left(\alpha\left(s^T\Delta x + x^T\Delta s\right)\right) &- \alpha\lambda^2 - \alpha\lambda - \log\left(1-\alpha\lambda\right) \\ &= -\alpha(\lambda^2+\lambda) - \log(1-\alpha\lambda).\end{aligned} \tag{5.24}$$

□

Claim 5.3.3. *Each iteration of Algorithm 1 will reduce Ψ by at least $\eta\beta\frac{\lambda^2}{1+\lambda}$.*

Proof. It suffices to prove that the step size $\alpha = \frac{1}{1+\lambda}$ is feasible and satisfies the stopping condition. Then the smallest possible step size the algorithm will take is $\beta\alpha$ and the result follows.

If $\alpha = \frac{1}{\lambda+1}$ then the distance $\|w - (w+\alpha\Delta w)\|_w$ satisfies

$$\begin{aligned}\|w-(w+\alpha\Delta w)\|_w &= \left(\|x+\alpha\Delta x - x\|_x^2 + \|s+\alpha\Delta s - s\|_s^2\right)^{1/2} \\ &= \alpha\left(\|\Delta x\|_x^2 + \|\Delta s\|_s^2\right)^{1/2} \\ &= \alpha\lambda = \frac{\lambda}{1+\lambda} < 1.\end{aligned}$$

Hence any step size smaller than $\frac{1}{1+\lambda}$ keeps the next iterate in the Dikin ellipsoid centered at x, s and is therefore feasible. From Lemma 5.3.2 the damped Newton step achieves the decrease

$$\Psi(x+\alpha\Delta x, s+\alpha\Delta s) - \Psi(x,s) \leq -\alpha(\lambda^2+\lambda) - \log(1-\alpha\lambda) \tag{5.25}$$

$$= -\lambda + \log(1+\lambda) \tag{5.26}$$

$$\leq -\frac{\lambda^2}{1+\lambda} \tag{5.27}$$

$$< -\eta\frac{\lambda^2}{1+\lambda} = -\alpha\eta\lambda^2. \tag{5.28}$$

This implies that the backtracking linesearch will choose $\frac{\beta}{1+\lambda} < \alpha$, and therefore

$$\Psi(x + \alpha\Delta x, s + \alpha\Delta s) - \Psi(x, s) \leq -\frac{\beta\eta\lambda^2}{1+\lambda}.$$

□

To obtain the worst-case complexity bounds we show that for the choice $\rho = \nu + \sqrt{\nu}$, at every iteration, λ is bounded below by a constant. Then from Lemma 5.2.5 it follows that the worst-case complexity is $\mathcal{O}\left(\sqrt{\nu}\log\left(\frac{1}{\varepsilon}\right)\right)$.

Theorem 5.3.4. *If $\rho = \nu + \sqrt{\nu}$ then λ is bounded away from zero by a constant.*

Proof. Assume that for some iterate $\lambda < 0.5$, for if not there is nothing to prove.

Denote by $\phi(x, s, \theta) = \frac{1}{\mu}\mu_0\nu\theta + f(x) + f^\star(s)$ the objective function for the barrier problem with parameter μ. Denote by $x^\star, s^\star$ the minimizer of ϕ, i.e. the point on the central path with parameter μ.

Since, by Lemma 5.2.5, the potential reduction direction is the Newton step for the barrier problem with parameter $\mu = \frac{x^T s}{\rho}$, the bound $\phi(x, s, \theta) - \phi(x^\star, s^\star, \theta^\star) \leq \omega^\star(\lambda)$ holds. Therefore a lower bound for $\phi(x, s, \theta) - \phi(x^\star, s^\star, \theta^\star)$ implies a lower bound for $\omega^\star(\lambda)$.

We can evaluate $\phi(x, s, \theta) - \phi(x^\star, s^\star, \theta^\star)$. Observe that at the optimum,

$$\begin{aligned}\phi(w^\star) &= \frac{1}{\mu}\mu_0\nu\theta^\star + f(x^\star) + f^\star(s^\star)\\ &= \nu\log(\nu) - \nu\log({x^\star}^T s^\star)\\ &= \nu\log(\nu) - \nu\log(\mu\nu)\\ &= \nu\log(\nu) - \nu\log(x^T s) - \nu\log(\nu/\rho),\end{aligned} \tag{5.29}$$

where the second equality follows from the definition of Ω and the fact that $x^\star, s^\star$ is on the central path. The third follows from ${x^\star}^T s^\star = \mu\nu$ for any point on the central path, and the last from the choice $\mu = x^T s/\rho$.

The objective function at (x, s, θ) takes the value

$$\phi(x, s, \theta) = \frac{1}{\mu}\mu_0\nu\theta + f(x) + f^\star(s) = \rho + f(x) + f^\star(s),$$

and therefore

$$\begin{aligned}\phi(x, s, \theta) - \phi(x^\star, s^\star, \theta^\star) &= \rho + f(x) + f^\star(s) - \nu\log(\nu) + \nu\log(x^T s) + \nu\log(\nu/\rho)\\ &= \rho + f(x) + f^\star(s) + \nu\log(x^T s) - \nu\log(\rho)\\ &= \rho - \nu + \nu\log(\nu) - \nu\log(\rho) + \Omega(x, s)\end{aligned} \tag{5.30}$$

Since $\Omega > 0$, we have $\rho - \nu + \nu\log(\frac{\nu}{\rho}) \leq \phi(w) - \phi(w^\star) \leq \omega^\star(\lambda)$, and a constant lower bound for $\rho - \nu + \nu\log(\frac{\nu}{\rho})$ will imply a constant lower bound for λ. If we let $\rho = \nu + \sqrt{\nu}$ then $\rho - \nu + \nu\log(\frac{\nu}{\rho}) = \sqrt{\nu} - \nu\log\left(1 + \frac{1}{\sqrt{\nu}}\right)$. The

function $\nu \to \sqrt{\nu} - \nu \log\left(1 + \frac{1}{\sqrt{\nu}}\right)$ is monotonically increasing, at $\nu = 1$ takes the value $0 < 1 - \log(2)$, and as $\nu \to \infty$ approaches the value 0.5. Therefore the choice $\rho = \nu + \sqrt{\nu}$ implies that $0.3 < 1 - \log(2) \leq -\lambda - \log(1 - \lambda)$; for example if $\lambda < 0.5$ then $-\lambda - \log(1-\lambda) > -0.5 - \log(0.5) \approx 0.19$. And therefore $\lambda > 0.5$. □

5.4 Reducing the system size using the Nesterov-Todd scaling point

One of the algorithmic advantages of symmetric cone problems over unsymmetric cone programming problems stems from the existence of Nesterov-Todd scaling points [44]. With these scaling points, search directions that are similar to the potential reduction directions can be calculated from linear systems of half the size.

Recall that for the potential reduction algorithm for general conic programming, the search directions are defined as the solution to the equations

$$\begin{pmatrix} \nabla^2 F(z) & B^T \\ B & \end{pmatrix} \begin{pmatrix} \Delta z \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla\psi(z) \\ 0 \end{pmatrix},$$

where F is the objective function of the barrier problem, B encodes all the linear constraints, and $\Psi(z)$ is the potential function. Here F is interpreted as a function of all the variables, in the order $z = (y, x, \theta, s)$, and the matrix $\nabla^2 F$ is given by

$$\nabla^2 F(z) = \begin{pmatrix} 0 & & & \\ & H(x) & & \\ & & 0 & \\ & & & H^\star(s) \end{pmatrix}.$$

We will refer to these directions as the potential reduction directions.

Recall that the Nesterov-Todd scaling point for (x, s) satisfies the relations $H(w)x = s$ and $H(w)g^\star(s) = g(x)$. Using the properties of conjugate barriers we have that $H^\star(-g(w)) = H^{-1}(w)$ for any $w \in \operatorname{int} \mathcal{K}$. Therefore, if w is the Nesterov-Todd scaling point for (x, s), then the point $w^\star = -g(w)$ satisfies $H^\star(w^\star)s = x$ and $H^\star(w^\star)g(x) = g^\star(s)$.

Denote by $\nabla^2 \tilde{F}$ the Hessian of F evaluated at the point $x = w$, $s = w^\star$:

$$\nabla^2 \tilde{F} = \begin{pmatrix} 0 & & & \\ & H(w) & & \\ & & 0 & \\ & & & H^\star(w^\star) \end{pmatrix}.$$

The symmetric potential reduction directions are the solution to

$$\begin{pmatrix} \nabla^2 \tilde{F} & B^T \\ B & \end{pmatrix} \begin{pmatrix} \Delta z \\ \lambda \end{pmatrix} = \begin{pmatrix} -\nabla\psi(z) \\ 0 \end{pmatrix}, \tag{5.31}$$

where the system matrix differs from the potential reduction directions in the point at which $\nabla^2 F$ is evaluated. Observe that these directions are descent directions for the potential function, since $\Delta z^T \nabla \Psi = -\Delta z^T (\nabla^2 \tilde{F}) \Delta z \leq 0$ because $\nabla^2 \tilde{F}$ is positive semidefinite.

We now show that these directions can be calculated from the smaller system

$$\begin{aligned} H(w)\Delta x + \Delta s &= -\frac{\rho}{x^T s} s - g(x), \\ B\Delta z &= 0. \end{aligned} \tag{5.32}$$

Theorem 5.4.1. *The solution Δz to system* (5.32) *is identical to the Δz component of the solution to* (5.31).

Proof. First observe that from $H(w)\Delta x + \Delta s = -\frac{x^T s}{\rho} s - g(x)$ and the properties of the Nesterov-Todd scaling point $H^{-1}(w)\Delta s + \Delta z = -\frac{\rho}{x^T s} x - g^\star(s)$, and $H^\star(w^\star)\Delta s + \Delta z = -\frac{\rho}{x^T s} x - g^\star(s)$ hold.

Observe that the second equation of (5.32) is identical to the second equation of (5.31). Therefore we need to show that there exists a λ such that $\nabla^2 \tilde{F} \Delta z + B^T \lambda = -\nabla \Psi$. Write

$$B\Delta z = G \begin{pmatrix} \Delta y \\ \Delta x \\ \Delta \theta \end{pmatrix} - \begin{pmatrix} 0 \\ \Delta s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

where G is the appropriate skew-symmetric matrix. Let

$$\lambda = \begin{pmatrix} \lambda_y \\ \lambda_x \\ \lambda_\theta \end{pmatrix} = \begin{pmatrix} -\Delta y \\ -\Delta x \\ -\Delta \theta \end{pmatrix},$$

where $\Delta x, \Delta y$ are the components of Δz; then

$$B^T \lambda = \begin{pmatrix} G^T \begin{pmatrix} -\Delta y \\ -\Delta x \\ -\Delta \theta \end{pmatrix} \\ \Delta x \end{pmatrix} = \begin{pmatrix} G \begin{pmatrix} \Delta y \\ \Delta x \\ \Delta \theta \end{pmatrix} \\ \Delta x \end{pmatrix} = \begin{pmatrix} 0 \\ \Delta s \\ 0 \\ \Delta x \end{pmatrix}.$$

Finally observe that

$$\nabla^2 \tilde{F} \Delta z + B^T \lambda = \begin{pmatrix} 0 \\ H(w)\Delta x + \Delta s \\ 0 \\ H^\star(w^\star)\Delta s + \Delta x \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{\rho}{x^T s} s + g(x) \\ 0 \\ -\frac{\rho}{x^T s} x + g^\star(s) \end{pmatrix} = -\nabla \Psi(z).$$

□

Nesterov and Todd show that with these directions the potential function can be reduced by a constant at every iteration and therefore the same computational complexity results hold. We do not prove all the elements but sketch the proof from [44].

Using the equality derived above, it is easy to see that

$$\|\Delta x\|_w^2 + \|\Delta s\|_{H^\star(w^\star)}^2 = -\frac{\rho}{x^T s}\left(s^T\Delta x + x^T\Delta s\right) - \Delta x^T g(x) - \Delta s^T g^\star(s).$$

From the bounds for self-scaled barriers (4.4) we can derive the bound for the change in the value of Ψ along Δz:

$$\begin{aligned}\Psi(z+\alpha\Delta z) - \Psi(z) &= \rho\log\frac{s^T\Delta x + x^T\Delta s}{x^T s} + f(x+\alpha\Delta x) + f^\star(s+\alpha\Delta s)\\ &\le \frac{\rho}{x^T s}\left(s^T\Delta x + x^T\Delta s\right) + \alpha\Delta x^T g(x) + \Delta s^T g^\star(s)\\ &+ \frac{\sigma_x(w)^2\left\|\Delta x\right\|_x^2}{\sigma_x^2(\Delta x)}\left(-\alpha\sigma_x(\Delta x) - \log\left(1-\alpha\sigma_x(\Delta x)\right)\right)\\ &+ \frac{\sigma_s^\star(w)^2\left\|\Delta s\right\|_{H^\star(s)}^2}{\sigma^\star{}_s^2(\Delta s)}\left(-\alpha\sigma_s(\Delta s) - \log\left(1-\alpha\sigma_s^\star(\Delta s)\right)\right).\end{aligned} \tag{5.33}$$

By letting $\bar\sigma = \max\left\{\sigma_s^\star(\Delta s), \sigma_x(\Delta x)\right\}$, denoting by $\gamma^2 = \|\Delta x\|_w^2 + \|\Delta s\|_{H^\star(w^\star)}$, and using the inequality $\sigma_x(\Delta x) \le \|\Delta x\|_x \le \sigma\|\Delta x\|_w \le \sigma\gamma$, we see that $\bar\sigma \le \sigma\gamma$, which implies that

$$\tau = \frac{\bar\sigma}{\sigma^2} \le \frac{\gamma}{\sigma}. \tag{5.34}$$

Using the fact that for all $0 \le \tau \le 1$ the function $\frac{-\tau-\log(1-\tau)}{\tau^2}$ is monotonically increasing and that $\sigma_x(w) = \sigma_s^\star(w)$, we can show that the following bound holds [44, Theorem 8.2]:

$$\begin{aligned}\Psi(z+\alpha\Delta z) - \Psi(z) &\le -\alpha\gamma^2 + \frac{\sigma^2\gamma^2}{\bar\sigma^2}\left(-\alpha\bar\sigma - \log\left(1-\alpha\bar\sigma\right)\right)\\ &= -\alpha\sigma^2\gamma^2\left(\frac{1}{\sigma^2} + \frac{1}{\bar\sigma}\right) - \frac{\sigma^2\gamma^2}{\bar\sigma^2}\log(1-\alpha\bar\sigma).\end{aligned} \tag{5.35}$$

With $\alpha = \frac{1}{\sigma^2+\bar\sigma}$ this bound becomes

$$\Psi(z+\alpha\Delta z) - \Psi(z) \le -\frac{\gamma^2}{\bar\sigma} + \frac{\sigma^2\gamma^2}{\bar\sigma^2}\log\left(1+\frac{\bar\sigma}{\sigma^2}\right) = -\frac{\gamma^2}{\sigma^2}\left[\frac{\tau-\log(1+\tau)}{\tau^2}\right],$$

where $\tau = \frac{\bar\sigma}{\sigma^2} \le \frac{\gamma}{\sigma}$. Since the function in brackets is monotonically decreasing, the upper bound for τ yields the final form of the bound:

$$\Psi(z+\alpha\Delta z) - \Psi(z) \le -\left(\frac{\gamma}{\sigma} - \log\left(1+\frac{\gamma}{\sigma}\right)\right).$$

The last element of the proof is to show that $\frac{\gamma}{\sigma} \ge \frac{\sqrt{3}}{2}$, [44, theorem 5.2] and therefore each step of the symmetric primal-dual potential reduction direction can be reduced by a constant.

Chapter 6

Algorithms for the full homogeneous embedding with small linear systems

As we have seen, it is possible to design algorithms with excellent worst-case complexity bounds for unsymmetric conic programming. These algorithms require the availability of a conjugate pair of barriers f for the primal cone $\mathcal{K}$ and $f^\star$ for the dual cone $\mathcal{K}^\star$ and with cheaply computable Hessians and gradients. Furthermore they require the solution of linear systems that are twice as large as those for symmetric conic programming algorithms.

In reality it is common to have an explicit barrier for the primal cone f while not being able to compute the conjugate barrier $f^\star$ [42]. Also, the large Newton systems pose a real limitation for the linear solvers, as doubling the size of the systems could increase the cost of the linear solves by up to a factor of eight.

In this section we analyze strategies for solving the homogeneous self-dual embedding problem without explicit derivatives of the dual barrier. In the process we define algorithms that solve smaller Newton systems that are comparable in size to those of the more practical symmetric cone algorithms. However, we will pay by having to maintain the iterates close to the central path and therefore being forced to accept shorter step-lengths and higher iteration counts.

As in the previous sections the symbol x represents the concatenation of the primal slacks with the variable τ, the symbol s represents the concatenation of the dual slacks with the variable κ, the symbol f is redefined as $f - \log(\tau)$ where f is the barrier of the primal slacks, and the symbol $f^\star$ is redefined to mean $f^\star - \log(\kappa)$ where $f^\star$ is the conjugate function to the original f. The symbols for the primal and dual cones are redefined accordingly. The symbols $g(x)$ and $H(x)$ denote the gradient and Hessian of the new f while the symbols $g^\star(s)$ and $H^\star(s)$ denote the gradient and Hessian of the new $f^\star$. We let $\|v\|_{\star x} = \sqrt{v^T H^{-1}(x) v}$ and $\|v\|_x = \sqrt{v^T H(x) v}$. Whenever we need to use the norm induced by the dual Hessian we use the symbol $\|v\|_{H^\star(s)} = \sqrt{v^T H^\star(s) v}$. Finally we use the

symbol z to mean the concatenation of all variables $z = (y, x, s, \theta)$, and redefine ν to mean $\nu + 1$, since now f is $(\nu + 1)$-logarithmically homogeneous.

The algorithms described in this section are built on a variation of the centering direction

$$\begin{gathered} \mu H(x)\Delta x + \Delta s = -s - \mu g(x) \\ G\begin{pmatrix}\Delta y\\ \Delta x\\ \Delta\theta\end{pmatrix} - \begin{pmatrix}0\\ \Delta s\\ 0\end{pmatrix} = \begin{pmatrix}0\\ 0\\ 0\end{pmatrix}, \end{gathered} \tag{6.1}$$

which differs from the direction used in symmetric cone programming in the argument of the Hessian. This search direction does not incorporate information about the dual barrier $f^\star$ and therefore maintaining s feasible with respect to the dual cone requires special care. In fact, we show that using the centrality measure

$$\eta(x, s, \mu) = \frac{1}{\mu} \left\| s + \mu g(x) \right\|_{H^{-1}(x)}$$

and maintaining

$$\eta(x, s, \mu) < \bar{\eta} < 1 \tag{6.2}$$

ensures feasibility of the dual slacks s. We also show that (6.2) ensures that direction (6.1) reduces the distance to the central path *quickly*. This characteristic is essential for the worst-case bounds of the algorithms we present.

Similar statements about the search direction (6.1) are shown in [51], where the authors define a predictor-corrector algorithm, verify computationally its efficiency, and show that a direction similar to (6.1) with the choice $\mu = x^T s/\nu$ reduces the centrality quickly. We extend this result slightly and show that for any μ that keeps (x, s) well centered the direction (6.1) reduces the centrality quickly. We also show that the choice $\mu = x^T s/\rho$ (where $\rho > \nu$ is chosen judiciously) will maintain the centering condition (6.2) while guaranteeing a step-size bounded away from zero.

Using these results we derive two algorithms. The first treats the barrier parameter μ as an independent scalar (which is modified at every iteration within certain bounds), and the second chooses a fixed scalar $\rho > \nu$ and sets $\mu = x^T s/\rho$ at every iteration. In both cases we show that the resulting algorithm achieves a worst-case complexity of order $\mathcal{O}(\sqrt{\nu} \log \frac{1}{\varepsilon})$ iterations to achieve a precision of ε.

This area is very well explored and similar results to the ones derived here, are known. Both the statements here and the results from [51] can be derived from the work on self-concordant monotone mappings presented by Nesterov and Nemirovski in [43]. We prefer to use the theory of self-concordant functions because the exposition will be clearer and we wish to avoid presenting the monotone-mapping machinery. We should also mention that the primal-dual potential reduction algorithm in [43] uses primal-dual search directions that are very similar to the ones used in this section. However, since that algorithm requires moving either the primal slacks or the dual variables, it is incompatible with the homogeneous embedding, and therefore we believe our algorithms are

a contribution.

We begin by recalling the following properties of the derivatives of conjugate pairs of barriers, which hold for all strictly feasible $x \in \operatorname{int} \mathcal{K}$ and $s \in \operatorname{int} \mathcal{K}^\star$:

$$-g(x) \in \operatorname{int} \mathcal{K}^\star, \tag{6.3a}$$
$$-g(-g^\star(s)) = s, \tag{6.3b}$$
$$H^\star(-g(x)) = H^{-1}(x) \tag{6.3c}$$

and the following properties of ν-logarithmically homogeneous self-concordant barrier functions

$$g(\gamma x) = \frac{1}{\gamma} g(x), \tag{6.4a}$$
$$H(\gamma x) = \frac{1}{\gamma^2} H(x). \tag{6.4b}$$

Because of the symmetry between conjugate function pairs, all these relations hold when the roles of f and $f^\star$ are reversed.

Since G is skew-symmetric we can show that

$$\Delta x^T \Delta s = 0, \tag{6.5a}$$
$$(x + \alpha \Delta x)^T (s + \alpha \Delta s) = x^T s + \alpha (s^T \Delta x + x^T \Delta s). \tag{6.5b}$$

By right-multiplying the linear constraints in (6.1) by $\left(\Delta y^T, \Delta x^T, \Delta\theta\right)$ and using the skew symmetry of G we get the result. From the orthogonality of Δx and Δs we can show that if $\frac{1}{\mu} \left\| s + \mu g(x) \right\|_{\star x} \leq \eta$ then

$$\begin{aligned} \left\| \Delta x \right\|_x^2 + \frac{1}{\mu^2} \left\| \Delta s \right\|_{\star x}^2 &\leq \eta^2, \\ \left\| \Delta x \right\|_x &\leq \eta, \\ \left\| \Delta s \right\|_{\star x} &\leq \eta. \end{aligned} \tag{6.6}$$

This is a simple consequence of the definition of the search directions. Since

$$\frac{1}{\mu} \left\| \mu H(x) \Delta x + \Delta s \right\|_{\star x} = \frac{1}{\mu} \left\| s + \mu g(x) \right\|_{\star x} \leq \eta,$$

expanding the left-hand side implies that

$$\left\| \Delta x \right\|_x^2 + \frac{1}{\mu^2} \left\| \Delta s \right\|_{\star x}^2 \leq \eta^2 \tag{6.7}$$

and the rest follow.

Lemma 6.0.2. *Whenever* $\frac{1}{\mu} \left\| s + \mu g(x) \right\|_{\star x} < 1$, $s \in \operatorname{int} \mathcal{K}^\star$.

Proof. Since $-\mu g(x) \in \operatorname{int} \mathcal{K}^\star$ and $f^\star$ is a self-concordant function, then the Dikin ellipsoid centered at $-\mu g(x)$ is contained in the dual cone. Therefore $\left\| s + \mu g(x) \right\|_{H^\star(-\mu g(x))} < 1$ implies $s \in \operatorname{int} \mathcal{K}^\star$, from (6.3c) and (6.4b). Because from $\left\| s + \mu g(x) \right\|_{H^\star(-\mu g(x))} = \frac{1}{\mu} \left\| s + \mu g(x) \right\|_{\star x}$, the result follows. □

6.1 A substitute for the dual barrier and the unsymmetric centering directions

In this section we motivate the search directions (6.1) by showing that they can be used to efficiently solve the barrier problem

$$\begin{aligned} &\text{minimize } \psi(x,s,\theta) = \frac{1}{\mu}\mu_0\nu\theta + f(x) + f^\star(s) \\ &\text{subject to} \\ &\qquad G\begin{pmatrix} y \\ x \\ \theta \end{pmatrix} - \begin{pmatrix} 0 \\ s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\mu_0\nu \end{pmatrix}. \end{aligned} \tag{PB}$$

Let $q_{\bar{x}}(s)$ be the quadratic approximation to $f^\star(s)$ centered at $-\mu g(\bar{x})$:

$$q_{\bar{x}}(s) = \frac{1}{2}\left\|s + \mu g(\bar{x})\right\|^2_{H^\star(-\mu g(\bar{x}))} + g^\star(-\mu g(\bar{x}))^T(s + \mu g(\bar{x})) + f^\star(-\mu g(\bar{x})).$$

If we substitute the $f^\star(s)$ term in ψ (PB) we form a new primal-dual barrier problem

$$\begin{aligned} &\text{minimize } \bar{\psi}(x,s,\theta) = \frac{1}{\mu}\mu_0\nu\theta + f(x) + q_{\bar{x}}(s) \\ &\text{subject to} \\ &\qquad G\begin{pmatrix} y \\ x \\ \theta \end{pmatrix} - \begin{pmatrix} 0 \\ s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\mu_0\nu \end{pmatrix}. \end{aligned} \tag{6.8}$$

We will show that whenever $\bar{x} = x$ the Newton direction for (6.8) can be calculated by solving system (6.1), and because the objective function of (6.8) is self-concordant the Newton directions will reduce $\bar{\psi}$ significantly. We will also show that when η is small, $\bar{\psi}$ and ψ are similar and reducing $\bar{\psi}$ implies reducing ψ. We can conclude that the objective of the original barrier problem can be reduced.

To show that direction (6.1) coincides with the Newton directions for (6.8), observe from (6.4b), (6.3c), (6.4a), and (6.3b) that

$$q_{\bar{x}}(s) = \frac{1}{2\mu^2}\left\|s + \mu g(\bar{x})\right\|^2_{\star\bar{x}} - \frac{1}{\mu}\bar{x}^T(s + \mu g(\bar{x})) + f^\star(-\mu g(\bar{x})), \tag{6.9}$$

$$\nabla_s q_{\bar{x}}(s) = \frac{1}{\mu^2}H(\bar{x})^{-1}(s + \mu g(\bar{x})) - \frac{1}{\mu}\bar{x}, \tag{6.10}$$

$$\nabla^2_s q_{\bar{x}}(s) = \frac{1}{\mu^2}H^{-1}(\bar{x}). \tag{6.11}$$

Therefore the gradient and Hessian of $q_{\bar{x}}(s)$ depend only on the primal barrier.

Since the Newton system for (6.8) is the set of equations

$$H(x)\Delta x + G^T \begin{pmatrix} \lambda_y \\ \lambda_x \\ \lambda_\theta \end{pmatrix} = - \begin{pmatrix} 0 \\ g(x) \\ \frac{1}{\mu}\mu_0\nu \end{pmatrix} \tag{6.12a}$$

$$\frac{1}{\mu^2} H^{-1}(x)\Delta s - \lambda_s = -\nabla q(s) \tag{6.12b}$$

$$G \begin{pmatrix} \Delta y \\ \Delta x \\ \Delta\theta \end{pmatrix} - \begin{pmatrix} 0 \\ \Delta s \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \tag{6.12c}$$

it is clear that the solution to (6.1) satisfies (6.12c). To show that (6.1) solves (6.12) we have to show that there exist Lagrange multipliers for which (6.12a) and (6.12b) are satisfied.

Lemma 6.1.1. *The solution to* (6.1) *together with the choice of Lagrange multipliers* $\lambda_y = \frac{1}{\mu}(-y - \Delta y)$, $\lambda_x = \frac{1}{\mu}(-x - \Delta x)$, *and* $\lambda_\theta = \frac{1}{\mu}(-\theta - \Delta\theta)$ *solves* (6.12).

Proof. Observe that the specified λ satisfies

$$G^T \begin{pmatrix} \lambda_y \\ \lambda_x \\ \lambda_\theta \end{pmatrix} = \frac{1}{\mu} G \begin{pmatrix} y + \Delta y \\ x + \Delta x \\ \theta + \Delta\theta \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{s+\Delta s}{\mu} \\ -\frac{1}{\mu}\mu 0\nu \end{pmatrix}.$$

Therefore $\mu H(x)\Delta x + \Delta s = -s - \mu g(x)$ implies that $H(x)\Delta x + \frac{s+\Delta s}{\mu} = -g(x)$, and (6.12a) is satisfied. As for (6.12b), multiply $\mu H(x)\Delta x + \Delta s = -s - \mu g(x)$ by $\frac{1}{\mu^2} H(x)^{-1}$ and add $\frac{1}{\mu} x$ to both sides of the equality to get (6.12b). □

Since $f^\star$ is a self-concordant function the error induced by a quadratic approximation is bounded by

$$|f^\star(s) - q(s)| \leq \frac{\frac{1}{\mu^3} \|s + \mu g(x)\|^3_{\star x}}{3(1 - \frac{1}{\mu} \|s + \mu g(x)\|_{\star x})} = \frac{\eta^3}{3(1-\eta)} \tag{6.13}$$

whenever $\eta < 1$ (see Theorem (4.1.3)). Therefore as η becomes small, $q_{\bar{x}}(s)$ approaches $f^\star(s)$ and in particular on the central path ($\eta = 0$) they are identical. Since we established that directions (6.1) are the Newton directions for (6.8) and that whenever η is small both barrier problems are similar, we can expect that these directions can be used to solve (PB). Before we state the main result of this section we need to show the following.

Lemma 6.1.2. *If* $\eta \leq 1$ *then for all* $\alpha < 1$ *we have* $\frac{1}{\mu} \|s + \alpha\Delta s + \mu g(x)\|_{\star x} < \eta$.

Proof. Observe that

$$\begin{aligned}\|s+\alpha\Delta s+\mu g(x)\|_{\star x}^2 &= \|s+\mu g(x)\|_{\star x}^2 \\ &\quad + 2\alpha\Delta s^T H(x)^{-1}(s+\mu g(x)) \\ &\quad + \alpha^2\|\Delta s\|_{\star x}^2 \\ &= \|s+\mu g(x)\|_{\star x}^2 \\ &\quad + \alpha(\alpha-2)\frac{1}{\mu^2}\|\Delta s\|_{\star x}^2 < \eta^2,\end{aligned} \tag{6.14}$$

where the second equality holds because $\Delta s^T H^{-1}(x)(s+\mu g(x)) = -\|\Delta s\|_{\star x}^2$. □

We are in a position to make a precise statement about how directions (6.1) reduce the primal-dual objective of (PB). We use the symbols ψ^+ and $\bar{\psi}^+$ to denote the value of ψ respectively $\bar{\psi}$ when evaluated at the point $(x+\alpha\Delta x, s+\alpha\Delta s, \theta+\alpha\Delta\theta)$.

Theorem 6.1.3. *If $\eta = \frac{1}{\mu}\|s+\mu g(x)\|_{\star x} \leq \frac{1}{2}$ then exact minimization of ψ along directions* (6.1) *will reduce ψ by at least $-\frac{1}{3}\eta^2$.*

Proof. Because $\bar{\psi}$ is self-concordant it admits the upper bound

$$\begin{aligned}\bar{\psi}^+ - \bar{\psi} &\leq \nabla\psi\Delta z - \alpha\lambda - \log(1-\alpha\lambda), \\ &= \frac{\alpha}{\mu}\mu_0\nu\Delta\theta + \alpha g(x)^T\Delta x + \alpha\nabla q_{\bar{x}}(s)^T\Delta s \\ &\quad - \alpha\lambda - \log(1-\alpha\lambda),\end{aligned}$$

where λ is the Newton decrement $\lambda^2 = \|\Delta x\|_x^2 + \|\Delta s\|_{\nabla^2 q(s)}^2$. From the definition of the search directions the equality $\frac{1}{\mu^2}\|\mu H(x)\Delta x + \Delta s\|_{\star x}^2 = \frac{1}{\mu^2}\|s+\mu g(x)\|_{\star x}^2 = \eta^2$ holds. Expanding the left-hand side and using $\Delta x^T\Delta s = 0$ and $\frac{1}{\mu}\|v\|_{\star x} = \|v\|_{\frac{1}{\mu^2}H^{-1}(x)}$ gives $\eta^2 = \|\Delta x\|_x^2 + \frac{1}{\mu^2}\|\Delta s\|_{\star x}^2 = \|\Delta x\|_x^2 + \|\Delta s\|_{\nabla^2 q(s)}^2 = \lambda^2$, and therefore $\eta = \lambda$.

Since the search directions are the Newton equations for problem (6.8), we have $\eta^2 = \Delta z^T\nabla^2\bar{\psi}\Delta z = -\nabla\bar{\psi}^T\Delta z$, and therefore

$$\bar{\psi}^+ - \bar{\psi} \leq -\alpha(\eta^2+\eta) - \log(1-\alpha\eta).$$

We now use this bound to derive a bound for the reduction achievable on ψ. Using Lemma 6.1.2 and bound (6.13) we know that both $|f^\star(s) - q(s)| \leq \frac{\eta^3}{3(1-\eta)}$ and $|f^\star(s+\alpha\Delta s) - q(s+\alpha\Delta s)| \leq \frac{\eta^3}{3(1-\eta)}$ hold. Therefore

$$\psi^+ - \psi \leq -\alpha(\eta^2+\eta) - \log(1-\alpha\eta) + \frac{2\eta^3}{3(1-\eta)}.$$

Exact minimization will achieve a reduction at least as good as that achievable by the damped Newton step $\alpha = \frac{1}{1+\eta}$, for which we have

$$\psi^+ - \psi \leq -\eta + \log(1+\eta) + \frac{2\eta^3}{3(1-\eta)} \leq \frac{-\eta^2}{2(1-\eta)} + \frac{2\eta^3}{3(1-\eta)}.$$

And since $\eta = \eta \leq \frac{1}{2}$, it follows that $\psi^+ - \psi \leq -\frac{\eta^2}{3}$. □

We have shown that the search directions (6.1) are good descent directions for the original primal-dual barrier problem (PB) because they are the Newton directions for a similar optimization problem and when the centrality is small, in particular for $\eta \leq \frac{1}{2}$, a decrease of $\frac{1}{3}\eta^2$ is guaranteed.

6.2 A short-step path-following algorithm for the unsymmetric homogeneous self-dual formulation

To build an algorithm out of directions (6.1) we need to define how the parameter μ is updated and when. The algorithm we present here reduces μ at every iteration as much as possible without violating the centrality condition. We show that, when the centering condition (6.2) holds, a backtracking linesearch along direction (6.1) can reduce the function $\eta(x+\alpha\Delta x, s+\alpha\Delta s, \mu)$ by a multiplicative constant. Once this distance is reduced, μ is reduced as much as possible keeping $\eta(x^+, s^+, \mu^+) < \bar{\eta}$. We show that the reduction in μ is bounded above by some multiplicative constant γ, namely $\mu^+/\mu \leq \gamma$, and in this way the algorithm produces a sequence of iterates that maintain the centrality condition while reducing the barrier parameter at a linear rate. Finally we show that γ is bounded above in such a way that the number of iterations to reach $\mu \leq \varepsilon$ is $\mathcal{O}(\sqrt{\nu}\log\frac{1}{\varepsilon})$, and that because of the centrality condition, this implies $x^T s/\nu \leq 2\varepsilon$.

We start the analysis of Algorithm 2 by deriving an upper bound for $\eta(x+\alpha\Delta x, s+\alpha\Delta s, \mu)$ as a function of α. To derive the result we need to bound the size of the step direction, the change in the norm induced by the inverse Hessian, and the error induced by approximating the gradient by a linearization.

Lemma 6.2.1. *If f is a self-concordant nondegenerate function, then for any x, y such that $\|y-x\|_x < 1$ the following inequality holds*

$$\|v\|_{\star y} \leq \frac{\|v\|_{\star x}}{1 - \|y-x\|_{\star x}}. \tag{6.15}$$

Proof. From Theorem [48, 2.2.1], if f is a nondegenerate self-concordant function then

$$\left\|H(y)^{-1}H(x)\right\|_{H(x)} \leq \frac{1}{\left(1-\|y-x\|_x\right)^2},$$

Algorithm 2 Unsymmetric primal dual path-following

Given a centrality parameter $\bar{\eta} < \frac{1}{3}$, a feasible $z_0 = (x_0, y_0, s_0, \theta_0)$ for which $\eta(x_0, s_0, \mu) \leq \frac{1}{2}\bar{\eta}^2$ and a backtracking parameter $0 < \beta < 1$.
while $\mu \geq \varepsilon$ **do**
  Solve for μ^+ such that $\eta(x, s, \mu^+) = \bar{\eta}$
  $\mu \leftarrow \mu^+$
  solve for the search directions $\Delta z = (\Delta y \Delta x, \Delta s, \Delta\theta)$ by (6.1)
  $\eta \leftarrow \eta(x, s, \mu)$
  $\alpha \leftarrow 1$ ▷ Backtracking linesearch
  while $\eta(x + \alpha\Delta x, s + \alpha\Delta s, \mu) > \eta + \frac{1}{9}\alpha\eta(\eta - 1)$ **do**
    $\alpha \leftarrow \beta\alpha$
  end while
  $w \leftarrow w + \alpha\Delta w$
end while

where the matrix norm is the induced norm. Therefore we can derive the inequality

$$\begin{aligned}
\|v\|_{\star y}^2 &= v^T H^{-1}(y) v \\
&= v^T H(x)^{-1/2} H(x)^{1/2} H^{-1}(y) H(x) H^{-1}(x) H(x)^{1/2} H(x)^{-1/2} v \\
&= v^T H(x)^{-1/2} H(x)^{1/2} H^{-1}(y) H(x) H^{-1/2}(x) H(x)^{-1/2} v \\
&\leq \|H(x)^{-1/2} v\|_2^2 \|H(x)^{1/2} H^{-1}(y) H(x) H^{-1/2}\|_2 \\
&= \|v\|_{\star x}^2 \left\|H^{-1}(y) H(x)\right\|_{H(x)} \\
&= \frac{\|v\|_{\star x}^2}{\left(1 - \|y - x\|_{H(x)}\right)^2},
\end{aligned} \tag{6.16}$$

as required. □

Lemma 6.2.2. *Let f be a self-concordant nondegenerate function and let x, y be such that $\|y - x\|_x < 1$. Then*

$$\|g(y) - g(x) - H(x)(y - x)\|_{\star x} \leq \frac{r^2}{1 - r},$$

where $r = \|y - x\|_x$.

Proof. See Nesterov [42]. □

We are now in a position to upper bound the centrality as a function of the step size.

Lemma 6.2.3. *The function of α given by $\eta(x + \alpha\Delta x, s + \alpha\Delta s, \mu)$ admits the upper bound*

$$\eta(x + \alpha\Delta x, s + \alpha\Delta s, \mu) \leq \frac{(1 - \alpha)\eta}{1 - \alpha\eta} + \frac{\alpha^2\eta^2}{(1 - \alpha\eta)^2}.$$

Proof. Let $x^+ = x + \alpha\Delta s$, $s^+ = s + \alpha\Delta s$, $r = \|\Delta x\|_x$ and $\eta = \eta(x, s, \mu)$. From Lemma 6.2.1, we have

$$\frac{1}{\mu}\left\|s^+ + \mu g(x^+)\right\|_{\star x^+} \leq \frac{\frac{1}{\mu}\left\|s^+ + \mu g(x^+)\right\|_{\star x}}{1 - \alpha r}.$$

Writing the gradient as $g(x^+) = g(x) + \alpha H(x)\Delta x + e_r$, where e_r is an error term, and using the bound for the norm of e_r of Lemma 6.2.2 we get the bound

$$\frac{1}{\mu}\left\|s^+ + \mu g(x^+)\right\|_{\star x} \leq \frac{1}{\mu}\left\|s + \alpha\Delta s + \mu g(x) + \alpha\mu H(x)\Delta x\right\|_{\star x} + \frac{\alpha^2 r^2}{1 - \alpha r}.$$

Using the definition of the search directions yields the bound

$$\frac{1}{\mu}\left\|s^+ + \mu g(x^+)\right\|_{\star x} \leq |1 - \alpha|\frac{1}{\mu}\left\|s + \mu g(x)\right\|_{\star x} + \frac{\alpha^2 r^2}{1 - \alpha r},$$

and since $r \leq \eta$,

$$\begin{aligned}\eta(x^+, s^+, \mu) &\leq \frac{(1-\alpha)\eta(x, s, \mu)}{1 - \alpha r} + \frac{\alpha^2 r^2}{(1 - \alpha r)^2} \\ &\leq \frac{(1-\alpha)\eta}{1 - \alpha\eta} + \frac{\alpha^2\eta^2}{(1 - \alpha\eta)^2}.\end{aligned} \tag{6.17}$$

□

Given this upper bound we can show that when the backtracking linesearch terminates, the centrality measure $\eta(x, s, \mu)$ has been reduced by a multiplicative constant. To show the worst-case complexity bound, the choice of backtracking parameter β is not very important. In fact as long as $0 < \beta < 1$ the resulting step size will achieve $\eta^+ \leq c\eta$ for some constant smaller than one. For simplicity we choose $\beta = \frac{1}{2}$ and show that the step size is bounded below by $\frac{\beta}{1+\eta} \leq \alpha$ and that the backtracking linesearch will terminate with $\eta(x + \alpha\Delta x, s + \alpha\Delta s, \mu) \leq \frac{35}{36}\eta$.

Lemma 6.2.4. *The backtracking linesearch of Algorithm 2 will terminate with* $\alpha \geq \frac{\beta}{1+\eta}$ *and*

$$\eta(x^+, s^+, \mu) \leq \frac{35}{36}\eta.$$

Proof. The step size $\alpha = \frac{1}{1+\eta}$ satisfies the linesearch termination criteria because

$$\eta(x^+, s^+, \mu) \leq \frac{(1-\alpha)\eta}{1 - \alpha\eta} + \frac{\alpha^2\eta^2}{(1 - \alpha\eta)^2} = 2\eta^2 \leq \frac{2}{3}\eta$$

and $\eta + \frac{1}{9}\alpha(\eta(\eta-1)) = \eta\left(1 + \frac{\eta-1}{9\eta+9}\right) \geq \frac{2}{3}\eta$. Therefore the linesearch terminates with $\alpha \geq \frac{\beta}{1+\eta}$ and

$$\eta(x^+, s^+, \mu) \leq \eta + \frac{1}{9}\alpha(\eta(\eta-1)) \leq \eta\left(1 - \frac{\beta(1-\eta)}{9(\eta+1)}\right) \leq \frac{35}{36}\eta. \tag{6.18}$$

□

After updating the variables, the algorithm finds the smallest value for μ^+ that satisfies $\eta(x^+, s^+, \mu^+) = \bar{\eta}$. This operation is not very costly, for observe that solving $\eta(x, s, \mu)^2 = \bar{\eta}^2$ is the same as solving for t such that $t^2 \|s\|_{\star x}^2 - 2tx^T s + \nu = \bar{\eta}^2$, and setting $\mu = \frac{1}{t}$. We now establish that there exists a constant $\bar{\gamma}$ such that $\mu^+ \leq \bar{\gamma}\mu$.

Lemma 6.2.5. *If $\eta(x, s, \mu) \leq \dot{\eta}$, where $\dot{\eta}$ is some constant, then $\|s\|_{\star x} \leq \mu(\dot{\eta} + \sqrt{\nu})$.*

Proof. Using the triangle rule and the properties of the ν-logarithmically homogeneous self-concordant barriers we get the bound

$$\frac{1}{\mu}\|s\|_{\star x} = \frac{1}{\mu}\|s + \mu g(x) - \mu g(x)\|_{\star x} \leq \bar{\eta} + \|g(x)\|_{\star x} = \bar{\eta} + \sqrt{\nu}.$$

□

Lemma 6.2.6. *If $\eta(x, s, \mu) \leq \dot{\eta}$ then*

$$\eta(x, s, \gamma\mu) \leq \left(\dot{\eta} + \left(\frac{1-\gamma}{\gamma}\right)(\dot{\eta} + \sqrt{\nu})\right),$$

and if γ is chosen so that $\frac{1-\gamma}{\gamma}(\dot{\eta} + \sqrt{\nu}) \leq \frac{1}{35}\dot{\eta}$ then

$$\eta(x, s, \mu\gamma) \leq \frac{36}{35}\dot{\eta}.$$

Proof. If $\delta = \frac{1-\gamma}{\mu\gamma}$ then $\frac{1}{\mu^+} = \frac{1}{\mu} + \delta$. Write

$$\begin{aligned}
\eta(x, s, \mu\gamma)^2 &= \left\|\frac{1}{\mu}s + \delta s + g(x)\right\|_{\star x}^2 \\
&= \eta(x, s, \mu)^2 + 2\delta s^T H^{-1}(x)\left(\frac{1}{\mu}s + g(x)\right) + \delta^2 \|s\|_{H^{-1}(x)}^2 \\
&\leq \eta(x, s, \mu)^2 + 2\delta \|s\|_{H^{-1}(x)} \eta(x, s, \mu) + \delta^2 \|s\|_{H^{-1}(x)}^2 \qquad (6.19) \\
&\leq \dot{\eta}^2 + 2\frac{1-\gamma}{\gamma}\dot{\eta}(\dot{\eta} + \sqrt{\nu}) + \left(\frac{1-\gamma}{\gamma}\right)^2 (\dot{\eta} + \sqrt{\nu})^2 \\
&= \left(\dot{\eta} + \left(\frac{1-\gamma}{\gamma}\right)(\dot{\eta} + \sqrt{\nu})\right)^2.
\end{aligned}$$

Using the choice of γ we get the second part of the lemma. □

After updating the iterate z, Algorithm 2 will have achieved $\eta \leq \frac{35}{36}\bar{\eta}$ therefore the updated μ^+ will be smaller than $\mu\gamma$, where γ is as chosen in Lemma 6.2.6.

To derive the worst-case complexity bound observe that at each iteration the algorithm maintains the centrality condition (6.2) while reducing the barrier parameter to $\mu_k \leq \mu_0\gamma^k$. The algorithm stops (at the latest) when k is large enough for $\mu_0\gamma^k \leq \varepsilon$ to hold. This is satisfied for all $k \geq N$, where

$$N = \left(\log \frac{1}{\gamma}\right)^{-1} \left(\log \frac{1}{\varepsilon} + \log \mu_0\right).$$

Using the bound $\log 1/\gamma \geq 1 - \gamma$ and since $\frac{1}{1-\gamma} = \frac{35}{\bar{\eta}}\sqrt{\nu} + 36$ we can conclude that if

$$k \geq \left(\frac{35}{\bar{\eta}}\sqrt{\nu} + 36\right) \left(\log \frac{1}{\varepsilon} + \log \mu_0\right),$$

then $\mu \leq \varepsilon$ and therefore the algorithm terminates in $\mathcal{O}\left(\sqrt{\nu} \log\left(\frac{1}{\varepsilon}\right)\right)$ iterations.

6.3 Moving the barrier parameter continuously

In this section we show that the barrier parameter can be set to $\mu = \frac{x^T s}{\rho}$ for some $\rho > \nu$. The algorithm moves the iterates along the direction (6.1) and chooses a step size that satisfies the centrality condition

$$\eta\left(x^+, s^+, \frac{{x^+}^T s^+}{\rho}\right) \leq \bar{\eta}.$$

We show that a step size of $\alpha = \frac{1}{1+\eta}$ is acceptable, and that for a correct choice of ρ such an algorithm achieves $\mu \leq \epsilon$ in $\mathcal{O}\left(\sqrt{\nu} \log \frac{1}{\varepsilon}\right)$ iterations. To remove some of the guesswork we set $\rho = \nu + p\sqrt{\nu}$ for some positive scalar p. And to simplify notation in this section we use the symbols $\mu = \frac{x^T s}{\rho}$ and $\mu^+ = \frac{{x^+}^T s^+}{\rho}$.

We show that condition (6.2) implies the restriction $p \leq \eta \leq \bar{\eta}$. This means that p acts as a lower bound for the centrality η the algorithm can achieve.

Lemma 6.3.1. *If $\eta(x, s, \mu) \leq \bar{\eta}$ then $p \leq \eta$.*

Algorithm 3 Unsymmetric primal dual potential reduction

Given a centrality parameter $\bar{\eta} < \frac{1}{5}$, a feasible $z_0 = (x_0, y_0, s_0, \theta_0)$ for which $\eta(x_0, s_0, \mu) \leq \bar{\eta}$, a parameter ρ, and a backtracking parameter $0 < \beta < 1$.
while $\mu \geq \varepsilon$ **do**
  solve for the search directions $\Delta z = (\Delta y \Delta x, \Delta s, \Delta\theta)$ by (6.1)
  $\alpha \leftarrow 1$ ▷ Backtracking linesearch
  $\mu^+ \leftarrow (x + \alpha\Delta x)^T(s + \alpha\Delta s)/\rho$
  while $\eta(x + \alpha\Delta x, s + \alpha\Delta s, \mu^+) > \bar{\eta}$ **do**
    $\alpha \leftarrow \beta\alpha$
    $\mu^+ \leftarrow (x + \alpha\Delta x)^T(s + \alpha\Delta s)/\rho$
  end while
  $w \leftarrow w + \alpha\Delta w$
  $\mu \leftarrow \mu^+$
end while

Proof. Observe that if $\beta = x^T s/\nu$ and if we choose δ so $\mu = \beta + \delta$ then

$$\begin{aligned}\eta(x,s,\mu)^2 &= \frac{1}{(\beta+\delta)^2}\left\|s + (\beta+\delta)g(x)\right\|_{\star x}^2 \\ &= \frac{1}{(\beta+\delta)^2}\left\|s + \beta g(x)\right\|_{\star x}^2 \\ &\quad -\frac{2\delta x^T(s+\beta g(x))}{(\beta+\delta)^2} \\ &\quad +\frac{1}{(\beta+\delta)^2}\delta^2\nu \\ &= \frac{1}{(\beta+\delta)^2}\left\|s + \beta g(x)\right\|_{\star x}^2 + \frac{1}{(\beta+\delta)^2}\delta^2\nu\end{aligned} \tag{6.20}$$

because $x^T(s + \beta g(x)) = 0$. Therefore if $\eta(x, s, \mu) \leq \bar{\eta}$, then $\frac{1}{(\beta+\delta)^2}\delta^2\nu \leq \bar{\eta}^2$. From $\delta + \beta = \delta + \frac{x^T s}{\nu} = \frac{x^T s}{\rho}$ we get $\delta = \frac{\nu-\rho}{\rho\nu}x^T s$ and therefore

$$\frac{1}{(\beta+\delta)^2}\delta^2\nu = \left(\frac{\rho}{x^T s}\frac{\nu-\rho}{\rho\nu}x^T s\right)^2\nu = \left(\frac{\nu-\rho}{\nu}\right)^2\nu = p^2.$$

This means that the constraint $\frac{1}{(\beta+\delta)^2}\delta^2\nu \leq \bar{\eta}$ implies $p \leq \bar{\eta}$. □

We now prove the equality

$$x^T\Delta s + s^T\Delta x = -\mu\left\|\Delta x\right\|_{\star x}^2 - p\mu\sqrt{\nu}, \tag{6.21}$$

which is used in the following lemma. Since

$$\Delta x^T\left(\mu H(x)\Delta x + \Delta s\right) = -\Delta x^T(s + \mu g(x)),$$

we have

$$\begin{aligned}\mu \left\|\Delta x\right\|_{\star x}^2 + \Delta x^T s + \Delta s^T x &= -\mu \Delta x^T g(x) + \Delta s^T x \\ &= x^T(\mu H(x)\Delta x + \Delta s) \\ &= x^T(-s - \mu g(x)) \\ &= \mu(\nu - \rho),\end{aligned}$$

and (6.21) follows.

Lemma 6.3.2. *The ratio $\frac{\mu^+}{\mu}$ admits the bounds*

$$1 - \alpha\frac{\eta^2 + p\sqrt{\nu}}{\rho} \leq \frac{\mu^+}{\mu} \leq 1 - \alpha\frac{p\sqrt{\nu}}{\rho}.$$

Proof. Using (6.5b) we have $\mu^+ = \mu + \frac{\alpha}{\rho}\left(x^T\Delta s + s^T\Delta x\right)$, and from the definition of $\rho = \nu + p\sqrt{\nu}$ we can derive the bounds

$$\frac{\mu^+}{\mu} = 1 - \alpha\frac{\left\|\Delta x\right\|_x^2 + p\sqrt{\nu}}{\rho} \quad \geq 1 - \alpha\frac{\eta^2 + p\sqrt{\nu}}{\rho}, \tag{6.22}$$

where we used (6.6) and (6.21). □

In Lemma 6.2.3 we established the upper bound

$$\eta(x^+, s^+, \mu) \leq \frac{(1-\alpha)\eta}{1-\alpha\eta} + \frac{\alpha^2\eta^2}{(1-\alpha\eta)^2}.$$

Therefore for the step size $\alpha = \frac{1}{1+\eta}$ we have $\eta(x^+, s^+, \mu) \leq 2\eta^2$. Using the first part of Lemma 6.2.6 with $\dot{\eta} = 2\eta^2$ we can show that $\eta(x^+, s^+, \mu^+) \leq 2\eta^2 + \frac{1-\gamma}{\gamma}(2\eta^2 + \sqrt{\nu})$, where $\gamma = \mu^+/\mu$. And from Lemma 6.3.2 and the choice $\alpha = \frac{1}{1+\eta}$ we have

$$\frac{1-\gamma}{\gamma} \leq \frac{\eta^2 + p\sqrt{\nu}}{\rho(\eta+1) - \eta^2 - p\sqrt{\nu}} \leq \frac{\eta^2 + p\sqrt{\nu}}{\nu}.$$

Therefore, whenever $\nu \geq 4$ we have

$$\begin{aligned}\eta(x^+, s^+, \mu^+) &\leq 2\eta^2 + \frac{\eta^2 + p\sqrt{\nu}}{\nu}(2\eta^2 + \sqrt{\nu}) \\ &\leq 2\eta^2 + \eta^4 + p\eta^2 + p,\end{aligned} \tag{6.23}$$

and with the choices $\bar{\eta} \leq \frac{1}{5}$ and $p \leq \frac{1}{2}\bar{\eta}$, the inequality $\eta(x^+, s^+, \mu^+) \leq \bar{\eta}$ holds. This proves that a step size of $\alpha \leq \frac{1}{1+\eta}$ is always admissible. Therefore a backtracking linesearch would terminate with a step size satisfying $\frac{1}{4} \leq \frac{\beta}{1+\eta} \leq \alpha$.

Using the bound from (6.3.2) we have $\mu^+/\mu \leq 1 - \frac{p\sqrt{\nu}}{4\rho}$. Therefore Algorithm 3 will reduce μ at least as fast as $\mu \leq (1 - \frac{p\sqrt{\nu}}{4\rho})^k \mu_0$, which in turn implies that for $k \geq \left(\frac{4\rho}{p\sqrt{\nu}}\right)\left(\log\frac{1}{\varepsilon} - \log\mu_0\right) \geq \frac{\log\frac{1}{\varepsilon} - \log\mu_0}{\log\left(1-\frac{p\sqrt{\nu}}{4\rho}\right)}$ it achieves the precision $\mu \leq \varepsilon$. Finally, since $\frac{4\rho}{p\sqrt{\nu}} = \frac{4}{p}\sqrt{\nu} + 4$, we conclude that the algorithm terminates in order

$$k \geq \left(\frac{4}{p}\sqrt{\nu} + 4\right)\left(\log\frac{1}{\varepsilon} - \log\mu_0\right) = \mathcal{O}\left(\sqrt{\nu}\log\frac{1}{\varepsilon}\right)$$

iterations.

Chapter 7

Linearly infeasible algorithms and the simplified homogeneous embedding

In this chapter we analyze algorithms for unsymmetric conic optimization with three changes relative to those of chapter 6. The first change is that these algorithms use the simplified homogeneous embedding. The second is that we use either the *centering* direction or the *affine scaling* direction. The final change is that we measure the distance to the central path with the functional centrality measure $\Omega(x, s)$ instead of $\frac{1}{\mu} \|s + \mu g(x)\|_{\star x}$. These modifications bring us closer to the implementation of the solver that we describe in the next chapter.

We establish some notation to be consistent with the simplified homogeneous embedding, see section 3.4.3 for more details. The simplified homogeneous embedding eliminates the variable θ; therefore we redefine G with one less row and column and we let:

$$G = \begin{pmatrix} & A & -b \\ -A^T & & c \\ b^T & -c^T & \end{pmatrix}. \tag{7.1}$$

As in the previous section the variables τ and κ are part of the vectors x and s. For the rest of this section we use the symbol μ to mean the complementarity measure $\mu = x^T s/\nu$.

Recall that we are interested in solving the *feasibility* problem

$$\begin{aligned} &\text{minimize } 0 \\ &\text{subject to} \\ &G\begin{pmatrix} y \\ x \end{pmatrix} - \begin{pmatrix} 0 \\ s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &x \in \mathcal{K}, s \in \mathcal{K}^\star, \end{aligned} \tag{sHSD}$$

for which we are given an initial point (x_0, s_0, y_0), with $x_0 \in \operatorname{int} \mathcal{K}$ and $s_0 \in \operatorname{int} \mathcal{K}^\star$. The residuals for the linear equality constraints are

$$\begin{pmatrix} r_p \\ r_d \end{pmatrix} = -G\begin{pmatrix} y_0 \\ x_0 \end{pmatrix} + \begin{pmatrix} 0 \\ s_0 \end{pmatrix},$$

and the central path is the solution of the nonlinear equations

$$s + \mu g(x) = 0, \tag{7.2a}$$

$$G\begin{pmatrix} y \\ x \end{pmatrix} - \begin{pmatrix} 0 \\ s \end{pmatrix} = -\mu \begin{pmatrix} r_p \\ r_d \end{pmatrix}. \tag{7.2b}$$

The affine scaling direction $(\Delta x, \Delta y, \Delta s)$ [51, 8] is derived by assuming the present iterates are on the central path and linearizing the equations (7.2):

$$\mu H(x)\Delta x + \Delta s = -s, \tag{7.3a}$$

$$G\begin{pmatrix} \Delta y \\ \Delta x \end{pmatrix} - \begin{pmatrix} 0 \\ \Delta s \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \end{pmatrix}. \tag{7.3b}$$

The centering direction $(\Delta x, \Delta y, \Delta s)$ is defined as

$$\mu H(x)\Delta x + \Delta s = -s - \mu g(x), \tag{7.4a}$$

$$G\begin{pmatrix} \Delta y \\ \Delta x \end{pmatrix} - \begin{pmatrix} 0 \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{7.4b}$$

Some of the most efficient interior-point symmetric conic programming algorithms [36, 7, 52, 4] use search directions that are a linear combination of (the analogous) affine and centering directions. We use directions satisfying

$$\mu H(x)\Delta x + \Delta s = -s - \sigma\mu g(x) \tag{7.5a}$$

$$G\begin{pmatrix} \Delta y \\ \Delta x \end{pmatrix} - \begin{pmatrix} 0 \\ \Delta s \end{pmatrix} = (1-\sigma)\begin{pmatrix} r_p \\ r_d \end{pmatrix}, \tag{7.5b}$$

where $0 < \sigma < 1$ is a scalar that determines the linear combination between the directions.

We now establish some notation that will be used for the bounds that follow. Let $\lambda = \left(\|\Delta x\|_x^2 + \|\Delta s\|_{\star x}^2\right)^{1/2}$, $\eta_\sigma = \frac{1}{\mu}\|s + \sigma\mu g(x)\|_{\star x}$, and $\eta = \frac{1}{\mu}\|s + \mu g(x)\|_{\star x}$. The following relates the two scalars η and η_σ.

Claim 7.0.3. *The equality $\eta_\sigma^2 = \eta^2 + (1-\sigma)^2\nu$ holds.*

Proof.

$$\begin{aligned}\eta_\sigma^2 &= \frac{1}{\mu}\left\|s + \sigma\mu g(x) - (1-\sigma)\mu g(x)\right\|_{\star x}^2 \\ &= \frac{1}{\mu}\left\|s + \mu g(x)\right\|_{\star x}^2 - 2\frac{1}{\mu}x^T(s+\mu g(x)) + (1-\sigma)^2\nu \\ &= \eta^2 + (1-\sigma)^2\nu,\end{aligned} \tag{7.6}$$

where we used $x^T s + \mu x^T g(x) = 0$. □

The relationship $\Delta x^T \Delta s = 0$ that held for the homogeneous embedding does not hold for the simplified case. However, the following equality will allow us to derive some useful bounds.

Claim 7.0.4.

$$\Delta x^T \Delta s = -(1-\sigma)\Delta x^T(s + \mu g(x)).$$

Proof.

$$(x+\Delta x)^T(s+\Delta s) = \sigma \begin{pmatrix} y+\Delta y \\ x+\Delta x\end{pmatrix}^T \begin{pmatrix} p \\ d\end{pmatrix} \tag{7.7}$$

$$= \sigma\left(x^T s + \Delta x^T s + \begin{pmatrix} y \\ x\end{pmatrix}^T G \begin{pmatrix} \Delta y \\ \Delta x\end{pmatrix}\right) \tag{7.8}$$

$$= \sigma\left(x^T s + \Delta x^T s + x^T \Delta s + (1-\sigma)x^T s\right) \tag{7.9}$$

$$\Delta x^T \Delta s + (1-\sigma)\left(x^T \Delta s + s^T \Delta x + x^T s\right) = \sigma(1-\sigma)x^T s$$

From (7.5b) we have $-\mu g(x)^T \Delta x + x^T \Delta s = -(1-\sigma)x^T s$, and therefore

$$\begin{aligned}\Delta x^T \Delta s + (1-\sigma)\left(\sigma x^T s + \mu g(x)^T \Delta x + s^T \Delta x\right) &= \sigma(1-\sigma)x^T s \\ \implies \Delta x^T \Delta s &= -(1-\sigma)\Delta x^T(s+\mu g(x))\end{aligned}$$

□

Claim 7.0.5. *The following bounds hold:*

$$\left\|\Delta x\right\|_x \le \eta(1-\sigma) + \sqrt{\eta^2 + (1-\sigma)^2(\eta^2+\nu)}, \tag{7.10}$$

$$\frac{1}{\mu}\left\|\Delta s\right\|_{\star x} \le \eta(1-\sigma) + \sqrt{\eta^2 + (1-\sigma)^2(\eta^2+\nu)}, \tag{7.11}$$

$$\left|\frac{\Delta x^T \Delta s}{\mu}\right| \le \eta^2(1-\sigma)^2 + (1-\sigma)\eta\sqrt{\eta^2 + (1-\sigma)^2(\eta^2+\nu)}, \tag{7.12}$$

$$\begin{aligned}\lambda &\le (1-\sigma)\eta + \sqrt{(1-\sigma)^2\eta^2 + \eta^2 + (1-\sigma)^2\nu}. \\ &\le \eta + \sqrt{2\eta^2 + (1-\sigma)^2\nu}.\end{aligned}$$

Proof. We have $\lambda^2 + 2\frac{1}{\mu}\Delta x^T \Delta s = \|\Delta x\|_x^2 + 2\frac{1}{\mu}\Delta x^T \Delta s + \frac{1}{\mu^2}\|\Delta s\|_{\star x}^2 = \eta^2 + (1-\sigma)^2\nu$ and therefore

$$\|\Delta x\|_x^2 \leq \eta^2 + (1-\sigma)^2\nu + 2\|\Delta x\|_x (1-\sigma)\eta, \tag{7.13}$$

$$\frac{1}{\mu^2}\|\Delta s\|_{\star x}^2 \leq \eta^2 + (1-\sigma)^2\nu + 2\|\Delta x\|_x (1-\sigma)\eta. \tag{7.14}$$

By solving the quadratic equation on $\|\Delta x\|_x$ we get bounds (7.10) and (7.11). From Claim 7.0.3 we get $\frac{1}{\mu}|\Delta x^T \Delta s| \leq (1-\sigma)\|\Delta x\|_x \eta$, which implies the bound (7.12). Finally, $\lambda^2 = \|\Delta x\|_x^2 + \frac{1}{\mu^2}\|\Delta s\|_{\star x}^2$ and $\|\Delta x\|_x \leq \lambda$ give

$$\begin{aligned}\lambda^2 &= \eta^2 + (1-\sigma)^2\nu - 2\frac{\Delta x^T \Delta s}{\mu} \\ &= \eta^2 + (1-\sigma)^2\nu + (1-\sigma)\frac{2}{\mu}\Delta x^T(s + \mu g(x)) \\ &\leq \eta^2 + (1-\sigma)^2\nu + 2(1-\sigma)\lambda\eta,\end{aligned} \tag{7.15}$$

which implies the bound

$$\begin{aligned}\lambda &\leq (1-\sigma)\eta + \sqrt{(1-\sigma)^2\eta^2 + \eta^2 + (1-\sigma)^2\nu} \\ &\leq \eta + \sqrt{2\eta^2 + (1-\sigma)^2\nu}.\end{aligned}$$

□

Now, we recall the definitions for the centrality measures

$$\Omega(x,s) = \nu\log(x^T s) + f(x) + f^\star(s) - \nu\log(\nu) + \nu, \tag{7.16}$$

$$\eta(x,s) = \frac{1}{\mu}\|s + \mu g(x)\|_{\star x}, \tag{7.17}$$

and establish some consistency between them.

Recalling that here $\mu = \frac{x^T s}{\nu}$, we show that the value of Ω induces an upper bound for the centrality measure $\eta(x,s)$ and that whenever $\eta < 1$ then η induces an upper bound on Ω.

Theorem 7.0.6. *The centrality measures* (7.16) *and* (7.17) *satisfy*

$$\eta - \log(1+\eta) \leq \Omega$$

and whenever $\eta < 1$

$$\Omega \leq -\eta - \log(1-\eta).$$

Proof. By definition, the conjugate function $f^\star(s) = -\inf\{f(x) + x^T s\}$. Therefore $-f^\star(s/\mu) = \inf\left\{f(x) + \frac{x^T s}{\mu}\right\}$. If $h(x) = f(x) + \frac{x^T s}{\mu}$ and $h^\star$ is the minimizer of h, then $-f^\star(s/\mu) = h^\star$.

The Newton step for $h(x)$ is $\Delta x = -\nabla^2 h(x)\nabla h(x) = -H(x)^{-1}\left(g(x) + \frac{1}{\mu}s\right)$ and the Newton decrement $\eta = \|\Delta x\|_x = \frac{1}{\mu}\|s + \mu g(x)\|_{\star x}$. Since h is self-concordant we have the bound

$$\eta - \log(1+\eta) \leq h(x) - h^\star$$

and for $\lambda < 1$ the bound

$$h(x) - h^\star \leq -\eta - \log(1-\eta).$$

Finally observe that

$$h(x) - h^\star = f(x) + \frac{x^T s}{\mu} + f^\star(s/\mu) = \nu\log(x^T s) + f(x) + f^\star(s) - \nu\log(\nu) + \nu = \Omega.$$

□

7.0.1 A predictor-corrector algorithm for the simplified homogeneous embedding and the functional proximity measure

In this section we show that an algorithm that alternates between prediction ($\sigma = 0$) and correction ($\sigma = 1$) steps can maintain a functional proximity measure Ω below some threshold $\bar{\Omega}$ and take prediction steps of order $\frac{1}{\sqrt{\nu}}$. We choose the threshold $\Omega \leq \frac{1}{4} - \log(1+\frac{1}{4})$, which will imply $\eta \leq \frac{1}{4}$. We establish a second threshold $\bar{\bar{\Omega}} = -\frac{1}{6} - \log(1-\frac{1}{6})$, which implies $\frac{1}{6} \leq \eta$. Whenever Ω is in the range $\bar{\bar{\Omega}} \leq \Omega \leq \bar{\Omega}$ we have $\frac{1}{6} \leq \eta \leq \frac{1}{4}$ and in that case the algorithm will take a correction step. After each correction step and whenever $\Omega \leq \bar{\bar{\Omega}}$ the algorithm will take a prediction step until $\Omega = \bar{\Omega}$.

Claim 7.0.7. *The following bound holds:*

$$g(x)^T\Delta x + \nabla q(s)^T\Delta s \leq -\frac{1}{\mu}(x^T\Delta s + s^T\Delta x) + (1-\sigma)\sqrt{\nu}\eta_\sigma - \eta_\sigma^2.$$

Proof. From $\mu H(x)\Delta x + \Delta s = -s - \sigma\mu g(x) = -s - \mu g(x) + (1-\sigma)\mu g(x)$, we get

$$g(x)^T\Delta x = -\frac{1}{\mu}s^T\Delta x + (1-\sigma)g(x)^T\Delta x - \|\Delta x\|_x^2 - \frac{1}{\mu}\Delta x^T\Delta s,$$

and from

$$\begin{aligned}
\nabla q(s) &= \frac{1}{\mu^2}H(x)^{-1}(s + \mu g(x)) - \frac{1}{\mu}x \\
&= \frac{1}{\mu^2}H(x)^{-1}(s + \sigma\mu g(x)) - \frac{1}{\mu}x - (1-\sigma)\frac{1}{\mu}x \\
&= -\frac{1}{\mu}\Delta x - \frac{1}{\mu^2}H(x)^{-1}\Delta s - \frac{1}{\mu}x - (1-\sigma)\frac{1}{\mu}x
\end{aligned}$$

we get the equality

$$\nabla q(s)^T \Delta s = -\frac{1}{\mu}\Delta x^T \Delta s - \frac{1}{\mu^2}\left\|\Delta s\right\|_{\star x} - \frac{1}{\mu}x^T \Delta s - (1-\sigma)\frac{1}{\mu}x^T \Delta s.$$

Thus

$$\begin{aligned} g(x)^T \Delta x + \nabla q(s)^T \Delta s = &-\frac{1}{\mu}(x^T \Delta s + s^T \Delta x) - (1-\sigma)\frac{1}{\mu}x^T(\mu H(x)\Delta x + \Delta s) \\ &- \frac{1}{\mu^2}\left\|\mu H(x)\Delta x + \Delta s\right\|_{\star x}^2 \end{aligned}$$

□

We now derive some bounds for the change in the functional centrality measure Ω along the combined search direction. Let $\beta = \left(\left\|\Delta x\right\|_x^2 + \left\|\Delta s\right\|_{H^\star(s)}^2\right)^{1/2}$. Since $f(x) + f^\star(s)$ is self-concordant, the following bound holds:

$$\begin{aligned} \Omega^+ - \Omega \leq &\nu \log\left(\frac{(x+\alpha\Delta x)^T(s+\alpha\Delta s)}{x^T s}\right) \\ &+ g(x)^T \Delta x + g^\star(s)^T \Delta s \\ &- \alpha\beta - \log(1-\alpha\beta). \end{aligned}$$

We must remove the dependency on the derivatives of $f^\star$. Therefore we bound the norm $\left\|\Delta s\right\|_{H^\star(x)}$ in terms of the centrality measure η, and relate $g^\star(s)$ to $\nabla q_{\bar{x}}$. Beginning with the norm of Δs we use the definition of self-concordant functions to get

$$\left\|\Delta s\right\|_{H^\star(s)} \leq \frac{\left\|\Delta s\right\|_{H^\star(-\mu g(x))}}{1 - \left\|s + \mu g(x)\right\|_{H^\star(-\mu g(x))}} \leq \frac{\frac{1}{\mu}\left\|\Delta s\right\|_{\star x}}{1 - \frac{1}{\mu}\left\|s + \mu g(x)\right\|_{\star x}} \leq \frac{\frac{1}{\mu}\left\|\Delta s\right\|_{\star x}}{1-\eta}. \tag{7.18}$$

To simplify the resulting bounds, we bound $\left\|\Delta x\right\|_x \leq \frac{\left\|\Delta x\right\|_x}{1-\eta}$, and from

$$\beta^2 = \left\|\Delta x\right\|_x^2 + \left\|\Delta s\right\|_{H^\star(s)}^2 \leq \frac{\left\|\Delta x\right\|_x^2 + \frac{1}{\mu}\left\|\Delta s\right\|_{\star x}^2}{(1-\eta)^2} = \frac{\lambda^2}{(1-\eta)^2} \tag{7.19}$$

we conclude that $\beta = \frac{\lambda}{1-\eta}$.

To relate $g^\star(s)$ to $\nabla q_{\bar{x}}(s)$ we can write

$$g^\star(s) = g^\star(-\mu g(x)) + H^\star(-\mu g(x))(s + \mu g(x)) + e_g,$$

where e_g is some error vector. Since

$$g^\star(-\mu g(x)) + H^\star(-\mu g(x))(s+\mu g(x)) = -\frac{1}{\mu}x + \frac{1}{\mu^2}H^{-1}(x)(s+\mu g(x)) = \nabla q_{\bar{x}}(s),$$

we get that $g^\star(s) = \nabla q_{\bar{x}} + e_g$.

It is now easy to show the following sequence of bounds:

$$\begin{aligned}
\Omega^+ - \Omega \leq\ & \nu \log\left(\frac{(x+\alpha\Delta x)^T(s+\alpha\Delta s)}{x^T s}\right) \\
& + g(x)^T\Delta x + g^\star(s)^T\Delta s \\
& - \alpha\beta - \log(1-\alpha\beta) \\
\leq\ & \nu \log\left(\frac{(x+\alpha\Delta x)^T(s+\alpha\Delta s)}{x^T s}\right) \\
& + g(x)^T\Delta x + \nabla q_{\bar{x}}(s)^T\Delta s + e_g^T\Delta s \\
& - \alpha\frac{\lambda}{1-\eta} - \log(1-\alpha\frac{\lambda}{1-\eta}) \\
\leq\ & \frac{\alpha^2}{\mu}\Delta x^T\Delta s + \alpha\eta_\sigma((1-\sigma)\sqrt{\nu} - \eta_\sigma) \\
& + \alpha\frac{1}{\mu}\left\|\Delta s\right\|_{\star x}\mu\left\|e_r\right\|_x \\
& - \alpha\frac{\lambda}{1-\eta} - \log(1-\alpha\frac{\lambda}{1-\eta}).
\end{aligned} \tag{7.20}$$

We have explicit bounds for the norm of e_g. Recall that from (6.2.2) we have $\|g(y) - g(x) - H(x)(y-x)\|_{\star x} \leq r^2/(1-r)$, where $r = \|y-x\|_x$. This translates to

$$\|g^\star(s) - g^\star(-\mu g(x)) - H^\star(-\mu g(x))(s+\mu g(x))\|_{(H^\star(-\mu g(x)))^{-1}} \leq r^2/(1-r),$$

where $r = \|s+\mu g(x)\|_{H^\star(-\mu g(x))} = \frac{1}{\mu}\|s+\mu g(x)\|_{\star x} = \eta$. Since $H^\star(-\mu g(x)) = \frac{1}{\mu^2}H^{-1}(x)$, we have

$$\mu\left\|g^\star(s) - g^\star(-\mu g(x)) - H^\star(-\mu g(x))(s+\mu g(x))\right\|_x \leq \eta^2/(1-\eta)$$

and therefore $\mu\|e_g\|_x \leq \frac{\eta^2}{1-\eta}$. It then follows that

$$\alpha\frac{1}{\mu}\left\|\Delta s\right\|_{\star x}\mu\left\|e_r\right\|_x \leq \frac{\eta^2}{1-\eta}\left(\eta(1-\sigma) + \sqrt{\eta^2 + (1-\sigma)^2(\eta^2+\nu)}\right).$$

We can specialize these bounds for the two extremes $\sigma = 1$ and $\sigma = 0$. For the centering direction we have $\Delta x^T\Delta s = 0$ and $\lambda = \eta$. We also have that $\frac{1}{\mu}\|\Delta s\|_{\star x} \leq \eta$, and the bound therefore becomes

$$\begin{aligned}
\Omega^+ - \Omega \leq\ & -\alpha\eta^2 - \alpha\frac{\eta}{1-\eta} - \log(1-\alpha\frac{\eta}{1-\eta}) + \alpha\frac{\eta^3}{1-\eta} \\
\leq\ & -\alpha\eta^2 + \frac{\alpha^2\left(\frac{\eta}{1-\eta}\right)^2}{2(1-\alpha\frac{\eta}{1-\eta})} + \alpha\frac{\eta^3}{1-\eta}.
\end{aligned} \tag{7.21}$$

For the affine direction $\sigma = 0$, we have $\lambda \leq \eta + \sqrt{2\eta^2 + \nu}$, $\left|\frac{\Delta x^T \Delta s}{\mu}\right| \leq \eta^2 + \eta\sqrt{2\eta^2 + \nu}$, and since $\sqrt{\nu} - \eta_\sigma < 0$ we get the bound

$$\begin{aligned}\Omega^+ - \Omega \leq & \alpha^2\left(\eta^2 + \eta\sqrt{2\eta^2+\nu}\right) - \alpha\frac{\eta + \sqrt{2\eta^2+\nu}}{1-\eta} - \log\left(1 - \alpha\frac{\eta+\sqrt{2\eta^2+\nu}}{1-\eta}\right) \\ & + \alpha\frac{\eta^2}{1-\eta}\left(\eta + \sqrt{2\eta^2+\nu}\right).\end{aligned} \tag{7.22}$$

Correction phase

Since at all times $\eta \leq \frac{1}{4}$ and for the correction stage $\frac{1}{6} \leq \eta$, by assuming $\alpha \leq 1$ it is easy to show that

$$\begin{aligned}\Omega^+ - \Omega \leq & -\frac{2}{3}\alpha\eta^2 + \alpha^2\frac{1}{12} && (7.23)\\ \leq & -\alpha\frac{1}{54} + \alpha^2\frac{1}{12}. && (7.24)\end{aligned}$$

Therefore we can always reduce the value of Ω by a constant. For example, for a step size $\alpha = \frac{6}{52}$ then Ω can be reduced by -0.001.

The prediction phase

After a correction step, the functional centrality Ω is at least a constant C below the upper threshold $\bar{\Omega}$. We now use the bound (7.22) to show that in this case we can take a step that is bounded below by a multiple of $\frac{1}{\sqrt{\nu}}$. Because (7.22) is somewhat involved, it is useful to define some notation. Let $\gamma = \eta + \sqrt{2\eta^2 + \nu}$. Then recalling that $\eta \leq \frac{1}{4}$ it is easy to show that

$$\Omega^+ - \Omega \leq \alpha^2\eta\gamma + \frac{(\alpha\frac{\gamma}{1-\eta})^2}{2(1-\alpha\frac{\gamma}{1-\eta})} + \alpha\frac{\gamma}{12}. \tag{7.25}$$

If $\alpha \leq \frac{1-\eta}{2\gamma}$ then

$$\begin{aligned}\Omega^+ - \Omega \leq & \alpha\gamma + \alpha\frac{\gamma}{1-\eta} + \alpha\frac{\gamma}{12} && (7.26)\\ \leq & \alpha\frac{19}{12}\gamma && (7.27)\end{aligned}$$

Therefore if $\Omega \leq \bar{\Omega} - C$, a step of size $\alpha = \min\{\frac{1-\eta}{2\gamma}, \frac{12C}{19}\}\frac{1}{\gamma}$ will maintain $\Omega \leq \bar{\Omega}$, and since $\gamma = \eta + \sqrt{2\eta^2 + \nu}$ we have the result. To conclude, we have shown that a predictor-corrector algorithm can be constructed that uses the functional centrality measure and is able to take prediction steps of order $\frac{1}{\sqrt{\nu}}$.

Chapter 8

Conjugate barriers for the exponential cone

In this chapter we define a conjugate pair of barriers for the exponential cone and its dual. Here the function $f(x, y, z)$ is the barrier for the exponential cone defined in [12]:

Note that f is LHCSB with v=3

$$f(x, y, z) = -\log\left(z \log\left(\frac{y}{z}\right) - x\right) - \log(y) - \log(z),$$

and $f^\star(u, v, w)$ is the conjugate function:

$$f^\star(u, v, w) = -2\log(-u) - \log(v) - \log\left(\frac{(1-\bar{\omega})^2}{\bar{\omega}}\right) - 3,$$

where $\bar{\omega} = \omega(2 - \frac{w}{u} - \log(-u) + \log(v))$ and ω is the Wright Omega function.

8.1 The Wright Omega function

The Wright Omega function [13] denoted $\omega(\beta) : \mathbb{R} \to \mathbb{R}_+$ is defined as the unique solution to the equation

$$\omega(\beta) + \log(\omega(\beta)) = \beta. \tag{8.1}$$

The function $\omega(\beta)$ is continuous, defined over all the reals and takes positive values in all its domain. Trivially $\omega(1) = 1$, when $\beta \to -\infty$ then $\omega(\beta) \to 0$, and as $\beta \to \infty$, $\omega \to \infty$.

Differentiating (8.1) yields $\omega'(\beta) + \frac{\omega'(\beta)}{\omega(\beta)} = 1$ and therefore

$$\omega'(\beta) = \frac{\omega(\beta)}{1 + \omega(\beta)}. \tag{8.2}$$

Differentiating again yields the expression for $\omega''(\beta) = \frac{\omega(\beta)}{(1+\omega(\beta))^3}$. We conclude that $\omega(\beta)$ is monotonically increasing and convex.

8.2 The conjugate function

In this section we prove that f and $f^\star$ are conjugate. We start by the following lemma, which shows that the gradient of f is invertible and we can calculate its inverse.

Lemma 8.2.1. *The mapping*

$$\tilde{g}(u,v,w) = -\begin{pmatrix} \frac{3-\frac{w}{u}-2\bar{\omega}}{u(1-\bar{\omega})} \\ \frac{-\bar{\omega}}{v(1-\bar{\omega})} \\ \frac{1}{u(1-\bar{\omega})} \end{pmatrix}, \tag{8.3}$$

with $\bar{\omega} = \omega\left(2 - \frac{w}{u} - \log(-u) - \log(v)\right)$ *is such that* $-g(-\tilde{g}(u,v,w)) = (u,v,w)^T$.

Proof. Let g be the gradient of f. It is easy to see that

$$-g(x,y,z) = \begin{pmatrix} \frac{1}{r} \\ \frac{1}{y} - \frac{z}{yr} \\ \frac{1}{z} - \frac{\ell-1}{r} \end{pmatrix}$$

where $\ell = \log\frac{y}{z}$ and $r = x - z\ell$.

Substituting $x = -\tilde{g}_1$, $y = -\tilde{g}_2$ and $z = -\tilde{g}_3$ in the definition of ℓ and r yields

$$\ell = \log(\frac{-u}{v}) + \log(\bar{\omega}) \tag{8.4}$$

$$r = \frac{3-\frac{w}{u}-2\bar{\omega}}{u(1-\bar{\omega})} - \frac{\log(\frac{-u}{v}) + \log(\bar{\omega})}{u(1-\bar{\omega})} \tag{8.5}$$

$$= \frac{3-\frac{w}{u}-2\bar{\omega} - \log(\frac{-u}{v}) - \log(\bar{\omega})}{u(1-\bar{\omega})} = \frac{1}{u}, \tag{8.6}$$

where for the last equality we used that $\bar{\omega} + \log(\bar{\omega}) = 2 - \frac{w}{u} - \log(-u) - \log(v)$. By substituting into the second entry of $-g$ we get

$$\frac{1}{y} - \frac{z}{yr} = \frac{1}{y}(1 - zu) = \frac{v(1-\bar{\omega})}{-\bar{\omega}}\left(1 - \frac{u}{u(1-\bar{\omega})}\right) = v,$$

and by substituting into the third entry we get

$$\frac{1}{z} - \frac{\ell-1}{r} = \frac{1}{z} - (\ell-1)u = u(2 - \bar{\omega} - \ell) = u(2 - \bar{\omega} - \log\frac{-u}{v} - \log\bar{\omega}) = w.$$

Therefore $-g(-\tilde{g}(u,v,w)) = (u,v,w)^T$. □

Recalling the definition of $f^\star(u,v,w) = -\inf_{x,y,z}\{f(x,y,z) + xu + yv + zw\}$, and noting that the optimality condition for the minimization is given by the equation $-g(x^\star,y^\star,z^\star) = (u,v,w)^T$, we can conclude that because $-g(-\tilde{g}(u,v,w)) = (u,v,w)^T$, the entries of $-\tilde{g}(u,v,w)$ are the minimizing $x^\star, y^\star, z^\star$.

From the properties of conjugate pairs of functions (4.3.5) we know that $-g^\star = \arg\min_{x,y,z}\{f(x,y,z) + xu + yv + zw\}$, so that $g^\star = \tilde{g}$. Finally to show that $f^\star$ is conjugate to f it remains to show that $f^\star = -f(-g^\star) - x^\star u - y^\star v - z^\star w$.

Theorem 8.2.2. *The function $f^\star$ is conjugate to f.*

Proof. We now evaluate $-f(x^\star, y^\star, z^\star) - x^\star u - y^\star v - z^\star w$. Observe that $f = -\log(-r) - \log(y) - \log(z)$ and therefore

$$f(x,y,z) = -\log(-\frac{1}{u}) - \log\left(\frac{-\bar{\omega}}{v(1-\bar{\omega})}\right) - \log\left(\frac{1}{u(1-\bar{\omega})}\right) \tag{8.7}$$

$$= \log(-u) + \log(v) + \log\left(\frac{1-\bar{\omega}}{-\bar{\omega}}\right) + \log\left(u(1-\bar{\omega})\right) \tag{8.8}$$

$$= 2\log(-u) + \log(v) + \log\left(\frac{(1-\bar{\omega})^2}{\bar{\omega}}\right). \tag{8.9}$$

On the other hand,

$$x^\star u + y^\star v + z^\star w = \frac{3 - \frac{w}{u} - 2\bar{\omega} - \bar{\omega} - \frac{w}{u}}{1-\bar{\omega}} = 3. \tag{8.10}$$

Therefore

$$-f(x^\star, y^\star, z^\star) - x^\star u - y^\star v - z^\star w = -2\log(-u) - \log(v) - \log\left(\frac{(1-\bar{\omega})^2}{\bar{\omega}}\right) - 3.$$

□

8.3 A second pair of conjugate functions

In [12] the author notes that there exists a linear transformation $B : \mathbb{R}^3 \to \mathbb{R}^3$ such that $\operatorname{int}\mathcal{K}^\star = B[\operatorname{int}\mathcal{K}]$ given by the matrix

$$B = \begin{pmatrix} & & -1 \\ & 1/e & \\ -1 & & \end{pmatrix}. \tag{8.11}$$

Therefore the function $\tilde{f}^\star(u,v,w) = f(B^{-1}(u,v,w)^T)$ is a self-concordant 3-logarithmically homogeneous barrier for the dual cone and $\tilde{f}(x,y,z) = f^\star(B(x,y,z)^T)$ is its conjugate barrier for the primal exponential cone. These functions take the form

$$\tilde{f}(x,y,z) = -2\log(z) - \log(y) - \log\left(\frac{(1-\bar{\omega})^2}{\bar{\omega}}\right) - 3$$

with $\bar{\omega} = \omega(1 - \frac{x}{z} - \log z - \log y)$, and

$$\tilde{f}^\star(u,v,w) = -\log(w - u - u\log\frac{-v}{u}) - \log(-u) - \log(v).$$

For notational convenience, let r = log(-v/u) and t = u - w + u*r.
One may write f_exp_tilde_dual(u,v,w) = -log(-t)-log(-u)-log(v).

8.4 Evaluating the Wright Omega function

In [31] Laurence et al. describe an algorithm for evaluating of the Wright Omega function over the complex plane. The authors separate the complex plane into 7 different regions and define an approximation algorithm for each region. Since we are only interested in the evaluation of ω for real numbers and all arguments in conic optimization with exponential cones will be positive and larger than 1, only two of these approximation schemes are of interest. We have implemented our own version of their algorithm, restricted to real positive arguments larger than 1.

Algorithm 4 describes our version of the algorithm. When the argument is between 1 and $1+\pi$ we initialize w with the value of the Taylor series about 1. When the argument is larger than $1+\pi$ we initialize w with the value of the series about ∞. After initializing we proceed to do two rounds of the refinement procedure described in [31] and return the resulting value.

Algorithm 4 Wright omega function for $z > 1$

if $1 \leq z < 1+\pi$ **then**
  $w \leftarrow 1 + \frac{1}{2}(z-1) + \frac{1}{16}(z-1)^2 - \frac{1}{192}(z-1)^3 - \frac{1}{3072}(z-1)^4 + \frac{13}{61440}(z-1)^5$
else if $z > 1+\pi$ **then**
  $w \leftarrow z - \log z + \frac{\log z}{z} + \frac{\log z}{z^2}\left(\frac{1}{2}\log z - 1\right) + \frac{\log z}{z^3}\left(\frac{1}{3}\log^2 z - \frac{3}{2}\log z + 1\right)$
else the argument is outside the range
  return -1
end if
▷ Refinement
$k \leftarrow 0$
$r_0 \leftarrow z - w - \log w$
$w_0 \leftarrow w$
for $k < 2$ **do**
  $w_{k+1} = w_k\left(1 + \frac{r_k}{1+w_k}\frac{(1+w_k)(1+w_k+2/3r_k)-r_k/2}{(1+w_k)(1+w_k+2/3r_k)-r_k}\right)$
  $r_{k+1} = \frac{2w_k^2-8w_k-1}{72(1+w_k)^6}r_k^4$
  $k \leftarrow k+1$
end for
Return w_k

8.4.1 Numerical evaluation of the Wright Omega Real implementation

We sampled 10^9 different values for z, and for each we calculated the relative residual $\mathrm{err}(z) = |z - w(z) - log(w(z))|/|z|$. The points were generated by the iteration $z_k = sz_k$ with s of the form $s = (1+\epsilon)^p$, where ϵ is the machine precision and $p = 2^{27}$. By leaving larger gaps between samples of larger magnitude, this sampling strategy chooses sample points that are equispaced in the sequence of all representable floating-point numbers. With this particular choice of s the first two values sampled are $z_1 = 1$ and $z_2 \approx 1 + 3 \times 10^{-8}$. The largest after

10^9 samples was close to 8.7×10^{12}. The largest resulting relative error was $\mathrm{err}(1 + 10^{-6}) \approx 4.5 \times 10^{-16}$.

Chapter 9

Modeling convex problems with the exponential cone

In this chapter we define conically representable functions and conically representable problems. Our objective is to show how some important convex optimization problems can be transformed into conic programming problems expressed in terms of the positive orthant, the Lorentz cone, and the exponential cone. We do not wish to describe the full family of problems that can be transformed into conic programming problems; instead we show the value of allowing for the exponential cone in the formulation of conic programming problems. For a more complete treatment we direct the reader to the work of Michael Grant [23], where *disciplined convex programming (DCP)* is defined (a series of rules to build and transform convex problems into equivalent problems for which solvers exist), and also to the work of Nemirovski [38, 8], where $\mathcal{K}$-representable sets are defined (all sets representable from simple operations on cones).

9.1 Conically representable functions

We say that $f : \mathbb{R}^n \to \mathbb{R}$ is conically representable if the epigraph $\operatorname{Epi} f = \{(x,t) \mid f(x) \leq t\}$ is the same as the set of pairs (x,t) (where $x \in \mathbb{R}^n$ and $t \in \mathbb{R}$) such that

$$A \begin{pmatrix} x \\ u \\ t \end{pmatrix} = b, \quad G \begin{pmatrix} x \\ u \\ t \end{pmatrix} + h \in \mathcal{K} \tag{9.1}$$

for some matrices A, G and vectors b, h and possibly some auxiliary variables u. We call the set of constraints $Ax = b$ equality constraints and $Gx + h \in \mathcal{K}$ conic constraints. For completeness we adopt the convention from [23, 49] that a convex function evaluated outside its domain takes the value ∞ and therefore there is no pair $(x,t) \in \operatorname{Epi}(f)$ for $x \notin \operatorname{Dom}(f)$ with finite t.

We now describe some simple operations on conically representable functions that maintain them conically representable.

9.1.1 Sums of conically representable functions

Assume f_1 and f_2 are conically representable with $A_1, A_2, G_1, G_2, b_1, b_2, h_1, h_2, K_1, K_2$ and auxiliary variables u_1 and u_2. Then $g(x) = f_1(x) + f_2(x) \leq t$ iff

$$\begin{aligned} A_1 \begin{pmatrix} x \\ u_1 \\ t_1 \end{pmatrix} = b_1, \quad G_1 \begin{pmatrix} x \\ u_1 \\ t_1 \end{pmatrix} + h_1 \in \mathcal{K}_1, \\ A_2 \begin{pmatrix} x \\ u_2 \\ t_2 \end{pmatrix} = b_2, \quad G_2 \begin{pmatrix} x \\ u_2 \\ t_2 \end{pmatrix} + h_2 \in \mathcal{K}_2, \\ -t_1 - t_2 + t \geq 0. \end{aligned} \tag{9.2}$$

By reinterpreting t_1 and t_2 as auxiliary variables and defining the cone $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \mathbb{R}_+$, we can reorganize those conditions into the form (9.1), so that $g(x)$ is conically representable.

9.1.2 Affine transformation of the arguments

If f is conically representable, then the function $g(x) = f(Bx+d)$, where B is a matrix and d is a vector, is conically representable. Observe that the epigraph of $g(x)$ is the set of all (x,t) such that there exists a $y = Bx+d$ with $(y,t) \in \mathrm{Epi}(f)$. However, this is equivalent to the conditions

$$A \begin{pmatrix} y \\ u \\ t \end{pmatrix} = b, \qquad G \begin{pmatrix} y \\ u \\ t \end{pmatrix} + h \in \mathcal{K}, \qquad Bx + d = y,$$

which is the same as

$$\begin{aligned} A \begin{pmatrix} B & & \\ & I & \\ & & I \end{pmatrix} \begin{pmatrix} x \\ u \\ t \end{pmatrix} = b - A \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix}, \\ G \begin{pmatrix} B & & \\ & I & \\ & & I \end{pmatrix} \begin{pmatrix} x \\ u \\ t \end{pmatrix} + G \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix} + h \in \mathcal{K}. \end{aligned}$$

This is equivalent to (9.1) and therefore g is conically representable.

9.1.3 Sums of functions defined over different variables

Assume $f_1(x_1)$ and $f_2(x_2)$ are functions defined for variables x_1 and x_2. We can trivially redefine $f_1(x_1)$ and $f_2(x_2)$ to be a function of both variables

$\hat{f}_1(x_1, x_2) = f_1(x_1)$ and $\hat{f}_2(x_1, x_2) = f_2(x_2)$. These are affine transformations of the argument of f_1 and f_2 and therefore $\hat{f}_1$ and $\hat{f}_2$ are conically representable. Finally the sum $g(x_1, x_2) = f_1(x_1) + f_2(x_2) = \hat{f}_1(x_1, x_2) + \hat{f}_2(x_1, x_2)$ is conically representable.

9.1.4 Multiple by positive constant

If $f(x)$ is conically representable, then $\lambda f(x)$ is conically representable for $\lambda > 0$. Any $(x, t) \in \text{Epi}(\lambda f)$ if and only if $(x, \frac{t}{\lambda}) \in \text{Epi}(f)$. Hence $(x, t) \in \text{Epi}(\lambda f)$ iff

$$A \begin{pmatrix} I & & \\ & I & \\ & & \frac{1}{\lambda} \end{pmatrix} \begin{pmatrix} x \\ u \\ t \end{pmatrix} = b,\ G \begin{pmatrix} I & & \\ & I & \\ & & \frac{1}{\lambda} \end{pmatrix} \begin{pmatrix} x \\ u \\ t \end{pmatrix} + h \in \mathcal{K}.$$

9.1.5 Maxima of conically representable functions

If $f_1(x)$ and $f_2(x)$ are conically representable, then $g(x) = \max\{f_1(x), f_2(x)\}$ is conically representable. Observe that $g(x) \leq t$ if and only if $f_1(x) \leq t_1 \leq t$ and $f_2(x) \leq t_2 \leq t$, which is the same set as

$$\begin{aligned} A_1 \begin{pmatrix} x \\ u_1 \\ t_1 \end{pmatrix} = b_1,\ G_1 \begin{pmatrix} x \\ u_1 \\ t_1 \end{pmatrix} + h_1 \in \mathcal{K}_1, \\ A_2 \begin{pmatrix} x \\ u_2 \\ t_2 \end{pmatrix} = b_2,\ G_2 \begin{pmatrix} x \\ u_2 \\ t_2 \end{pmatrix} + h_2 \in \mathcal{K}_2, \\ t_1 \leq t,\ t_2 \leq t. \end{aligned}$$

9.2 Examples of conically representable functions

9.2.1 Negative Entropy

Let $\phi(x) = x\log(x)$ for $x \in \mathbb{R}$. We call ϕ the negative entropy function on one variable. This function is convex in the interval $[0, \infty)$.

Observe that the triplet $(t, u, x) \in \mathcal{K}_e$ if and only if $-t \geq x \log\left(\frac{x}{u}\right)$, and every triplet $(t, u, x) \in \mathcal{K}_e$ such that $u = 1$ has the property that $(x, -t) \in \text{Epi}(\psi)$. Therefore we can write $\text{Epi}(\psi)$ as

$$\begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} t \\ u \\ x \end{pmatrix} = 1 \qquad \begin{pmatrix} -1 & & \\ & 1 & \\ & & 1 \end{pmatrix} \begin{pmatrix} t \\ u \\ x \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \in \mathcal{K}_e.$$

We can then define the negative entropy function in n variables $\phi_n(x) : \mathbb{R}^n \to \mathbb{R}$ as $\phi_n(x) = \sum x_i \log(x_i)$. Since this is the sum of n conically representable variables, it is itself conically representable.

9.2.2 Kullback-Leibler divergence

The Kullback-Leibler divergence (KL) between p and q, denoted $D(p||q)$ and defined for probability distributions over finite sets, is conically representable as a function of p. If p and q are defined by vectors $p, q \in \mathbb{R}^K_+$, then $D(p||q) = \sum_i^K p_i \log \frac{p_i}{q_i}$. Let $d(p_i, q_i) = p_i \log \frac{p_i}{q_i} = p_i \log p_i - p_i \log q_i$. It is clear that $d(p_i, q_i)$ is conically representable, for it is the sum of the negative entropy function and a term linear in p_i. Writing $D(p||q) = \sum_i d(p_i, q_i)$ we conclude that $D(p||q)$ is conically representable as a function of p.

The exponential function is conically representable. Observe that any triplet $(x, t, u) \in \mathcal{K}_e$ satisfies $\exp(\frac{x}{u}) \leq \frac{t}{u}$ and therefore any triplet $(x, t, 1) \in \mathcal{K}_e$ satisfies $\exp(x) \leq t$. Therefore $(x, t) \in \text{Epi}(e^x)$ for any $(x, t, u) \in \mathcal{K}_e$ with $u = 1$. And we can write the epigraph of the exponential function as

$$\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ t \\ u \end{pmatrix} = 1, \qquad \begin{pmatrix} x \\ t \\ u \end{pmatrix} \in \mathcal{K}_e.$$

9.2.3 Logarithm of sum of exponentials

Define by $\psi(x) : \mathbb{R}^n \to \mathbb{R}$ the function $\psi(x) = \log\left(\sum_i e^{x_i}\right)$. This function is also conically representable, for observe that $\psi(x) \leq t$ if and only if $\sum_i e^{x_i - t} \leq 1$. To represent $\psi(x)$ as a set of conic constraints, for each x_i we introduce the auxiliary triplet u_i, v_i, w_i, and establish the constraints $w_i = 1$, $x_i - t = u_i$ and $\left(u_i, v_i, w_i\right)^T \in \mathcal{K}_e$ for all i, and the constraint $-v_1 - \cdots - v_n + 1 \geq 0$.

9.2.4 Negative logarithm

The function $-\log(x)$ is convex and conically representable. Because $\left(t, x, w\right)^T \in \mathcal{K}_e$ if and only if $-t/w \geq -\log(x/w)$ and $w > 0$, then $-\log(x) \leq t$ if and only if $\left(-t, x, 1\right)^T \in \mathcal{K}_e$.

9.2.5 Two norm

The function $\psi(x) = \|x\|_2$ is conically representable, albeit using the SOCP cone and not the exponential cone, for $\psi(x) \leq t$ iff $\left(x, t\right) \in \mathcal{L}^\star$.

9.2.6 Two norm squared

The function $\psi(x) = \|x\|_2^2$ is also conically representable, for

$$\sqrt{x^T x + \left(\frac{1}{4} - z\right)^2} \leq \frac{1}{4} + z \tag{9.3}$$

if and only if $x^T x \leq z$. Finally, constraint (9.3) is equivalent to $\left(x, \frac{1}{4} - z\right)^T \in \mathcal{L}^\star$, where $\mathcal{L}^\star$ is the SOCP cone.

9.2.7 One norm

The function $\psi(x) = \|x\|_1$ is conically representable, for $\psi(x) \leq t$ iff $u_i - v_i = x_i$ and $u_i \geq 0$ and $v_i \geq 0$ and $\sum_i x_i + v_i \leq t$.

9.2.8 Linear functions

The function $\psi(x) = c^T x + b$ is linearly representable, for $\psi(x) \leq t$ iff $(c^T, -1)(x^T, t)^T = -b$.

9.3 An alternative standard form

The standard form we use here differs from the one used previously but it is consistent with the input format for ECOS, CVXOPT and other solvers. Some of the analysis of the previous section has to be adapted, but the changes are mostly cosmetic. In fact, any problem in the standard form used in the previous chapters has an equivalent problem in the format described here, and conversely any problem in this new format can be transformed into an equivalent problem in the original standard form.

In the next chapter we describe our extension to ECOS in more detail. Meanwhile we refer to [17] for the description of ECOS and to [7] for the description of the CVXOPT algorithm used by ECOS.

ECOS solves problems of the form

$$\begin{aligned} \min\ & c^T x \\ \text{subject to } & Ax = b, \\ & Gx + s = h, \\ & s \in \mathcal{K}. \end{aligned} \tag{Sn}$$

Note that here s denotes the *primal* slacks.

If a problem is defined in the original standard form

$$\begin{aligned} \text{minimize } & \hat{c}^T \hat{x} \\ \text{subject to } & \hat{A}\hat{x} = \hat{b}, \\ & \hat{x} \in \hat{\mathcal{K}}, \end{aligned} \tag{So}$$

then choosing $G = -I$ and $h = 0$ in problem (Sn) yields an equivalent problem. On the other hand if a problem is given in format (Sn), then with

$$\begin{aligned} \hat{A} &= \begin{pmatrix} A & -A & \\ G & -G & I \end{pmatrix}, \\ \hat{x} &= (x_+, x_-, s)^T, \\ \hat{c} &= (c, -c, 0)^T, \\ \hat{\mathcal{K}} &= \mathbb{R}^n_+ \times \mathbb{R}^n_+ \times \mathcal{K}, \end{aligned}$$

problem (So) will be equivalent. (Here the length of the zero vector in $\hat{c}$ is the same as the dimension of s, the n in the Cartesian cone is the same as the dimension of x, and the symbols x_- and x_+ refer to the positive and negative parts of x.)

9.4 Conic programming problems

A *conically representable problem* is a convex problem of the form

$$\begin{aligned} \text{minimize } & f_0(x), \\ \text{subject to } & f_1(x) \leq 0, \\ & f_2(x) \leq 0, \\ & \vdots \\ & f_p(x) \leq 0, \\ & A_{p+1}x = b_{p+1}, \\ & G_{p+1}x + h_{p+1} \in \mathcal{K}_{p+1}, \end{aligned} \tag{CP}$$

where all the $f_i(x)$ are conically representable. As we show now, conically representable problems are easily transformed into problems in the conic standard form (Sn).

Since the functions f_i in the constraints are conically representable, the constraints $f_i(x) \leq t_i \leq 0$ are equivalent to

$$A_i \begin{pmatrix} x \\ u_i \\ t_i \end{pmatrix} = b_i,\ G_i \begin{pmatrix} x \\ u_i \\ t_i \end{pmatrix} + h_i \in \mathcal{K}_i,\ t_i \leq 0.$$

On the other hand the constraint $f_0(x) \leq t$ is equivalent to

$$A_i \begin{pmatrix} x \\ u_0 \\ t_0 \end{pmatrix} = b_0, \; G \begin{pmatrix} x \\ u_0 \\ t_0 \end{pmatrix} + h_0 \in \mathcal{K}_0,$$

and it is easy to see that there exists a matrix A and vector b such that $A\left(x, u, t\right)^T = b$ if and only if $A_i\left(x, u_i, t_i\right)^T = b_i$ for all $i = 0, \ldots, p+1$. And that there exist a matrix G and cone $\mathcal{K}$ such that $G\left(x, u, t\right)^T + h \in \mathcal{K}$ if and only if $G_i x + h_i \in \mathcal{K}_i$ for all $i = 0, \ldots, p+1$, and $t_i \leq 0$ for all $i > 0$. Using these matrices A, G and vectors b and h, the final conic problem is of the form

$$\begin{aligned} &\text{minimize } t_0 \\ &\text{subject to } A \begin{pmatrix} x \\ u \\ t \end{pmatrix} = b, \\ &\qquad\qquad G \begin{pmatrix} x \\ u \\ t \end{pmatrix} + h \in \mathcal{K}. \end{aligned} \tag{9.4}$$

It is worth noting that this transformation is not unique but serves our objective of showing that all conically representable problems are in fact equivalent to a conic programming problem.

9.4.1 Logistic regression

Logistic regression is a model for the probability that a vector $x \in \mathbb{R}^p$ belongs to one of K classes, given a linear function of the covariates x [26].

We use the $K = 2$ instance as an example, and assume that we have access to a training set formed by N samples $(x, y) \in \mathbb{R}^p \times \{0, 1\}$, where $y_i \in \{0, 1\}$ and x_i takes values in $\mathbb{R}^p$. We assume that the probability that $y_i = 1$ given x is given by

$$P(Y = 1 \mid X = x) = \frac{\exp\left(\beta^T x + \beta_0\right)}{1 + \exp\left(\beta^T x + \beta_0\right)},$$

where β is a vector and β_0 a scalar.

Let I_1 be the indices of the samples where $y_i = 1$ and let I_0 be the indices where $y_i = 0$, trivially $I_1 \cup I_0 = 1, \ldots, N$. The likelihood of any observed sequence of samples is given by

$$L(\beta, \beta_0) = \prod_{i \in I_0} \frac{1}{1 + \exp\left(\beta^T x_i + \beta_0\right)} \prod_{i \in I_1} \frac{\exp\left(\beta^T x_i + \beta_0\right)}{1 + \exp\left(\beta^T x_i + \beta_0\right)},$$

and the log-likelihood is given by

$$\ell(\beta, \beta_0) = -\sum_{i=1}^{N} \log\left(1 + \exp\left\{\beta^T x_i + \beta_0\right\}\right) + \sum_{i \in I_1} (\beta^T x_i + \beta_0).$$

We are interested in fitting β, β_0 using a maximum likelihood estimator. Therefore we need to maximize $\ell(\beta, \beta_0)$.

We now show that the above can be solved by an exponential conic programming problem. Letting $\hat{x} = \sum_{i \in I_0} x_i$ and $t \in \mathbb{R}_N, w \in \mathbb{R}_N$, and recalling that $\log \sum_j e^{x_j} \leq -u_j$ iff $\sum_j e^{x_j + u_j} \leq 1$, we can see that the maximum likelihood problem is equivalent to

$$\begin{aligned} \text{minimize } & -\sum_i u_i - \hat{x}^T \beta - |I_0| \beta_0 \\ \text{subject to } & \beta^T x_i + \beta_0 = w_i \qquad\qquad i \in 1, \ldots, N \\ & \exp(u_i) + \exp(w_i + u_i) \leq 1, \quad i \in 1, \ldots, N. \end{aligned} \tag{9.5}$$

Since (9.5) is formed by conically representable functions, it is conically representable.

9.4.2 Sparse logistic regression

Sparse logistic regression adds a regularization term $\|\beta\|_1$ to the maximum likelihood objective in order to achieve a sparser model β [26]. The regularization term is weighted by the constant $\lambda > 0$, which modifies its relative importance. The regularized maximum likelihood objective takes the form

$$\ell_r(\beta, \beta_0, \lambda) = \sum_{i=1}^{N} \log\left(1 + \exp\left\{\beta^T x_i + \beta_0\right\}\right) - \sum_{i \in I_1} (\beta^T x_i + \beta_0) + \lambda \|\beta\|_1 .$$

Because both $\ell(\beta, \beta_0)$ and $\lambda \|\beta\|_1$ are conically representable, the minimization of ℓ_r can be done via an equivalent conic programming problem.

Other important regularization terms $\psi(\beta, \beta_0)$ exist, and as long as ψ is a conically representable function the resulting regularized maximum likelihood objective can be maximized using a conic programming solver, for in that case the function

$$\ell_\psi(\beta, \beta_0) = \sum_{i=1}^{N} \log\left(1 + \exp\left\{\beta^T x_i + \beta_0\right\}\right) - \sum_{i \in I_1} (\beta^T x_i + \beta_0) + \psi(\beta, \beta_0)$$

is conically representable.

9.4.3 Minimum Kullback-Leibler divergence

Assume that a probability distribution p over the set $\mathcal{X} = \{1, \ldots, K\}$ is known to belong to a set P that can be written as $P = \{p \mid f(p) \leq 0\}$, for some conically representable f. And assume we know a probability distribution $q \notin P$ that is a good approximation to some optimal $p^\star$. The problem of finding the distribution $p \in P$ that minimizes the Kullback-Leibler divergence is given by [11]

$$\begin{aligned} \text{minimize } & \sum_i^K p_i \log \frac{p_i}{q_i} \\ \text{subject to } & f(p) \leq 0, \end{aligned} \tag{9.6}$$

and as we have shown, the objective and constraints are conically representable.

9.4.4 Geometric programming

Geometric programming problems are a family of nonlinear optimization problems that have applications in many areas. For a detailed tutorial on geometric programming, see [10] and references therein for applications in circuit design, chemical engineering, information theory, probability, computational finance, etc.

Geometric programming problems are not directly conically representable. However, after a nonlinear transformation of their variables the resulting optimization problems are. These problems are written in terms of mathematical objects called monomials.

Let $x_1, \ldots, x_p$ be positive variables and $x = (x_1, \ldots, x_p)$ be a vector. The *monomial* $m(x)$ is defined as the function $m(x) = cx_1^{a_1}x_2^{a_2}\cdots x_p^{a_p}$, where all the a_i are real and $c > 0$. A sum of monomials $f(x) = \sum_i^k c_i \prod_j x_j^{a_j}$ is called a posynomial.

A geometric programming problem is a problem of the form

$$\begin{aligned} \text{minimize } & f_0(x) \\ \text{subject to } & f_i(x) \leq 1, \quad i = 1, \ldots, m \\ & g_i(x) = 1, \quad i = 1, \ldots, p \\ & 0 < x_i, \qquad i = 1, \ldots, n \end{aligned} \tag{9.7}$$

where $f_0, \ldots, f_m$ are posynomials, and $g_i, \ldots, g_p$ are monomials.

Even though this problem is not convex and therefore not conically representable, the variable transformation $\exp(y_i) = x_i$ yields a convex problem on y_i. For observe that in the new variables any monomial $g_i(e^y) = \exp\left(\sum_j a_j y_j + \log(c_i)\right)$ results from the affine transformation of the arguments of the exponential function, and is therefore conically representable. Furthermore, any posynomial is itself the sum of conically representable functions and therefore conically representable. We conclude that the problem

$$\begin{aligned} \text{minimize } & f_0(e^y) \\ \text{subject to } & f_i(e^y) \leq 1, \quad i = 1, \ldots, m \\ & g_i(e^y) = 1, \quad i = 1, \ldots, p \end{aligned} \tag{9.8}$$

is conically representable and equivalent to the original geometric programming problem. Therefore a conic solver that supports the exponential cone can solve all geometric programing problems, and furthermore solve any problem from the larger class of problems for which the transformation $e^y = x$ yields conically representable problems. This class is defined in [10] under the name *generalized geometric programming*.

Chapter 10

Extending ECOS to solve problems with the exponential cone

ECOS [16] is an ANSI-C implementation of an interior-point method designed to solve symmetric cone problems modeled with products of second-order cones and linear constraints. We have extended ECOS into ECOS-Exp, an implementation that can solve problems modeled with products of second-order cones, exponential cones and linear constraints. In this section we describe our extension and some of the implementation choices made to build an efficient solver.

Recall that the difficulty of a conic programming problem is given in terms of ν (the complexity of the barrier). ECOS is based on the exceedingly successful Mehrotra predictor-corrector algorithm (MPC), which increases its iteration count very slowly as ν grows. This Mehrotra predictor-corrector algorithm approximates the central path by a second-order Taylor expansion. At the same time it can take long steps within the feasible region as it does not require that its iterates stay close to the central path. In this section we explore our variant of the Mehrotra predictor-corrector. This variant has two important differences with respect to the original MPC. It restricts the iterates to a region close to the central path, and since, for the variables corresponding to the exponential cones no second-order information is available, it only approximates the direction of the central path with respect to the variables in the exponential cone to first order. Remarkably, we have discovered that our extension also increases its iteration count very slowly as ν increases, yielding an implementation that is practical to use.

In Chapter 5 we described the potential reduction algorithms for general conic programming. These algorithms can be used to solve any conic programming problem, provided conjugate, self-concordant barriers are known for the primal and dual cone. Their downside is that the Hessian of both barriers must be used when solving for the search direction, essentially doubling the size of the

Newton system. This potentially increases the iteration cost by up to a factor of 8. In ECOS-Exp we keep the size of the Newton system comparable to that of a symmetric cone problem, meaning that the size of the matrix is the same as for an alternative problem with the same number of variables and constraints but only with symmetric cones. To achieve this, our method keeps the iterates close to the central path in a similar way to that described in Chapter 7. The restriction of the iterates to a vicinity of the central path implies that the initialization procedure ECOS uses can no longer be used. Instead, all variables are initialized on a *central ray*.

This section is organized as follows. We begin by describing the format of the problems ECOS solves. We then describe the variant of the Mehrotra predictor-corrector algorithm on which ECOS is based. We proceed to describe how we extended ECOS into ECOS-Exp. There we define the search directions and we explain how we enforce the centrality condition and how we initialize the iterates.

We then proceed with numerical experiments to evaluate ECOS-Exp by solving random conic programming problems, negative entropy minimization problems, and geometric programming problems. We also compare against PDCO [50] and MOSEK [5].

10.1 ECOS for symmetric cones

A full description in ECOS and the MPC algorithm can be found in [16, 7] and [17]. Our description is more succinct for we only wish to give the minimum necessary background for the ECOS-Exp extension.

The standard form used by ECOS differs from the definition used in the previous chapters and the notation has to be updated. However, the changes are mostly cosmetic and don't pose important theoretical modifications. ECOS solves problems of the form

$$\begin{aligned} \min\, & c^T x \\ \text{subject to } & Ax = b, \\ & Gx + s = h, \\ & s \in \mathcal{K}, \end{aligned} \tag{Sn}$$

with dual

$$\begin{aligned} \max\, & -b^T y - h^T z, \\ \text{subject to } & G^T z + A^T y + c = 0, \\ & z \in \mathcal{K}^\star. \end{aligned} \tag{10.1}$$

For this primal-dual pair, s denotes the *primal* slacks and z denotes the *dual* slacks. The simplified homogeneous self-dual embedding for the standard form used by ECOS is the problem of finding a nonzero solution of the feasibility

problem

$$\begin{aligned}&\text{minimize } 0\\&\text{subject to}\\&\begin{pmatrix} & G^T & A^T & c\\ -A & & & b\\ -G & & & h\\ -c^T & -b^T & -h^T & \end{pmatrix}\begin{pmatrix}x\\y\\z\\\tau\end{pmatrix}-\begin{pmatrix}0\\0\\s\\\kappa\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\end{pmatrix}\\&z\in\mathcal{K}^\star,\ s\in\mathcal{K},\ \tau>0,\ \kappa>0.\end{aligned}\tag{10.2}$$

The changes apply to all subsequent coefficient matrices

Given an initial point $x_0, s_0, y_0, z_0, \tau_0, \kappa_0$ we define the residuals

$$\begin{pmatrix}r_x\\r_y\\r_z\\r_\tau\end{pmatrix}=-\begin{pmatrix} & G^T & A^T & c\\ -A & & & b\\ -G & & & h\\ -c^T & b^T & h^T & \end{pmatrix}\begin{pmatrix}x\\y\\z\\\tau\end{pmatrix}+\begin{pmatrix}0\\0\\s\\\kappa\end{pmatrix}.$$

The central path is the unique solution of the equations

$$s-\mu g^\star(z)=0\tag{10.3a}$$

$$\tau\kappa-\mu=0\tag{10.3b}$$

$$\begin{pmatrix} & G^T & A^T & c\\ -A & & & b\\ -G & & & h\\ -c^T & b^T & h^T & \end{pmatrix}\begin{pmatrix}x\\y\\z\\\tau\end{pmatrix}-\begin{pmatrix}0\\0\\s\\\kappa\end{pmatrix}=-\mu\begin{pmatrix}r_x\\r_y\\r_z\\r_\tau\end{pmatrix}.\tag{10.3c}$$

By construction, the initial point satisfies (10.3c).

The centering search directions maintain the linear residuals unchanged and reduce the distance to the central path. They are the solution to the equations

$$H^\star(w)\Delta z+\Delta s=-s-\mu g^\star(z)\tag{10.4a}$$

$$\tau\Delta\kappa+\kappa\Delta\tau=-\tau\kappa+\mu\tag{10.4b}$$

$$\begin{pmatrix} & G^T & A^T & c\\ -A & & & b\\ -G & & & h\\ -c^T & b^T & h^T & \end{pmatrix}\begin{pmatrix}\Delta x\\\Delta y\\\Delta z\\\Delta\tau\end{pmatrix}-\begin{pmatrix}0\\0\\\Delta s\\\Delta\kappa\end{pmatrix}=\begin{pmatrix}0\\0\\0\\0\end{pmatrix},\tag{10.4c}$$

where w denotes the Nesterov-Todd scaling point for the ordered pair s, z.

In other words, H(w)s=z

When the iterates are exactly on the central path, the affine direction is tangent to it and points in the direction along which μ reduces. Moving the iterate along this direction corresponds to a first-order path-following method. The affine direction is calculated from the solution of

$$\begin{aligned}&H(w)\Delta z_a+\Delta s_a=-s\\&\tau\Delta\kappa_a+\kappa\Delta\tau_a=-\tau\kappa\\&\begin{pmatrix} & G^T & A^T & c\\ -A & & & b\\ -G & & & h\\ -c^T & b^T & h^T & \end{pmatrix}\begin{pmatrix}\Delta x_a\\\Delta y_a\\\Delta z_a\\\Delta\tau_a\end{pmatrix}-\begin{pmatrix}0\\0\\\Delta s\\\Delta\kappa\end{pmatrix}=\begin{pmatrix}r_x\\r_y\\r_z\\r_\tau\end{pmatrix}.\end{aligned}\tag{10.5}$$

As discussed before, the Mehrotra predictor-corrector algorithm uses a second-order approximation to the central path. This second-order direction is calculated from equations of the form

$$\begin{aligned} H(w)\Delta z + \Delta s &= -s - \frac{1}{1-\sigma} T(\Delta z_a, \Delta s_a) \\ \tau\Delta\kappa + \kappa\Delta\tau &= -\tau\kappa - \frac{1}{1-\sigma}\Delta\tau_a\Delta\kappa_a \\ \begin{pmatrix} & G^T & A^T & c \\ -A & & & b \\ -G & & & h \\ -c^T & b^T & h^T & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \Delta s \\ \Delta\kappa \end{pmatrix} &= \begin{pmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{pmatrix}, \end{aligned} \tag{10.6}$$

where $T(\Delta z_a, \Delta s_a)$ is a function of the affine direction. The derivation of this term is outside the scope of this thesis, but we will mention that for linear programming problems it takes the form of $T(\Delta z_a, \Delta s_a) = \Delta S_a \Delta z_a$, where ΔS_a is the diagonal matrix with the entries of Δs_a in the diagonal. Similar terms exist for symmetric conic programming problems with self-scaled barriers. However, a generalization for an unsymmetric cone problem could not be derived for this implementation, and remains an important area of opportunity for research.

The final search direction is a linear combination of the second-order search direction (10.6) and the centering direction (10.4). By smartly ordering the solution of the linear systems, the MPC algorithm solves only two linear systems instead of the expected three (we would expect to solve two for the second-order approximation of the central path and one for the centering direction). The strategy is to solve for the affine scaling direction, then determine the coefficient of the linear combination, then form and solve for a right-hand side crafted to yield the desired search direction [7]. Once the linear combination is determined and the terms from the affine search direction $\Delta z_a, \Delta s_a$ are known, the combined search direction is the solution of

$$\begin{aligned} H(w)\Delta z + \Delta s &= -s - \sigma\mu g(z) - T(\Delta z_a, \Delta s_a) \\ \tau\Delta\kappa + \kappa\Delta\tau &= -\tau\kappa + \sigma\mu - \Delta\tau_a\Delta\kappa_a \\ \begin{pmatrix} & G^T & A^T & c \\ -A & & & b \\ -G & & & h \\ -c^T & b^T & h^T & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \Delta s \\ \Delta\kappa \end{pmatrix} &= (1-\sigma) \begin{pmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{pmatrix}. \end{aligned} \tag{10.7}$$

The selection of the scalar σ that determines the linear combination between the affine and centering direction is done by first computing the largest *feasible* step size α_a along the affine scaling direction, then choosing $\sigma = (1-\alpha_a)^3$. This has the following effect: if the affine unit step is admissible then the algorithm will set $\sigma = 0$ and use the affine scaling direction. This is the ideal behavior, for a feasible unit step along the affine scaling direction reduces the residuals to zero and achieves optimality. However, if the feasible step along the affine scaling direction is zero or nearly zero then $\sigma \approx 1$. In this case the algorithm has

Algorithm 5 Mehrotra predictor-corrector

```
    k ← 0
2:  x ← x_0, y ← y_0, s ← s_0, z ← z_0
    τ ← 1
4:  κ ← 1
    while not converged do
6:      Compute the Nesterov-Todd scaling points
        Form and factor the system matrix of equation 10.5
8:      Solve for the affine search direction Δw_a of equation 10.5
        Find the largest α_a such that z + α_a Δz_a and s + α_a Δs_a are feasible
10:     Set σ = (1 − α_a)^3
        Solve for the combined second-order search directions Δw_c of equation
    10.7
12:     Find the largest α such that z + αΔz_c and s + αΔs_c are feasible
        α ← 0.98α
14:     w ← w + αΔw
        k ← k + 1
16: end while
```

strayed far from the central path and the next step should be mostly a centering step. A similar heuristic for the selection of σ was confirmed experimentally in [36] and this particular version is defined in [7] and used by ECOS.

Listing 5, describes the basic Mehrotra predictor-corrector algorithm. The two linear system solves for the search directions occur in lines 8 and 11. Lines 9–10 are concerned with computing the coefficient of the linear combination, and 12–14 with the final linesearch and updating the iterates. A notable aspect of the algorithm is that the final step length is a constant multiple of the step to the boundary, and generally this constant is chosen like 0.95 to 0.99, even 0.995!

10.2 ECOS for the exponential cone

Following the theoretical development of chapter 7 we know that for unsymmetric conic programming problems, the centering directions arising from

$$\mu H^\star(z)\Delta z + \Delta s = -s - \mu g^\star(z) \tag{10.8a}$$

$$\frac{\mu}{\tau^2}\Delta\tau + \Delta\kappa = -\kappa + \mu\frac{1}{\tau} \tag{10.8b}$$

$$\begin{pmatrix} & G^T & A^T & c \\ -A & & & b \\ -G & & & h \\ -c^T & b^T & h^T & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \Delta s \\ \Delta\kappa \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{10.8c}$$

are efficient at reducing the distance to the central path. However, this is true provided that s, z are already close to the central path in the sense that the

value of the function $\Omega(s, z, \tau, \kappa) = (\nu + 1)\log(s^T z + \tau\kappa) + f(s) + f^\star(z) - \log(\tau) - \log(\kappa) + (\nu + 1)\log(\nu + 1) + \nu + 1$ is small. (Here $f(s)$ and $f^\star(z)$ are conjugate pairs of barriers for the cones $\mathcal{K}$ and $\mathcal{K}^\star$ respectively.) Since we know explicit conjugate barriers for any Cartesian product of second-order cones and the positive orthant, and have derived a conjugate pair of barriers for the exponential cone and its dual, we can define the appropriate function Ω for problems modeled with Cartesian products of any of these cones.

To extend ECOS we restrict the iterates to the set $\mathcal{N} = \{\Omega \leq \bar{\Omega}\}$, where $\bar{\Omega}$ is a threshold chosen so that the conditions for the efficiency of the centering direction from (10.8) are maintained. In this manner ECOS-Exp will be able to calculate useful centering directions at any iterate. Given an initial point $x_0, s_0, z_0, y_0\tau_0, \kappa_0$ that satisfies $\Omega(s_0, z_0, \tau_0, \kappa_0) \leq \bar{\Omega}$, the algorithm proceeds in a similar manner to the original MPC, with the following exceptions: linesearches are used to find the largest affine scaling and combined steps that reach the edge of the set $\mathcal{N}$ instead of the edge of the feasible set; a first-order Taylor approximation is used for the variables that correspond to the unsymmetric cones; and the Hessian of the *unsymmetric* barrier is evaluated at the dual iterate instead of the Nesterov-Todd scaling point.

We must pause for a moment and mention again the work of Ye and Skajaa [51]. Where the authors derive a similar predictor-corrector algorithm for unsymmetric cones. This algorithm differs from the one here in the following fronts: First, the predictor and corrector directions are used in alternation while we use linear combinations. Their predictor steps correspond to our affine scaling directions, and their corrector to our centering directions. Second: the measure of centrality $\eta(s, z) = \|s + \mu g^\star(z)\|_{H^{-\star}(z)}$ is used instead of Ω. Their algorithm is guaranteed to converge if the iterates are maintained in the region where $\eta(s, z) < 1$ holds. Third: the Hessian is evaluated in the primal iterate for symmetric and unsymmetric variables, while we evaluate the section of the Hessian corresponding to the symmetric variables at the Nesterov-Todd scaling point.

We now introduce some notation to describe the ECOS-Exp algorithm more thoroughly. We denote the Cartesian product of all symmetric cones by $\mathcal{K}_s$, of all exponential cones by $\mathcal{K}_e$, and of all dual exponential cones by $\mathcal{K}_e^\star$. With this notation we have that $\mathcal{K} = \mathcal{K}_s \times \mathcal{K}_e$ and $\mathcal{K}^\star = \mathcal{K}_s \times \mathcal{K}_e^\star$. We will partition the primal (dual) variables into the symmetric variables $s_s \in \mathcal{K}_s$ ($z_s \in \mathcal{K}_s$) and the exponential variables $s_e \in \mathcal{K}_e$, ($z_e \in \mathcal{K}_e^\star$).

The centering directions are now the solution of the equations

$$\begin{aligned} \mu H^\star(z_{ec})\Delta z_{ec} + \Delta s_{ec} &= -s_{ec} - \mu g^\star(z_{ec}) \\ H(w)\Delta z_{sc} + \Delta s_{sc} &= -s_{sc} - \mu g(z_{sc}) \\ \tau\Delta\kappa + \kappa\Delta\tau &= -\tau\kappa + \mu \\ \begin{pmatrix} & G^T & A^T & c \\ -A & & & b \\ -G & & & h \\ -c^T & b^T & h^T & \end{pmatrix} \begin{pmatrix} \Delta x_c \\ \Delta y_c \\ \Delta z_c \\ \Delta\tau_c \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \Delta s_c \\ \Delta\kappa_c \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}. \end{aligned} \tag{10.9}$$

The c in the subscript is used to indicate that these are centering directions. Here w is the Nesterov-Todd scaling point for the dual-primal pair z_s, s_s. The affine directions are the solution to

$$\begin{aligned} \mu H^\star(z_e)\Delta z_{ea} + \Delta s_{ea} &= -s_e \\ H(w)\Delta z_{sa} + \Delta s_{sa} &= -s_s \\ \tau\Delta\kappa + \kappa\Delta\tau &= -\tau\kappa \\ \begin{pmatrix} & G^T & A^T & c \\ -A & & & b \\ -G & & & h \\ -c^T & b^T & h^T & \end{pmatrix} \begin{pmatrix} \Delta x_a \\ \Delta y_a \\ \Delta z_a \\ \Delta\tau_a \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \Delta s_a \\ \Delta\kappa_a \end{pmatrix} &= \begin{pmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{pmatrix}, \end{aligned} \tag{10.10}$$

where the a subscript indicates affine scaling directions. The combined directions are the solution to

$$\begin{aligned} \mu H^\star(z_e)\Delta z_e + \Delta s_e &= -s_e - \sigma\mu g^\star(z_e) \\ H(w)\Delta z_s + \Delta s_s &= -s_s - \sigma\mu g(z_s) + T(\Delta z_{sa}, \Delta s_{sa}) \\ \tau\Delta\kappa + \kappa\Delta\tau &= -\tau\kappa + \sigma\mu - \Delta\tau\Delta\kappa \\ \begin{pmatrix} & G^T & A^T & c \\ -A & & & b \\ -G & & & h \\ -c^T & b^T & h^T & \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \Delta s \\ \Delta\kappa \end{pmatrix} &= (1-\sigma) \begin{pmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{pmatrix}. \end{aligned} \tag{10.11}$$

First order affine search direction for exponential cone part with correction

Second-order correction equation for quadratic part

See Equation (38) in CVXOPT guide

Algorithm 6 ECOS-Exp Mehrotra predictor-corrector

$k \leftarrow 0$
$x \leftarrow x_0, y \leftarrow y_0, s \leftarrow s_0, z \leftarrow z_0$
$\tau \leftarrow 1$
$\kappa \leftarrow 1$
while not converged **do**
  Compute the Nesterov-Todd scaling point for z_s, s_s
  Form and factor the matrix of equations (10.9)
  Solve for the affine search direction $(\Delta x_a, \Delta y_a, \Delta s_a, \Delta z_a, \Delta\tau_a, \Delta\kappa_a)$
  Find the largest α_a such that $z + \alpha_a\Delta z_a$ and $s + \alpha_a\Delta s_a$ are contained in the set $\mathcal{N}$
  Set $\sigma = (1-\alpha_a)^3$
  Solve for the combined second-order search directions Δw_c using (10.11)
  Find the largest α such that $z + \alpha\Delta z_c$ and $s + \alpha\Delta s_c$ are contained in the set $\mathcal{N}$
  $\alpha \leftarrow 0.98\alpha$
  $(x, y, s, z, \tau, \kappa) \leftarrow (x, y, s, z, \tau, \kappa) + \alpha(\Delta x_c, \Delta y_c, \Delta s_c, \Delta z_c, \Delta\tau_c, \Delta\kappa_c)$
  $k \leftarrow k + 1$
end while

10.2.1 The barriers for the exponential cone

Due to the formulation used by ECOS, the Hessian $H^\star(z_e)$ in (10.9)–(10.11) is the dual barrier of the exponential cones. We have chosen to use the barrier which in chapter 8 was denoted $\tilde{f}^\star(u, v, w)$ and given by

$$\tilde{f}^\star(u, v, w) = -\log(w - u - u\log\frac{-v}{u}) - \log(-u) - \log(v).$$

10.2.2 Initializing ECOS-Exp

Because all iterates are expected to stay within the region $\mathcal{N}$ we must select an initial iterate inside $\mathcal{N}$. One possible strategy is to find an initial iterate strictly on the central path. Some solvers, for example, SEDUMI [52] initialize variables that are not conically constrained (in this case x, y) to zero and the rest (in this case s, z, τ, κ) to some vector ι that depends on the cone of the problem and satisfies $\iota = -g(\iota)$, where g is the gradient of the barrier. The conjugacy of the barriers implies that $-g^\star(\iota) = \iota$ and therefore ι is primal-dual strictly feasible. Because $\iota + g(\iota) = 0$, this vector is on the central path (for $\mu = 1$). Observe that $-\gamma^2 g(\gamma\iota) = -\gamma g(\iota) = \gamma\iota$ holds and therefore any scaling $\gamma\iota$ with $\gamma > 0$ can also be perfectly centered with $\mu = \gamma^2$.

For symmetric cones, ι is the identity in the Jordan algebra where the cone is defined. This corresponds to the following: for the positive orthant, ι is the vector of all ones; for the Lorentz cone, $\iota = (1, 0, \cdots, 0)$ where the 1 corresponds to the root variable in the second-order cone. For semidefinite cones (though ECOS does not solve systems with them) ι is the identity matrix. Unsymmetric cones are not domains of positivity of a Jordan algebra and therefore this definition of ι as an identity vector in a ring does not make sense. However, for the three symmetric cases listed here, the vector ι can also be constructed by finding some sort of central ray in the cone. This idea extends to unsymmetric cones and will allow us to define such an ι for the exponential cone.

The equation $\iota = -g(\iota)$ can also be solved for barriers of unsymmetric cones. If we find a solution the resulting ι will satisfy $-g^\star(\iota) = \iota$, will be strictly primal-dual feasible and the pair (ι, ι) will be on the central path for $\mu = 1$.

To find ι we need to solve the optimization problem

$$\text{minimize } f(s) \tag{10.12}$$
$$\text{subject to } \|s\|^2 \leq 1. \tag{10.13}$$

Whose optimality conditions are

$$\begin{aligned} \|s\|^2 &\leq 1, \\ -g(s) &= \lambda s, \\ \lambda &\geq 0, \\ s &\in \mathcal{K}. \end{aligned}$$

If the inequality is active then $\lambda = 0$ and if it is active $\lambda > 0$. However, for any recession direction Δs of the cone $\mathcal{K}$ we have that $s + \alpha\Delta s$ is feasible for all $\alpha > 0$,

then $f(\alpha\Delta s) = -\nu\log(\alpha) + f(\Delta s) \to -\infty$ and we conclude that the barrier functions for the cones are unbounded below. Therefore the solution $s^\star$ satisfies $\|s^\star\|^2 = 1$ and $g(s^\star) + \lambda s^\star = 0$ for some positive Lagrange multiplier. From the ν-logarithmic homogeneity of f we have that $0 = s^{\star T} g(s^\star) + \lambda = -\nu + \lambda = 0$ and we conclude that $\lambda = \nu$.

To calculate the ι for the exponential cone we need to solve problem 10.13 using the barrier of the exponential cone and then set $\iota = \frac{s^\star}{\sqrt{\nu}}$ for then $\iota + g(\iota) = \frac{1}{\sqrt{\nu}} s^\star + \sqrt{\nu} g(s^\star) = \frac{1}{\sqrt{\nu}} \left(s^\star + \lambda g(s^\star)\right) = 0$. We used projected gradient descent and hard-coded the entries of ι into the implementation. For the exponential cone,

$$\iota \approx \begin{pmatrix} -1.051 & 1.259 & 0.556 \end{pmatrix}.$$

Symmetric cone solvers, where the iterates don't have to start close to the central path, can take advantage of another initialization strategy. This is due to Mehrotra, and the form described here is the same as implemented in ECOS [16] and CVXOPT [7]. We refer to this as Mehrotra's initialization strategy, in which the initial points are selected by solving the least-norm problem

$$\begin{aligned} \text{minimize } & c^T x + \frac{1}{2}\|s\|_2^2 \\ \text{subject to } & Ax = b, \\ & Gx + s = h, \end{aligned}$$

with solution $\hat{x},\ \hat{s}$, and the least-norm problem

$$\begin{aligned} \text{maximize } & -h^T z - b^T y - \frac{1}{2}\|z\|_2^2, \\ \text{subject to } & A^T y + G^T z + c = 0, \end{aligned}$$

with solution $\hat{y}$, and $\hat{z}$, and adding a multiple of a vector ι to both $\hat{s}$ and $\hat{z}$ so that $s_0 = \hat{s} + \alpha_s \iota \in \operatorname{int}\mathcal{K}$ and $z_0 = \hat{z} + \alpha_z \iota \in \operatorname{int}\mathcal{K}^\star$. The initial point is $x_0 = \hat{x}, y_0 = \hat{y},\ s_0, z_0, \tau_0 = 1$, and $\kappa_0 = 1$. The least-norm problems can be readily solved by one linear system. The selection of the α_s and α_z is done as follows: first the smallest $\hat{\alpha}_s$ such that $s_0 + \hat{\alpha}_s \in \mathcal{K}$ is found, then $\alpha_s = \hat{\alpha}_s + 1$. The same is done to find α_z.

10.2.3 Stopping criteria

The termination criteria for ECOS-Exp are designed to detect when the solver has reached an optimal point or has generated a certificate of primal or dual unboundedness. We have observed that it is not uncommon for problems defined in terms of exponential cones to have solutions with large norms. Therefore we have adapted the termination criteria of ECOS.

When an iterate is identified as optimal, the vectors x, y, s, z are divided by τ to form the solution. Therefore, the solution returned to the user has residuals

norms

$$\begin{aligned} r_x &= \frac{\|A^T y + G^T z + \tau c\|}{\tau}, \\ r_y &= \frac{\|Ax - \tau b\|}{\tau}, \\ r_z &= \frac{\|Gx + s - \tau h\|}{\tau}. \end{aligned} \tag{10.14}$$

Since, under floating-point arithmetic, we can only guarantee small residuals with respect to the size of the inputs, we must normalize the residuals by some measure of A, G, b, c, h and x, y, s, z. ECOS-Exp achieves this by computing the scaled residual norms

$$\begin{aligned} \hat{r}_x &= \frac{\|A^T y + G^T z + \tau c\|}{\tau \max(\|c\| + \|y\| + \|z\|, 1)}, \\ \hat{r}_y &= \frac{\|Ax - \tau b\|}{\tau \max(\|b\| + \|x\|, 1)}, \\ \hat{r}_z &= \frac{\|Gx + s - \tau h\|}{\tau \max(\|x\| + \|s\| + \|h\|, 1)}. \end{aligned} \tag{10.15}$$

It declares an iterate feasible with respect to the linear constraints when $r_x \leq \varepsilon_f$, and $\max(\hat{r}_y, \hat{r}_z) \leq \varepsilon_f$. An iterate is declared optimal if it is linear feasible (in the above sense) and if the gap $s^T z \leq \varepsilon_{abs}$, or if the gap satisfies the bound $s^T z / \max\left(\frac{|c^T x|}{\tau}, \frac{|b^T y + h^T z|}{\tau}\right) \leq \varepsilon_{rel}$. In the definition of these stopping criteria, there is an implicit assumption that the matrices A, G are of moderate norm. In practice a scaling procedure applied to A and G before solving.

An approximate certificate of primal unboundedness is found when the equations $Ax = 0$, $Gx + s = 0$ are approximately satisfied and $c^T x$ is sufficiently negative. On the other hand a certificate of primal or dual unboundedness is found when the equations $A^T y + G^T z = 0$ are approximately satisfied and $-b^T y - h^T z$ is sufficiently positive. To this end ECOS-Exp calculates the set of scaled residual norms

$$\begin{aligned} hr_x &= \frac{\|A^T y + G^T z\|}{\max(\|y\| + \|z\|, 1)}, \\ hr_y &= \frac{\|Ax\|}{\max(\|x\|, 1)}, \\ hr_z &= \frac{\|Gx + s\|}{\max(\|x\| + \|s\|, 1)}, \end{aligned} \tag{10.16}$$

and declares a problem unbounded if it is not optimal, $c^T x / \|x\| \leq -\varepsilon_a$, and $\max(hr_y, hr_z)/(\|y\| + \|z\|) \geq \varepsilon_f$. It declares the problem dual unbounded (infeasible) when it is not optimal, $hr_x \leq \varepsilon_f$, and $-b^T y - h^T z > \varepsilon_a$.

10.3 Empirical evaluation of ECOS

In this section we document our numerical experiments using ECOS-Exp. We first study the growth of the iteration count as the complexity of the problem

(defined by ν) increases. In section 10.3.1 we show that, for a set of random problems, the increase in complexity causes a slow growth in iteration count. In fact, the rate of increase in iteration counts is similar to that observed for symmetric problems, and we confirm the same result for families of unbounded and infeasible random problems.

We observe that for two versions of the same random problem —one formed of symmetric cones and one formed of a mixture of symmetric an exponential cones— the iteration count is larger for the problem that includes exponential cones. This is not surprising because the purely symmetric problem can take advantage of the second-order path-following strategy and the Mehrotra initialization scheme. We explore this issue further and modify the *symmetric* version of ECOS to use the ι-initialization (see (10.2.2)) and disable the second-order path-following scheme. We observe that (at least for this set of random problems) the second-order approximation to the central path seems to be the heuristic that most reduces the iteration count. As mentioned before we were not able to extend this heuristic to the exponential cone case. However, knowing that this can bring such benefits highlights the importance of defining a generalization.

We are aware that random problems tend to be easy to solve. Therefore in section 10.4 we test ECOS-Exp on a set of problems defined on data from the LPnetlib library [14]. We extract 75 linear programming problems given in the standard form: minimize $c^T x$ subject to $Ax = b,\ 0 \leq x$, define the negative-entropy problem

$$\begin{aligned} &\text{minimize} \sum_i x_i \log(x_i) \\ &\text{subject to } Ax = b, \end{aligned} \tag{10.17}$$

and solve it using ECOS-Exp. This problem is ideally suited for the solver PDCO [50] and for the exponential optimization solver of MOSEK [5]. We solve each of these problems with them and compare the iteration counts.

Geometric programming problems are an important family of optimization problems that can be modeled with the exponential cone. A public test set is not available, however the authors of [51] shared with us the problems used for the results in their paper. In section 10.5 we list the iterations required by ECOS-Exp.

10.3.1 Growth in iteration count as a function of complexity

We now describe the strategy used to explore the growth in iteration count as the complexity of the problem increases. We wish to create a family of problems with increasing complexity ν and increasing number of variables. Furthermore for each ν and problem size we wish to generate two problems, one purely symmetric and the other a mixture of symmetric and exponential cones. Our strategy is the following: For each $k = 1, \ldots, k_{\max}$ we determine the values

$n = 12 \times k$, $m = n$, $p = \text{floor}(n/10)$. We set $G = -I_n$ (the $n \times n$ identity) and generate random sparse matrix $A \in \mathbb{R}^{p \times n}$. (The matrix A is generated using the *sprand* command in MATLAB with sparsity parameter 0.5.)

The problems are fully determined by specifying the values of c, b and h and the sizes of the cones. To generate pairs of problems with the same complexity, for each k we generate symmetric problems with $6 \times k$ linearly constrained variables and $2 \times k$ SOCP cones of dimension 3 each. This yields a problem of complexity $\nu = 8 \times k$ on $n = m = 12 \times k$ conically constrained variables. The mixed symmetric-exponential problems are formed by $2 \times k$ SOCP cones of dimension 3 and two exponential cones, yielding a complexity of $\nu = 8 \times k$, and again $n = m = 12 \times k$ conically constrained variables.

To generate primal-dual feasible problems we use the following strategy. Once the sizes of the cones have been determined, we generate the random vector z_0 strictly in the dual cone and s_0 strictly in the primal cone. We also generate random normal vector y_0. We then let $b \leftarrow Ax_0$, $h \leftarrow Gx_0 + s_0$ and $c \leftarrow -A^T y_0 - G^T z_0$. That way we know that the feasible set contains at least the strictly feasible point x_0, y_0, z_0, s_0 and therefore the problem is solvable.

For each value of k, 10 random problem pairs are generated; one purely symmetric and solved with ECOS and one containing both symmetric and exponential constraints and solved by ECOS-Exp. The iteration count over these 10 problems is averaged to yield the values reported in this experiment. ECOS is used to solve each instance of the purely symmetric problem 4 times: Once with the Mehrotra initialization strategy and second-order path following (this is the default in ECOS), a second time with ι-initialization and second-order path-following, a third time with Mehrotra initialization and first-order path-following, and a fourth time with ι-initialization and first-order path-following.

The first experiment compares ECOS (second-order path-following and Mehrotra initialization) with ECOS-Exp. Figure 10.1 shows the average iteration count for each complexity value ν. The iteration count of ECOS-Exp is larger than that of ECOS however, for this range of complexities the growth rates of both iteration counts seems to be similar. In Figure 10.2 we plot $\log_{10}$ of the series, and observe that the growth rate of the log series seems to be close to linear or possibly sublinear. Importantly, the sequences for ECOS and ECOS-Exp seem to grow at similar rates. For these random problems both ECOS and ECOS-Exp seem to increase their iteration count close to logarithmically in the complexity of the problem.

We are interested in why ECOS solves in less iterations than ECOS-Exp. We hypothesize that both the Mehrotra initialization strategy and the second-order path-following play a part. To explore this issue further we modify ECOS to be able to initialize it with the ι-initialization strategy.

Figure 10.3 extends Figure 10.1 and incorporates a new series with the average iteration count for ECOS when initialized with the ι vector. We observe that the average iteration count is very similar to the version with Mehrotra initialization. If anything, the ι-initialization seems to slightly reduce the average number of iterations. We do not claim that the ι-initialization strategy is superior, as this set of problems is randomly generated and likely to be well

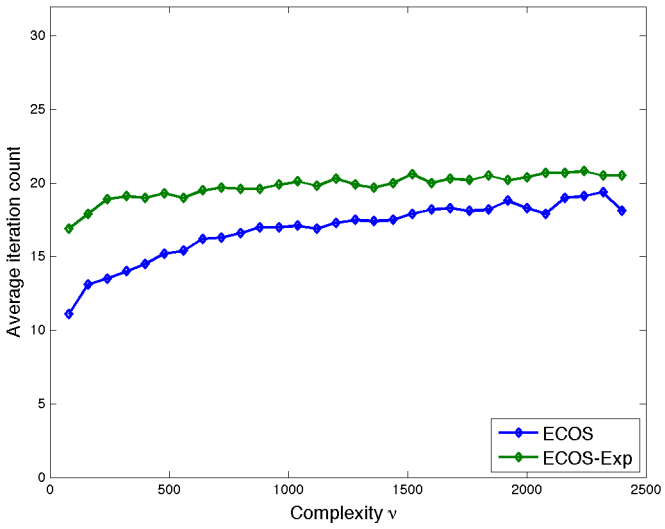

Figure 10.1: Average iteration count versus complexity. ECOS with second-order path-following and Mehrotra initialization vs ECOS-Exp.

scaled. However, this experiment indicates that the difference between ECOS and ECOS-Exp cannot be explained by the difference in initialization strategy.

In the third experiment we disable the second-order path-following of ECOS and plot the average iteration count when ECOS is initialized with the ι-initialization strategy and with Mehrotra's strategy. Disabling the second-order path-following amounts to setting the term $T(\Delta s, \Delta z)$ to zero in the second equation of (10.11), and modifying the third to $\tau\Delta\kappa + \kappa\Delta\tau = -\tau\kappa + \sigma\mu$. Because ECOS-Exp uses second-order path-following for the symmetric variables we plot two series for ECOS-Exp: one with second-order path-following and one without. In Figure 10.4 we observe that disabling the second-order path-following from ECOS increases the average iteration count enough to surpass ECOS-Exp by a large margin, even when the second-order path-following is disabled for ECOS-Exp. This seems to indicate that the second-order path is responsible for the smaller iteration count that ECOS achieves. It also seems to indicate, though we have not confirmed it, that for first-order path-following methods, maintaining the iterates close to the central path can be beneficial.

10.3.2 Detection of unbounded problems

We repeat the first experiment of the previous section but generate problems we know are primal unbounded. The problems have the same size as those

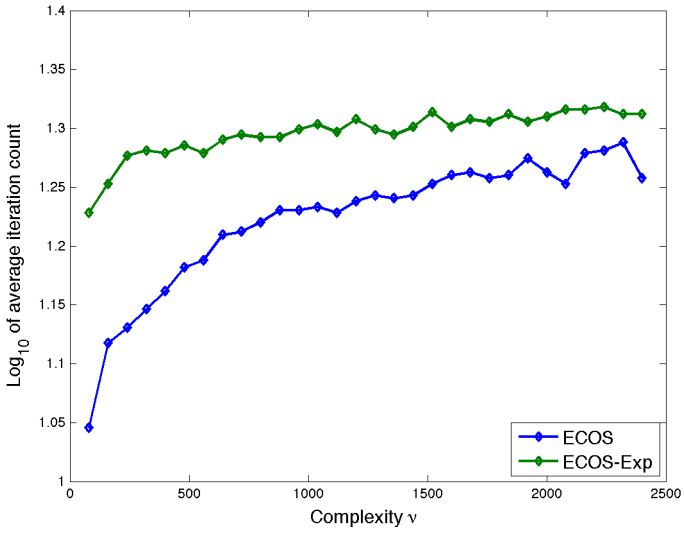

Figure 10.2: $\log_{10}$ of average iteration count versus complexity. ECOS with second-order path-following and Mehrotra initialization vs ECOS-Exp

in the previous section. ECOS is set to solve with second-order path-following and Mehrotra's initialization, while ECOS-Exp uses ι-initialization with second-order path-following for the symmetric cones.

To generate unbounded problems we generate a feasible problem as in the previous experiment, and then change the sign of c. This yields an unbounded problem with very high probability. To be certain, we check the exit conditions of ECOS and ECOS-Exp. If the problem is not unbounded we discard the problem and generate a new one. In Figure 10.5 we observe the same behavior we observed for solvable problems. The iteration count for ECOS-Exp is larger than that of ECOS, but the growth in iteration counts is very similar.

10.3.3 Detection of infeasible problems

We repeat the experiment, this time generating dual unbounded (primal infeasible) problems. The sizes and complexities are equal to the previous two experiments. To generate primal infeasible problems we use the following strategy: After generating matrices A and G (with the same scheme as described for the feasible problem) we generate random normal $x_0 \in \mathbb{R}^n$, $y_{res} \in \mathbb{R}^{m-1}$ and random s_0 and z_0 strictly feasible with respect to the primal and dual cones. We then calculate $d = A^T(:, 1 : \text{end} - 1)y_{res} + G^T z_0$, and form the new matrix $\hat{A}^T = [A^T d]$ and vector $\hat{y}_{res} = [y_{res}; -1]$. By construction the equation $\hat{A}^T \hat{y}_{res} + G^T z_0 = 0$ holds. We then generate random normal vectors b and h. If $-b^T \hat{y}_{res} - h^T z_0 < 0$ we multiply b and h by -1. This strategy yields a problem

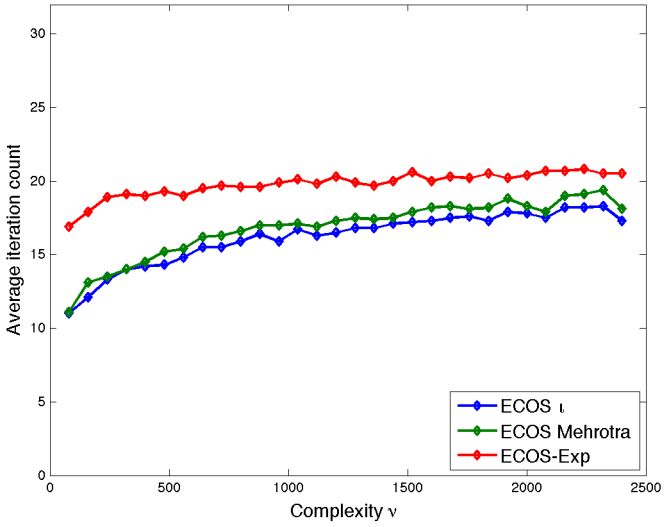

Figure 10.3: Average iteration count versus complexity for ECOS (with second-order path-following and Mehrotra initialization, with second-order path-following and ι-initialization) and ECOS-Exp

for which the vectors y_{res}, z_0 are a certificate of dual unboundedness and therefore the problems are infeasible. The vector c is then generated by choosing another random normal y_0 and setting $c \leftarrow -\hat{A}^T y_0 - G^T z_0$. This yields a dual feasible and unbounded problem.

In Figure 10.6 we observe, the same behavior as for solvable and unbounded problems. The iteration count of ECOS-Exp is larger than ECOS, but with similar growth rate.

10.4 Negative entropy problems

We generate test problems by using the linear systems A, b from the LPnetlib [14] collection and form the problem

$$\begin{aligned} &\text{minimize} \sum_i x_i \log(x_i) \\ &\text{subject to } Ax = b. \end{aligned} \tag{10.18}$$

These problems can be solved by PDCO and MOSEK. This allows us to compare the iteration counts for the three solvers.

PDCO and ECOS-Exp can be sensitive to the scaling of the problem data, and the systems from the LPnetlib library can be badly scaled. Therefore, we modify the problem by using a pair of diagonal scaling matrices R, C calculated

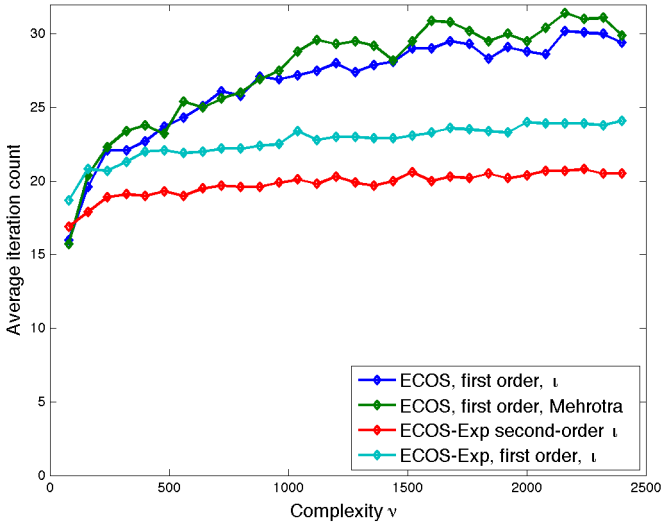

Figure 10.4: Average iteration count versus complexity for ECOS (with first-order path-following, Mehrotra initialization, with first-order path-following and ι-initialization) and ECOS-Exp (with first-order path-following for the symmetric variables and second-order path-following for the symmetric variables)

from the geometric mean scaling heuristic implemented in the MATLAB code gmscale.m [19, 50], and instead solve the equivalent problem

$$\text{minimize} \sum_i c_i^{-1}\bar{x}_i \log(c_i^{-1}\bar{x}_i) \text{ subject to } \bar{A}\bar{x} = \bar{b}, \tag{10.19}$$

where $\bar{A} = R^{-1}AC^{-1}$, $\bar{b} = R^{-1}b$, and $Cx = \bar{x}$ and we expect $\bar{A}$ and $\bar{b}$ to be better scaled. Because the geometric mean scaling heuristic finds scalings for the rows and columns of a matrix and we wish to scale the right-hand side b, we used the heuristic on the matrix $[A, b]$. This produces a diagonal matrix C of size $n + 1$ and a diagonal matrix R of size p (here A is $p \times n$). For this experiment we discarded the last entry of C to produce a scaling matrix C of the correct size.

For this experiment the three solvers were set to solve to the same precision: 10^{-7} for the linear residuals and 10^{-7} for the complementarity. The stopping criteria of the three solvers is slightly different and therefore these three are not identical. PDCO has the capacity to use an estimate of the norm of the solution to re-scale the problem. When such an estimate is known this feature can greatly improve its behavior. In these tests we do not use PDCO's scaling feature because we try to give ECOS-Exp and PDCO identically scaled data. On the other hand, we have very little insight into the heuristics MOSEK uses and have no knowledge of the presolve techniques or their effect.

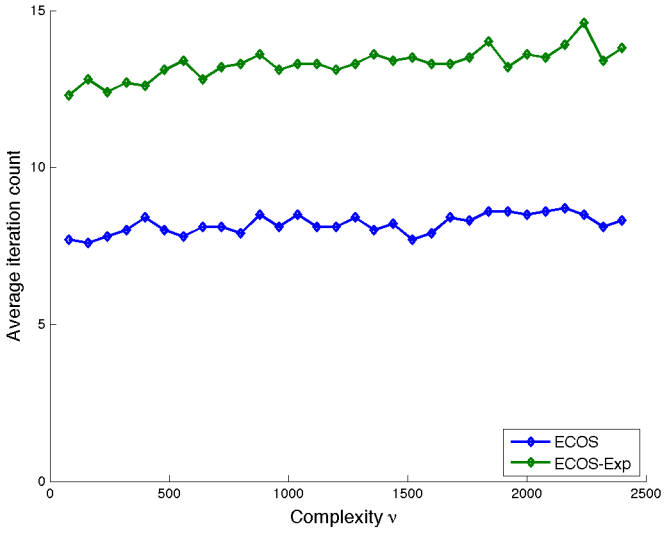

Figure 10.5: Average iteration count versus complexity. ECOS and ECOS-Exp, unbounded problems

In Table 10.1 we highlight some problems that proved difficult for ECOS-Exp. Problem *lp agg* was declared infeasible by ECOS-Exp, yet we know that the original LP is feasible and therefore so is the negative entropy problem. This seems to be a relatively hard problem because neither PDCO nor MOSEK manages to solve it. The sequence of *osa* problems were solved to a moderate precision, but the precision was not high enough for the solution to be declared close to optimal, and instead ECOS-Exp indicated a failure. PDCO was also unable to solve these problems, while MOSEK indicates the problems were solved with linear residuals two orders of magnitude smaller than those of ECOS-Exp, but still far from the requested precision. It is very likely that the norm of the solution is large and the normalized residuals are in fact acceptably small.

We explore the above phenomenon further by extracting the convergence history of ECOS-Exp for a particularly complicated problem. We use the negative entropy problem generated with A, b from *lp agg* as an example. Even though we know the problem is feasible, ECOS-Exp reaches its limit of iterations and concludes that the problem is close to infeasible. We know that one of τ or κ will tend to zero upon convergence, but it is possible that initially both variables become very small, one dominating the other only at the later stages of the solution. This is the signature of an almost infeasible or almost unbounded problem.

For the problem in question, the norms of the iterates at the solution are $\|x\| \approx 10^8$, $\|y\| \approx 10^9$, $\|s\| \approx 10^8$, $\|z\| \approx 10^7$. Figure 10.7 plots the norm of the linear residuals $r_y = \|Ax - \tau b\|_2$, $r_x = \left\|A^T y + G^T z + \tau c\right\|_2$ and $r_z =$

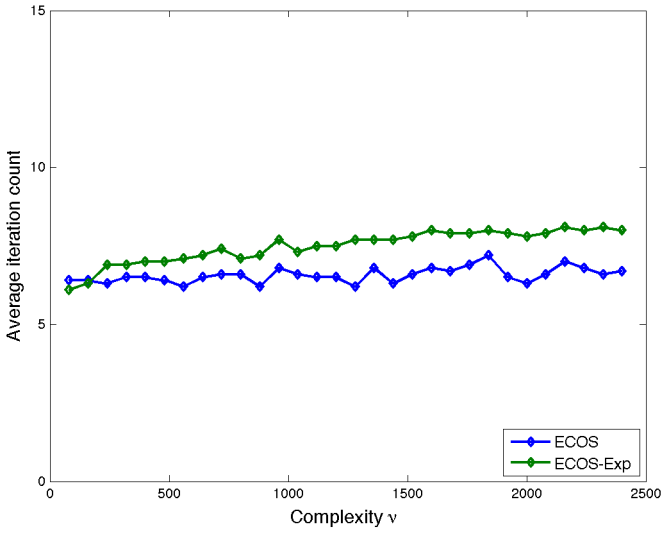

Figure 10.6: Average iteration count versus complexity. ECOS and ECOS-Exp, infeasible problems

$\|Gx + s - \tau h\|_2$ as a function of the iteration count. We observe initial rapid convergence followed by stalled progress. Figure 10.8 shows the iteration history of τ and κ. We observe that both become small quickly and only at the very late stages of the solve does κ become significantly smaller than τ.

To declare a problem feasible, ECOS-Exp evaluates the residual norms divided by τ and scaled by some function of the norm of the iterates (see (10.16)). In this case the stopping criterion depends on residuals that are close to $\frac{1}{\tau \max(\|x\|+\|s\|,1)} \approx 10^{-2}$ times the ones plotted. And therefore $\hat{r}_y \approx 10^{-4}$.

To declare a problem infeasible the residual $\hat{h}r_x = \frac{\|A^T y + G^T z\|_2}{\max(\|y\|+\|z\|,1)}$ of (10.16) must be small. This residual is independent of the value of τ and κ. Figure 10.9

Table 10.1: Problems where ECOS-Exp was unable to achieve the requested precision

Name	ECOS-Exp			PDCO			MOSEK		
	Iter	Flag	Lin res	Iter	Flag	Lin res	Iiter	Flag	Lin res
lp agg	300	CPi	2.3e-02	135	Fail	1.2e+07	1	Fail	1.53+10
lp d2q06c	300	CO	1.2e-04	300	Fail	1.2e+04	46	Opt	1.14-02
lp osa 07	300	Fail	1.4e-04	300	Fail	8.2e+03	59	Opt	1.78-06
lp osa 14	300	Fail	2.3e-03	36	Fail	7.1e+05	117	Opt	2.30-04
lp osa 30	300	Fail	1.4e-02	35	Fail	2.6e+06	114	Opt	8.83-04
lp osa 60	300	Fail	1.2e-01	35	Fail	7.8e+06	100	Opt	3.60-03

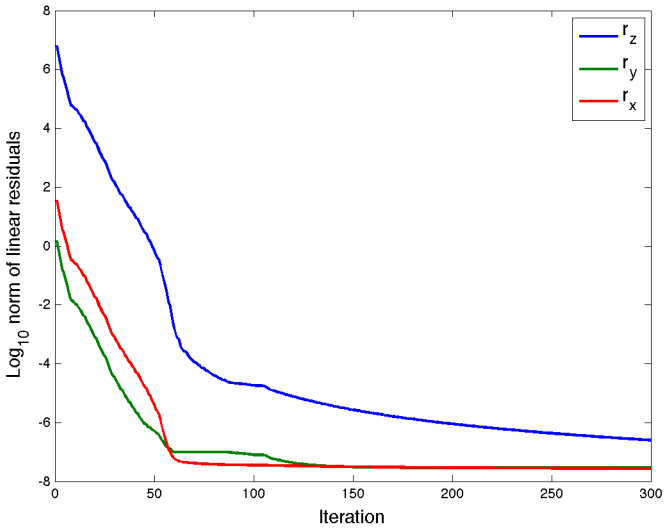

Figure 10.7: Convergence history of the linear residuals for problem lp agg

shows the norm of the unscaled homogeneous residuals (without the denominator of the definition of hr_x). For this problem the norm of hr_x will converge to close to 10^{-13}.

We can dramatically improve the behavior of ECOS-Exp for this problem by re-scaling the data A, G, c, b, h with scalars $\gamma_x = 10^2, \gamma_y = 10^3, \gamma_\tau = 10^{-5}$ and forming $\hat{A} = \gamma_x\gamma_y A$, $\hat{G} = \gamma_x G$, $\hat{c} = \gamma_x\gamma_\tau c$ $\hat{h} = \gamma_\tau h$, $\hat{b} = \gamma_y\gamma_\tau b$. This yields the alternative and equivalent conic problem with solution $\hat{x} = \frac{\gamma_\tau}{\gamma_x}x^\star$, $\hat{y} = \frac{\gamma_\tau}{\gamma_y}y^\star$, $\hat{s} = \gamma_\tau s^\star$, $\hat{z} = \gamma_\tau z^\star$, where the variables $x^\star, y^\star, s^\star, z^\star$ correspond to the solution of the unscaled problem. For this re-scaled problem ECOS-Exp identifies an optimal solution in 68 iterations. Figure 10.11 shows the iteration history of $\hat{\tau}$ and $\hat{\kappa}$, where we observe that for the re-scaled problem only $\hat{\kappa}$ tends to zero while $\hat{\tau}$ stays well away from it. Figure 10.10 shows the norm of the unscaled residuals $\hat{r}_x, \hat{r}_y, \hat{r}_z$ for the re-scaled problem and we observe the expected fast convergence of a Newton method.

Table 10.2 lists the set of problems where ECOS-Exp detected an infeasible solution. For some of these PDCO failed to converge; for some others both MOSEK and PDCO failed to converge. However, for all problems the linear residuals achieved by both PDCO and MOSEK are large. For example the set of *klein* problems are identified as optimal by MOSEK but the linear residuals at the solution are of order 10^6 to 10^8. Again different stopping criteria between ECOS-Exp and MOSEK can explain this difference.

In Table 10.3 we include the iteration counts, status flags, and linear residuals for the three solvers. This table is summarized in the performance profile of Figure 10.12. We use an example to interpret the performance profile: The

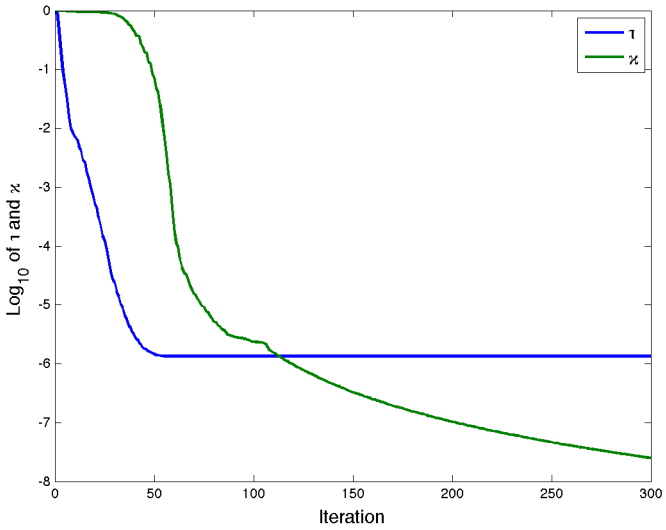

Figure 10.8: Convergence history of the homogeneous variables τ and κ for lp agg

vertical line at ordinate 10 intersects the line for ECOS-Exp close to 0.75. This means that for 75% of the problems ECOS-Exp took less than 10 times as many iterations to solve the problem than the best solver for each problem.

The performance profile shows that MOSEK is the best solver on all the problems (in terms of iteration counts). However, there is a factor that has to be considered: MOSEK incorporates heuristics that we do not have access to, therefore we are comparing ECOS-Exp against a method that potentially uses a pre-solver and some other well tuned and tested scaling strategy.

Name	ECOS-Exp			PDCO			MOSEK		
	Iter	Flag	Lin res	Iter	Flag	Lin res	Iiter	Flag	Lin res
lp 25fv47	56	Opt	2.4e-07	42	Opt	2.4e-08	22	Opt	3.92e-11
lp adlittle	28	Opt	5.5e-10	32	Opt	8.7e-09	9	Opt	2.09e-06
lp afiro	30	Opt	1.4e-11	25	Opt	1.1e-09	12	Opt	8.76e-12
lp agg	300	CPi	2.3e-02	135	Fail	1.2e+07	1	Fail	1.53e+10
lp agg2	93	Opt	1.9e-08	300	Fail	2.6e+06	16	Opt	7.54e-03
lp agg3	117	Opt	1.1e-08	154	Fail	2.7e+06	15	Opt	6.02e-02
lp bandm	40	Opt	1.3e-06	55	Opt	7.2e-09	17	Opt	9.42e-06
lp beaconfd	38	Opt	1.7e-09	166	Fail	5.0e+03	12	Opt	2.54e-06
lp blend	32	Opt	3.8e-09	35	Opt	1.9e-10	15	Opt	1.90e-14
lp bnl1	258	Opt	7.6e-07	75	Opt	1.1e-06	61	Opt	3.63e-03
lp bnl2	102	Opt	1.2e-06	74	Opt	1.5e-06	109	Opt	1.11e-04
lp brandy	48	Opt	1.4e-07	49	Opt	1.2e-07	16	Opt	7.91e-11
lp cre a	77	Opt	6.8e-05	50	Opt	4.6e-07	25	Opt	1.10e-10

lp cre b	188	Opt	3.1e-05	63	Opt	9.5e-07	24	Opt	1.52e-10
lp cre c	96	Opt	2.8e-05	57	Opt	8.3e-07	35	Opt	7.33e-11
lp cre d	181	Opt	7.7e-05	72	Opt	1.6e-06	69	Opt	3.16e-11
lp d2q06c	300	CO	1.2e-04	300	Fail	1.2e+04	46	Opt	1.14e-02
lp degen2	61	Opt	1.6e-07	43	Opt	1.3e-07	17	Opt	1.13e-10
lp degen3	111	Opt	6.8e-07	45	Opt	3.9e-07	25	Opt	9.11e-09
lp e226	46	Opt	5.3e-07	44	Opt	1.0e-08	15	Opt	1.45e-12
lp fffff800	112	Opt	2.2e-03	300	Fail	3.2e+05	29	Opt	3.16e+01
lp israel	58	Opt	1.1e-06	252	Fail	7.9e+05	22	Opt	2.50e+02
lp lotfi	35	Opt	8.9e-10	37	Opt	2.1e-06	13	Opt	2.39e-05
lp maros r7	36	Opt	1.5e-10	45	Opt	6.5e-07	19	Opt	6.69e-09
lp modszk1	81	Opt	7.5e-11	101	Fail	2.5e+05	21	Opt	1.01e-09
lp osa 07	300	Fail	1.4e-04	300	Fail	8.2e+03	59	Opt	1.78e-06
lp osa 14	300	Fail	2.3e-03	36	Fail	7.1e+05	117	Opt	2.38e-04
lp osa 30	300	Fail	1.4e-02	35	Fail	2.6e+06	114	Opt	8.83e-04
lp osa 60	300	Fail	1.2e-01	35	Fail	7.8e+06	100	Opt	3.60e-03
lp qap12	13	Opt	1.2e-08	6	Opt	7.8e-11	8	Opt	9.89e-11
lp qap15	14	Opt	3.1e-10	6	Opt	1.2e-10	7	Opt	1.21e-10
lp qap8	14	Opt	7.1e-09	6	Opt	3.3e-11	7	Opt	4.27e-10
lp sc105	26	Opt	1.0e-09	29	Opt	2.3e-09	13	Opt	3.98e-13
lp sc205	29	Opt	2.2e-09	39	Opt	2.0e-09	14	Opt	4.82e-13
lp sc50a	26	Opt	8.8e-10	28	Opt	3.0e-09	11	Opt	3.08e-13
lp sc50b	29	Opt	1.9e-09	33	Opt	1.5e-09	11	Opt	6.83e-13
lp scagr25	29	Opt	1.0e-11	41	Opt	5.6e-08	13	Opt	7.693-03
lp scagr7	28	Opt	6.5e-12	36	Opt	3.5e-08	16	Opt	1.13e-10
lp scfxm1	85	Opt	5.7e-07	300	Fail	1.0e+03	22	Opt	2.16e-04
lp scfxm2	84	Opt	5.1e-07	239	Fail	1.5e+03	24	Opt	8.19e-05
lp scfxm3	85	Opt	3.4e-07	257	Fail	1.9e+03	24	Opt	3.80e-03
lp scorpion	43	Opt	1.3e-07	42	Opt	2.1e-08	11	Opt	4.27e-15
lp scrs8	66	Opt	5.8e-07	50	Opt	6.1e-08	18	Opt	1.84e-03
lp scsd1	16	Opt	3.1e-11	13	Opt	1.2e-12	5	Opt	7.36e-12
lp scsd6	17	Opt	3.3e-12	14	Opt	1.5e-11	5	Opt	6.32e-10
lp scsd8	38	Opt	5.8e-12	20	Opt	6.9e-10	9	Opt	5.25e-14
lp sctap1	38	Opt	1.7e-07	36	Opt	3.4e-09	400	Fail	9.09e-13
lp sctap2	41	Opt	5.0e-08	34	Opt	4.0e-09	16	Opt	5.29e-13
lp sctap3	50	Opt	9.4e-09	35	Opt	7.9e-09	279	Opt	5.92e-13
lp share1b	48	Pi	1.0e+00	227	Fail	5.8e+03	1	Fail	3.42e+06
lp share2b	22	Opt	2.0e-08	32	Opt	3.0e-08	19	Opt	6.53e-11
lp ship04l	54	Opt	8.5e-08	45	Opt	3.9e-08	11	Opt	2.07e-08
lp ship04s	48	Opt	8.1e-08	52	Opt	9.5e-09	11	Opt	7.35e-09
lp ship08l	97	Opt	7.9e-07	49	Opt	3.9e-08	16	Opt	1.58e-09
lp ship08s	71	Opt	8.0e-07	47	Opt	2.7e-08	13	Opt	2.48e-09
lp ship12l	91	Opt	3.9e-07	49	Opt	4.7e-08	15	Opt	6.51e-10
lp ship12s	70	Opt	4.1e-07	47	Opt	1.9e-08	14	Opt	1.08e-11
lp stocfor1	26	Opt	4.3e-09	33	Opt	8.1e-09	13	Opt	1.42e-07
lp stocfor2	48	Opt	1.9e-07	52	Opt	1.1e-07	24	Opt	1.89e-05
lp stocfor3	88	Opt	1.0e-06	69	Opt	9.3e-07	32	Opt	1.55e-06
lp truss	73	Opt	5.6e-09	28	Opt	1.4e-08	12	Opt	5.41e-13
lp wood1p	147	Opt	3.8e-06	56	Opt	2.0e-05	24	Opt	3.95e-12
lp woodw	112	Opt	1.3e-04	57	Opt	1.6e-05	18	Opt	2.45e-07

lpi bgindy	13	Pi	3.7e-02	113	Opt	6.1e+04	14	Opt	1.22e+08
lpi bgprtr	13	Pi	1.2e+01	69	Opt	4.0e+03	10	Opt	4.71e+08
lpi ceria3d	25	Pi	5.3e-03	300	Fail	6.2e+00	0	Opt	1.37e+04
lpi gosh	141	Pi	4.7e-02	300	Fail	3.6e+02	400	Fail	1.47e+04
lpi itest2	10	Pi	4.2e-03	300	Fail	3.5e+00	8	Opt	2.68e+01
lpi itest6	10	Pi	3.2e-01	20	Fail	3.4e+05	10	Opt	2.34e+08
lpi klein1	30	Pi	5.0e-01	300	Fail	4.7e+00	21	Opt	9.77e+06
lpi klein2	32	Pi	7.9e+00	300	Fail	9.9e+02	33	Opt	9.69e+06
lpi klein3	41	Pi	1.4e+01	300	Fail	2.0e+03	54	Opt	8.59e+06

Table 10.3: Iteration counts, result status, and linear residuals for ECOS-Exp, PDCO, and Mosek

10.5 Geometric programming problems

To conclude Table (10.4) lists the iteration count and resulting flag that ECOS-Exp returns when solving a set of geometric programming problems. These were provided by Erling Andersen and used by Ye and Skajaa in [51]. The transformation into a conic programming problem yields the number of linear constraints in the column denoted *Lin*, the number of exponential cones in the column *Exp*, and the total number of conically constrained variables in *Conic vars*. The column *Flag* indicates the status of the solver at the solution, where Opt indicates an optimal point was detected, Pi that the primal is infeasible, and CO that the desired precision was not achieved but to a lower precision the problem seems to be Optimal. The requested accuracy was 10^{-7} for the linear residuals and a complementarity measure of less than 10^{-8}.

Table 10.2: Negative entropy problems where ECOS-Exp found a certificate of infeasibility

Name	ECOS-Exp			PDCO			MOSEK		
	Iter	Flag	Lin res	Iter	Flag	Lin res	Iter	Flag	Lin res
lp share1b	48	Pi	1.0e+00	227	Fail	5.8e+03	1	Fail	3.42e+06
lpi bgindy	13	Pi	3.7e-02	113	Opt	6.1e+04	14	Opt	1.22e+08
lpi bgprtr	13	Pi	1.2e+01	69	Opt	4.0e+03	10	Opt	4.71e+08
lpi ceria3d	25	Pi	5.3e-03	300	Fail	6.2e+00	0	Opt	1.30e+03
lpi gosh	141	Pi	4.7e-02	300	Fail	3.6e+02	400	Fail	1.47e+04
lpi itest2	10	Pi	4.2e-03	300	Fail	3.5e+00	8	Opt	2.68e+02
lpi itest6	10	Pi	3.2e-01	20	Fail	3.4e+05	10	Opt	2.34e+08
lpi klein1	30	Pi	5.0e-01	300	Fail	4.7e+00	21	Opt	9.77e+06
lpi klein2	32	Pi	7.9e+00	300	Fail	9.9e+02	33	Opt	9.69e+06
lpi klein3	41	Pi	1.4e+01	300	Fail	2.0e+03	54	Opt	8.59e+06

Table 10.4: Iteration counts, result status and problem size for a set of Geometric programming problems

Problem name	Iteration count	Flag	Lin	Exp	Conic vars
beck751	36	Opt	4	18	58
beck752	32	Opt	4	18	58
beck753	33	Opt	4	18	58
bss2	17	Opt	1	4	13
car	35	Opt	19	142	445
demb761	300	CO	3	31	96
demb762	34	Opt	3	31	96
demb763	32	Opt	3	31	96
demb781	15	Opt	1	4	13
fang88	32	Opt	3	28	87
fiac81a	64	Pi	36	73	255
fiac81b	41	Pi	7	20	67
gptest	30	Opt	2	6	20
jha88	65	Pi	40	305	955
mra01	72	Opt	83	906	2801
mra02	127	Opt	245	3621	11108
rijc781	30	Opt	2	6	20
rijc782	26	Opt	1	9	28
rijc783	15	Pi	1	12	37
rijc784	146	Pi	3	8	27
rijc785	19	Opt	7	12	43
rijc786	20	Opt	7	12	43
rijc787	81	Opt	7	48	151

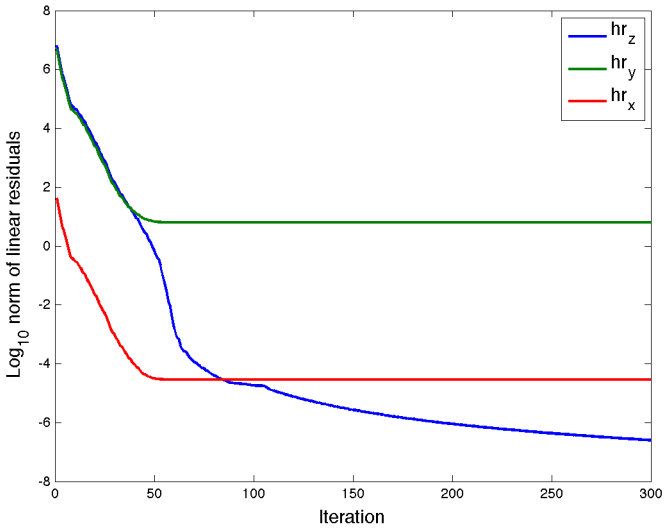

Figure 10.9: Convergence history of the homogeneous residuals for lp agg

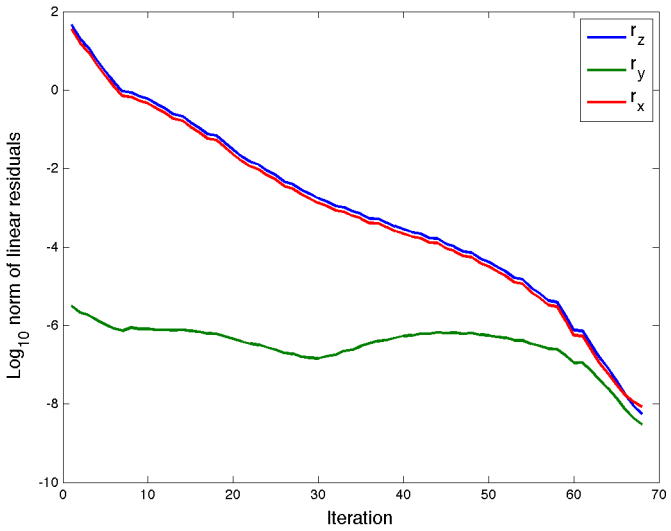

Figure 10.10: Convergence history of the linear residuals for problem lp agg after re-scaling

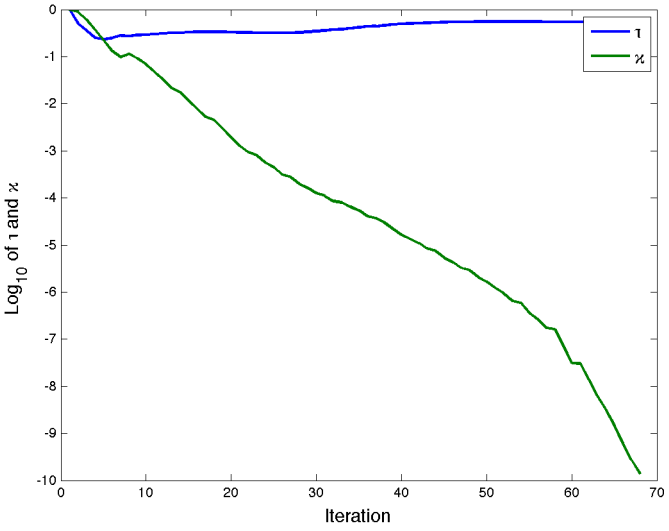

Figure 10.11: Convergence history of the homogeneous variables τ and κ for lp agg after re-scaling

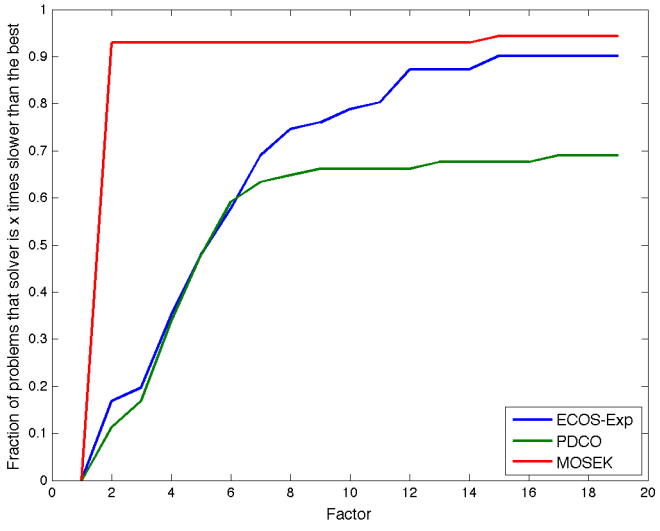

Figure 10.12: Perfomance profile for iteration count of ECOS-Exp, PDCO and MOSEK over the 72 negative-entropy problems

Chapter 11

Conclusions and future directions

We have derived a variant of the primal-dual interior methods for the full homogeneous embedding that can be used whenever a cheaply computable self-concordant barrier for an unsymmetric cone is available. This method solves systems of size comparable to the Nesterov-Todd methods for primal-dual symmetric cone programming. We have also derived a version for the simplified homogeneous embedding and shown that the functional centrality measure defined by Nesterov can be used to define a region where the Hessian of the primal barrier evaluated at the primal iterate serves to define primal-dual search directions, thus maintaining the size of the systems comparable to the symmetric versions of these algorithms. These algorithms achieve the theoretical state of the art complexity of $\mathcal{O}(\sqrt{\nu})$.

The robustness of our implementation of a Mehrotra predictor-corrector like algorithm for the exponential cone, in the extension of ECOS to support Cartesian products of symmetric cones with exponential cones, shows that these ideas are not only of theoretical use but also of practical importance.

11.1 Contributions

11.1.1 Predictor-corrector algorithms with small Newton systems

The algorithms of Chapters 6 and 7 are new variants for the family of primal-dual conic programming algorithms. Similar primal-dual potential reduction algorithms exist: by Skajaa and Ye [51], Nesterov and Nemirovski [43], Ye[55], and more. Our variants differ by the measure of centrality, the selection of directions, or the choice of homogeneous embedding. As far as we know, no algorithm for the full homogeneous embedding has been published that uses the search directions we defined. And no algorithm for the simplified homogeneous

embedding uses the centrality measure we chose. For these reasons we can count these algorithms as an original contribution.

11.1.2 The conjugate pair of functions

As far as we know, the conjugate pair of the barrier for the exponential cone from [12] defined in (8) has not been published before and we consider this a contribution of the present work. Since the definition of the centrality measure used in ECOS-Exp depends on it, the discovery of this function and the fact that it is computable are of practical importance.

11.1.3 Proofs and alternative interpretations

Although the results were already known, we derived a new proof for a key result to show that the potential reduction algorithm for general conic programming achieves the state of the art complexity of $\mathcal{O}(\sqrt{\nu})$. The proof for the complexity bound of (5) of section 5.3, Theorem (5.3.4).

The interpretation of the search directions of (6.1) as the Newton direction of a quadratic approximation to the conjugate barrier, as described in Section (6.1), is also a minor contribution.

11.1.4 Extension of ECOS

Finally, for showing that a Mehrotra predictor-corrector type algorithm is practical for unsymmetric cone problems, the extension of ECOS into ECOS-Exp amounts to another contribution. Furthermore, during the development of ECOS-Exp we had to extend the definition of the identity vectors used for symmetric cone programming. We believe that the ι vector used to initialize the exponential cone — see section 10.2.2— is a contribution in itself. In particular it is a natural way to generalize the derivation of the identity vector of a Jordan algebra into the unsymmetric exponential cone.

11.2 Future work

Of the many avenues that are left to explore, generalizing Nesterov-Todd scaling points to unsymmetric cones seems the most important. We know that these scaling points do not exist, but it is not inconceivable that a weaker version can be defined. For example it is possible that for some conjugate pairs of barriers and for any pair of primal dual feasible points x, s there exists a $\hat{w}$ such that $H(\hat{w})x = s$ but that $H(\hat{w})g(s) = g(x)$ does not hold, or otherwise that the second equation holds while the first does not. Given such a point, it is feasible to solve for the Newton direction of a modified Newton problem like that of section 6.1, using one factorization and two different right-hand sides. This could yield methods for which no centrality condition has to be enforced!

11.2.1 Conjugate pairs of barriers for other cones

Defining an algorithm that requires a conjugate pair of barriers limits its applicability and the possibility of extending ECOS further. Power cones can be used to represent a wide variety of problems. An efficient self-concordant barrier was defined by [12]. It would be of great interest to derive its conjugate dual to define an algorithm like ECOS-Exp that can also handle power cones.

11.2.2 An automatic scaling for the exponential cone

We have observed that the scaling of the solution is pretty critical, and that when the solution norms are of moderate size the iteration count is greatly reduced. We believe that implementing a strategy to rescale the problem as the iterations proceed and the norm of the solution becomes apparent will have a good impact on the practical performance of ECOS-Exp.

Bibliography

[1] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95:3–51, 2001.

[2] F. Alizadeh and S. H. Schmieta. Optimization with semidefinite, quadratic and linear constraints. Technical report, RUTCOR, Rutgers University, 1997.

[3] 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 Journal on Optimization*, 5:13–51, 1994.

[4] Erling D. Andersen and Knud D. Andersen. The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In Hans Frenk, Kees Roos, Tamas Terlaky, and Shuzhong Zhang, editors, *High Performance Optimization*, volume 33 of *Applied Optimization*, pages 197–232. Springer US, 2000.

[5] Erling D. Andersen and Yinyu Ye. A computational study of the homogeneous algorithm for large-scale convex optimization. *Computational Optimization and Applications*, 10(3):243–269, 1998.

[6] Erling D. Andersen and Yinyu Ye. On a homogeneous algorithm for the monotone complementarity problem. *Mathematical Programming*, 84(2):375–399, 1999.

[7] M. S. Andersen, J. Dahl, and L. Vandenberghe. CVXOPT: A python package for convex optimization, version 1.1.5, 2012.

[8] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization*. SIAM, Philadelphia, 2001.

[9] Jonathan M. Borwein and Adrian S. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer, 2006.

[10] Stephen Boyd, Seung-Jean Kim, Lieven Vandenberghe, and Arash Hassibi. A tutorial on geometric programming. *Optimization and Engineering*, 8(1):67–127, 2007.

[11] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[12] Peter Rober Chares. *Cones and Interior-Point Algorithms for Structured Convex Optimization involving Powers and Exponentials*. PhD thesis, Université Catholique de Louvain, 2007.

[13] Robert M. Corless and D. J. Jeffrey. The Wright Omega function. In Jacques Calmet, Belaid Benhamou, Olga Caprotti, Laurent Henocque, and Volker Sorge, editors, *Artificial Intelligence, Automated Reasoning, and Symbolic Computation*, volume 2385 of *Lecture Notes in Computer Science*, pages 76–89. Springer, Berlin and Heidelberg, 2002.

[14] T. A. Davis and Y. Hu. The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software*, 38(1), 2011.

[15] E. de Klerk, C. Roos, and T. Terlaky. Initialization in semidefinite programming via a self-dual skew-symmetric embedding. *Operations Research Letters*, 20(5):213–221, 1997.

[16] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076, 2013.

[17] Alexander Domahidi. *Methods and Tools for Embedded Optimization and Control*. PhD thesis, ETH Zurich, Switzerland, 2013.

[18] Mirjam Dür. Copositive programming a survey. In Moritz Diehl, Francois Glineur, Elias Jarlebring, and Wim Michiels, editors, *Recent Advances in Optimization and its Applications in Engineering*, pages 3–20. Springer Berlin Heidelberg, 2010.

[19] Robert Fourer and Michael Saunders. Geometric-Mean Scaling. `http://stanford.edu/group/SOL/software/pdco`.

[20] Robert M. Freund. Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function. *Mathematical Programming*, 51(1-3):203–222, 1991.

[21] Donald Goldfarb, Shucheng Liu, and Siyun Wang. A logarithmic barrier function algorithm for quadratically constrained convex quadratic programming. *SIAM Journal on Optimization*, 1(2):252–267, 1991.

[22] Clovis C. Gonzaga. Path-following methods for linear programming. *SIAM Rev.*, 34(2):167–224, June 1992.

[23] Michael Charles Grant. *Disciplined Convex Programming*. PhD thesis, Stanford University, 2005.

[24] Osman Güler. Barrier functions in interior point methods. *Mathematics of Operations Research*, 21:860–885, 1996.

[25] Osman Gler and Yinyu Ye. Convergence behavior of interior-point algorithms. *Mathematical Programming*, 60(1-3):215–228, 1993.

[26] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2008.

[27] F. Jarre. On the convergence of the method of analytic centers when applied to convex quadratic programs. *Mathematical Programming*, 49(1-3):341–358, 1990.

[28] F. Jarre and M. A. Saunders. A practical interior-point method for convex programming. *SIAM Journal on Optimization*, 5(1):149–171, 1995.

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

[30] Masakazu Kojima, Shinji Mizuno, and Akiko Yoshise. An $\mathcal{O}(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems. *Mathematical Programming*, 50(1-3):331–342, 1991.

[31] Piers W. Lawrence, Robert M. Corless, and David J. Jeffrey. Algorithm 917: Complex double-precision evaluation of the Wright Ω function. *ACM Transactions on Mathematical Softwtware*, 38(3):20:1–20:17, April 2012.

[32] Miguel Sousa Lobo, Lieven Vandenberghe, Stephen Boyd, and Herv Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(13):193–228, 1998. International Linear Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing.

[33] Z-Q. Luo, J. F. Sturm, and S. Zhang. Conic convex programming and self-dual embedding. Econometric Institute Research Papers EI 9815, Erasmus University Rotterdam, Erasmus School of Economics (ESE), Econometric Institute, 1998.

[34] Zhi-Quan Luo, Jos F. Sturm, and Shuzhong Zhang. Duality results for conic convex programming. Technical report, Erasmus University, 1997.

[35] Nimrod Megiddo. Pathways to the optimal set in linear programming. In Nimrod Megiddo, editor, *Progress in Mathematical Programming*, pages 131–158. Springer, New York, 1989.

[36] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

[37] Renato D. C. Monteiro. Polynomial convergence of primal-dual algorithms for semidefinite programming based on the Monteiro and Zhang family of directions. *SIAM Journal on Optimization*, 8(3):797–812, 1998.

[38] Arkadi Nemirovski. Advances in convex optimization: Conic programming. In Marta Sanz-Sol, Javier Soria, Juan L. Varona, and Joan Verdera, editors, *Proceedings of International Congress of Mathematicians, August 22–30, 2006 Madrid*, volume 1, pages 413–444. European Mathematical Society Publishing House, April 2007.

[39] Arkadi Nemirovski and Katya Scheinberg. Extension of Karmarkar's algorithm onto convex quadratically constrained quadratic problems. *Mathematical Programming*, 72(3):273–289, 1996.

[40] Yurii Nesterov. Long-step strategies in interior-point primal-dual methods. *Mathematical Programming*, 76(1):47–94, 1997.

[41] Yurii Nesterov. *Introductory Lectures on Convex Optimization. A Basic Course*. Kluwer Academic Publishers, 2004.

[42] Yurii Nesterov. Towards nonsymmetric conic optimization. Core discussion paper, Université Catholique de Louvain, 2006.

[43] Yurii Nesterov and Arkadi Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.

[44] Yurii Nesterov and Michael. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22(1):1–42, 1997.

[45] Yurii. Nesterov, Michael. J. Todd, and Yinyu. Ye. Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems. *Mathematical Programming*, 84(2):227–267, 1999.

[46] Florian A. Potra and Rongqin Sheng. On homogeneous interrior-point algorithms for semidefinite programming. *Optimization Methods and Software*, 9(1-3):161–184, 1998.

[47] James Renegar. A polynomial-time algorithm, based on Newton's method, for linear programming. *Mathematical Programming*, 40(1-3):59–93, 1988.

[48] James Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*. SIAM, Philadelphia, 2001.

[49] Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[50] Michael Saunders. PDCO: Primal-dual interior method for convex objectives. `http://stanford.edu/group/SOL/software/pdco`.

[51] Anders Skajaa and Yinyu Ye. A homogeneous interior-point algorithm for nonsymmetric convex conic optimization. *Mathematical Programming*, pages 1–32, 2014.

[52] Jos F. Strum. Implementation of interior point methods for mixed semidefinite and second-order cone optimization problems. Technical report, Tilburg University, The Netherlands, 2002.

[53] Xiaojie Xu, Pi-Fang Hung, and Yinyu Ye. A simplified homogeneous and self-dual linear programming algorithm and its implementation. *Annals of Operations Research*, 1996.

[54] Yinyu Ye. An $\mathcal{O}(n^3L)$ potential reduction algorithm for linear programming. *Mathematical Programming*, 50(1-3):239–258, 1991.

[55] Yinyu Ye. *Interior-Point Algorithms: Theory and Practice*. John Wiley & Sons, New York, NY, 1997.

[56] Yinyu Ye, Michael J. Todd, and Shinji Mizuno. An $\mathcal{O}(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research*, 19(1):53–67, 1994.