The Q Method for Second-Order Cone Programming

Yu Xia * Farid Alizadeh †

July 25, 2005

Key words. Second-order cone programming, infeasible interior point method, the Q method

Abstract

We develop the Q method for the Second Order Cone Programming problem. The algorithm is the adaptation of the Q method for semidefinite programming originally developed by Alizadeh, Haeberly and Overton, [3] and [2]. We take advantage of the special algebraic structure associated with second order cone programs to formulate the Q method. Furthermore we prove that our algorithm is globally convergent. Finally Some numerical results are presented.

1 Introduction

The Second Order Cone Programming (SOCP) problem is a family of convex optimization problems more general than linear programming; however it is a special case of semidefinite programming (SDP) problem. In [1] a review of this problem and its analytic, algorithmic and numerical properties, as well a number of its applications in various areas are presented. We refer the reader to this paper and references in it for further information.

In this paper we present a new interior point algorithm which is an adaptation of the Q method for semidefinite programming introduced in [3] and [2]. However we will not rely on the knowledge of SDP algorithms and will develop the Q method for the SOCP problem from ground up. We only mention that in [8] we analyze the Q method for optimization problems over any symmetric cone which include both the cone of positive semidefinite matrices and the second order cone. In the following sections we first develop the algebraic foundations in §2. In §3 we develop the Q method and derive the associated Newton direction. The global convergence analysis of this method is given in Appendix A. In §4 numerical results for both randomly generated problems, and also problems arising from Steiner tree problems are presented. Finally in §5 a modification of the Q method along with additional computational results are presented. We also present global convergence of the modified algorithm in Appendix A.

Notation

For the most part we use the notation of [1]. In particular, boldface letters such as \( a, b, x \), etc. represent column vectors and capital letters such as \( A, X \), etc. represent matrices. To refer to the \( i \)th entry of the vector \( x \) we write \( x_i \). For transpose of matrix (or vector) \( A \) we write \( A^\top \).
vector of all zeros is written as $\mathbf{0}$, and the vector of all ones as $\mathbf{1}$. If we need to join vectors and matrices, we adapt the syntax of MATLAB and similar programming languages. Thus “;” is used to join matrices and vectors in a row and “,” is used to join them in a column. For example,

$$\begin{pmatrix} x^\top, y^\top \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} = (x; y) \quad \text{and} \quad \begin{pmatrix} a \\ c \\ b \\ d \end{pmatrix} = (a, b; c, d)$$

If a vector is partitioned into several blocks we write $\mathbf{x} = (\mathbf{x}_1; \ldots; \mathbf{x}_r)$; therefore, $\mathbf{x}_i$ refers to the $i^{th}$ block and should not be confused with $x_i$. Also $\mathbf{x}_{ij}$ means the $j^{th}$ entry of the $i^{th}$ block.

For norms we use $\|\cdot\|$. In particular, $\|\mathbf{x}\|$, $\|\mathbf{x}\|_1$ and $\|\mathbf{x}\|_{\infty}$ are the Euclidean, the 1 and the infinity norms of $\mathbf{x}$, respectively.

For a vector $\mathbf{x}$, Diag$\mathbf{x}$ is a diagonal matrix whose diagonal is $\mathbf{x}$. In general, if $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are a sequence of matrices, then Diag($\mathbf{X}_1, \ldots, \mathbf{X}_n$) is a block diagonal matrix made up of $\mathbf{X}_i$.

The simple second order cone $\mathcal{Q}_{n+1}$ is defined by

$$\mathcal{Q}_{n+1} = \{ (x_0; \mathbf{x}) \mid x_0 \geq \|\mathbf{x}\| \}$$

Vectors $\mathbf{x}$ in $\mathcal{Q}_{n+1}$ are indexed from zero, and the subvector consisting of $x_1$ through $x_n$ is represented by $\mathbf{x}$. We also right $\mathbf{x}^R$ for the subvector formed from entries $x_2$ through $x_n$. Also a matrix multiplied from left by $\mathbf{x}$ has its columns indexed from zero.

The identity matrix is $I$ and the matrix $R_{n+1} = (1, \mathbf{0}^\top; \mathbf{0}, -I)$; $R$ is indexed from zero in both rows and columns. The vector $\mathbf{e}_{n+1} = (1; \mathbf{0})$, a zero indexed $(n+1)$-dimensional vector will play a fundamental role. For reasons to become clear shortly $\mathbf{e}$ is called the identity element.

More generally, we call any direct sum of simple second order cones the second order cone and write $\mathcal{Q}$. Thus,

$$\mathcal{Q} = \{ (x_1; \ldots; x_r) \mid x_i \in \mathcal{Q}_{n_i+1} \quad \text{for} \quad i = 1 \ldots r \quad \text{for simple second order cones} \quad \mathcal{Q}_{n_i+1} \}$$

In this context, $\mathbf{e} = (\mathbf{e}_{n_1+1}; \ldots; \mathbf{e}_{n_r+1})$, $R = \text{Diag}(R_{n_1+1}, \ldots, R_{n_r+1})$, and when we say $\mathbf{x} \in \mathcal{Q}$ it is understood that $\mathbf{x}$ is decomposed into blocks: $\mathbf{x} = (\mathbf{x}_1; \ldots; \mathbf{x}_r)$ consistent with $\mathcal{Q}$ and each $\mathbf{x}_i \in \mathcal{Q}_{n_i}$.

## 2 Foundations

The standard second order cone programming problem is defined as a pair of mutually dual optimization problems

**Primal:**

$$\begin{align*}
\text{min} & \quad c_1^\top x_1 + \cdots + c_r^\top x_r \\
\text{s.t.} & \quad A_1 x_1 + \cdots + A_r x_r = b \\
& \quad x_i \in \mathcal{Q}_{n_i+1} \quad i = 1, \ldots, r
\end{align*}$$

**Dual:**

$$\begin{align*}
\text{max} & \quad b^\top y \\
\text{s.t.} & \quad A_i y + z_i = c_i \quad i = 1, \ldots, r \\
& \quad z_i \in \mathcal{Q}_{n_i+1} \quad i = 1, \ldots, r
\end{align*}$$

where $\mathcal{Q}_{n_i+1}$ are simple second order cones. This problem can also be written more succinctly as

**Primal:**

$$\begin{align*}
\text{min} & \quad c^\top x \\
\text{s.t.} & \quad A x = b \\
& \quad x \in \mathcal{Q}
\end{align*}$$

**Dual:**

$$\begin{align*}
\text{max} & \quad b^\top y \\
\text{s.t.} & \quad A y + z = c \\
& \quad z \in \mathcal{Q}
\end{align*}$$

where $\mathcal{Q}$ is direct sum of $\mathcal{Q}_{n_i+1}$ and $\mathbf{x}$, $\mathbf{c}$ and $\mathbf{z}$ can be decomposed into blocks $\mathbf{x}_i$, $\mathbf{c}_i$ and $\mathbf{z}_i$ as in [1]. In this paper we switch back and forth between [1] and [2].
2.1 Duality and complementarity

It can be shown that under certain constraint qualifications the optimal values of Primal and Dual problems are equal. That is at the optimal point \((x; y; z)\) we have \(c^\top x - b^\top y = x^\top z = 0\).

Furthermore, at the optimal point a series of complementary slackness conditions analogous to those in linear programming hold. To express this complementarity conditions we first define an algebraic binary operation “◦” as follows: For \(x, z \in \mathbb{R}^{n_1+1}\), both of which are indexed from zero define

\[
x \circ z = \begin{pmatrix} x^\top z \\ x_0 z_1 + x_1 z_0 \\ \vdots \\ x_n z_0 + x_0 z_n \end{pmatrix}
\]

Note that the operation “◦” is bilinear in \(x\) and \(z\) and it is commutative, that is \(x \circ z = z \circ x\). Bilinear property of “◦” implies that there is a matrix \(X\) such that \(x \circ z = Xz\) for every \(z\). This matrix is nothing other than the “arrow-shaped” matrix representation of “Lorentz transformation” and is defined by

\[
Arw(x) = \begin{pmatrix} x_0 & x^\top \\ x & x_0 I \end{pmatrix}
\]

Thus, \(x \circ z = Arw(x)z = Arw(x)Arw(z)\), where \(e = (1; 0)\). Notice that \(e\) acts as the identity element with respect to “◦”, that is \(e \circ x = x \circ e = x\).

In the case of multiple blocks where \(x = (x_1; \ldots; x_r)\) and \(z = (z_1; \ldots; z_r)\)

\[
x \circ z = (x_1 \circ z_1; \ldots; x_r \circ z_r)
\]

Also, for a multiple block \(x\), we set \(Arw(x) = \text{Diag}(Arw(x_1), \ldots, Arw(x_r))\).

Now the complementary slackness theorem for SOCP can be formulated as follows:

**Proposition 2.1 (Complementary slackness)** If \(x = (x_1; \ldots; x_r)\) is optimal for Primal problem in (2) and \((y; z)\) with \(z = (z_1; \ldots; z_r)\) is optimal for Dual, then \(x_i \circ z_i = 0\) for \(i = 1, \ldots, r\).

The significance of the complementary slackness theorem is that the equalities for primal and dual feasibility and complementarity

\[
\begin{align*}
Ax &= b \\
A^\top y + z &= c \\
x \circ z &= 0
\end{align*}
\]

constitutes a square system of equations which in the absence of various degeneracies, determine the optimal solution uniquely, see [1].

2.2 The traditional interior point methods in a nutshell

Let \(Q_{n+1}\) be a simple second order cone and let \(x \in \text{Int } Q_{n+1}\). Then the function \(b(x) = -\ln(x_0^2 - \|x\|^2)\) is a barrier function defined on the interior of \(Q_{n+1}\). This means that \(b(x)\) is defined on \(\text{Int } Q_{n+1}\) and tends to infinity as \(x\) tends to any point on the boundary of \(Q_{n+1}\). Also observe that \(b(x)\) is strictly and strongly convex on \(\text{Int } Q_{n+1}\). When \(Q\) is the direct sum of simple second order cones \(Q_{n+1}\), define \(b(x) = \sum_i b(x_i)\).
In traditional interior point methods, \textbf{Primal} in (2) is replaced by
\begin{equation}
\min \quad c^\top x + \mu b(x)
\end{equation}

\text{s.t.} \quad Ax = b

with the understanding that \( x \) is restricted to the interior of \( Q \). Applying Karush-Kuhn-Tucker conditions to (4) and simplifying we arrive at the following system of equations:
\begin{align}
Ax &= b \\
A^\top y + z &= c \\
x \circ z &= \mu e
\end{align}

where \( e = (e_1; \ldots; e_k) \) and each \( e_i = (1; 0) \).

**Definition 1** The set of points \((x_\mu; y_\mu; z_\mu)\) that satisfy (5) as \( \mu \) varies traverse a path called the central path.

Note that \((x_\mu)^\top y_\mu = \mu \) and thus \( x_\mu^\top z_\mu = \mu r \); therefore as \( \mu \) tends to zero, the duality gap \( x_\mu^\top z_\mu \) also tends to zero, and thus \( x_\mu, y_\mu \) and \( z_\mu \) tend to the optimal solution.

The next step in solving SOCP problems in traditional interior point methods is to apply Newton’s method to the nonlinear system of equations (5). This amounts to replacing \((x; y; z)\) by \((x + \Delta x; y + \Delta y; z + \Delta z)\) in (5), with \((\Delta x; \Delta y; \Delta z)\) as our unknowns. In this approach, all nonlinear terms in the \( \Delta \)'s are removed and the Newton system, that is the following linear system is solved:
\begin{align}
A \Delta x &= b - Ax \\
A^\top \Delta y + \Delta z &= c - A^\top y - z \\
z \circ \Delta x + x \circ \Delta z &= \mu e - x \circ z
\end{align}

(Most interior point algorithms apply some linear transformation to \( x \) and \( z \) before this step in order to simplify the analysis; refer to [1] and the relevant references there for details.)

Note that the last equation can be written as
\[ Arw(z) \Delta x + Arw(x) \Delta z = \mu e - x \circ z \]

Thus in block matrix form the system of equations (6) can be written as
\[ \begin{pmatrix}
A & 0 & 0 \\
0 & A^\top & I \\
Arw(z) & 0 & Arw(x)
\end{pmatrix}
\begin{pmatrix}
\Delta x \\
\Delta y \\
\Delta z
\end{pmatrix} = \begin{pmatrix}
\Delta x \\
\Delta y \\
\Delta z
\end{pmatrix}
\]

where \( r_p = b - Ax \)
\( r_d = c - A^\top y - z \)
\( r_c = \mu e - x \circ z \)

The algorithms proceed as follows: At each iteration the system [9] is solved for a given value of \( \mu \) and a new solution \((x_{\text{new}} = x + \alpha \Delta x; y_{\text{new}} = y + \beta \Delta y; z_{\text{new}} = z + \beta \Delta y)\) is calculated. Then \( \mu \) is reduced by some factor, and \((x_{\text{new}}; y_{\text{new}}; z_{\text{new}})\) is used in place of \((x; y; z)\) in [6]. The new system is solved again, and the process is continued until we get sufficiently close to the optimal solution. With judicious choice of step lengths, reduction factor of \( \mu \) and scaling of \((x, y, z)\) it can be shown that the resulting algorithm (or class of algorithms) converge to the optimal solution in polynomial time; see references in [1].

There are two issues with this class of algorithms just described. When solving the system [6] we will end up solving equations with the \textit{Schur complement}
\[ A Arw(z)^{-1} Arw(x) A^\top \]
Without any scaling of \( x \) and \( z \), \( \text{Arw}(x) \) and \( \text{Arw}(z) \) do not commute and the Schur complement is not a symmetric matrix for points not on the central path. With certain scaling applied to \( x \) and \( z \) at each iteration, it is possible to transform the problem to an equivalent one where \( \text{Arw}(x) \) and \( \text{Arw}(z) \) in the transformed space commute making the Schur complement symmetric and positive semidefinite; see [1] for details and references. However, since the Newton method is applied to a different system at each iteration, technically it is not Newton method any more, and in particular the Newton system may converge to a singular system.

The \( Q \) method is devised to ensure that we deal with a symmetric Schur complement at each iteration while keeping the Newton system nonsingular even at the optimum.

### 2.3 Further algebraic properties of second order cones

Before describing the \( Q \) method we need to develop some machinery that simplifies presentation. For the moment let us assume that we are dealing with a simple second order cone \( Q_{n+1} \) and vectors are assumed to have just one block. As we saw the binary operation \( x \circ z \) is commutative, however the associative law does not hold in general. Instead a weaker condition holds. Writing \( x^2 = x \circ x \) it is straightforward to verify that

\[
x^2 \circ (x \circ y) = x \circ (x^2 \circ y)
\]

This identity is equivalent to stating that \( \text{Arw}(x) \) and \( \text{Arw}(x^2) \) commute.

An important fact is that the second order cone \( Q \) is the cone of squares under “\( \circ \)” operation:

\[
Q_{n+1} = \{x^2 \mid x \in \mathbb{R}^{n+1}\}
\]

There are two elementary but very important identities that are satisfied by all vectors. The first one is the quadratic identity

\[
x^2 - 2x_0x + (x_0^2 - \|x\|^2)e = 0.
\]

The polynomial \( \lambda^2 - 2x_0\lambda + (x_0^2 - \|x\|^2) \) is called the characteristic polynomial of \( x \) and plays a similar role as the characteristic polynomials of symmetric matrices. The roots of this polynomials are \( x_0 \pm \|x\| \) and are called the eigenvalues of \( x \). Only when \( x = 0 \), that is when \( x \) is a scalar multiple of the identity element \( e \), the two eigenvalues are equal.

The second important identity is

\[
x = \frac{1}{2}(x_0 + \|x\|) \left(\frac{1}{\|x\|} \right) + \frac{1}{2}(x_0 - \|x\|) \left( -\frac{1}{\|x\|} \right).
\]

Defining

\[
c_1 \overset{\text{def}}{=} \frac{1}{2} \left(\frac{1}{\|x\|} \right), \quad c_2 \overset{\text{def}}{=} \frac{1}{2} \left( -\frac{1}{\|x\|} \right), \quad \lambda_1 = x_0 + \|x\|, \quad \lambda_2 = x_0 - \|x\|,
\]

it follows that \(11\) can be written as

\[
x = \lambda_1 c_1 + \lambda_2 c_2.
\]

The relationships \(11\) and \(13\) are spectral decomposition of \( x \). They play much the same role as the eigenvalue decomposition of symmetric matrices. Observe that

\[
c_1 \circ c_2 = 0, \quad \text{and} \quad c_1^\top c_2 = 0
\]

\[
c_1^2 = c_1 \quad \text{and} \quad c_2^2 = c_2
\]

\[
c_1 + c_2 = e
\]

\[
c_1 = Rc_2 \quad \text{and} \quad c_2 = Rc_1
\]

\[
c_1, c_2 \in \text{bd} \, Q.
\]
Any pair of vectors \( \{ c_1, c_2 \} \) that satisfies properties (14), (15), and (16) is called a Jordan frame. It can be shown that every Jordan frame \( \{ c_1, c_2 \} \) is of the form (12), that is Jordan frames are pairs of vectors \( (\frac{1}{2}; \pm \| c \|) \) with \( \| c \| = \frac{2}{3} \).

When a vector has distinct eigenvalues (that is when \( x \neq 0 \)), the Jordan frame is determined uniquely. Therefore, the only vectors whose associated Jordan frames are ambiguous are multiples of the identity vector \( e \). For these vectors both eigenvalues are equal, and it turns out that we can take any Jordan frame as their associated Jordan frame. This issue will cause a slight difficulty in our algorithm which we will address when we discuss the \( Q \) method.

![Figure 1: Construction of Jordan frames from a vector \( x \).](image)

We can also define the trace and determinant of \( x \):

\[
\begin{align*}
\text{tr}(x) &= \lambda_1 + \lambda_2 = 2x_0 \quad \text{and} \\
\text{det}(x) &= \lambda_1\lambda_2 = x_0^2 - \| x \|^2.
\end{align*}
\]

Thus the barrier function defined earlier can be written as \( b(x) = -\ln \det x \).

From the spectral decomposition we can prove that the binary operation “\( \circ \)” is power associative, that is as long as we are multiplying many copies of \( x \) by itself, the order of multiplication does not matter. Thus for any nonnegative integer \( p \) we can write \( x^p \) unambiguously.

Another consequence is that the second order cone consists of those vectors \( x \) whose both eigenvalues are nonnegative and the interior of \( Q \) consists of vectors whose both eigenvalues are positive.

Using the spectral decomposition one can extend any real valued continuous functions \( f(\cdot) \) defined on \( \mathbb{R} \) to vectors:

\[
x = \lambda_1 c_1 + \lambda_2 c_2 : \quad f(x) \overset{\text{def}}{=} f(\lambda_1) c_1 + f(\lambda_2) c_2 \quad \text{whenever} \quad f(\lambda_1), f(\lambda_2) \text{ are defined}
\]

In particular, when \( x = \lambda_1 c_1 + \lambda_2 c_2 \), we may define

\[
x^{-1} \overset{\text{def}}{=} \frac{1}{\lambda_1} c_1 + \frac{1}{\lambda_2} c_2 \quad \text{when} \quad \lambda_1 \neq 0 \quad \text{and} \quad \lambda_2 \neq 0
\]

It is easily verified that \( x \circ x^{-1} = e \). Notice that the condition \( x \circ z = e \) for a given \( x \) may be satisfied by an infinite number of vectors. In other words for the binary operation “\( \circ \)” there is no unique “inverse” in general. Therefore when we say “inverse” we mean the one defined by (19).

Also note that the definition of the central path in (5) is not ambiguous. It turns out that the only
vectors \( \mathbf{x} \) for which there are more than one \( \mathbf{z} \) with \( \mathbf{x} \odot \mathbf{z} = \mu \mathbf{e} \) are those with \( x_0 = 0 \). Since in (5) we assume that \( \mathbf{x} \in \text{Int}(\mathcal{Q}) \) it follows that \( x_0 > 0 \) and \( \mathbf{z} \) is unique.

As we saw earlier, for two vectors \( \mathbf{x} \) and \( \mathbf{z} \), it is not necessarily the case that the matrices \( \text{Arw}(\mathbf{x}) \) and \( \text{Arw}(\mathbf{z}) \) commute. It turns out that the case when \( \text{Arw}(\mathbf{x}) \) and \( \text{Arw}(\mathbf{z}) \) commute is an important property that can be exploited in design of interior point algorithms, including the \( Q \) method. Therefore we present the following

**Definition 2** Two vectors \( \mathbf{x} \) and \( \mathbf{z} \) operator commute if \( \text{Arw}(\mathbf{x}) \) and \( \text{Arw}(\mathbf{z}) \) commute.

However two matrices commute if and only if they share a system of common eigenvectors. Moreover, eigenvalues and eigenvectors of \( \text{Arw}(\mathbf{x}) \) are described by the Jordan frame associated with \( \mathbf{x} \).

**Lemma 2.1** Let the spectral decomposition of \( \mathbf{x} = \lambda_1 \mathbf{c}_1 + \lambda_2 \mathbf{c}_2 \), where \( \lambda_{1,2} \) and \( \mathbf{c}_{1,2} \) are as in (11). Then eigenvalues and eigenvectors of \( \text{Arw}(\mathbf{x}) \) are given as follows:

i. \( x_0 + \|\mathbf{x}\| \) is an eigenvalue with corresponding eigenvector \( \mathbf{c}_1 \).

ii. \( x_0 - \|\mathbf{x}\| \) is an eigenvalue with corresponding eigenvector \( \mathbf{c}_2 \).

iii. All other eigenvalues are equal to \( x_0 \) and their corresponding eigenvectors are any set of orthonormal vectors, orthogonal to \( \mathbf{c}_1 \) and \( \mathbf{c}_2 \).

It is fairly simple to verify this lemma. As a direct consequence we get

**Theorem 2.1** Two vectors \( \mathbf{x} \) and \( \mathbf{z} \) operator commute if and only if they share a common Jordan frame.

In other words if \( \mathbf{x} \) and \( \mathbf{z} \) operator commute, then there is a Jordan frame \( \{\mathbf{c}_2, \mathbf{c}_2\} \) such that \( \mathbf{x} = \lambda_1 \mathbf{c}_1 + \lambda_2 \mathbf{c}_2 \) and \( \mathbf{z} = \omega_1 \mathbf{c}_1 + \omega_2 \mathbf{c}_2 \).

On the central path defined by (3) the vectors \( \mathbf{x} \) and \( \mathbf{z} \) operator commute since \( \mathbf{x} = \mu \mathbf{z}^{-1} \) for \( \mu > 0 \). Similarly it is not hard to show that at the optimum \( \mathbf{x} \) and \( \mathbf{z} \) operator commute. Therefore, the complementary slackness theorem may be restated as:

**Theorem 2.2** \( \mathbf{x} = (\mathbf{x}_1; \ldots; \mathbf{x}_r) \) is primal optimal and \( (\mathbf{y}, \mathbf{z}) \) with \( \mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_r) \) is dual optimal solutions of (1) if and only if

i. there exist Jordan frames \( \{\mathbf{c}_i, \mathbf{c}_i\} \) for \( i = 1, \ldots, r \) and pairs of real numbers \( \lambda_{i1}, \lambda_{i2} \) and \( \omega_{i1}, \omega_{i2} \) such that \( \mathbf{x}_i = \lambda_{i1} \mathbf{c}_{i1} + \lambda_{i2} \mathbf{c}_{i2} \) and \( \mathbf{z}_i = \omega_{i1} \mathbf{c}_{i1} + \omega_{i2} \mathbf{c}_{i2} \), for \( i = 1, \ldots, r \), and

ii. \( \lambda_{ij} \omega_{ij} = 0 \) for \( i = 1, \ldots, r \) and \( j = 1, 2 \).

It is possible to apply an orthogonal transformation to any Jordan frame \( \{\mathbf{c}_1, \mathbf{c}_2\} \) to change it to another Jordan frame \( \{\mathbf{f}_1, \mathbf{f}_2\} \). Such transformations are essential for the \( Q \) method. More specifically if \( \{\mathbf{c}_1, \mathbf{c}_2\} \) and \( \{\mathbf{f}_1, \mathbf{f}_2\} \) are two pairs of Jordan frames, then we wish to find a linear transformation that maps \( \mathbf{c}_1 \) to \( \mathbf{f}_1 \), \( \mathbf{c}_2 \) to \( \mathbf{f}_2 \) and the second order cone \( \mathcal{Q}_{n+1} \) back to itself. However, in general there may be many such transformations. To make this choice unique we choose to make this transformation the unique rotation in the plane spanned by \( \mathbf{c}_1 \) and \( \mathbf{f}_1 \) fixing the identity element \( \mathbf{e} \). This rotation will also map \( \mathbf{c}_2 \) to \( \mathbf{f}_2 \) and thus the frame \( \{\mathbf{c}_1, \mathbf{c}_2\} \) to \( \{\mathbf{f}_1, \mathbf{f}_2\} \). In particular when we use the standard frame \( \{\mathbf{d}_1, \mathbf{d}_2\} \), defined by setting \( \mathbf{d}_1 = \frac{1}{2}(1;1;0) \) and \( \mathbf{d}_2 = \frac{1}{2}(1;-1;0) \), then the rotation that maps the frame \( \{\mathbf{c}_1, \mathbf{c}_2\} \) to \( \{\mathbf{d}_1, \mathbf{d}_2\} \) is

\[
(20) \quad Q_\mathbf{x} \overset{\text{def}}{=} \begin{pmatrix} 1 & 0 & 0^\top \omega_1 \mathbf{e}_1 \mathbf{e}_1^\top \omega_1 \mathbf{e}_1 + x_1 \\ 0 & \frac{1}{\|\mathbf{e}_1\|} & \\ 0 & \frac{1}{\|\mathbf{e}_1\|+x_1} & \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0^\top \\ 0 & 2(\mathbf{c}_1)_1 & -2\mathbf{c}_1 \mathbf{e}_1^\top \\ 0 & 2\mathbf{c}_1 & \mathbf{e}_1^\top \end{pmatrix}.
\]
 Regardless of the dimension, a two dimensional rotation $\theta$ in the plane spanned by $\mathbf{c}_1$ and $\mathbf{d}_1$ can map a frame $\{\mathbf{c}_1, \mathbf{c}_2\}$ to frame $\{\mathbf{d}_1, \mathbf{d}_2\}$.

Figure 2: Regardless of the dimension, a two dimensional rotation $\theta$ in the plane spanned by $\mathbf{c}_1$ and $\mathbf{d}_1$ can map a frame $\{\mathbf{c}_1, \mathbf{c}_2\}$ to frame $\{\mathbf{d}_1, \mathbf{d}_2\}$.

where $\mathbf{x} = (x_2; \ldots; x_n)$, and similarly for $\mathbf{c}_1$. Also $(\mathbf{c}_1)_1$ is the entry of $\mathbf{c}_1$ indexed by 1 (noting that $\mathbf{c}_1 = (c_0; c_1; \ldots)$ is indexed from 0). Observe that this transformation really depends on the Jordan frame of $\mathbf{x}$ rather than $\mathbf{x}$ itself. Thus all vectors sharing the same Jordan frame will have the same $Q_x$.

We also note two special cases. First, if $\mathbf{c}_1 = \mathbf{d}_1, \mathbf{c}_2 = \mathbf{d}_2$, then $Q_x = I$. Second if $\mathbf{c}_1 = \mathbf{d}_2, \mathbf{c}_2 = \mathbf{d}_1$ then $Q_x = Q_{\text{flip}}$, where

\[
Q_{\text{flip}} \triangleq \begin{pmatrix}
1 & 0 & 0 & 0^T \\
0 & -1 & 0 & 0^T \\
0 & 0 & -1 & 0^T \\
0 & 0 & 0 & I
\end{pmatrix}.
\]

Thus, $Q_{\text{flip}}$ corresponds to rotation of $\pi$ which effectively maps $\mathbf{d}_1$ to $\mathbf{d}_2$ and $\mathbf{d}_2$ to $\mathbf{d}_1$.

It is well-known that if $S$ is a skew-symmetric matrix then the matrix $\exp(S) = \sum_{k=0}^{\infty} \frac{S^k}{k!}$ is orthogonal. However we are not interested in all orthogonal matrices, but only those that are of the form $Q_x$ for some $\mathbf{x}$. It turns out that a particular subset of skew-symmetric matrices generates all $Q_x$ through the matrix exponential map. More precisely, let

\[
L \triangleq \begin{cases}
1 & 0 & 0 & 0^T \\
0 & 2c_1 & -2\mathbf{c}_1^T & 0^T \\
0 & 2\mathbf{c}_1 & I - \frac{2\mathbf{c}_1^T}{1+2c_1} & 0^T
\end{cases} \mid \mathbf{c} = (c_0; c_1; \mathbf{c}) \text{ and } \{\mathbf{c}, R\mathbf{c}\} \text{ is a Jordan frame}
\]

Now define the set $I$

\[
I \triangleq \left\{ S_s = \begin{pmatrix}
0 & 0 & 0^T \\
0 & 0 & s^T \\
0 & -s & 0
\end{pmatrix} \mid s \in \mathbb{R}^{n-1} \right\}, \quad \text{and } I_{\pi} = \{S_s \in I \mid \|s\| < \pi\}.
\]

Then we have

**Proposition 2.2** The exponential map $\exp(\cdot)$ maps $I$ to $L$ and $I_{\pi}$ to $L\backslash\{Q_{\text{flip}}\}$; the latter is one-to-one.
Proof: For any $S_a \in l$,

$$S_a^2 = \begin{pmatrix} 0 & 0 & 0^T \\ 0 & -s^T & 0^T \\ 0 & 0 & -ss^T \end{pmatrix},$$

and

$$S_a^{2k+1} = (-s^T s)^k S_a, \quad S_a^{2k+2} = (-s^T s)^k S_a^2.$$ 

Hence for $\|s\| \neq 0$,

$$\exp(S_a) = I + \frac{s^2}{\|s\|^2} \left[ \sum_{i=1}^{\infty} (-1)^{i+1} \frac{\|s\|^{2i}}{(2i)!} \right] + \frac{S_a}{\|s\|} \left[ \sum_{i=0}^{\infty} (-1)^i \frac{\|s\|^{2i+1}}{(2i+1)!} \right] = I + 1 - \cos(\|s\|) \frac{S_a^2}{\|s\|^2} + \sin(\|s\|) \frac{S_a}{\|s\|}.$$ 

When $s = 0$, then $S = 0$ and $\exp(0) = I$. Now let $c \in \mathbb{R}^{n+1}$ such that $\{c, Rc\}$ is a Jordan frame; thus in particular, $\|c\| = 1/2$. Assume $|c_1| < 1/2$. Then there is a unique $0 < \alpha < \pi$ such that $\cos \alpha = 2c_1$ and $\sin \alpha = 2\|c\|$. Since $\|c\| \neq 0$, let $s = -2\alpha c$, then $\exp(S_a) = Q_c$. On the other hand, $\forall S_a \in l, S_a \neq 0$, let

$$c = \frac{1}{2} \left( 1; \cos(\|s\|); -\frac{\sin(\|s\|)}{\|s\|} s \right).$$

Then $\exp(S_a) = Q_c \in L$. For $c_1 = 1/2 \ Q_c = I$ and for $c_1 = -1/2$ note that as $\|s\| \to \pi$, the matrix $Q_c$ tends to $Q_{\text{flip}}$.

Corollary 2.1 For any $Q$ in $L$, $Q(Q) = Q$.

Proof: It is well-known that the set of orthogonal transformations that fix the coordinate $x_0$ and act on $\mathbb{R}$ of vectors $x \in Q$ are part of the automorphism group of $Q$, see [4]. And $L$ is a subset of the orthogonal transformations that fix $e$.

Instead of the exponential map we may use the Cayley transform to map skew symmetric matrices to orthogonal matrices:

$$\text{cay}(S) \overset{\text{def}}{=} \left( I + \frac{1}{2} S \right) \left( I - \frac{1}{2} S \right)^{-1}.$$ 

For the special case of $S_a$ we write $\text{cay}(s)$ for $\text{cay}(S_a)$.

Proposition 2.3 Under the Cayley transform $l$ is mapped onto $L$. Furthermore, this map is one-to-one.

Proof: When $\|\frac{1}{2}s\| < 1$, all eigenvalues of $\frac{1}{2} S_a$ are smaller than two in modulus. Therefore, the power series expansion of the $\text{cay}(s)$ is defined and can be expanded by the following power series:

$$(I + \frac{1}{2} S_a)(I - \frac{1}{2} S_a)^{-1} = I + \sum_{k=1}^{\infty} \frac{1}{2^{k+1}} S_a^k$$

(25)

Since the right hand side of (25) is well defined even for $\|s\| \geq 2$, we use the right hand side of (25) as the definition of Cayley transformation for any $S \in l$. It is not hard to see that given $S \in l$, the Cayley transformation of $S$ is $Q_c$ where

$$c = \frac{1}{2} \left( 1; \frac{4 - \|s\|^2}{4 + \|s\|^2}; -\frac{4s}{4 + \|s\|^2} \right).$$

Next let $Q_c \in L$, where $\{c, Rc\}$ is a Jordan frame, $\|c\| = 1/2$ and $c_1 \neq -1$. Set $s = -\frac{c_1}{c_0 + 1}$. Then $\text{cay}(s) = Q_c$. Note that as $s_1 \to \infty$ and $\|\frac{1}{2}s\| \to 1$, $\text{cay}(s)$ converges to $Q_d$ where $d = \frac{1}{2}(1; -1; 0)$. Propositions 2.2 and 2.3 show that the tangent space to $L$ at the identity $I$ is $l$. 

9
3 Description of the $Q$ Method

As stated earlier, the traditional interior point methods for SOCP are based on applying Newton’s method to the system of equations $[9]$. Assume that $(x; y; z)$ is our current estimate of the optimal solution. In general there is no reason for $x$ and $z$ to operator commute, even though the optimal $x^*$ and $z^*$ do. In many traditional interior point algorithms a transformation is applied to $x$ and $z$ so that in the transformed space they operator commute. The price paid for this operation is that the algorithm deviates from Newton’s method. In particular the Schur complement $A r w(z)^{-1} A r w(x) A^T$ may be undefined or singular at the optimal point.

3.1 Change of variables by spectral decomposition

The $Q$ method imposes the requirement that $x$ and $z$ should operator commute all the time. To achieve this, we make a change of variables by replacing blocks $x_i$ and $z_i$ with their spectral decomposition. Furthermore, we require that in their spectral decomposition $x_i$ and $z_i$ share a common Jordan frame $\{c_{i1}, c_{i2}\}$. Thus, if $x = (x_1, \ldots, x_r)$ and $z = (z_1, \ldots, z_r)$ where each $x_i$ and $z_i$ are in the simple second order cone $Q_{n_i + 1}$, then

$$
\lambda_{i1} c_{i1} + \lambda_{i2} c_{i2} \leftarrow x_i
$$

$$
\omega_{i1} c_{i1} + \omega_{i2} c_{i2} \leftarrow z_i
$$

With this change of variables the system $[5]$ becomes

$$
\sum_{i=1}^{r} A_i (\lambda_{i1} c_{i1} + \lambda_{i2} c_{i2}) = b
$$

(27)

$$
A_i^T y + \omega_{i1} c_{i1} + \omega_{i2} c_{i2} = c_i \text{ for } i = 1, \ldots, r
$$

$$
\lambda_{i1} \omega_{i1} = \lambda_{i2} \omega_{i2} = \mu \text{ for } i = 1, \ldots, r
$$

The difficulty with this change of variables is that we have to somehow ensure that $\{c_{i1}, c_{i2}\}$ is a Jordan frame. To this end we write $c_{i1} = Q_{c_{i1}} d_{i1}, c_{i2} = Q_{c_{i2}} d_{i2}$, where $\{d_{i1}, d_{i2}\}$ is the standard frame, and $Q_{c_{i1}} \in L$. Next we express each $Q_{c_{i}}$ by either $\exp(S_{c_{i}})$ or $\text{cay}(S_{c_{i}})$ where $S_{c_{i}} \in l$. With these modifications the system $[27]$ can be written as

$$
\sum_{i} A_i \exp(S_{c_{i}}) (\lambda_{i1} d_{i1} + \lambda_{i2} d_{i2}) = b \text{ for } i = 1, \ldots, r
$$

(28)

$$
A_i^T y + \exp(S_{c_{i}}) (\omega_{i1} d_{i1} + \omega_{i2} d_{i2}) = c_i \text{ for } i = 1, \ldots, r
$$

$$
\lambda_{i1} \omega_{i1} = \lambda_{i2} \omega_{i2} = \mu \text{ for } i = 1, \ldots, r
$$

3.2 The Newton direction

The next step is to apply Newton’s method to $[28]$. First, for multiple block system, as usual, we set $\lambda = (\lambda_{11}; \lambda_{12}; \ldots; \lambda_{r1}; \lambda_{r2}), \omega = (\omega_{11}; \omega_{12}; \ldots; \omega_{r1}; \omega_{r2}), s = (s_1; \ldots; s_r), S_{c_{i}} = \text{Diag}(S_{c_{1}}, \ldots, S_{c_{r}})$, and $Q = (Q_{1}, \ldots, Q_{r})$. The same applies to $\Delta \lambda, \Delta \omega$, etc.

To derive the the next approximation to the optimal solution using Newton’s method we make the following substitution in the primal and dual feasibility and complementarity relations $[28]$

$$
\lambda \leftarrow \lambda + \Delta \lambda
$$

$$
y \leftarrow y + \Delta y
$$

$$
\omega \leftarrow \omega + \Delta \omega
$$

$$
Q \leftarrow Q \exp(S_{c_{i}}) \text{ or } Q \leftarrow Q \text{cay}(S_{c_{i}})
$$

10
To update $Q$, we may use $\text{cay}(s)$ instead of the exponential map.

Once we make this substitution in (28) we drop all nonlinear terms involving $s$, $\Delta \lambda$, $\Delta y$, and $\Delta \omega$ and solve the resulting linear system in these variables. For $\exp(S_s)$ we simply write $I + S_s$ and drop higher order terms of its power series expansion. Since $\text{cay}(\cdot)$ and $\exp(\cdot)$ have the same first two terms in their power series expansion we make the same replacement if we are using the Cayley transform. After simplification the primal feasibility, dual feasibility and complementarity conditions result in the following linear Newton system:

\[
\begin{align*}
\sum_i B_i \left( \begin{array}{c}
\Delta \lambda_{i1} + \Delta \lambda_{i2} \\
\Delta \lambda_{i1} - \Delta \lambda_{i2} \\
\frac{\Delta \omega_{i1} + \Delta \omega_{i2}}{2} \\
\frac{\Delta \omega_{i1} - \Delta \omega_{i2}}{2}
\end{array} \right) s_i &= r_p \\
B_i \Delta y + \frac{1}{2} \left( \begin{array}{c}
\Delta \omega_{i1} + \Delta \omega_{i2} \\
\Delta \omega_{i1} - \Delta \omega_{i2} \\
\frac{\omega_{i1} + \omega_{i2}}{2} \\
\frac{\omega_{i1} - \omega_{i2}}{2}
\end{array} \right) &= Q_i^T (r_d)_i & \text{for } i = 1, \ldots, r \\
\lambda_{ij} \Delta \omega_{ij} + \omega_{ij} \Delta \lambda_{ij} &= \mu - \lambda_{ij} \omega_{ij} & \text{for } i = 1, \ldots, r, \quad j = 1, 2,
\end{align*}
\]

where

\[
\begin{align*}
B_i &= A_i Q_i \\
r_p &= b - A x \\
r_d &= c - A^T y - z \\
r_c &= \mu 1 - \lambda \cdot \omega
\end{align*}
\]

Here $\lambda \cdot \omega$ is a vector which is the component-wise product of entries of $\lambda$ and $\omega$. Also $r_d = ((r_d)_1; \ldots; (r_d)_r)$. Note that when $x$ is primal-feasible then $r_p = 0$. Similarly when $(y; z)$ is dual-feasible then $r_d = 0$. In general even if $(x; y; z)$ is feasible the solution to the Newton system (29) is not necessarily primal or dual feasible direction. This is due to the fact that the change of variables has turned both primal and dual feasibility equations nonlinear. In contrast, in traditional interior point methods, if $(x; y; z)$ is primal (respectively dual) feasible, then the Newton direction is primal (respectively dual) feasible.

Let $P = \text{Diag}(P_1, \ldots, P_r)$ where $P_i = \frac{1}{2}(1, 1; 1, -1)$. Note that $P^{-1} = 2P$. Finally let $\Lambda = \text{Diag}(\lambda)$ and $\Omega = \text{Diag}(\omega)$.

After collecting the first two columns of each $B_i$ into $\hat{B}$, the remaining columns of each $B_i$ into $\tilde{B}$, and splitting $Q^T r_d$ accordingly to $\hat{r}_d$ and $\tilde{r}_d$, we can rewrite the Newton system as

\[
\begin{align*}
P \Delta \omega + (\hat{B})^T \Delta y &= \hat{r}_d, \\
\frac{\omega_{i2} - \omega_{i1}}{2} s_i + ((\hat{B}_i))^T \Delta y &= (\hat{r}_d)_i & \text{for } i = 1, \ldots, n, \\
\hat{B} \Delta \lambda + \sum_{i=1}^n \left( \frac{\lambda_{i2} - \lambda_{i1}}{2} \hat{B}_i s_i \right) &= r_p, \\
\Lambda \Delta \omega + \Omega \Delta \lambda &= r_c.
\end{align*}
\]

Let $E_i \overset{\text{def}}{=} \frac{(\omega_{i2} - \omega_{i1})}{2} I$, $E = \text{Diag}(E_1, \ldots, E_r)$, $D_i \overset{\text{def}}{=} \frac{(\lambda_{i2} - \lambda_{i1})}{2} I$, and $D = \text{Diag}(D_1, \ldots, D_r)$. Then the solution to the Newton linear system is given by

\[
\begin{align*}
\Delta y &= \left( 2\hat{B} P \Omega^{-1} \Lambda P^T \hat{B}^T + \hat{B} D E^{-1} \hat{B}^T \right)^{-1} \left( r_p - \hat{B} P \Omega^{-1} r_c - \hat{B} D E^{-1} \hat{r}_d + 2\hat{B} P \Omega^{-1} \Lambda P^T \hat{r}_d \right) \\
\Delta \omega &= P^{-1} (\hat{r}_d - \hat{B}^T \Delta y) \\
\Delta \lambda &= \Omega^{-1} (r_c - \Lambda \Delta \omega) \\
s &= 2E^{-1} (\hat{r}_d - \hat{B}^T \Delta y).
\end{align*}
\]
Comparison of Newton systems in the $Q$ and traditional methods

1. In traditional methods our variables are $\Delta x, \Delta y, \Delta z$. When we have $r$ blocks, $\sum_i n_i = n$ and $A$ has $m$ rows then we will have a total $2n + m$ variables. In the $Q$ method for each block of size $n_i$ we have $\lambda_{i1}, \lambda_{i2}, \omega_{i1}, \omega_{i2}$ and $s_i$, for a total of $n + 2r + m$ variables.

2. The Schur complement $(BP2\Omega^{-1}AP^T \B - \hat{B}DE^{-1} \hat{B})$ is a symmetric positive definite matrix. This can be seen by writing the Schur complement as

$$AQ \text{Diag} (2P\Omega_i^{-1} \Lambda_i P^T, -D_i E_i^{-1}) Q^T A^T,$$

The traditional interior point methods require a scaling in order to get a symmetric Schur complement.

### 3.3 Centrality requirement

To ensure that the sequence of iterates in the $Q$ method converges to an optimal solution when one exists, we define a region that includes the optimal solution and require that each new approximation remain in this region. The region or neighborhood $\mathcal{N}(\gamma_c, \gamma_p, \gamma_d)$ depends on several parameters. Let $\epsilon_p, \epsilon_d,$ and $\epsilon_c$ be our tolerance for, respectively, primal infeasibility, dual infeasibility, and duality gap. In Appendix A we will explain the details of dependence of $\gamma_p, \gamma_d, \gamma_c$ on $\epsilon_p, \epsilon_d, \epsilon_c$ and other parameters derived from the data. For now let us simply assume that the $\gamma$’s are given. Then the neighborhood $\mathcal{N}$ is defined as

$$\mathcal{N}(\gamma_c, \gamma_p, \gamma_d) \overset{\text{def}}{=} \{ (\lambda, \omega, y, Q) \mid \lambda \in \mathbb{R}^{2r}, \omega \in \mathbb{R}^{2r}, y \in \mathbb{R}^m, Q \in L, \lambda > 0, \omega > 0,$$

$$\lambda_{ij} \omega_{ij} \geq \gamma_c \frac{\lambda_i \omega_j}{2r} \quad (i = 1, \ldots, r, j = 1, 2) \quad \text{and}$$

$$(\lambda^T \omega \geq \gamma_p \| AQ\hat{P} \lambda - b \| \quad \text{or} \quad \| AQ\hat{P} \lambda - b \| \leq \epsilon_p) \quad \text{and}$$

$$(\lambda^T \omega \geq \gamma_d \| A^T y + Q\hat{P} \omega - c \| \quad \text{or} \quad \| A^T y + Q\hat{P} \omega - c \| \leq \epsilon_d) \}$$

The first inequality is the centrality condition; it ensures that none of the products $\lambda_{ij} \omega_{ij}$ approach zero much faster than others. The second and third inequalities guarantee that the complementarity is not attained before primal or the dual feasibility. Obviously, when $(\gamma_c', \gamma_p', \gamma_d') \leq (\gamma_c, \gamma_p, \gamma_d), \mathcal{N}(\gamma_c', \gamma_p', \gamma_d') \subseteq \mathcal{N}(\gamma_c, \gamma_p, \gamma_d),$ and

$$\bigcup_{(\gamma_c, \gamma_p, \gamma_d) > 0} \mathcal{N}(\gamma_c, \gamma_p, \gamma_d) = \{ (\lambda, \omega, y, Q) : \lambda > 0, \omega > 0 \}.$$
Algorithm 1
Choose $0 < \sigma_1 < \sigma_2 < \sigma_3 < 1$ and $\Upsilon > 0$. To start from an arbitrary point $(\lambda^0, \omega^0, y^0, Q^0)$, one may select $0 < \gamma_c, \gamma_p > 0, \gamma_d > 0$, so that $(\lambda^0, \omega^0, y^0, Q^0) \in \mathcal{N}(\gamma_c, \gamma_p, \gamma_d)$.

Do

1. Set $\mu = \sigma_1 \lambda^T \omega^k$.
2. Compute the search direction $(\Delta \lambda, \Delta \omega, \Delta y, s)$ from (31).
3. Choose step sizes $\alpha, \beta, \gamma$ and set
   
   $\lambda^{k+1} = \lambda^k + \alpha \Delta \lambda$,
   $y^{k+1} = y^k + \beta \Delta y$,
   $\omega^{k+1} = \omega^k + \gamma \Delta \omega$,
   $Q^{k+1} = Q_{c^{k+1}}$ where $c_{k+1}$ is computed from (24) for $\exp(\gamma_c S_{s_k})$ and (26) for $\text{cay}(\gamma_c s_k)$.

4. $k \leftarrow k + 1$.

Until

either: \[ \| r_p^k \| < \epsilon_p, \| s_d^k \| < \epsilon_d, \text{ and } \lambda^T \omega^k < \epsilon_c; \]

or: \[ \| (\lambda^k, \omega^k) \|_1 > \Upsilon. \]

End

(eigenvalues of different blocks may be equal and this will not cause any difficulty). By careful choice of step sizes we can always ensure that $\lambda^{k+1}_i > \lambda^{k+1}_j > 0$ and $\omega^{k+1}_i > \omega^{k+1}_j > 0$ for each block $i$. First, if $\omega^{k+1}_i > \omega^{k+1}_j$, or $\lambda^{k+1}_i > \lambda^{k+1}_j$, then we simply swap the corresponding Jordan frame. This means that $c_{i1}, c_{i2}$ is replaced by $c_{i2}, c_{i1}$. This transformation geometrically is equivalent to a rotation of $\pi$ and algebraically equivalent to multiplication by $Q_{\text{flip}}$. Now let $\omega^{k+1}_i > \omega^{k+1}_j$. Only when $\Delta \omega_1 \neq \Delta \omega_2$ and $\beta = \frac{\omega^{k+1}_2 - \omega^{k+1}_1}{\Delta \omega_1 - \Delta \omega_2}$, it is possible that $\omega^{k+1}_i + \beta \Delta \omega_2 = \omega^{k+1}_i + \beta \Delta \omega_1$. In this case, instead of taking a uniform step size of length $\beta$ we simply take a smaller step size $\beta'$ for each block $i$ (Thus $\beta'$ are not necessarily equal). For instance $\beta'_i$ can be set to at least $\frac{\beta}{2}$. In this way we guarantee that in each block eigenvalues are distinct for both the $x_i$ and $z_i$.

Note that Algorithm 1 does not require a large number of numerical operations to calculate updates of $Q_i$, since we will use (25) for the Cayley transform and (23) for the exponential map.

Let $\alpha^k$ be the maximum of $\alpha \in [0, 1]$, so that for any $\alpha \in [0, \alpha^k]$, and

\[ \left( \lambda^k + \alpha \Delta \lambda, \omega^k + \alpha \Delta \omega, y^k + \alpha \Delta y, Q^k(I + \frac{\alpha}{2} S)(I - \frac{\alpha}{2} S)^{-1} \right) \in \mathcal{N}, \]

\[ (\lambda^k + \alpha \Delta \lambda)^T (\omega^k + \alpha \Delta \omega) \leq [1 - \alpha(1 - \sigma_2)] \lambda^k \omega^k. \]

The step sizes $\alpha \in (0, 1]$, $\beta \in (0, 1]$, $\gamma < (0, 1]$ are chosen so that

$\lambda^{k+1} = \lambda^{k+1} + \gamma \Delta \lambda$,
$\lambda^{k+1} \omega^{k+1} \leq [1 - \alpha^k(1 - \sigma_3)] \lambda^k \omega^k$.

Since $\sigma_1 < \sigma_2 < \sigma_3$, the primal and dual step sizes are not necessarily equal.

One important property of the $Q$ method is that in the absence of primal and dual nondegeneracy and presence of strict complementarity (see [1] for definitions) the Newton system (29) is nonsingular.
at the optimum. A similar result holds for the so-called AHO method for SOCP (see [1]) but not for many other traditional methods. We will not prove nonsingularity of the Newton system here since it is proved in a more general setting in [8].

In Appendix A we prove that with appropriate choice of step lengths the algorithm converges to an \((\epsilon_p, \epsilon_d, \epsilon_c)\)-optimal solution in finite number of iterations.

4 Numerical Results

To test the \(Q\) method, we have implemented a version of Algorithm 1 in MATLAB. Below we present the results of our test on 1,000 randomly generated problems with known solutions.

We test the algorithm using the Cayley transform for the most part. For the step sizes we set \(\alpha = \min(1, \tau \alpha')\), \(\beta = \min(1, \tau \beta')\), \(\gamma = \sqrt{\alpha \beta}\), where \(\alpha'\) is the maximum possible step size such that \(x + \alpha' \Delta x \in Q\). Similarly \(\beta'\) is the maximum step size such that \(z + \beta' \Delta z \in Q\). These choices are in line with implementation of traditional methods.

4.1 Test results on randomly generated problems

We use \(x_0^i = (2; 1; 0)\), \(s_0^i = (2; -1; 0)\), \(y^0 = 0\) as starting point. We pick \(\sigma = 0.25\), \(\tau = 0.99\). Our code reduced the \(l_2\) norm of primal infeasibility, \(l_2\) norm of dual infeasibility, and the duality gap to less than \(5.0e - 12\) for all problems. The data for \(A\), \(b\) and \(c\) were randomly generated between \(-0.5\) and \(0.5\). Thus, in most cases the magnitude of the solutions were comparable and there is no need to use relative accuracy.

<table>
<thead>
<tr>
<th>bk</th>
<th>dimension of each block</th>
<th>type of each block</th>
<th>m</th>
<th>(r_p^0)</th>
<th>(r_d^0)</th>
<th>it</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>[2,2,2,2,2,2,2,2,2,2]</td>
<td>[b,i,o,b,i,b,o,i,b,b]</td>
<td>12</td>
<td>342.20</td>
<td>45.59</td>
<td>27.07</td>
</tr>
<tr>
<td>10</td>
<td>[10,10,10,10,10,10,10,10,10,10]</td>
<td>[b,o,i,b,i,o,b,b,b]</td>
<td>30</td>
<td>299.69</td>
<td>142.30</td>
<td>34.16</td>
</tr>
<tr>
<td>10</td>
<td>[3,10,8,9,12,4,6,3,14,8]</td>
<td>[b,i,o,b,i,b,o,i,b,o]</td>
<td>45</td>
<td>539.07</td>
<td>146.97</td>
<td>31.46</td>
</tr>
<tr>
<td>10</td>
<td>[20,10,8,9,12,15,6,3,14,8]</td>
<td>[b,i,b,i,o,b,i,b,o]</td>
<td>55</td>
<td>861.28</td>
<td>193.32</td>
<td>33.31</td>
</tr>
<tr>
<td>10</td>
<td>[20,15,15,15,15,15,15,15,15,15]</td>
<td>[b,b,b,i,o,b,i,b,b,b]</td>
<td>75</td>
<td>1341.71</td>
<td>295.07</td>
<td>32.16</td>
</tr>
<tr>
<td>12</td>
<td>[10,10,10,10,10,10,10,10,10,10]</td>
<td>[b,o,i,b,b,i,o,b,b,o,b]</td>
<td>50</td>
<td>420.43</td>
<td>197.80</td>
<td>31.96</td>
</tr>
<tr>
<td>15</td>
<td>[10,10,10,10,10,10,10,10,10,10]</td>
<td>[b,o,i,b,b,b,i,b,b,b]</td>
<td>70</td>
<td>558.19</td>
<td>262.10</td>
<td>32.46</td>
</tr>
<tr>
<td>15</td>
<td>[15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15,15]</td>
<td>[b,i,o,b,b,i,o,b,b,b]</td>
<td>100</td>
<td>1748.47</td>
<td>375.81</td>
<td>33.46</td>
</tr>
<tr>
<td>20</td>
<td>[10,20,20,20,20,20,3,8,26,20,9,12,21,3,11,23,5,2,20,18]</td>
<td>[b,i,b,i,o,b,b,o,b,b,b,i,b,b,b,b]</td>
<td>130</td>
<td>1478.57</td>
<td>496.35</td>
<td>31.97</td>
</tr>
<tr>
<td>20</td>
<td>[20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20]</td>
<td>[b,i,b,i,o,b,b,o,b,b,b,i,b,b,b]</td>
<td>130</td>
<td>1348.60</td>
<td>572.31</td>
<td>33.94</td>
</tr>
</tbody>
</table>

Figure 3: Computational test results of performance of the \(Q\) method for 1000 randomly generated SOCP’s. For each of 10 different classes the size and type of each block along with the average number of iterations for the 100 generated instances of that type is given.

For our tests, we generated 100 random instances of “similar” classes of SOCP problems. In each class, the number of blocks \(r\), dimensions \(n_i\) for \(i = 1, \ldots, r\), number of rows of matrix \(A\), and the type of each block at the optimum is fixed. Each block can be one of three types:

**Type i**: \(x_i \in \text{Int } Q_{n_i + 1}, z_i = 0\),

**Type o**: \(x_i = 0, z_i \in \text{Int } Q_{n_i + 1}\),
Type b: \( x_i, z_i \in \text{bd} \, \mathbb{Q}_{n+1} \).

In the table in Figure 4.1 "bk" stands for the number of blocks, “type of each block” is a string made of 'i', 'o' and 'b', and “m” is the number of constraints. “\( r_p0 \)” and “\( r_d0 \)” give the average \( l_2 \) norm of initial primal and dual infeasibility for the 100 instances in each row, and “it” is the average number of iterations for the 100 instances.

Every instance was solved to optimality in fewer than 50 iterations, indicating the robustness of the algorithm. Notice that the problem type and size have little effect on the average number of iterations, which is typical of most interior point methods for SOCP.

In Figure 4.1 a plot indicating progress of one run of the algorithm is presented. The logarithm of primal and dual feasibility (as represented by euclidean norm of \( r_p \) and \( r_d \)) and duality gap are plotted against iteration count. The plot shows that, at least in this case, the order of convergence is well above 1. We have observed that this is a typical behavior for almost all runs.

It is possible to adapt Mehrotra’s predictor-corrector method (developed for linear programming in [7]) to the \( Q \) method for SOCP. However, once this predictor-corrector method was added to our code, we observed an inconsistent behavior. In most cases the number of iterations was reduced by as much as one third. In other cases though the number of iterations was increased significantly and some times, no convergence was attained at all. We believe that adaptation of the predictor-corrector method for the \( Q \) method requires further research and experimentation.

Finally we have tested the algorithm using the exponential map instead of the Cayley transform. The results were entirely compatible with Cayley transform version of the algorithm.

4.2 Testing the \( Q \) method on the Steiner minimal tree problem

We also tested the \( Q \) method on problems arising from euclidean minimal Steiner tree problems. Details can be found in the paper of Xue and Ye, [9]. Here we briefly mention that in the Steiner tree problem we are searching for a tree consisted of a set of \( N \) given points (called regular points) and possibly additional vertices (called Steiner points) such that the sum of (euclidean) length of edges in the tree is minimum. The problem may be decomposed into two subproblems. In the first subproblem we need to find a particular topology, that is a set of Steiner points. In the second subproblem, given the topology we need to find the optimal position of the Steiner points. The first
subproblem—finding a particular topology—is combinatorial in nature, the second can be formulated as an SOCP problem. It can be shown that in the optimal tree the number of Steiner points is at most $N - 2$ and the degree of each Steiner point is three.

We tested our code on Example 1 in [9]. We use the same starting points as we did in the randomly generated SOCP's of the previous section. The progress of the $Q$ method in each iteration is summarized in Figure 4.2.

<table>
<thead>
<tr>
<th>it</th>
<th>network-cost</th>
<th>$r_p$</th>
<th>$r_d$</th>
<th>gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>67.4046273974</td>
<td>5.744563e+000</td>
<td>2.631707e+001</td>
<td>1.020000e+002</td>
</tr>
<tr>
<td>1</td>
<td>46.4511882048</td>
<td>1.612290e+000</td>
<td>1.531418e+001</td>
<td>4.650618e+001</td>
</tr>
<tr>
<td>2</td>
<td>54.7067952019</td>
<td>2.814354e+000</td>
<td>1.253095e+001</td>
<td>2.718293e+001</td>
</tr>
<tr>
<td>3</td>
<td>46.3246649054</td>
<td>2.594265e+000</td>
<td>1.055171e+001</td>
<td>2.233545e+001</td>
</tr>
<tr>
<td>4</td>
<td>51.9713738708</td>
<td>2.814354e+000</td>
<td>9.177738e+000</td>
<td>1.840966e+001</td>
</tr>
<tr>
<td>5</td>
<td>46.4651882048</td>
<td>1.612290e+000</td>
<td>1.531418e+001</td>
<td>4.650618e+001</td>
</tr>
<tr>
<td>6</td>
<td>39.0432902564</td>
<td>3.149857e+000</td>
<td>8.887436e+000</td>
<td>1.223603e+001</td>
</tr>
<tr>
<td>7</td>
<td>30.2772082577</td>
<td>1.664164e+000</td>
<td>3.598923e+000</td>
<td>5.686488e+000</td>
</tr>
</tbody>
</table>

In contrast to the algorithm used in Xue and Ye [9] we start ours from an infeasible point. We achieve the same accuracy as Xue and Ye did at iteration number 27.

4.3 Taking advantage of sparsity

Our implementation of Algorithm 1 in MATLAB does not take advantage of sparsity in the input matrix $A$. Here we briefly state how sparse input matrices $A$ can be exploited and compare the situation with traditional interior point methods.

Let both $A$ and $AA^\top$ be sparse matrices. It is well-known that in interior point methods for linear programming the Schur complement in all versions are of the form $AF A^\top$ for some diagonal matrix $F$. Also the “sparsity pattern of $AA^\top$ and $AF A^\top$ are identical. The most difficult part for a sparse Cholesky factorization is to find a good order for pivoting. Once such an order is found for $AA^\top$, the same ordering may be used for all subsequent iterations for FACTORING $AF A^\top$.
The situation for SOCP is more involved, see [1] for a detailed analysis for the case of traditional interior point methods. However, in all methods we have the fact that the Schur complement at each iteration is of the form $AFA^\top + L$ where $L$ is of rank $r$, $2r$ or $3r$, and is symmetric for most methods (except for the so-called AHO method).

We now show that the Schur complement for the $Q$ method is also of the same format. For convenience let us temporarily assume that $r = 1$. In this case, as stated in [3] the Schur complement is given by

$$AQ\text{Diag}(2P\Omega_1^{-1}Λ, P^\top, -D_iE_i^{-1})Q^TA^\top,$$

The Cholesky factorization of this matrix is the main computational effort at each iteration. However it is easily verified that the matrices $Q_e \in L$ are all of the form of $I + \text{rank-2}$ matrix. Furthermore, with some algebraic manipulation it can be verified that the

$$Q\text{Diag}(2P\Omega_1^{-1}Λ, P^\top, -D_iE_i^{-1})Q^T$$

is also sum of a diagonal matrix with positive entries and a positive semidefinite rank-2 matrix. Thus, the Schur complement is of the form $AFA^\top + \text{a rank-2}$ matrix. With $r$ blocks, the Schur complement is $AFA^\top + \text{a rank-2}$ matrix. (In both cases $F$ is a diagonal.) As in linear programming, sparse factorization of $A^\top A$ will yield sparse factorization of $ADA^\top$. Sophisticated numerical techniques are available to extend this sparse factorization to ones with low rank updates, see [1] for details and references.

5 A Modified $Q$ Method

In this section, we give a variant of the $Q$ method for SOCP, which has similar properties and convergence results as Algorithm 1.

5.1 The Newton system

We can write dual and primal feasibility and complementarity relations alternatively in the following form

$$Q\hat{P}\omega + A^\top y = c,$$

$$AQ\hat{P}\lambda = b,$$

$$\lambda \cdot \omega = \mu 1$$

where $\hat{P} = \text{Diag}(\hat{P}_1, \ldots, \hat{P}_r)$ where each $\hat{P}_i = (d_{i1}, d_{i2})$, made up of the standard frame in $\mathbb{R}^{n_i}$. Thus each block $\hat{P}_i$ is an $n_i \times 2$ matrix with only the $2 \times 2$ top part nonzero. Now in (32) The matrix $Q$ appears only as $AQ\hat{P}$. Thus for each block $i$ only columns indexed zero and 1 of each $Q_i$ is involved. Since each $Q_i$ is of the form $Q_e \in L$, the 0th column is all zeros except the first entry. The column indexed 1 is of the form $(0; 2R)$ where $\{c, R\}$ is a Jordan frame. In particular, defining $q = 2R$ it follows that the only relevant part of $Q_i$ in (32) is the vector $q_i$, of the column indexed 1. Furthermore $q_i^\top q_i = 1$.

Partitioning $A_i$ into $A_i = ((A_i)_{00}, \hat{A}_i)$ we write $A \overset{\text{def}}{=} (\hat{A}_1, \ldots, \hat{A}_n)$. Observe that $x_i$ and $z_i$ may be written as

$$x_i = \left(\frac{\lambda_{i1} + \lambda_{i2}}{2}, \frac{\lambda_{i1} - \lambda_{i2}}{2} q_i\right), \quad z_i = \left(\frac{\omega_{i1} + \omega_{i2}}{2}, \frac{\omega_{i1} - \omega_{i2}}{2} q_i\right).$$

We set $q_i = (1; 0)$ when $\tilde{x}_i = 0$. Then the decomposition (33) is unique when $\lambda_{i1} \geq \lambda_{i2}$ and $\omega_{i1} \leq \omega_{i2}$. We now substitute (33) into (32), and add constraints $q_i^\top q_i = 1$ for $i = 1, \ldots, r$. The
residuals \( r_p, r_d \) and \( r_c \) are defined as before. Let \((\tilde{r}_d)_{i}^{k}\) be the first element of \((r_d)_{i}^{k}\), and \((\tilde{r}_d)_{i}^{k}\) the remaining subvector. Then, similar to Algorithm 1 we make the substitution \( \lambda \leftarrow \lambda + \Delta \lambda, \omega \leftarrow \omega + \Delta \omega, \) and \( y \leftarrow y + \Delta y \). But now, instead of exponential map or the Cayley transform we simply substitute \( q \leftarrow q + \Delta q \). After making these substitutions in (32) and dropping nonlinear terms in \( \Delta \)'s we get the linear Newton system

\[
\frac{\Delta \omega_{i2}}{2} + (A_i)_0^\top \Delta y = (\tilde{r}_d)_{i}^{k}
\]

\[
\frac{\Delta \omega_{i2}}{2} \Delta q_i + \omega_{i}^{k} \Delta q_i + (\tilde{A}_i)^\top \Delta y = (\tilde{r}_d)_{i}^{k}
\]

\[
\sum_{i=1}^{n} \left( \frac{\Delta \lambda}{2} + \Delta \lambda \right) (A_i)_0 + \tilde{A}_i \frac{\Delta \lambda}{2} q_i^k + \tilde{A}_i \frac{\lambda}{2} \Delta q_i^k = r_p^k
\]

\[
q_i^k \Delta q_i = 0 \quad (i = 1, \ldots, n)
\]

\[
A^k \Delta \omega + \Omega^k \Delta \lambda = r_c^k.
\]

where, as before \( \Lambda = \text{Diag}(\lambda) \) and \( \Omega = \text{Diag}(\omega) \). The modified algorithm is similar to Algorithm 1 with the exception that the new \( q_i \) should be normalized:

\[
q_i^{k+1} = \frac{q_i^{k+\gamma \Delta q_i}}{\|q_i^{k+\gamma \Delta q_i}\|_2}.
\]

### 5.2 The solution to modified \( Q \) method

Let

\[
u_i \overset{\text{def}}{=} \frac{\lambda_{i1}}{\omega_{i1}} + \frac{\lambda_{i2}}{\omega_{i2}}, \quad v_i \overset{\text{def}}{=} \frac{\lambda_{i1}}{\omega_{i1}} - \frac{\lambda_{i2}}{\omega_{i2}}.
\]

\( E_i \) and \( D_i \) are defined as that in section 3 but with dimensions adjusted properly. Omitting the superscript \( k \), the Schur complement \( M \) can be written as

\[
M = \sum_{i=1}^{n} \frac{\nu_i}{2} \left( ((A_i)_0 q_i^k \tilde{A}_i^\top + \tilde{A}_i q_i^k (A_i)_0^\top) + \tilde{A}_i \text{Diag} \left( \frac{\nu_i}{2}, (D_i E_i^{-1} + \frac{\nu_i}{2} I) q_i^k q_i^k - D_i E_i^{-1} \right) \right) \tilde{A}_i^\top.
\]

The solution to (34) is

\[
\Delta y = M^{-1} \left( r_p + \frac{1}{2} \sum_{i=1}^{n} \left( - (A_i)_0 (r_c)_i^\top \omega_{i1}^{-1} - \tilde{A}_i q_i^k (r_c)_i^\top (1 - 1) \omega_{i1}^{-1} + v_i (A_i)_0 \tilde{A}_i q_i^k + v_i (A_i)_0 q_i^k (\tilde{A}_i)_i^\top + (u_i + 2 D_i E_i^{-1}) \tilde{A}_i q_i^k (\tilde{r}_d)_i - 2 D_i E_i^{-1} \tilde{A}_i (\tilde{r}_d)_i^\top) \right) \right)
\]

\[
\Delta \omega_{i2} = -q_i^k ((\tilde{r}_d)_i - q_i (\tilde{r}_d)_i) + (q_i^k A_i^\top - (A_i)_0^\top) \Delta y
\]

\[
\Delta q_i = E_i^{-1} ((\tilde{r}_d)_i - q_i (\tilde{r}_d)_i) + q_i \Delta \omega_{i2} - (A_i^\top - q_i (A_i)_0^\top) \Delta y
\]

\[
\Delta \omega_{i1} = 2(\tilde{r}_d)_i - 2(A_i)_0^\top \Delta y - \Delta \omega_{i2}
\]

\[
\Delta (\lambda) = \Omega^{-1} (r_c - \Lambda \Delta \omega).
\]

### Properties of the solution

1. The number of unknowns in the modified \( Q \) method is the same as the \( Q \) method: Instead of the vector \( s_i \) we now have \( q_i \) in each block \( i \).

2. (a) The dimension of each block \( M_i \) of \( M \) is one smaller than the Schur complement of the traditional methods.
(b) To keep each iterate in $Q$, we only need to compute $\alpha^{-1} \geq \max\{-\Delta \lambda_i / \lambda_i : \Delta \lambda_i < 0\}$. In comparison, in traditional methods we need to solve a quadratic equation for each block.

(c) When $\lambda_{i1} > \lambda_{i2} > 0$ and $\omega_{i2} > \omega_{i1} > 0$, we have $u_i > v_i > 0$, and $-D_i E_i^{-1}$ is a positive scalar matrix. Therefore it is easily verified that each $M_i$ is symmetric positive definite and we may use Cholesky factorization to solve systems of equations coefficient matrix $M$.

3. **Accuracy and stability.** It can be shown that at the optimum the Newton system (34) is nonsingular if primal and dual nondegeneracy and strict complementarity holds. This in turn implies that the algorithm converges quadratically near the optimal solution.

4. We can show that with careful choice of parameters the modified $Q$ method converges to the optimal solution in finite number of iterations. This is argued in Appendix [A].

5.3 Numerical examples

We have implemented the basic algorithm for the modified $Q$ method in MATLAB and have tested on 1,000 randomly generated problems. Step sizes $\alpha, \beta, \gamma$ are chosen similar to our implementation of Algorithm 1. We applied our modified $Q$ code to the same set 10 classes of SOCP problems each with 100 instances. The results are summarized in Figure 5.3. Although the algorithm finds an $(\epsilon_p, \epsilon_d, \epsilon_c)$-optimal solution for all of the 1,000 problems, a small portion of them need more than 100 iterations to reach the required accuracy.

Finally we applied our implementation of the modified $Q$ method to the same instance of the Steiner tree problem we used for our implementation of Algorithm 1. The progress of the algorithm is reported in Figure 5.3. The performance is similar to Algorithm 1.

6 Conclusions

We have developed and analyzed the $Q$ method and a modified variant for SOCP. Preliminary numerical results show that the algorithm is robust.

We conclude by mentioning some future research directions. First, it remains open whether the $Q$ method has polynomial time iteration complexity. The finite convergence result presented in this paper depends on the condition number of the input matrix $A$.

For practical and stable implementation of the method, Mehotra’s predictor-corrector method should be adjusted to perform more robustly. Finally, a serious implementation should take advantage of sparse factorization of the Schur complement.
<table>
<thead>
<tr>
<th>it</th>
<th>network-cost</th>
<th>$r_p$</th>
<th>$r_d$</th>
<th>gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>67.404627397</td>
<td>5.744563e+000</td>
<td>2.631707e+001</td>
<td>1.020000e+002</td>
</tr>
<tr>
<td>1</td>
<td>46.4651882048</td>
<td>1.570830e+000</td>
<td>1.536167e+001</td>
<td>4.650618e+001</td>
</tr>
<tr>
<td>2</td>
<td>57.9523360177</td>
<td>2.854242e+000</td>
<td>1.322494e+001</td>
<td>2.962662e+001</td>
</tr>
<tr>
<td>3</td>
<td>36.5711471818</td>
<td>2.257227e+000</td>
<td>8.998654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>4</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>5</td>
<td>57.9523360177</td>
<td>2.854242e+000</td>
<td>1.322494e+001</td>
<td>2.962662e+001</td>
</tr>
<tr>
<td>6</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>7</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>8</td>
<td>57.9523360177</td>
<td>2.854242e+000</td>
<td>1.322494e+001</td>
<td>2.962662e+001</td>
</tr>
<tr>
<td>9</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>10</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>11</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>12</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>13</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>14</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>15</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>16</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>17</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>18</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>19</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
<tr>
<td>20</td>
<td>36.5711471818</td>
<td>2.264570e+000</td>
<td>8.885654e+000</td>
<td>1.977556e+001</td>
</tr>
</tbody>
</table>

Figure 5: Progress of the modified $Q$ method for one instance of a Steiner tree problem with given topology.

A Appendix: Convergence Analysis of the $Q$ and Modified $Q$ Methods

A.1 Convergence analysis

Global convergence of Algorithm 1 can be established by adopting ideas of Kojima, Megiddo and Mizuno, \cite{6} for infeasible interior point linear programming.

**Theorem A.1** If Algorithm 1 does not stop after a finite number of steps, the smallest singular value of the Jacobian of \eqref{30} must converge to zero.

**Proof:** Let $J$ be the Jacobian of \eqref{30}. The key to the proof is to show that the step sizes are bounded below. Assume the algorithm doesn’t stop after a finite number of steps. Define $\epsilon^* \overset{\text{def}}{=} \min(\epsilon_c, \gamma_p \epsilon_p, \gamma_d \epsilon_d)$. Then for each iteration $k$,

$$(\lambda^k)^T \omega^k \geq \epsilon^*, \quad \text{and} \quad \|\lambda^k; \omega^k\|_1 \leq \Upsilon,$$

otherwise, the termination criterion will force the algorithm to stop. Boundedness of $\mathbf{y}^k$ is explained by the dual feasibility constraint. Also $Q^k$ is orthogonal and the set of orthogonal matrices is compact. Assume the smallest singular value of $J$ does not converge to zero. Then there must exist a positive scalar $d$, and a subsequence $\{(\lambda^m, \omega^m, \mathbf{y}^m, Q^m)\}_{i=1}^\infty$ such that for all $m_i$, the largest singular value of $J^{-1}$ is at most $\frac{1}{d}$. Both the right hand side and the left hand side of \eqref{30}
depend continuously on \((\lambda, \omega, y, Q)\). Since the set \((\lambda, \omega, y, Q)\) derived from the Newton system forms a compact set the Newton direction in (30) is a continuous function of \((\lambda, \omega, y, Q)\). Therefore, solution of (30) is uniformly bounded for the subsequence \(\{m_i\}\). Hence, there exists a positive constant \(\eta\), such that the search direction computed by (30) satisfies

\[
\left| \Delta \lambda_{ij} \Delta \omega_{ij} - \frac{\gamma_c}{2n} \Delta \lambda^T \Delta \omega \right| \leq \eta, \quad \left| \Delta \lambda^T \Delta \omega \right| \leq \eta, \quad \| \Delta \lambda \| \leq \eta,
\]

\[
\| \Delta \omega \| \leq \eta, \quad \| s_i \| \leq \eta, \quad \text{for } i = 1, \ldots, n; \ j = 1, 2.
\]

Note that \(\| S_i \|_2 = \| s_i \|\) for \(i = 1, \ldots, n\), \(\| S \|_2 = \max_i \| s_i \|\). For \(k \in \{ m_i \}_{i=1}^\infty\), adopting the notation used in [6], we define

\[
f_{ij}(\alpha) \overset{\text{def}}{=} \lambda_{ij}^k + \alpha (\Delta \lambda_{ij}) \left[ \omega_{ij}^k + \alpha (\Delta \omega_{ij}) - \frac{\gamma_c}{2n} (\lambda^k + \alpha \Delta \lambda)^T (\omega^k + \alpha \Delta \omega) \right],
\]

\[
g_p(\alpha) \overset{\text{def}}{=} (\lambda^k + \alpha \Delta \lambda)^T (\omega^k + \alpha \Delta \omega) - \gamma_p \left\| AQ^k (I + \alpha S - \frac{\alpha}{2} S) - \frac{\alpha}{2} \right\| - \nu_k - \frac{1}{2} P \left( \lambda^k + \alpha \Delta \lambda \right) - b \right\|,
\]

\[
g_d(\alpha) \overset{\text{def}}{=} (\lambda^k + \alpha \Delta \lambda)^T (\omega^k + \alpha \Delta \omega) - \gamma_d \left\| A^T (y^k + \Delta y) + Q^k (I + \alpha S - \frac{\alpha}{2} S) - \frac{1}{2} P \left( \omega^k + \alpha \Delta \omega \right) - c \right\|,
\]

\[
h(\alpha) \overset{\text{def}}{=} \left[ 1 - \alpha (1 - \sigma_2) \right] \lambda^T \omega - (\lambda^k + \alpha \Delta \lambda)^T (\omega^k + \alpha \Delta \omega).
\]

Therefore, \(\alpha^k\) is the largest \(\alpha\) satisfying the following inequalities:

\[
f_{ij}(\alpha) \geq 0 \quad (i = 1, \ldots, n; \ j = 1, 2),
\]

\[
g_p(\alpha) \geq 0 \quad \text{or} \quad \left\| AQ^k \lambda^k - b \right\| \leq \epsilon_p,
\]

\[
g_d(\alpha) \geq 0 \quad \text{or} \quad \left\| A^T y^k + Q^k \omega - c \right\| \leq \epsilon_d,
\]

\[
h(\alpha) \geq 0.
\]

Next, we give a lower bound for the set \(\{ \alpha^k \}\). Recall that each block of the Cayley transformation is given by

\[
I + \alpha S_i - \frac{\alpha^3}{4 + \alpha^2 \| s_i \|^2} S_i^2 + \frac{2\alpha^2}{4 + \alpha^2 \| s_i \|^2} S_i^2.
\]

The inequalities for \(f_{ij}\) and \(h\) can be obtained by similar arguments as those in [6]:

\[
f_{ij}(\alpha) \geq \sigma_1 \frac{\epsilon^*}{2n} (1 - \gamma_c) \alpha - \eta \alpha^2,
\]

\[
h(\alpha) \geq (\sigma_2 - \sigma_1) \epsilon^* \alpha - \eta \alpha^2.
\]

Next, we estimate \(g_p(\alpha)\) and \(g_d(\alpha)\). Note that the first column of \(S_i^2\) is zero; and the only nonzero entry of its second column is \(-s_i^T s_i\). Let \(Q^k_2\) be the matrix consisting of only the 2nd column of each block of \(Q^k\), \(\lambda_1\) be the vector made of all of the larger of the two eigenvalues of \(x_i\) and \(\lambda_2\) be the vector made up of all the smaller of the two eigenvalues of \(x_i\), for \(i = 1, \ldots, r\). When

---

1. Recall that \(\| S \|_2\), the spectral norm of \(S\), is given by the modulus of the largest-modulus eigenvalue of \(S\).
\[ \lambda^k \omega^k \geq \gamma_p \| A Q^k \tilde{P} \lambda^k - b \|, \]

\[ (36) \quad g_p(\alpha) \geq (1 - \alpha) \lambda^k \omega^k + \alpha \sigma_1 \lambda^k \omega^k + \alpha^2 \Delta \lambda^k \Delta \omega - \gamma_p (1 - \alpha) \| A Q^k \tilde{P} \lambda^k - b \| \]

\[ - \gamma_p \alpha^2 \left[ \| A Q^k S \tilde{P} \Delta \lambda \| + \frac{\max_i \| s_i \|^2}{4} \| A Q^k (\lambda^k - \Delta \lambda_1 - \alpha \Delta \lambda_2) \| \right. \]

\[ + \frac{\alpha \max_i \| s_i \|^2}{4} \| A Q^k S \tilde{P} (\lambda^k + \alpha \Delta \lambda) \| \right] \]

\[ \geq \alpha \sigma_1 \epsilon_* - \alpha^2 \eta - \gamma_p \alpha^2 \| A \| \left( \frac{1}{\sqrt{2}} \eta^{32} + \frac{1}{2} \eta \eta + \frac{1}{2 \sqrt{2}} \eta^{32} \eta + \frac{1}{4 \sqrt{2}} \eta^{32} \right). \]

The first inequality follows from the Newton linear system for the search directions \((30)\), the fact that \((\lambda^k)^\top \omega \geq \gamma_p \| A Q^k \tilde{P} \lambda - b \|\), and the Cayley transform \((35)\). The second inequality follows from the bound on the variables and search directions, the fact that \(\alpha \leq 1\), and

\[ \| \tilde{P} \lambda \| = \frac{1}{\sqrt{2}} \| \lambda \| \leq \frac{1}{\sqrt{2}} \| \lambda \|_1. \]

If \( \| A Q^k \tilde{P} \lambda^k - b \| \leq \epsilon_p \), then

\[ (37) \quad \| A Q^k (I + \frac{\alpha}{2} S)^{-1} \tilde{P} (\lambda^k + \alpha \Delta \lambda) - b \| \leq (1 - \alpha) \| A Q^k \tilde{P} \lambda^k - b \| \]

\[ + \alpha^2 \left[ \| A Q^k S \tilde{P} \Delta \lambda \| + \frac{\max_i \| s_i \|^2}{4} \| A Q^k (\lambda^k - \Delta \lambda_1 - \alpha \Delta \lambda_2) \| \right. \]

\[ + \frac{\alpha \max_i \| s_i \|^2}{4} \| A Q^k S \tilde{P} (\lambda^k + \alpha \Delta \lambda) \| \right] \]

\[ \leq (1 - \alpha) \epsilon_p + \alpha^2 \| A \| \left( \frac{1}{\sqrt{2}} \eta^{32} + \frac{1}{2} \eta \eta + \frac{1}{2 \sqrt{2}} \eta^{32} \eta + \frac{1}{4 \sqrt{2}} \eta^{32} \right). \]

Therefore when \( \alpha \leq \epsilon_p \)

we must have \( \| A Q^{k+1} \tilde{P} \lambda^{k+1} - b \| \leq \epsilon_p \). Next, we will consider the dual constraints.

When \((\lambda^k)^\top \omega^k \geq \gamma_d \| A^\top y^k + Q^k \tilde{P} \omega^k - c \|\),

\[ (38) \quad g_d(\alpha) \geq (1 - \alpha) \lambda^k \omega^k + \alpha \sigma_1 \lambda^k \omega^k + \alpha^2 \Delta \lambda^k \Delta \omega \]

\[ - \gamma_d (1 - \alpha) \| A^\top y^k + Q^k \tilde{P} \omega^k - c \| - \gamma_d \alpha^2 \left[ \| Q^k S \tilde{P} \Delta \omega \| \right] \]

\[ + \frac{2}{4} \| Q^k S \tilde{P} (\omega^k + \alpha \Delta \omega) \| + \frac{\alpha \max_i \| s_i \|^2}{4} \| Q^k S \tilde{P} (\omega^k + \alpha \Delta \omega) \| \]

\[ \geq \alpha \sigma_1 \epsilon_* - \alpha^2 \eta - \gamma_d \alpha^2 \left( \frac{1}{\sqrt{2}} \eta^{32} + \frac{1}{2} \eta \eta + \frac{1}{2 \sqrt{2}} \eta^{32} \eta + \frac{1}{4 \sqrt{2}} \eta^{32} \right). \]
When \( \| A^T y^k + Q^k \tilde{P} \omega^k - c \| \leq \epsilon_d \),

\[
\| A^T (y^k + \alpha \Delta y) + Q^k (I + \frac{\alpha}{2} S)(I - \frac{\alpha}{2} S)^{-1} \tilde{P} (\omega^k + \alpha \Delta \omega) - c \|
\leq (1 - \alpha) \| A^T y^k + Q^k \tilde{P} \omega^k - c \| + \alpha^2 \left( \| Q^k S \tilde{P} \Delta \omega \| + \frac{\max_i \| s_i \|^2}{4} \right). 
\]

Thus, the lower bound on \( \hat{\alpha} \)'s is

\[
\alpha^* = \min \left\{ 1, \frac{(\sigma_2 - \sigma_1) \epsilon^*}{\eta}, \frac{\sigma_1 (1 - \epsilon_c) \epsilon^*}{2n \eta}, \frac{\sigma_1 \epsilon^*}{\eta + \gamma_p \| A \| \left( \frac{1}{\sqrt{2}} \eta^{3/2} + \frac{1}{4} \eta Y + \frac{1}{2\sqrt{2}} \eta^2 Y + \frac{1}{4\sqrt{2}} \eta^{3/2} Y + \frac{1}{4\sqrt{2}} \eta^{5/2} \right)}, \frac{\epsilon_d}{\frac{1}{\sqrt{2}} \eta^{3/2} + \frac{1}{4} \eta Y + \frac{1}{2\sqrt{2}} \eta^2 Y + \frac{1}{4\sqrt{2}} \eta^{3/2} Y + \frac{1}{4\sqrt{2}} \eta^{5/2}} \right\}.
\]

After adjusting step sizes to ensure \( \lambda_1 > \lambda_2 \) and \( \omega_2 > \omega_1 \) (as discussed at the end of [3]), the lower bound on \( \alpha^k \) is at least \( \frac{\alpha^*}{2} \). The algorithm tracks the decrease of the sequence \( \left\{ (\lambda^j)^\top \omega^j \right\}_{j=1}^\infty \). Therefore, for each \( m_i \) in the subsequence, by \( h(\alpha) \geq 0 \), we have

\[
(\lambda^{m_{i+1}})^\top \omega^{m_{i+1}} \leq \left( 1 - \frac{\alpha^*}{2} (1 - \sigma_3) \right)^2 (\lambda^{m_i})^\top \omega^{m_i} \leq \left( 1 - \frac{\alpha^*}{2} (1 - \sigma_3) \right)^2 (\lambda^{m_{i-1}})^\top \omega^{m_{i-1}} \leq \cdots \leq \left( 1 - \frac{\alpha^*}{2} (1 - \sigma_3) \right)^i (\lambda^{m_1})^\top \omega^{m_1}.
\]

Hence the entire sequence \( \left\{ (\lambda^j)^\top \omega^j \right\}_{j=1}^\infty \) converges to 0, which contradicts our assumption. We have proved that if the smallest singular value of [30] doesn’t converge to zero, either the algorithm finds an \((\epsilon_p, \epsilon_d, \epsilon_c)\)-optimal solution in finite iterations, or the sequence of solutions is unbounded. \( \blacksquare \)

**A.2 Finite convergence**

Algorithm 1 may abort due to unboundedness of eigenvalues or singularity of Jacobian. In this section, we give some conditions under which Algorithm 1 converges to an \((\epsilon_p, \epsilon_d, \epsilon_c)\)-optimum in finite iterations. To achieve this we need to derive conditions that ensure boundedness of estimates \((\lambda, \omega, y, Q)\). To this end we adopt some of the ideas of Freund and Jarre [5], which were stated originally for linear programming.

\[
23
\]
Let $\rho$ represent a positive scalar not larger than the smallest singular value of $A$. Suppose the SOCP problem (2) has a primal-dual feasible solution $(\bar{x}, \bar{z}, \bar{y})$ in the interior of $Q$ and $\hat{\lambda}$ and $\hat{\omega}$, respectively, be their eigenvalues. Assume $\nu_p 1 \leq \lambda \leq \chi_p 1$, $\nu_d 1 \leq \omega \leq \chi_d 1$, where $\nu_p > 0$, $\nu_d > 0$ are some constants. We require the feasibility constraints to be calculated to a certain accuracy. That is, each iterate satisfies

$$Q\hat{P}\omega + A^T\gamma = \bar{c} + \delta$$

$$AQ\hat{P}\lambda = \bar{b} + \delta,$$

with $\|\delta\| \leq \frac{1}{2\sqrt{2}} \nu_p \rho$, $\|\delta\| \leq \frac{1}{2\sqrt{2}} \nu_d$.

Friedland and Jarre in [5] show that by an appropriate transformation, we can make the smallest singular value of a matrix larger than 1, while preserving the structure of the problem. This implies that $\|\bar{b}\|$ and $\|\delta\|$ are not too small.

If $\epsilon_p > \frac{1}{2\sqrt{2}} \rho \nu_p$, we replace $\epsilon_p$ with $\frac{1}{2\sqrt{2}} \rho \nu_p$; if $\epsilon_d > \frac{1}{2\sqrt{2}} \nu_d$, we replace $\epsilon_d$ with $\frac{1}{2\sqrt{2}} \nu_d$. We modify Algorithm 1 so that $(x^k; y^k; z^k)$ is now forced to stay in the neighborhood $\mathcal{N}$ for every $k$:

$$\mathcal{N} \overset{\text{def}}{=} \{(\lambda, \omega, y, Q) : \lambda \in \mathbb{R}^{2n}, \omega \in \mathbb{R}^{2n}, y \in \mathbb{R}^m, Q \in L, \lambda > 0, \omega > 0,$$

$$\lambda_{ij} \omega_{ij} \geq \gamma_p \frac{\lambda^T \omega}{2n} (j = 1, 2; i = 1, \ldots, n),$$

$$\lambda^T \omega \geq \gamma_p \|AQ\hat{P}\lambda - \bar{b}\|,$$

$$\|AQ\hat{P}\lambda - \bar{b}\| \leq \frac{1}{2\sqrt{2}} \nu_p \rho \text{ or } \|AQ\hat{P}\lambda - \bar{b}\| \leq \epsilon_p \} \text{ and}$$

$$\lambda^T \omega \geq \gamma_d \|A^T y + Q\hat{P}\omega - \bar{c}\|,$$

$$\|A^T y + Q\hat{P}\omega - \bar{c}\| \leq \frac{1}{2\sqrt{2}} \nu_d \text{ or } \|A^T y + Q\hat{P}\omega - \bar{c}\| \leq \epsilon_d \} \}$$

The rest of the algorithm remains unchanged. Let us call this version Algorithm 2. Similar to the derivation given in [37] and [39], we find that when the step size $\alpha^k \leq \alpha^{**}$, with

$$\alpha^{**} \overset{\text{def}}{=} \min \left\{ \frac{\nu_p \rho}{\|\lambda\| \left(2\eta^{1/2} + \frac{\sqrt{2}}{2} \eta \gamma + \eta^2 + \frac{1}{2} \eta^{3/2} \gamma + \frac{1}{2} \eta^{5/2} \gamma \right)}, \frac{\nu_d}{\left(2\eta^{1/2} + \frac{\sqrt{2}}{2} \eta \gamma + \eta^2 + \frac{1}{2} \eta^{3/2} \gamma + \frac{1}{2} \eta^{5/2} \gamma \right)} \right\},$$

condition (40) is satisfied. Hence, $\hat{\alpha}$ in Algorithm 2 has a lower bound: $\min\{\alpha^*, \alpha^{**}\}$. Thus, by the results in § A.1 if the initial point is in $\mathcal{N}$, the smallest singular value of the Jacobian of the Newton System is at least $d$, and the sequence $(x^k, y^k, z^k, Q^k)$ is bounded, then the sequence of solutions generated by Algorithm 2 converges to a solution of (1) in finite iterations.

Of the assumptions above, we only need to show the boundedness. We use two lemmas to show this. Lemma A.1 guarantees the existence of a feasible solution in the interior of $Q$; subsequently Lemma A.2 guarantees the boundedness.

Consider the perturbed system:

$$z + A^T y = \bar{c} + \delta$$

$$Ax = \bar{b} + \delta$$

$$x \in Q, \; z \in Q$$

(41)
Lemma A.1 Suppose \((\bar{x}, \bar{z}, \bar{y})\) has a feasible solution with \(\bar{x}, \bar{z} \in \text{Int } Q\). Assume \(\nu_\rho \leq \hat{\lambda} \leq \chi_\rho \), \(\nu_d \leq \hat{\omega} \leq \chi_d \); then for all \(\|\hat{\mathbf{b}}\| \leq \frac{1}{2\sqrt{2}} \nu_p \rho \) and \(\|\hat{\mathbf{c}}\| \leq \frac{1}{2\sqrt{2}} \nu_d \), \((\mathbf{b}, \mathbf{c})\) has a feasible solution \((\hat{\lambda}, \hat{\omega}, \hat{\mathbf{y}}, \hat{\mathbf{Q}})\) with \(\frac{1}{2} \nu_\rho \leq \hat{\lambda} \leq \frac{3}{2} \chi_\rho \), \(\frac{1}{2} \nu_d \leq \hat{\omega} \leq \frac{3}{2} \chi_d \).

Proof: Let \(\mathbf{h} \overset{\text{def}}{=} A^+ \hat{\mathbf{b}}\), where \(A^+\) is the Moore-Penrose generalized inverse of \(A\). We can write the spectral decomposition of \(\mathbf{h}\) as \(\mathbf{h} = Q_h \tilde{\mathbf{P}} \lambda_h\). Then
\[
\|\lambda_h\|_\infty \leq \|\lambda_h\| = \sqrt{2} \|\mathbf{h}\| \leq \sqrt{2} \|A^+\| \|\hat{\mathbf{b}}\| \leq \frac{1}{2} \nu_p.
\]
Let
\[
\tilde{x} \overset{\text{def}}{=} \bar{x} + \mathbf{h}, \quad \tilde{y} \overset{\text{def}}{=} \bar{y}, \quad \tilde{z} \overset{\text{def}}{=} \bar{z} + \mathbf{c}.
\]
Write \((\hat{\lambda})_{\text{small}}\) for the smaller eigenvalue of \(\tilde{x}_i\). Then
\[
(\hat{\lambda})_{\text{small}} = (\bar{x}_i)_0 + (\mathbf{h}_i)_0 - \sqrt{\sum_{j=1}^{n_i-1} ((\bar{x}_i)_j + (\mathbf{h}_i)_j)^2} = (\bar{x}_i)_0 + (\mathbf{h}_i)_0 - \|\tilde{x}_i\| - \|\tilde{h}_i\|
\]
\[
\geq (\hat{\lambda})_{\text{small}} - \|\lambda_h\|_\infty \geq \frac{1}{2} \nu_p.
\]
The first inequality follows from the Cauchy-Schwartz-Boniakovky inequality. Similarly, write \((\hat{\lambda})_{\text{big}}\) for the larger eigenvalue of \(\tilde{x}_i\). Then
\[
(\hat{\lambda})_{\text{big}} = (\bar{x}_i)_0 + (\mathbf{h}_i)_0 + \sqrt{\sum_{j=1}^{n_i-1} ((\bar{x}_i)_j + (\mathbf{h}_i)_j)^2} \leq (\bar{x}_i)_0 + (\mathbf{h}_i)_0 + \|\tilde{x}_i\| + \|\tilde{h}_i\|
\]
\[
\leq (\hat{\lambda})_{\text{big}} + \|\lambda_h\|_\infty \leq \frac{3}{2} \chi_p.
\]
Thus, \(\frac{1}{2} \nu_\rho \leq \hat{\lambda} \leq \frac{3}{2} \chi_\rho \).
The inequalities \(\frac{1}{2} \nu_d \leq \hat{\omega} \leq \frac{3}{2} \chi_d \) can be proved similarly.

Lemma A.2 If \((\bar{x}, \bar{z}, \bar{y})\) has an interior feasible solution with \(\nu_\rho \leq \hat{\lambda} \leq \chi_\rho \), \(\nu_d \leq \hat{\omega} \leq \chi_d \); then there exists a positive scalar \(\Gamma\), such that for any \((\lambda, \omega, y, Q) \in \mathcal{N}\), \(\nu_p \|\lambda\|_1 + \nu_p \|\omega\|_1 \leq \Gamma\).

Proof: Given an \((\lambda, \omega, y, Q) \in \mathcal{N}\), there exists \(\hat{c}\) and \(\hat{b}\) such that \((\lambda, \omega, y, Q)\) is a solution to the following system of equations.
\[
Q \tilde{P} \omega + A^T y = \mathbf{c} + \hat{c}
\]
\[
A Q \tilde{P} \lambda = \mathbf{b} + \hat{b}
\]
Then, by Lemma A.1 there exists \((\hat{\lambda}, \hat{\omega}, \hat{y}, \hat{Q})\) satisfying the above perturbed constraints with \(\frac{1}{2} \nu_\rho \leq \hat{\lambda} \leq \frac{3}{2} \chi_\rho \), \(\frac{1}{2} \nu_d \leq \hat{\omega} \leq \frac{3}{2} \chi_d \). Therefore,
\[
A(x - \bar{x}) = 0, \quad A^T(y - \bar{y}) + z - \bar{z} = 0.
\]
Hence,
\[
(x - \bar{x})^T (z - \bar{z}) = -(x - \bar{x})^T A^T (y - \bar{y}) = 0.
\]
\[25\]
Therefore,

\[(42) \quad x^T z + \bar{\bar{x}}^T \bar{z} = x^T \bar{z} + \bar{\bar{x}}^T \bar{z} \geq \sum_{i=1}^{n} (\bar{\bar{x}}_i)_0 - \|\bar{x}_i\|_1 \|\bar{z}_i\|_1 + \sum_{i=1}^{n} (\bar{\bar{x}}_i)_0 - \|\bar{x}_i\|_1 \|\bar{z}_i\|_1 \]

\[
\geq \sum_{i=1}^{n} (\bar{\bar{x}}_i)_0 - \|\bar{x}_i\|_1 + (\bar{\bar{x}}_i)_0 + \|\bar{z}_i\|_1 \left( (\bar{\bar{z}}_i)_0 - \|\bar{\bar{z}}_i\|_1 \right) = \sum_{i=1}^{n} (\bar{\bar{x}}_i)_0 - \|\bar{z}_i\|_1 + (\bar{\bar{x}}_i)_0 + \|\bar{z}_i\|_1 \left( (\bar{\bar{z}}_i)_0 - \|\bar{\bar{z}}_i\|_1 \right) \geq \frac{1}{4} \|\lambda\|_1^1 \nu_d + \frac{1}{4} \|\omega\|_1^1 \nu_p.
\]

The first inequality follows from \((x_i)_0 \geq 0, (\bar{\bar{z}}_i)_0 \geq 0, (z_i)_0 \geq 0, (\bar{\bar{z}}_i)_0 \geq 0,\) and the Cauchy-Schwartz-Buniakovsky inequality. The second one follows from \((x_i)_0 \geq \|\bar{x}_i\|, (\bar{\bar{x}}_i)_0 \geq \|\bar{\bar{x}}_i\|_1\). Finally, the first one follows from the spectral decomposition and the lower bounds established on \(\lambda\) and \(\bar{\bar{\omega}}\). We also have

\[(43) \quad x^T z + \bar{\bar{x}}^T \bar{z} = \lambda^T \bar{\bar{P}}^T \bar{\bar{P}} \omega + \bar{\bar{x}}^T \bar{z} \leq \frac{1}{2} \lambda^T \omega + \sum_{i=1}^{n} \left( (\bar{\bar{x}}_i)_0 + \|\bar{\bar{z}}_i\|_1 \right) (\bar{\bar{z}}_0 + \|\bar{\bar{z}}_i\|_1) \leq \frac{1}{2} \lambda^0_1^2 \omega^0 + \frac{9}{4} \eta \nu_p \nu_d.
\]

Again the first two inequalities follow from the Cauchy-Schwartz-Buniakovsky inequality, and the second from \((\lambda^0)^T \omega^0 \geq (\lambda^k)^T \omega^k\), which is enforced by Algorithm 2. Combining \((42)\) and \((43)\), we get

\[
\|\lambda\|_1^1 \nu_d + \|\omega\|_1^1 \nu_p \leq 2 (\lambda^0)^T \omega^0 + 9 \eta \nu_p \nu_d.
\]

We have proved that Algorithm 2 will terminate at an \((\epsilon_c, \epsilon_p, \epsilon_d)\) solution to \(\square\) in finite iterations, provided that the smallest singular value of the Jacobian of the Newton’s system doesn’t converge to zero.

### A.3 Convergence analysis of the modified Q method

The analysis of Algorithm 1 can be modified to give a similar convergence result for the modified \(Q\) method. For example, replacing \(\|s_i\|^2 \leq \eta \|\Delta q_i\|^2 \leq \eta\) in the proof of Theorem \(1\) we get:

\[(44) \quad g_p(\alpha) = (\lambda^k + \alpha \Delta \lambda)^T (\omega^k + \alpha \Delta \omega) - \gamma_p \|A x - b\|
\]

\[
= (1 - \alpha) \lambda^k T \omega^k + \alpha \sigma^k T \omega^k + \alpha^2 \Delta \lambda^T \Delta \omega - \gamma_p \left\| \sum_{i=1}^{n} (A_i)_0 \lambda_{i1} + \frac{\lambda_{i2}}{2} q_i - b + \alpha (A_i)_0 \frac{\Delta \lambda_{i1}}{2} + \alpha \Delta \lambda_{i2} + \alpha A_i \frac{\Delta \lambda_{i1} - \Delta \lambda_{i2}}{2} q_i
\]

\[
+ \alpha A_i \frac{\Delta \lambda_{i1} - \Delta \lambda_{i2}}{2} q_i + \frac{\Delta \lambda_{i1} + \Delta \lambda_{i2}}{2} \|q_i + \alpha \Delta q_i\| \right. 
\]

\[
+ \frac{\lambda_{i1} - \lambda_{i2}}{2} A_i \left( \frac{1}{\|q_i + \alpha \Delta q_i\|} - 1 \right) \right\}^2.
\]
Since \( \frac{1}{\sqrt{1+a}} - 1 + \frac{a}{2} \) is increasing for \( a \geq 0 \), we see that
\[
-\frac{a}{2} \leq \frac{1}{\sqrt{1+a}} - 1 \leq 0.
\]

Also notice that since \( q_i^T \Delta q_i = 0 \) we have
\[
1 - \left\| q_i + \alpha \Delta q_i \right\| = \left| \frac{1}{\sqrt{1+\alpha^2 \Delta q_i^T \Delta q_i}} - 1 \right| \leq \frac{\alpha^2}{2} \Delta q_i^T \Delta q_i \leq \frac{\alpha^2 \eta^2}{2}.
\]

Hence,
\[
g_p(\alpha) \geq \alpha \sigma_1 e^* - \alpha^2 \eta - \gamma_p \| A \| \alpha^2 \left( \eta^2 + \left( \frac{\eta}{2} + \eta \right) \frac{\eta}{2} + \frac{1}{2} \Upsilon \eta^2 \right).
\]

In other words, a lower bound on \( \alpha \) for \( g_p(\alpha) \geq 0 \) is
\[
\frac{\sigma_1 e^*}{\eta + \gamma_p \| A \| \left( \eta^2 + \left( \frac{\eta}{2} + \eta \right) \frac{\eta}{2} + \frac{1}{2} \Upsilon \eta^2 \right)}.
\]

Therefore, conclusions of section sections A.1, A.2 carry over to the modified Q method.

References


