The Solution of Euclidean Norm Trust Region
SQP Subproblems via Second Order Cone Programs,
an Overview and Elementary Introduction

Florian Jarre, Felix Lieder,
Mathematisches Institut,
Heinrich-Heine Universität Düsseldorf, Germany.

Abstract

It is well known that convex sequential quadratic programming
(SQP) subproblems with a Euclidean norm trust region constraint
can be reduced to second order cone programs for which the theory
of Euclidean Jordan algebras leads to efficient interior-point algo-
rithms. Here, a brief and self-contained outline of the principles of
such an implementation is given. All identities relevant for the im-
plementation are derived from scratch and are compared to interior-
point methods for linear programs. Sparsity of the data of the SQP
subproblem can be maintained essentially in the same way as for
interior-point methods for linear programs. The presentation is in-
tended as an introduction for students and for colleagues who may
have heard about Jordan algebras but did not yet find the time to
get involved with them. A simple Matlab implementation is made
available and the discussion of implementational aspects addresses
a scaling property that is critical for SQP subproblems.

Key words: Second order cone program, Jordan algebra, SQP subproblem.
1 Introduction

The trust region subproblem for unconstrained minimization typically is stated in terms of minimizing a quadratic function subject to a Euclidean norm trust region constraint, see e.g. [9]. This trust region problem is “reasonably easy” to solve (see e.g. [25]) while minimizing a quadratic function subject to an $\infty$-norm trust region is NP-complete. Moreover, if the trust region constraint is active, then the solutions of trust region subproblems with a polyhedral trust region generally lie at a vertex or a low dimensional facet of the trust region, and there is no compelling reason why one would want to restrict the search directions to such low dimensional facets that depend on the particular choice of the trust region. Under mild conditions, the Euclidean norm trust region subproblem results in globally convergent and rapidly locally convergent algorithms.

The transfer of Euclidean norm trust region subproblems to equality constrained problems has been considered in a series of papers [7, 33, 42, 45, 8, 31, 35, 4].

For minimization subject to equality and inequality constraints the situation often is perceived differently: As examples by Meredith Goldsmith [17] illustrate, a nonconvex SQP trust region subproblem may generate search directions that (locally and globally) neither reduce the objective function nor the underlying measure for the constraint violation. One way to cope with this situation is to define convex sequential quadratic programming (SQP) trust region subproblems in a way that local superlinear convergence can still be anticipated. For such trust region subproblems, often a 1-norm or an $\infty$-norm trust region constraint is considered, e.g. [9, 14, 43]. This choice allows that the trust region problem can be converted to an equivalent convex QP which is also easily solvable. As far as the global convergence analysis is concerned, since all norms are equivalent, the choice of the norm is not crucial. Likewise, fast local convergence is expected only when the trust region constraint is not active in the final iterations, and thus, the choice of the norm is not crucial either. Nevertheless it is somewhat unsatisfactory to prefer the Euclidean norm trust region only in the case of unconstrained (or equality constrained) minimization. Here, the known fact is repeated that also in the presence of inequality constraints, a Euclidean norm trust region subproblem is easy to solve, not only in theory, but also in practice. To support the latter point, a self-contained derivation of an interior-point approach for second order cone programs is presented and its application to a Euclidean norm trust region subproblem.

The papers [33, 43] each present an appealing global convergence analysis for a $p$-norm SQP method. As an alternative to such methods several other SQP approaches that do not rely on $l_p$-norm trust region subproblems have been proposed such as constraint relaxation [39], constraint reduction [30], constraint lumping [7], or the filter-SQP approach [15]. For a detailed discussion we refer, for example, to the monograph [18].

Optimization algorithms over the second order cone (SOC) are well studied, see e.g. [12, 13, 28, 29, 36, 24, 34, 38, 5], and the underlying monograph [11]. A comprehensive survey with numerous further applications of second order cone programs is given in [3], see also [22]. In this paper, based on [19, 10], a brief and self-contained outline of the principles of an implementation is given. The results needed for the design and analysis of an interior-point algorithm are derived from scratch and in analogy to interior-point methods for linear
programs (LPs). As an application, the solution of penalty SQP subproblems with a Euclidean trust region constraint is discussed.

1.1 Notation

The components of a vector \( x \in \mathbb{R}^n \) are denoted by \( x_i \) for \( 1 \leq i \leq n \). The Euclidean norm of a vector \( x \in \mathbb{R}^n \) is denoted by \( \|x\| \) and \( \text{Diag}(x) \) denotes the \( n \times n \) diagonal matrix with diagonal entries \( x_i \) for \( 1 \leq i \leq n \). The identity matrix is denoted by \( I \), its dimension being evident from the context. Inequalities such as \( x \geq 0 \) are understood componentwise. The second order cone is given by

\[
\text{SOC} = \left\{ x \in \mathbb{R}^{1+n} \mid x = \begin{pmatrix} x_0 \\ x_N \end{pmatrix}, x_0 \in \mathbb{R}, x_N \in \mathbb{R}^n, \|x_N\| \leq x_0 \right\}.
\]

Here, \( N := \{1, \ldots, n\} \) spans the indices from 1 to \( n \). The partition of a vector \( x \in \mathbb{R}^{1+n} \) into a first component \( x_0 \in \mathbb{R} \) and a vector \( x_N \in \mathbb{R}^n \) will be used throughout sections 2 and 3 of this paper. The interior of \( \text{SOC} \) is denoted by \( \text{SOC}^o \).

2 Second order cone programs

A second order cone program (SOCP) is the problem of minimizing a linear objective function over the Cartesian product of second order cones and subject to linear equality constraints. In the following, the theory of second order cone programs is contrasted with linear programs and in particular with interior-point methods for solving linear programs.

A crucial feature in the design of long step interior-point methods for linear programs is the fact that the positive orthant \( \mathbb{R}_+^n \) is selfdual, i.e. \( x \in \mathbb{R}_+^n \) if, and only if, \( x^T s \geq 0 \) for all \( s \in \mathbb{R}_+^n \). This implies that the primal and dual conic constraints for linear programs take the same form, namely \( x \geq 0 \) and \( s \geq 0 \) where \( x \) and \( s \) are the primal and dual variables. As a direct consequence of the Cauchy-Schwarz inequality, also \( \text{SOC} \) is selfdual.

For the comparison with linear programs in the next section, problems involving a single second order cone of the form (1) are considered; the generalization to the Cartesian product of several second order cones is addressed in Section 3.3.

2.1 Relation to linear programs

First, using a notation that is suitable also for second order cone programs, a brief summary of some well known principles of interior-point methods for solving linear programs is presented.

Let \( A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n \) be given. For a linear program in standard form,

\[
(LP) \quad \min \left\{ c^T x \mid Ax = b, \ x \geq 0 \right\},
\]

a point \( x \geq 0 \) with \( Ax = b \) is optimal, if, and only if, there exists a dual variable \( s > 0 \) that satisfies \( A^T y + s = c \) for some \( y \), as well as the complementarity condition \( x^T s = 0 \).
In this subsection let “◦” denote the Hadamard-product (the componentwise product) of two vectors. Then, for a vector \( x > 0 \), powers of \( x \) (with respect to the multiplication “◦”) can be defined in an obvious fashion, for example \( x^{1/2} \geq 0 \) with \( x^{1/2} \circ x^{1/2} = x \). Evidently,

\[
R^n_+ := \{ x \in \mathbb{R}^n \mid x \geq 0 \} = \{ v \circ v \mid v \in \mathbb{R}^n \}
\]  

(2)

showing that the positive orthant is the cone of squares with respect to the multiplication “◦”. In the next subsection another product “◦” will be considered leading to \( SOC \) as a cone of squares. Interior point methods for linear programs use the following reformulation of the complementarity condition,

\[
x \geq 0, \quad s \geq 0, \quad x^T s = 0 \iff x \geq 0, \quad s \geq 0, \quad x \circ s = 0.
\]  

(3)

The right hand side converts the scalar complementarity condition to \( n \) bilinear equations. These are perturbed and linearized at each step of primal-dual interior-point algorithms. More precisely, at each iteration of an interior-point algorithm, given some iterate \( x, y, s \) with \( x, s > 0 \), systems of the following form for finding a correction \( \Delta x, \Delta y, \Delta s \) are considered:

\[
\begin{align*}
A \Delta x &= b - Ax \\
A^T \Delta y + \Delta s &= c - A^T y - s \\
s \circ \Delta x + x \circ \Delta s &= \mu e - x \circ s
\end{align*}
\]  

(4)

where, in this subsection, \( e = (1, \ldots, 1)^T \) is the neutral element with respect to the multiplication “◦”. Changing the multiplication “◦” in the next subsection also results in a change of the neutral element \( e \).

Operations such as \( s \circ \Delta x \) are typically expressed by a multiplication with a diagonal matrix, \( s \circ \Delta x \equiv \text{Diag}(s) \Delta x \); the above presentation is used to contrast the situation with the second order cone.

A crucial element in solving (4) is the fact that the last line can easily be solved for \( \Delta x \) (or for \( \Delta s \)):

\[
\Delta x = s^{-1} \circ (s \circ \Delta x) = s^{-1} \circ (\mu e - x \circ s - x \circ \Delta s),
\]  

(5)

where \( s^{-1} \) is the componentwise inverse of \( s \) satisfying \( s^{-1} \circ s = e \). Note that the first equation in (5) exploits the fact that the Hadamard-product is associative. Solving the second line for \( \Delta s \) and inserting both into the first row leads to a linear system of the form

\[
AD^2 A^T \Delta y = rhs
\]  

(6)

where \( D = D^T = \text{Diag}(x^{1/2} \circ s^{-1/2}) \) is a scaling matrix satisfying the equation

\[
Ds = D^{-1} x,
\]  

(7)

see e.g. [40].

### 2.2 Transfer to second order cone programs

We now turn from \( (LP) \) to second order cone programs where the self-dual cone \( R^n_+ := \{ x \in \mathbb{R}^n \mid x \geq 0 \} \) of the linear program is replaced with the second order cone \( SOC \); leading to

\[
\text{(SOCP)} \quad \quad \quad \min \{ c^T x \mid Ax = b, \ x \in SOC \}.
\]
Here, $A \in \mathbb{R}^{(1+n) \times m}$, $b \in \mathbb{R}^{m}$, and $c \in \mathbb{R}^{1+n}$. SOC being self-dual, it turns out that the optimality conditions can be written in the same fashion as for (LP), namely, if both, primal and dual, have a strictly feasible solution, then $x \in \text{SOC}$ with $Ax = b$ is optimal if, and only if, there exist $s \in \text{SOC}$ and $y$ satisfying $A^T y + s = c$, as well as the complementarity condition $x^T s = 0$, see e.g. [27], Theorem 4.2.1.

Interior-point methods for linear programs use the representation (2) of $\mathbb{R}^n_+$ as a cone of squares leading to the scaling matrix $D$ in (7). It turns out that (2) and (7) can be transferred to SOCPs (and to semidefinite programs) when changing the product “◦”. For $x, s \in \mathbb{R}^{1+n}$ partitioned as in (1), let the Jordan-product of $x$ and $s$ be given by

$$x \circ s := \begin{pmatrix} x^T s \\ x_0 s_N + s_0 x_N \end{pmatrix},$$

so that $x \circ s = L(x)s$ for all $x, s$. (For linear programs, the matrix $L(x)$ is replaced with the matrix $\text{Diag}(x)$.) Note that the Jordan-product “◦” is bilinear and commutative ($x \circ s = s \circ x$), but – unlike the Hadamard-product and unlike the standard matrix multiplication – it is not associative (in general, $x \circ (y \circ z) \neq (x \circ y) \circ z$). Let further $\tilde{x}_N := (0, x_N^T)$ and

$$e := (1, 0, \ldots, 0)^T.$$  

Then,

$$L(x) = x_0 I + e\tilde{x}_N^T + \tilde{x}_N e^T$$

from which it is easy to derive that $L(x)$ has the eigenvalues $x_0$ and $x_0 \pm \|x_N\|$. The Jordan-product replaces the Hadamard-product used in interior point methods for linear programs. In fact, it is easy to verify that in analogy to (2)

$$\text{SOC} = \{v \circ v \mid v \in \mathbb{R}^{1+n}\}.$$  

And, as in (3), the complementarity conditions can be rewritten as

$$x \in \text{SOC}, \quad s \in \text{SOC}, \quad x \circ s = 0.$$  

However, due to the lack of associativity of the Jordan-product, the first equation in (5) no longer holds true for the Jordan-product. The main difference in the design and analysis of interior-point methods for SOC problems compared to linear programs is to compensate for this lack of associativity.

The most evident way to compensate might seem to use the relation $s \circ \Delta s \equiv L(s)\Delta s$ and to replace (5) with

$$\Delta x = L(s)^{-1}((L(s)\Delta x) = L(s)^{-1} \circ (x e - x \circ s - x \circ \Delta s).$$

1Note that (SOCP) in the form stated here can be solved directly; the techniques outlined below, however, can be extended in a straightforward way to problems over the Cartesian product of several second order cones. A second order cone of dimension 1 (i.e. where $n = 0$ in (1)) is just a non-negative variable, so that the theory covers mixed linear - second order cone programs, referred to as (MSOCP) below, as well.
It will turn out, however, that there is a more natural way to transfer relation (5) to \( SOCPs \).

Note that analogously to \( e \) in (4), the vector \( e = (1,0,\ldots,0)^T \) used in (9) is the neutral element of the Jordan-product, i.e. \( L(e) = I \) the identity matrix. By definition, \( e^2 = e \), so that \( e \) lies in \( SOC \), the cone of squares. In addition, \( e \) is also a “central element” in \( SOC \), maximizing the Euclidean distance to the boundary of \( SOC \) within the set \( \{ x \in SOC \mid \|x\| \leq 1 \} \). This, and the fact that the Jordan product is “compatible” with the scalar product associated with the Euclidean norm, in the sense that \( x^T(y \circ z) = (x \circ y)^Tz \) for all \( x,y,z \in \mathbb{R}^{1+n} \), is essential for a theoretical analysis (and the practical efficiency) of the algorithm in Section 3.

2.3 Computations with the Jordan-product

It is also straightforward to transfer the notion of the inverse and the square root from the Hadamard-product to the Jordan-product: First, it is easy to verify that

\[
x \circ x = 2x_0 + \frac{\text{trace}(x)}{\det(x)}(x_0^2 - x_N^T x_N)e = 0
\]  
(11)

for all \( x \in \mathbb{R}^{1+n} \). Relation (11) defines the \( \text{trace} \) and \( \text{determinant} \) for the Jordan-product\(^2\), in particular,

\[
\det(x) := x_0^2 - x_N^T x_N
\]

for any \( x \in \mathbb{R}^{1+n} \).

From (9) and the Sherman-Morrison-Woodbury update formula one obtains

\[
L(x)^{-1} = \frac{1}{x_0} \left( I - \frac{1}{\det(x)} \left( x_0 (e \tilde{x}_N^T + \tilde{x}_N e^T) - \tilde{x}_N \tilde{x}_N^T - \|\tilde{x}_N\|^2 ee^T \right) \right)
\]  
(12)

for \( x \) with \( x_0 \neq 0 \) and \( \det(x) \neq 0 \), while the inverse element with respect to the Jordan-product of a vector \( x \) with \( \det(x) \neq 0 \) is defined by the characteristic equation \( x^{-1} \circ x = e \). This characteristic equation implies

\[
x^{-1} = L(x^{-1})e = \frac{1}{\det(x)} \left( \begin{array}{c} x_0 \\ -x_N \end{array} \right).
\]

and if \( x_0 \neq 0 \), also the relation \( x^{-1} = L(x)^{-1}e \) holds true. Note that in general, \( L(x^{-1}) \neq L(x)^{-1} \) and \( \det(x) \neq \det(L(x)) \). (The former is due to the lack of associativity of the Jordan-product.) Not only the inverse, but more generally, any analytic function \( \mathbb{R} \to \mathbb{R} \) can be extended to vectors in \( \mathbb{R}^{1+n} \) based on the \emph{eigenvalue decomposition} of a vector \( x \) with \( x_N \neq 0 \),

\[
x = \left( \begin{array}{c} x_0 \\ x_N \end{array} \right) = (x_0 + \|x_N\|) \underbrace{\left( \frac{1}{2} x_N \right)}_{=: u(1)} + (x_0 - \|x_N\|) \underbrace{\left( \frac{-x_N}{\|x_N\|} \right)}_{=: u(2)}.
\]

\(^2\)Relation (11) is called the characteristic polynomial of \( x \) transferring the notion of characteristic polynomial from the matrix multiplication to the Jordan-product. For the Jordan-product related to \( SOC \) the characteristic polynomial always has degree two.
Here, \( u^{(1)}, u^{(2)} \in SOC \) with \( u^{(1)} + u^{(2)} = e \), \( u^{(1)} \circ u^{(1)} = u^{(1)} \), \( u^{(2)} \circ u^{(2)} = u^{(2)} \), and \( u^{(1)} \circ u^{(2)} = 0 \) are the eigenvectors\(^3\) of \( x \) and \( x_0 = \|x_N\| \) are the associated eigenvalues, the product of which is the determinant, and the sum of which is the trace. Let

\[
\xi := \sqrt{x_0 + \|x_N\|} + \sqrt{x_0 - \|x_N\|} = \sqrt{2x_0 + 2\sqrt{x_0^2 - \|x_N\|^2}}.
\]

The orthonormality of the eigenvectors \( u^{(i)} \) with respect to the Jordan-product, and an inverse of a matrix referring to the inverse of a linear mapping. Let

\[
x_{1/2} := \sqrt{x_0 + \|x_N\|} u^{(1)} + \sqrt{x_0 - \|x_N\|} u^{(2)} = \left( \frac{\xi/2}{x_N/\xi} \right) \in SOC
\]

satisfies \( x_{1/2} \circ x_{1/2} = x \). (The equation \( v \circ v = x \) generally has four solutions; the one in (14) is called root of \( x \).) In the same manner as the square root, any other analytic function \( \mathbb{R} \rightarrow \mathbb{R} \) can be applied to the eigenvalues of \( x \), thus defining an analytic function \( \mathbb{R}^{1+n} \rightarrow \mathbb{R}^{1+n} \).

We stress that operations such as \( x_{1/2} \) depend on the Jordan-product \( \circ \); the square root of the Hadamard-product (for linear programs) is not to be confused with the square root in (14).

### 2.4 Quadratic representation

In the following, the derivative of the map \( f^{inv} : \mathbb{R}^{1+n} \setminus \{x \mid \det(x) = 0\} \rightarrow \mathbb{R}^{1+n} \) with \( x \mapsto x^{-1} \) will play an important role. \( f^{inv} \) being a rational function of the vector entries, it is differentiable on its domain and its derivative satisfies the identity

\[
e + o(||\Delta x||) \equiv (x + t\Delta x) \circ (x^{-1} + Df^{inv}(x)[t\Delta x]) = (x + t\Delta x)^{-1} + o(||\Delta x||)
\]

\[
e + t\Delta x \circ x^{-1} + x \circ Df^{inv}(x)[t\Delta x] + o(||\Delta x||)
\]

for all \( x \) with \( \det(x) \neq 0 \) and all \( \Delta x \) and all \( t \) sufficiently close to zero. Subtracting \( e \) from both sides, dividing by \( t \), and taking the limit as \( t \to 0 \) yields

\[
0 = \Delta x \circ x^{-1} + x \circ Df^{inv}(x)[\Delta x].
\]

This equation – being satisfied for any \( \Delta x \) – defines the linear map \( Df^{inv}(x) \). Due to the lack of associativity this equation cannot be solved for \( Df^{inv}(x)[\Delta x] \) by multiplying from left with \( x^{-1} \). Instead, denote the inverse\(^4\) of \( -Df^{inv}(x) \) by \( P(x) := -(Df^{inv}(x))^{-1} \) and \( y := Df^{inv}(x)[\Delta x] \). Then, above equation can be written as a characteristic equation for \( P \),

\[
(P(x)y) \circ x^{-1} = x \circ y.
\]

\(^3\)For semidefinite programs, the Jordan-product of two symmetric matrices \( X, S \) is \( X \circ S := \frac{1}{4}(XS + SX) \) where \( XS \) is the usual matrix multiplication. If some vectors \( u^{(i)} \) form a basis of orthonormal eigenvectors of \( X \), then the “eigenvectors” of \( X \) are given by \( u^{(i)}(u^{(i)})^T \) satisfying the relation \( X \circ u^{(i)}(u^{(i)})^T = \lambda u^{(i)}(u^{(i)})^T \). This latter relation corresponds to the above definition of eigenvectors associated with the Jordan-product in (8).

\(^4\)Here, the inverse refers to the linear operator \( -Df^{inv}(x) \). In general the form of the inverse will be evident from the context; an inverse of a vector referring to the inverse with respect to the Jordan-product, and an inverse of a matrix referring to the inverse of a linear mapping.
It turns out that
\[
P(x) = 2L(x)^2 - L(x^2) = \det(x)(I - 2e e^T) + 2xx^T \tag{16}
\]
satisfies this equation: Indeed, the second equality in (16) follows from (9), and it leads to
\[
P(x)y = 2x^Tyx + \det(x) \begin{pmatrix} -y_0 \\ y_N \end{pmatrix} \tag{17}
\]
and
\[
(P(x)y) \circ x^{-1} = 2 \frac{x^Ty}{\det(x)} \begin{pmatrix} x_0 \\ x_N \end{pmatrix} \circ \begin{pmatrix} x_0 \\ -x_N \end{pmatrix} + \begin{pmatrix} -y_0 \\ y_N \end{pmatrix} \circ \begin{pmatrix} x_0 \\ -x_N \end{pmatrix} = x \circ y
\]
for any \(x, y\) with \(\det(x) \neq 0\) so that also the characteristic equation (15) holds true.

Here, \(P\) is “some form of square” of \(x\) satisfying the equation \(P(x)x^{-1} = x\) and is called quadratic representation of the Jordan algebra associated with the Jordan-product “\(\circ\)”. A key relation is the property
\[
(P(u)v)^{-1} = P(u^{-1})v^{-1} \tag{18}
\]
for \(u, v\) with \(\det(u) \neq 0 \neq \det(v)\).

To prove (18) observe that the equation (18) is invariant under multiplication of \(u\) or \(v\) with a nonzero constant. Thus, for the proof, one can assume without loss of generality that \(\det(u) = \det(v) = 1\). In this case, the relation \(u^Tv = (u^{-1})^Tv^{-1}\) holds and thus, using (17), one obtains
\[
(P(u)v) \circ (P(u^{-1})v^{-1}) = (2u^Tv_0v - v^{-1}) \circ (2u^Tv_0 - v)
\]
\[= \left( \frac{2u^Tv_0v - v}{2u^Tv_0v + v} \right) \circ \left( \frac{2u^Tv_0 - v}{2u^Tv_0 - v} \right) = e. \tag*{\Box}
\]
Finally, using the Sherman-Morrison rank-one update formula for inverse matrices one can verify that
\[
P(u^{-1}) = P(u)^{-1} \text{ for } \det(u) \neq 0, \text{ and } P(u^{1/2}) = P(u)^{1/2} \text{ for } u \in SOC^o,
\]
where \(SOC^o\) denotes the interior of \(SOC\).

2.5 Scaling point
For \(x, s \in SOC^o\), the interior of \(SOC\), let \(\hat{x} := \frac{x}{(x^0 - x_N)^{1/2}}, \hat{s} := \frac{s}{(s^0 - s_N)^{1/2}}\) so that \(\det(\hat{x}) = \det(\hat{s}) = 1\). Set \(\gamma := \frac{1}{\sqrt{2(1 + x^2)}}\) and \(\hat{w} := \gamma(\hat{x} + \hat{s}^{-1})\).

It follows that also \(\det(\hat{w}) = 1\), and thus, \(\hat{w}^{1/2} = \frac{1}{\sqrt{2(1 + \hat{w}_0)}} \begin{pmatrix} 1 + \hat{w}_0 \\ \hat{w}_N \end{pmatrix}\). Let \(\eta := \left(\frac{\det(x)}{\det(s)}\right)^{1/4}\) and
\[
W := P((\eta \hat{w})^{1/2}) = \eta \begin{pmatrix} \hat{w}_0 \\ \hat{w}_N \end{pmatrix} \begin{pmatrix} \hat{w}_N^T \\ \hat{w}_N \end{pmatrix} I + \frac{\hat{w}_N \hat{w}_N^T}{1 + \hat{w}_0} \tag{20}
\]
\[ = \eta \left\{ \begin{array}{c} -1 \\ I \end{array} \right\} + \frac{1}{1 + \bar{w}_0} \begin{pmatrix} 1 + \bar{w}_0 \\ \bar{w}_N \end{pmatrix} \begin{pmatrix} 1 + \bar{w}_0 \\ \bar{w}_N \end{pmatrix}^T \right\}. \]

Since \( \det(\bar{w}) = 1 \) it follows from (19) that

\[ W^{-1} = P((\eta \bar{w})^{-1/2}) = \frac{1}{\eta} \begin{pmatrix} \bar{w}_0 & -\bar{w}_N \bar{w}_N^T \\ -\bar{w}_N & I + \frac{\bar{w}_N \bar{w}_N^T}{1 + \bar{w}_N} \end{pmatrix} \].

Using the definition of \( W \) one can derive the scaling condition

\[ W s = W^{-1} x \quad \text{or equivalently,} \quad W^2 s = P(\eta \bar{w}) s = x. \quad (21) \]

The first relation in (21) can be verified for the case that \( \det(x) = \det(s) = 1 \) – with somewhat tedious but straightforward calculations – and then, based on the definition of \( \eta \), for the general case. The second relation follows from (19).

The matrix \( W \) in (21) thus satisfies the same relation as \( D \) in (7). For this reason \( \eta \bar{w} \) is called the scaling point of \( x \) and \( s \) (in this order), see e.g. [19].

For general \( x, s \in SOC^o \) since \( W \) is a quadratic representation of the form (16) it follows

\[ u \circ v = \mu e \]

\[ \iff u = \mu v^{-1} \]

\[ \iff W^{-1} u = \mu W^{-1} v^{-1} = \mu (Wv)^{-1} \]

\[ \iff (W^{-1} u) \circ (Wv) = \mu e, \]

where (we did not use associativity and where) the second equation in line 3 uses the key relation (18). Thus, for given \( x, s \in SOC^o \), the condition of finding corrections \( \Delta x, \Delta s \) satisfying the complementarity relation

\[ (x + \Delta x) \circ (s + \Delta s) = \mu e \]

can be expressed in the form

\[ (W^{-1}(x + \Delta x)) \circ (W(s + \Delta s)) = \mu e \]

or

\[ (W^{-1} x) \circ (W s) + (W^{-1} \Delta x) \circ (W s) + (W^{-1} x) \circ (W \Delta s) + (W^{-1} \Delta x) \circ (W \Delta s) = \mu e. \]

Let \( v := W^{-1} x = W s \). Replacing \( v \) into the above equation leads to

\[ (W^{-1} \Delta x) \circ v + v \circ (W \Delta s) = \mu e - v \circ v - (W^{-1} \Delta x) \circ (W \Delta s). \]

Since the (left or right) Jordan-multiplication with \( v \) is expressed by the linear operator \( L(v) \), this is equivalent to

\[ L(v)(W^{-1} \Delta x) + L(v)(W \Delta s) = \mu e - v \circ v - (W^{-1} \Delta x) \circ (W \Delta s). \quad (22) \]

Now, the associativity of the matrix multiplication can be exploited by multiplying from left with \( L(v)^{-1} \). Using the identities \( v^{-1} \circ e = v^{-1} \) and \( v^{-1} \circ (v \circ v) = v \), it follows

\[ W^{-1} \Delta x + W \Delta s = \mu v^{-1} - v - L(v)^{-1} ((W^{-1} \Delta x) \circ (W \Delta s)). \quad (23) \]
The crucial point of the preceding derivations is that this equation can easily be solved for $\Delta x$ or $\Delta s$. Its derivation hinges on the scaling point $\eta \bar{w}$ in (21) which was first discovered in the context of semidefinite programs⁵ defining the NT direction [28, 29].

3 An interior point algorithm

3.1 Search direction

In analogy to linear programs (LP), one iteration of an interior-point method for solving (SOCP) is now derived: At the start let some $x, s \in SOC^\circ$ be given and some $y \in \mathbb{R}^m$. A correction $\Delta x, \Delta y, \Delta s$ is searched for such that

$$A(x + \Delta x) = b$$
$$A^T(y + \Delta y) + (s + \Delta s) = c$$
$$W^{-1}\Delta x + W\Delta s = \mu v^{-1} - v,$$

where the last line is obtained by linearizing (23) i.e. by ignoring the quadratic term $L(v)^{-1}(W^{-1}\Delta x) \circ (W\Delta s)$.

Setting $p := b - Ax$, $q := c - A^T y - s$, and $r := \mu v^{-1} - v = \mu (W s)^{-1} - W s$, the above system is of the form

$$A\Delta x = p$$
$$A^T \Delta y + \Delta s = q$$
$$W^{-1} \Delta x + W\Delta s = r.$$

Solving the second and third equation for $\Delta s$ and $\Delta x$ and substituting both in the first equation one obtains

$$Aw^2 A^T \Delta y = p + Aw(Wq - r) \quad (24)$$
$$\Delta s = q - A^T \Delta y \quad (25)$$
$$\Delta x = Wr - W^2 \Delta s. \quad (26)$$

Note that $W$ is a symmetric rank-1-perturbation of a diagonal matrix (as can be seen from the second line in (20)), and thus, for a dense matrix $A$, the evaluation of the $m \times n$-matrix $AW$ can be carried out with $3mn$ multiplications and additions. The most expensive part of the solution of (24) is forming the matrix $(AW)(AW)^T$ with $m^2n/2$ multiplications and additions. This is identical to the computational effort for solving dense linear programs.

But unlike the case of linear programs, multiplication with $W$ typically destroys any sparsity that may be present in $A$. For sparse problems, observing

⁵ For semidefinite programs, the scaling point of two matrices $X, S \succ 0$ is given by $\eta \bar{W} := W := S^{-1/2}(S^{1/2}X S^{1/2})^{1/2}S^{-1/2}$. Set $P := (W)^{-1/2}$ and replace $v$ in (22) with $V := P^{-1}SP^{-1}$ and $PXP$. Then, for some symmetric matrix $A$, define $L_V(A) := \frac{1}{2}(VA + AV)$. Now, the equivalent of (22) for semidefinite programs is given by the equation

$$L_V(P\Delta XP) + L_V(P^{-1}\Delta SP^{-1}) = \mu I - V^2 - \frac{1}{2}(P\Delta X\Delta SP^{-1} + P^{-1}\Delta S\Delta XP).$$
that $W^2$ is a rank-2-perturbation of the $(1+n) \times (1+n)$-matrix $y^2 I$, a sparse factorization of $AA^T$ and a rank-2 update formula for $W^2$ may be used – or more generally, as for example in [10], a sparse $LDL^T$ factorization of an indefinite KKT system may be used. For the more general (MSOCP) addressed below, such rank-2 update formula is to be applied to a sparse factorization of $AD^2A^T$ for some diagonal matrix $D$; and when the number $p_S$ of second order cones is small such update preserves sparsity in the same fashion as for linear programs.

We may now return to relation (10). Based on (10) the preceding transformations can also be applied, the main difference, however, being that then, the matrix $W^2$ in $AW^2A^T$ in (24) is to be replaced with a nonsymmetric matrix making the numerical computations about twice as expensive. For semidefinite programs, the resulting search direction is called AHO direction, [2]. Apart from the higher computational effort, under standard nondegeneracy assumptions the AHO direction is defined in a neighborhood of the optimal solution (while the NT direction is defined only for positive definite iterates) and in numerical implementations it returned fairly good results in terms of overall number of iterations see e.g. [37], but its theoretical properties in the interior-point context are slightly weaker than those of the NT direction, see e.g. [26, 37].

3.2 Step length

Equations (24) – (26) define some search direction. Given $x \in SOC^\circ$ and a search direction $\Delta x$, it is straightforward to compute the maximum step length $\alpha_x^{\max} > 0$ such that $x + \alpha \Delta x \in SOC^\circ$ for all $\alpha \in [0, \alpha_x^{\max})$:

1. If $\Delta x_0 \geq ||\Delta x_N||$, then, because $SOC$ is a convex cone, $\alpha_x^{\max} = \infty$.

2. Else, let the determinants $\delta_{\Delta x} := \Delta x_0^2 - \Delta x_N^T \Delta x_N$ and $\delta_x := x_0^2 - x_N^T x_N > 0$, and the mixed term $\delta_m := x_0 \Delta x_0 - x_N^T \Delta x_N$ be given. For $\delta_{\Delta x} \neq 0$ it follows from $0 = \det(x + \alpha \Delta x) = \delta_x + 2\alpha \delta_m + \alpha^2 \delta_{\Delta x}$ at the boundary of $SOC$ that

$$
\alpha_x^{\max} = -\frac{1}{\delta_{\Delta x}} \left( \delta_m \pm \sqrt{\delta_m^2 - \delta_{\Delta x} \delta_x} \right)
$$

(27)

where the sign of “±” is to be chosen such that $\alpha_x^{\max} > 0$, and when both are positive, then such that $\alpha_{\max}$ is the smaller of both numbers.

3. For very small $|\delta_{\Delta x}|$ there may be a cancellation error in evaluating (27). In this case consider first the sub-case that $\delta_m = 0$:

If $\delta_{\Delta x} \geq 0$, then $\alpha_x^{\max} = \infty$, and if $\delta_{\Delta x} < 0$, then,

$$
\alpha_x^{\max} = \sqrt{\delta_x/|\delta_{\Delta x}|}
$$

Now assume that $\delta_m \neq 0$:

When $\delta_m > 0$ and $\delta_{\Delta x} < 0$, the “+”-sign in formula (27) is to be taken, and then, there is no cancellation error so that the numerical computation is sufficiently stable.

If $\delta_m > 0$ and $\delta_{\Delta x} \geq 0$ then $\alpha_x^{\max} = \infty$. 
When $\delta_m < 0$, the Taylor approximation of the square root in (27) is used for numerical stability, leading to

$$\alpha_{x,\text{max}} \approx \frac{\delta_x}{2|\delta_m|}$$

where this formula is exact for $\delta_{\Delta x} = 0$.

### 3.3 A general mixed linear second order cone problem

To state the general algorithm, consider now a general (MSOCP)

\[
\text{(MSOCP)} \quad \min \{ c^T x \mid Ax = b, \ x \in K \},
\]

where $K$ is the Cartesian product of a nonnegative orthant and of several second order cones. (We do not consider free variables here.) More precisely, assume that there are $p_N$ nonnegative variables and $p_S$ second order cones of different dimensions, i.e.

$$K = \mathbb{R}^{p_N} \times \text{SOC}^{(1)} \times \ldots \times \text{SOC}^{(p_S)}.$$ Let $p_K := p_N + p_S$ and $n = p_N + \dim(SOC^{(1)}) + \ldots + \dim(SOC^{(p_S)})$, where $\dim(SOC^{(i)})$ denotes the dimension of $SOC^{(i)}$.

The matrix $W$ from above is then replaced with the block diagonal matrix having a $p_N \times p_N$ diagonal block of the form $D$ from (7) for the nonnegative variables and diagonal blocks of the form (20) for each $SOC^{(i)}$.

Mehrotra’s predictor-corrector algorithm [23] applied to (MSOCP) can then be stated as follows:

**MPC-MSOCP-Algorithm:**

(Mehrotra’s Predictor-Corrector Algorithm for solving (MSOCP))

**Input:** $(x^0, y^0, s^0)$ with $x^0, s^0 \in K^\circ$, $\varepsilon > 0$, and $M \gg 0$. Set $k := 0$.

1. Set $(x, y, s) := (x^k, y^k, s^k)$, $\mu_k := (x^k)^T s^k / p_K$.
2. If $\|Ax - b\| < \varepsilon$, $\|ATy + s - c\| < \varepsilon$, and $\mu_k < \varepsilon / p_K$: STOP.
3. If $\|x\| > M$ or $\|s\| > M$: STOP.
4. Solve system (24) – (26) with $\mu = 0$ in the definition of $r$. Denote the solution by $\Delta x^N$, $\Delta y^N$, and $\Delta s^N$.
5. Find the maximum feasible step lengths along $\Delta x^N$ and $\Delta s^N$,

$$\alpha_{x,\text{max}}^N := \min\{1, \min_{i \in \{1, \ldots, p_N\}} \frac{x_i}{\Delta x_i^N} \},$$

$$\alpha_{s,\text{max}}^N := \min\{1, \min_{i \in \{1, \ldots, p_N\}} \frac{s_i}{\Delta s_i^N} \},$$

where the maximum step lengths $\alpha_{x,i}^N$, $\alpha_{s,i}^N$ with respect to $SOC^{(i)}$ are computed as in (27).

6. Set

$$\mu_N^* = (x + \alpha_{x,\text{max}}^N \Delta x^N)^T (s + \alpha_{s,\text{max}}^N \Delta s^N) / p_K$$

and

$$\mu_C = \mu_k \cdot (\mu_N^*/\mu_k)^3.$$
7. Solve system (24) – (26) again (with the factorization of Step 4.) and with
\[ r := \mu v^{-1} - v - L(Ws)^{-1}((W^{-1}\Delta x^N) \circ (W\Delta s^N)). \]
Denote the solution by \( \Delta x^C, \Delta y^C, \) and \( \Delta s^C. \)

8. Choose a damping parameter \( \eta_k \in [0.5, 1.0) \) and set
\[ \alpha_{\text{max},x} := \min\{2, \min_{i \in \{1, \ldots, p_N\}} \frac{x_i}{\Delta x_i^{C}} \min_{i \in \{1, \ldots, p_K\}} \alpha_{x,i}^{\text{max}} \}, \]
\[ \alpha_{\text{max},s} := \min\{2, \min_{i \in \{1, \ldots, p_N\}} \frac{s_i}{\Delta s_i^{C}} \min_{i \in \{1, \ldots, p_K\}} \alpha_{s,i}^{\text{max}} \}, \]
and then set
\[ \alpha_x := \min\{1, \eta_k \alpha_{\text{max},x} \}, \quad \alpha_s := \min\{1, \eta_k \alpha_{\text{max},s} \}. \]

9. Set
\[ x^{k+1} := x^k + \alpha_x \Delta x^C, \quad (y^{k+1}, s^{k+1}) := (y^k, s^k) + \alpha_s (\Delta y^C, \Delta s^C), \]
and \( k := k + 1, \) and return to Step 1.

The input \( \varepsilon > 0 \) is a stopping parameter, and \( M \gg 0 \) an upper bound on the norm of the iterates. A stop in Step 2. returns an approximation to the optimal solution of (MSOCP), and a stop in Step 3. indicates that problem (MSOCP) either has no solution or it is poorly conditioned. The evaluation of \( r \) in Step 7. can be carried out with \( O(n) \) arithmetic operations using the formula (12).

A complexity analysis of several related methods with a feasible starting point is given, for example in [36, 24]. For a method with long steps and with search directions as used in the above algorithm, the computational effort of reducing the parameter \( \mu \) by a factor of \( \varepsilon \in (0, 1] \) is \( O(p_K \log(1/\varepsilon)) \) Newton steps. Assuming exact arithmetic, this estimate is independent of any condition numbers; an error message that the problem may be poorly conditioned (as in Step 3. above) will not occur. When the starting point is not feasible, the complexity estimate may worsen, and for the predictor-corrector method above, the choice of the step lengths may be too aggressive to allow for a polynomial complexity estimate. In particular, a stop in Step 3. of the method may be possible.

4 Considering an SQP subproblem

Let twice differentiable functions \( f : \mathbb{R}^n \to \mathbb{R}, f_E : \mathbb{R}^n \to \mathbb{R}^{m_E}, \) and \( f_I : \mathbb{R}^n \to \mathbb{R}^{m_I} \) be given and assume a problem of the form
\[ \min \{ f(x) \mid f_E(x) = 0, \quad f_I(x) \leq 0 \} \quad (28) \]
is to be solved. Assume further that a current iterate \( \bar{x} \) is given along with some approximate Hessian \( H \) of the Lagrangian. Let \( g \) be the gradient of \( f \) at \( \bar{x} \), and \( Df_E \) be the Jacobian of the equality constraints in (28) evaluated at \( \bar{x} \) and \( f_E \) be the function value of the equality constraints evaluated at \( \bar{x} \). (It shall always be evident from the context whether \( f_E \) refers to the vector of function values at \( \bar{x} \) or to the function: \( \mathbb{R}^n \to \mathbb{R}^{m_E} \)). Let \( f_I \) and \( Df_I \) be given
likewise. The Celis-Dennis-Tapia Euclidean norm trust region SQP subproblem [7] termed “CDT subproblem” then takes the form:

$$\begin{align*}
\min & \quad g^T \Delta x + \frac{1}{2} \Delta x^T H \Delta x \\
& \quad Df_E \Delta x - s_E = -f_E, \\
& \quad Df_I \Delta x - s_I \leq -f_I, \\
& \quad \|\Delta x\| \leq \delta_C, \\
& \quad \|\Delta x\| \leq \delta.
\end{align*}$$

(29)

$$\begin{align*}
\min & \quad t \\
& \quad Df_E \Delta x - s_E = -f_E, \\
& \quad Df_I \Delta x - s_I \leq -f_I, \\
& \quad \|s_E\| \leq t, \\
& \quad \|s_I\| \leq 0.9 \delta.
\end{align*}$$

(30)

where $\delta_C$ is the optimal value of

$$\begin{align*}
\min & \quad t \\
& \quad Df_E \Delta x - s_E = -f_E, \\
& \quad Df_I \Delta x - s_I \leq -f_I, \\
& \quad \|s_E\| \leq \delta_C, \\
& \quad \|s_I\| \leq \delta.
\end{align*}$$

(31)

The constant “0.9” in the last line of (30) can be replaced by some parameter $\rho \in [\rho_1, \rho_2]$ where $0 < \rho_1 \leq \rho_2 < 1$ are some fixed values, see e.g. [33]. For simplicity we state a fixed parameter “0.9” in (30).

Problems of the form (29), (30) have been proposed and analyzed for equality constrained problems in [7, 33]. The numerical computation of the solution of the SQP subproblem (for equality constrained problems) is discussed in [42, 45]. The papers [8, 31, 41] analyze optimality conditions for the nonconvex equality constrained case. In [1, 35] semidefinite relaxations (including the inequality constrained case) are considered. A recent survey on this topic is given in [44].

The present paper does not consider (computationally expensive) semidefinite relaxations as in [1, 35] but does consider the inequality constrained case as well for which MSOCPs offer a rather convenient tool if convexity can be enforced.

We note that the nonconvex case including inequality constraints is considerably more difficult than the equality constrained case. In fact the nonconvex SQP subproblem (29) in the inequality constrained case is NP-hard:

**Lemma 1**

Problem (29) is NP-hard.

**Proof:**

The proof is straightforward. First recall that the Max-Cut-Problem is NP-complete, see e.g. [16]. The Max-Cut problem can be written as a $\{-1, 1\}^n$-quadratic program see e.g. [32]. While the optimal value and the multipliers change, the (local or global) optimal solutions remain invariant when a multiple of the identity is subtracted from the Hessian of the quadratic objective function. After subtracting a suitable multiple (determined by Gershgorin’s theorem), the objective function is strictly concave, so that any local solution of the (continuous) $[-1, 1]^n$-quadratic program lies at a vertex and thus coincides with the solution of the $\{-1, 1\}^n$-quadratic program. The SQP subproblem (29) (with the exact Hessian of the Lagrangian about the point $\tilde{x} = 0$, with a sufficiently
large trust region radius \( \delta \geq \sqrt{n} \), and with \( \delta_{C} = 0 \) for such continuous \([-1, 1]^{n}\)-quadratic program is the \([-1, 1]^{n}\)-quadratic program itself, proving the claim of the lemma.

Lemma 1 underlines a contrast to equality constrained problems for which polynomial solvability of the CDT subproblem is shown in [4] also in the non-convex case. Lemma 1 – and the fact that nonconvex SQP subproblems may generate search directions that neither improve feasibility nor optimality (see e.g. [17]) – provides a practical justification for the following further assumption,

\[ H > 0 \quad \text{(i.e. } H \text{ is positive definite)}, \]

which allows for a polynomial time solution of the SQP subproblems (29)\(^6\).

In particular, for problems where the evaluation of the function values and derivative information is computationally expensive, second order cone programs offer an elegant way for solving the subproblems (29) at a predictable computational cost.

Second order cone programs also allow the computation of a second order correction step and a re-evaluation of the Lagrange multipliers – the multipliers obtained from (29) do not approximate the Lagrange multipliers of (28) when the trust region constraint \( \Delta x \leq \delta \) is active.

In the following we address two alternatives of converting problem (29) to a MSOCP. The first approach introduces two additional scalar variables and implicitly makes use of a “rotated second order cone” (as introduced e.g. in [34]) while the second approach only introduces one additional scalar variable and may seem more evident and natural at a first glance. In exact arithmetic both alternatives generate the same search directions.

**First approach**

As observed for example, in [6], Section 9.4, minimizing \( \bar{t} - 1 + g^{T} \Delta x \) subject to the constraints

\[
\begin{pmatrix}
\bar{t} \\
\sqrt{2}H^{1/2}\Delta x
\end{pmatrix} \in SOC
\]

is equivalent to minimizing \( g^{T} \Delta x + \frac{1}{2} \Delta x^{T}H \Delta x \). Indeed, the above SOC constraint implies

\[ \bar{t}^{2} \geq (\bar{t} - 2)^{2} + 2\|H^{1/2}\Delta x\|^{2}, \quad \bar{t} \geq 0, \]

which is equivalent to

\[ 4\bar{t} - 4 \geq 2\Delta x^{T}H \Delta x \]

\(^6\)Enforcing this assumption may result in some loss in the quality of the solution of the SQP subproblem. In the case of a dense matrix \( H \) it can be implemented in a straightforward fashion by first identifying inequality constraints which are anticipated as strongly active, i.e. which have a positive estimate of the Lagrange multiplier at \( \hat{x} \). If \( H \) is not positive semi-definite, then replace \( H \) with its projection onto the null space of the gradients of the equality constraints and of the strongly active inequality constraints. Thereafter, regularize by replacing \( H \) with \( H := H + \theta I \) (for some not too large \( \theta \)) such that the result is positive definite.

If the active constraints are estimated correctly, the projection of \( H \) onto the null space does not have any effect on the location of the optimal solution. If the trust region constraint is active at the optimal solution of the regularized problem (which is more likely, when \( \theta \) is small), then adding \( \theta I \) does not change the optimal solution either.
or to
\[ \bar{\ell} - 1 \geq \frac{1}{2} \Delta x^T H \Delta x. \]

For completeness, the reformulation of (29) to a MSOCP based on this approach is detailed next. For this reformulation denote the symmetric square root of \( H \succ 0 \) by \( H^{1/2} \), identify \( \bar{\ell} \equiv t_2 \) and \( \bar{\ell} - 2 \equiv t_3 \). Adding nonnegative slack variables \( \hat{s}_I \) to the second constraint in (29), and adding two further scalar fixed variables \( t_0, t_1 \) converting the Euclidean norm terms of (29) into SOC-constraints then leads to the equivalent problem

\[
\begin{align*}
\text{minimize} & \quad g^T \Delta x + \frac{1}{2} (t_2 + t_3) \\
\text{s.t.} & \quad Df_E \Delta x - s_E = -f_E, \\
& \quad Df_I \Delta x - s_I + \hat{s}_I = -f_I, \\
& \quad t_0 = \delta, \\
& \quad t_1 = \bar{\delta}_C, \\
& \quad t_2 - t_3 = 2, \\
& \quad \sqrt{2} H^{1/2} \Delta x - x^{(1)} = 0, \\
& \quad \hat{s}_I \geq 0, \quad \begin{pmatrix} t_0 \\ \Delta x \end{pmatrix} \in \text{SOC}, \quad \begin{pmatrix} t_1 \\ s_E \\ s_I \end{pmatrix} \in \text{SOC}, \quad \begin{pmatrix} t_2 \\ t_3 \\ x^{(1)} \end{pmatrix} \in \text{SOC}. 
\end{align*}
\]

(31)

where “equivalent” is understood in the sense that the \( \Delta x \)-part of the optimal solution also solves (29), but the objective values differ, in general.

Note that all variables of (31) are either nonnegative or in a SOC, and thus, (31) has exactly the format \((MSOCP)\). Moreover, as only three second order cones are involved, sparsity can be exploited in the same fashion as for interior-point methods for linear programs if \( H^{1/2}, Df_E, Df_I, \) and \( A \) are sparse. (Of course \( H^{1/2} \) can also be replaced with some other factor \( R, R^T R = H \) that possibly better preserves sparsity.)

While it is possible to retrieve not only the optimal solution of (29) from (31) but also the Lagrange multipliers (using suitable rescaling), the Lagrange multipliers only have limited relevance for problem (28); they also depend on the first two SOC constraints of (31) (namely \( \begin{pmatrix} t_0 \\ \Delta x \end{pmatrix} \in \text{SOC}, \begin{pmatrix} t_1 \\ s_E \\ s_I \end{pmatrix} \in \text{SOC} \)) that have no correspondence in (28). More appropriate estimates for the Lagrange multipliers of (28) can be generated from the solution of an associated MSOCP.

**Second approach**

As an alternative to the first approach above (see e.g. [22], equation (5)), the objective function term \( g^T \Delta x + \frac{1}{2} \Delta x^T H \Delta x \) of (29) can be written in the form

\[
(H^{-1/2}g)^T H^{1/2} \Delta x + \frac{1}{2} (H^{1/2} \Delta x)^T (H^{1/2} \Delta x) = \frac{1}{2} \| \bar{g} + H^{1/2} \Delta x \|^2 - \frac{1}{2} \| \bar{g} \|^2
\]

where \( \bar{g} = H^{-1/2}g \). Since the constant term “\( \frac{1}{2} \| \bar{g} \|^2 \)” does not influence the optimal solution \( \Delta x \), the objective function term can be replaced with \( \| \bar{g} + H^{1/2} \Delta x \|^2 \).
or with an artificial variable \( t \in \mathbb{R} \) and the additional constraints

\[
\begin{pmatrix} t \\ x^{(1)} \end{pmatrix} \in \text{SOC},
\]

\( H^{1/2} \Delta x - x^{(1)} = -\bar{g}. \)  \( \tag{32} \)

Again, the objective value changes with this transformation, but the optimal solution \( \Delta x \) does not.

For the second approach, the variables \( t_2, t_3 \) of (31) are replaced with a variable \( t \), the fifth equation \( “t_2 - t_3 = 2” \) of (31) is deleted and the sixth equation of (31) is replaced with (33). The objective function is the variable \( t \), and the last second order cone constraint in (31) is replaced with (32).

The second approach having one less variable may appear more natural at first glance, but as shown below, it is considerably more sensitive to ill-conditioning and rounding errors. We therefore do not present a complete statement of the SQP subproblem for the second approach.

As noted by Yin Zhang [45] with regards to equality constrained problems, “As far as efficiency is concerned, we believe that methods of any kind ... for solving the n-dimensional Euclidean norm CDT problem are not likely to be cheap ... On the other hand, solving the CDT problem supposedly produces better iterative steps and therefore enhances robustness and global convergence. ...” As pointed out in Lemma 1 above, the problem involving inequality constraints is somewhat more involved than the situation referred to in [45], so that solving the convexified inequality constrained problem via an interior point method for MSCOPs at a moderate computational may be suitable for problems with expensive evaluations of \( g, Df_E, \) or \( Df_I \).

5 Implementational aspects

For small scale dense problems (number of variables plus number of constraints up to 1000), a stand-alone implementation of the MPC-MSOCP-algorithm of the present paper is made available for general use at the site [20]. Given the data to problem (29) an (MSOCP) of the form (31) is formed and its primal-dual solution is approximated with an interior-point method as in Section 3.3. In the end, the dual variables of (31) are rescaled and an approximate primal-dual solution of problem (29) is generated.

5.1 Standard enhancements

Interior-point methods are known to be sensitive to ill-conditioning and to scaling of the data. To reduce this sensitivity, the algorithm includes some standard technical details not addressed in the conceptual presentation in Section 3.3, and some of which are suitable only for dense problems:

Prior to solving problem (MSOCP), linearly dependent rows are eliminated, a Cholesky factor \( L \) with \( LL^T = AA^T \) is computed and \( [A,b] \) is replaced with \( L^{-1}[A,b] \). (After this transformation \( A \) has orthonormal rows.) Then, \( c \) is replaced with \( c - A^TAc \), and \( b \) and (new) \( c \) are scaled to Euclidean norm 1 effecting that the origin has Euclidean distance 1 from the set \( \{ x \mid Ax = b \} \), and from the set \( \{ (y,s) \mid A^Ty + s = c \} \). (The case \( b = 0 \) cannot occur for the
trust region subproblem since the trust region radius \( \delta \) is positive, and when \( c = 0 \) the current iterate either is a KKT point, or one may replace \( c \) with \( \epsilon \) to generate a direction that reduces infeasibility.) Starting points are \( x = s = \epsilon \) and \( y = 0 \). A phase 1 of several centering steps aims at reducing the violation of the primal-dual equality constraints while keeping \( \mu \equiv 1 \). One step of iterative refinement for the solution of the linear systems is included when the back solve returns low accuracy results, and a safeguard is added in case that the Mehrotra predictor-corrector strategy does not result in a sufficient decrease of \( \mu \) or in case that \( \mu \ll \max\{\|Ax - b\|, \|A^T y + s - c\|\} \).

A “geometric stopping test” is used requiring the Euclidean distance to the primal-dual equality constraints to be at most \( \epsilon \), i.e., \( \|Ax - b\| \leq \epsilon \) and \( \|A^T y + s - c\| \leq \epsilon \), relying on the above rescaling. The geometric stopping test further requires the cosine of \( x \) and \( s \) to be less than \( \epsilon \),

\[
x^T s \leq \epsilon \|x\| \|s\|.
\]

This stopping test is motivated by the Euclidean scaling of the starting point and the desire of bounding the condition number of the linear systems to be solved at each iteration, see e.g. [21]. It implies, however, that in case of poorly scaled problems (and for problems that do not have a finite optimal solution) the cosine may be small before the duality gap is below a desired threshold.

In spite of the standard safeguards as outlined above, for sparse problems – and also for poorly conditioned problems – much more sophisticated and efficient implementations of SOCP-solvers such as SDPT3 [38] or ECOS [10] should be used; the implementation in [20] is intended as a simple stand-alone alternative for dense SQP subproblems.

5.2 Reducing rounding errors

Interior-point algorithms typically stop when the residuals and the complementarity gap are below a given threshold. Here, the norm of the residuals is compared to the norm of the right hand side. (In Section 5.1, for example, the right hand side is first scaled to norm 1.) If certain components of the right hand side are large compared to others, then the relative accuracy with respect to the small components may be low. For a composite right hand side as in (31) this might imply that the solution found is not very accurate. We list three cases, where this effect may reduce the accuracy of an approximate solution of (31) obtained from an interior point solver and discuss how to reduce this effect.

1. Recall that \( \bar{g} = H^{-1/2}g \) enters the right hand side of the equality constraints of the reformulation of (31) via (33) in the “second approach”.

The fact that the second approach (32), (33) is feasible for \( H > 0 \) but not for \( H \succeq 0 \) hints at possible cancellation errors when \( H \) has small positive eigenvalues. Such cancellation errors may indeed render an approximate solution of the transformed MSOCP associated with (29) useless: Note that the norm of \( g \) typically does not tend to zero in the final iterations of an SQP algorithm. When \( H \) has tiny eigenvalues the norm of \( \bar{g} = H^{-1/2}g \) typically is large. An algorithm that aims at finding an approximate solution with a small relative error in the right hand side may generate a solution with a larger error in the remaining components of the right hand
side of (31). In the next section a simple example is given with a condition number of $10^5$ for the KKT conditions, with a strictly feasible iterate at a distance of 1 to the optimal solution, and a matrix $H > 0$ with condition number $10^6$, and where the solution of (29) via a MSOCP with accuracy $10^{-10}$ (solved with both, SeDuMi [34], or with the solver in [20]) returns an ascent direction for the objective function (due to rounding errors). It is not known how to rescale the resulting MSOC to eliminate such severe rounding errors. The first approach uses one more scalar variable but does not suffer from any systematic ill-conditioning and is also applicable for the slightly weaker assumption $H \succeq 0$ (rather than $H > 0$).

2. Optimization problems sometimes come with large bounds such as "$x_i \leq 10^{20}$" for some $i \in \{1, \ldots, n\}$. Consider an SQP subproblem of the form (31) about a point $\hat{x}$ with moderate norm $\|\hat{x}\|$. When such bound is treated as an equality constraint with a slack variable $\hat{s}_i$ as in the second row of equations in (31) then such bound yields an entry of size about "$10^{20}$" in the right hand side $-f_i$ of the equality constraints in (31). As just discussed above, very large components in the right hand side of (31) typically reduce the overall accuracy of the approximate solution obtained from an interior-point solver. However, such loss of accuracy can easily be avoided by eliminating all inequalities "i" of the SQP-subproblem for which $-f_i(\hat{x}) > \delta \|\nabla f_i(\hat{x})\|$ before solving the SQP subproblem. (By the Cauchy Schwarz inequality such constraints will never be active.)

3. Similarly, as detailed with an example in the next section, when the trust region radius $\delta$ in the right hand side of the equality constraints of (31) is very large but the remaining parts of the right hand side are small, the overall accuracy of an approximate solution of (31) may degrade as well. Such situation might be possible in a trust region algorithm if the trust region constraint was not active for several successive successful steps. In such situation it would be practical to reduce $\delta$ to a value at most 10 times the norm of the last SQP step. Such reduction would balance the right hand side of (31) but is unlikely to have much effect on the overall number of iterations or on the local or global convergence analysis.

5.3 Numerical examples

To illustrate the sensitivity with respect to scaling and to give some intuition about the computational effort for solving CDT SQP subproblems, some numerical results are listed in this section.

The MCP-MSOCP-algorithm of this paper is similar to the approach used in SPDT3 [38] while SEDUMI [34] uses a different self-dual approach. The comparison of the MCP-MSOCP-algorithm with SEDUMI below suggests that the scaling issues are not due to the particular approach but rather to the conditioning of the problem. This impression is also supported by the comparison below replacing the Euclidean norm trust region problem with an $\infty$-norm trust region problem. For the latter, the variable $\Delta x$ is replaced with $\Delta \hat{x} := \Delta x - \delta e \geq 0$ and an additional slack variable $x^{(2)} \geq 0$ with $\Delta \hat{x} + x^{(2)} = 2\delta e$ is introduced, and the objective function is transformed to an SOC-constraint and a linear objective function as in (31).
In Table 1 below, the two approaches considered in Section 4 for replacing the quadratic objective function with a linear objective function and a second order cone constraint are compared. It turns out that the difference of the two approaches can be illustrated with a very simple instance of a quadratic program with two unknowns. More precisely, let \( \rho > 0 \) be a fixed number and consider

\[
\min \frac{\rho}{2}(x_1^2 + x_2^2) + 1000x_1 + 0.1x_2 \quad \text{where} \quad x \geq 0.
\] (34)

As initial point let \( x^{(0)} := (10^{-4}, 1)^T \).

Since the starting point is feasible, it follows that \( \delta_C = 0 \) in (30). Let \( \delta \) be given as some value \( \delta \geq 1 \). Then, for \( 0 \leq \rho \leq 0.1 \), the SQP correction in exact arithmetic is

\[
\Delta x^* = -x^{(0)}.
\]

In Table 1 below we report the results of solving the SQP subproblem with the first approach, with the second approach (both using the default settings of the Mehrotra predictor corrector method described in Section 3.3), and with the second approach using SeDuMi [34] with its default settings. The three values \( \rho = 10^{-2}, \rho = 10^{-4}, \) and \( \rho = 10^{-6} \) are compared for the above problem. Note that \( H^{-1/2} \) used in the right hand side of (33) is given by \( H^{-1/2} = \rho^{-1/2}I \). (Also recall that for all three values of \( \rho \) the optimal solution of the SQP subproblem remains the same.)

For each case, the norm \( \|\Delta x - \Delta x^*\| \) is listed, where \( \Delta x \) refers to the approximate solutions generated by the three algorithms.

<table>
<thead>
<tr>
<th>approach</th>
<th>( \rho = 10^{-2} )</th>
<th>( \rho = 10^{-4} )</th>
<th>( \rho = 10^{-6} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>first approach</td>
<td>6.7e-8</td>
<td>6.1e-9</td>
<td>1.3e-8</td>
</tr>
<tr>
<td>second approach</td>
<td>2.3e-3</td>
<td>9.6e-3</td>
<td>5.2</td>
</tr>
<tr>
<td>second appr., SeDuMi</td>
<td>4.5 e-3</td>
<td>4.3</td>
<td>4.3</td>
</tr>
</tbody>
</table>

Table 1, norm of the error \( \Delta x - \Delta x^* \) for (34).

The numbers in Table 1 make clear that the second approach is not reliable, independent of whether the MSOCOCP is solved via SeDuMi or with an implementation as in Section 3.3. A closer look at the results of the second approach for \( \rho = 10^{-6} \) shows that the MSOCOCP to be solved has a right hand side of norm 1.0e6. The primal residual of the solution generated by SeDuMi has a norm of 1.7e-4 (almost 10 digits relative accuracy), the primal solution returned by SeDuMi has norm 1.4e6, the \( \Delta x \)-part of this solution has norm 3.3e6. Even for this simple problem, looking at the solution generated by SeDuMi, it is not clear, how to rescale the problem to reduce rounding errors; except from using the first approach.

Table 2 refers to 100 randomly generated problems with 200 variables each, 100 equality constraints and 200 inequality constraints. The objective term \( g \) and 100 entries of \( f_I \) were generated uniformly distributed in \([0, 1]\), 50 entries of \( f_I \) were generated uniformly distributed in \([-1, 0]\) and the remaining entries of \( f_I \) were set to zero. All the remaining data for Table 2 was generated standard normally distributed. The matrix \( H \) generated this way was multiplied by its transpose to make it positive definite. To test the effect of an overly large trust region radius \( \delta \), an initial value \( \delta := 10^4 \) was chosen which happened to be
inactive for all 100 test problems. Then, a second run was performed for which
\( \delta \) was reduced to twice the norm of the solution \( \Delta x \) generated in the first run (to
make sure the trust region constraint is still inactive). The average reduction
of \( \delta \) was by a factor 1.4e-3. Table 2 lists the smallest and the largest norm of
the violation of the KKT conditions before and after reducing \( \delta \), as well as the
norm of the constraint violation before and after reducing \( \delta \).

\[
\begin{array}{ccc}
\text{result} \setminus \text{approach} & \text{MCP-MSOCP} & \text{SEDUMI} \\
\|KKT\text{viol.}\|, \text{before} & 1.2e-3 & 2.0e-3 \\
\|KKT\text{viol.}\|, \text{after} & 2.1e-5 & 1.2e-4 \\
\|\text{feas.viol.}\|, \text{before} & 1.0e-2 & 1.4e-4 \\
\|\text{feas.viol.}\|, \text{after} & 4.4e-7 & 7.3e-5 \\
\end{array}
\]

Table 2, average KKT / constraint violation before and after reducing \( \delta \).

The examples generated for Table 2 resulted in a solution \( \Delta x \) for which the
norm of \( H\Delta x \) typically was much larger than the norm of \( g \), accounting for a
cancellation error in the evaluation of the KKT conditions. The crucial part for
these problems, namely the violation of feasibility, is reduced effectively when
\( \delta \) is small.

To obtain some intuition about the running times for solving Euclidean
norm subproblems compared to \( \infty \)-norm subproblems, feasible random sparse
subproblems were generated and tested with SEDUMI. For feasible problems
\( \delta = 0 \) and the second SOC constraint in (31) (involving the variable \( t_1 \)) is
eliminated prior to solving the MSOCP. The problems included \( n \) variables,
\( n/2 \) equality constraints and \( n \) inequality constraints. The average running
times over 10 random problems are listed in Table 4:

\[
\begin{array}{ccc}
\text{approach} \setminus n & 20 & 200 & 2000 \\
\text{Euclidean Norm} & 0.13 & 0.24 & 171 \\
\text{\( \infty \)-Norm} & 0.10 & 0.29 & 226 \\
\end{array}
\]

Table 3, solution times in seconds

For the \( l_2 \) problem the dimension \( n = 20000 \) was tested as well. This dimen-
sion resulted in an MSOCP with 60000 variables and 50000 equations. (For the
\( l_\infty \) problem there are 70000 variables and 60000 equations.) The matrix \( A \) of the
MSOCP had 0.022 percent nonzero entries, the factor \( ADA^T \) had 0.79 percent
nonzero entries and its Cholesky factor had 18 percent nonzero entries (i.e. over
400 Million nonzeros) after eliminating 3 dense columns from \( A \). The numbers
indicate that the approach is not readily transferrable to large scale problems.
While large scale problem problems would need to be solved by different ap-
proaches the results do seem to indicate that there is no loss in efficiency when
solving Euclidean norm trust region subproblems with interior point approaches
rather than infinity norm trust region subproblems. The same would be true
for the comparison of Euclidean norm trust region subproblems with one-norm
trust region subproblems (having one additional equation compared to infinity
norm trust region subproblems).
6 Summary

While the theory of interior-point methods for linear and semidefinite programs is well established and implementations are used for many practical applications, there still seems to be a lack regarding the general acceptance of second order cone programs. To lessen the reservations toward SOCPs, in chapter 2 and 3, a self-contained presentation of practical aspects of an interior point method for solving SOCPs is given. One important application where SOCP can be useful are trust region SQP subproblems. This application is addressed in detail and some scaling/conditioning parameters that influence the numerical accuracy of an approximate solution are analyzed. Preliminary numerical examples illustrate the effects of the scaling parameters. As a future work the approach is to be used in an SQP method using finite differences for the approximations of the derivatives.

Acknowledgment

The author likes to thank two anonymous referees and the editors who helped to extend and to improve this paper.

References


