# The CVXOPT linear and quadratic cone program solvers

L. Vandenberghe

March 20, 2010

**Abstract**

This document describes the algorithms used in the `conelp` and `coneqp` solvers of CVXOPT version 1.1.2 and some details of their implementation.

## Contents

## 1 Introduction

Two problems are considered in these notes. The first is the *cone linear program* (cone LP)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Gx + s = h \\ & Ax = b \\ & s \succeq 0 \end{array} \tag{1a}$$

with variables $x$ and $s$, and its dual

$$\begin{array}{ll} \text{maximize} & -h^T z - b^T y \\ \text{subject to} & G^T z + A^T y + c = 0 \\ & z \succeq 0 \end{array} \tag{1b}$$

with variables $y$ and $z$. The inequalities $s \succeq 0$, $z \succeq 0$ are generalized inequalities with respect to a self-dual convex cone $C$. We restrict $C$ to be a Cartesian product

$$C = C_1 \times C_2 \times \cdots \times C_K, \tag{2}$$

where each cone $C_k$ can be a nonnegative orthant, second-order cone, or positive semidefinite cone. The second problem is the *cone quadratic program* (cone QP)

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + c^T x \\ \text{subject to} & Gx + s = h \\ & Ax = b \\ & s \succeq 0, \end{array} \tag{3a}$$

with $P$ positive semidefinite. The corresponding dual problem is

$$\begin{array}{ll} \text{maximize} & -(1/2)(G^T z + A^T y + c)^T P^\dagger (G^T z + A^T y + c) - h^T z - b^T y \\ \text{subject to} & G^T z + A^T y + c \in \text{range}(P) \\ & z \succeq 0, \end{array} \tag{3b}$$

where $P^\dagger$ is the pseudo-inverse of $P$. The dual problem can be written more simply by introducing an additional variable $w$:

$$\begin{array}{ll} \text{maximize} & -(1/2)w^T P w - h^T z - b^T y \\ \text{subject to} & G^T z + A^T y + c = Pw \\ & z \succeq 0. \end{array}$$

Although the cone LP can be solved as a special case of the cone QP, the absence of a quadratic term in the cost function makes it easier to implement methods that detect primal or dual infeasibility. This explains the difference between the `conelp` and `coneqp` solvers. The `conelp` solver does not require strict primal and dual feasibility and attempts to solve the problem (1) or establish primal or dual infeasibility. The `coneqp` solver can handle a quadratic term in the objective, but requires that the problem is strictly primal and dual feasible, and does not detect infeasibility.

The algorithms implemented in the two solvers are primal-dual path-following methods based on the Nesterov-Todd scaling. For more background, history, and analysis of the algorithms we refer the reader to the articles in the References (in particular, [ART03, Stu02, Stu03, TTT03]).

**Rank assumptions** We assume that

$$\mathbf{rank}(A) = p, \qquad \mathbf{rank}\left(\left[\begin{array}{ccc} P & A^T & G^T \end{array}\right]\right) = n$$

where $p$ is the row dimension of $A$ and $n$ is the dimension of $x$. This is equivalent to assuming that the matrix

$$\left[\begin{array}{ccc} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & -Q \end{array}\right]$$

is nonsingular for any positive definite $Q$.

If $\mathbf{rank}(A) < p$, then either the equality constraints in the primal problem are inconsistent (if $b \notin \mathrm{range}(A)$) or some of the equalities are redundant and can be removed. If $\mathbf{rank}(\left[\begin{array}{ccc} P & A^T & G^T \end{array}\right]) < n$, then either the equality constraints $G^T z + A^T y + c = Pw$ in the dual problem are inconsistent (if $c \notin \mathrm{range}(\left[\begin{array}{ccc} P & A^T & G^T \end{array}\right])$) or some are redundant and the number of primal variables can be reduced.

The CVXOPT solvers raise an exception if the rank conditions are not satisfied. They do not report which of the two rank assumptions does not hold and they do not detect whether this makes the primal or dual equalities inconsistent or not.

**Notation** We will often represent symmetric matrices as vectors that contain the lower triangular entries of the matrix. This operation is denoted **vec**: if $U \in \mathbf{S}^p$ (the symmetric matrices of order $p$), then

$$\mathbf{vec}(U) = \left(U_{11}, \sqrt{2}U_{21}, \ldots, \sqrt{2}U_{p1}, U_{22}, \sqrt{2}U_{32}, \ldots, \sqrt{2}U_{p2}, \ldots, U_{p-1,p-1}, \sqrt{2}U_{p,p-1}, U_{pp}\right).$$

The scaling of the off-diagonal entries ensures that inner products are preserved, *i.e.*, $\mathbf{tr}(UV) = \mathbf{vec}(U)^T \mathbf{vec}(V)$ for all $U$, $V$. The inverse operation is denoted **mat**: if $u$ is a vector of length $p(p+1)/2$, then

$$\mathbf{mat}(u) = \left[\begin{array}{cccc} u_1 & u_2/\sqrt{2} & \cdots & u_p/\sqrt{2} \\ u_2/\sqrt{2} & u_{p+1} & \cdots & u_{2p-1}/\sqrt{2} \\ \vdots & \vdots & & \vdots \\ u_p/\sqrt{2} & u_{2p-1}/\sqrt{2} & \cdots & u_{p(p+1)/2} \end{array}\right].$$

The scaling ensures that $u^T v = \mathbf{tr}(\mathbf{mat}(u)\,\mathbf{mat}(v))$.

The image of $\mathbf{S}^p_+$ (the positive semidefinite matrices of order $p$) under the **vec** operation is denoted $\mathcal{S}_p$:

$$\mathcal{S}_p = \left\{\mathbf{vec}(U) \mid U \in \mathbf{S}^p_+\right\} = \{u \in \mathbf{R}^{p(p+1)/2} \mid \mathbf{mat}(u) \succeq 0\}.$$

The second-order cone in $\mathbf{R}^p$ is denoted

$$\mathcal{Q}_p = \{(u_0, u_1) \in \mathbf{R} \times \mathbf{R}^{p-1} \mid \|u_1\|_2 \le u_0\}.$$

The notation $\mathbf{R}^p_+$ is used for the cone of nonnegative $p$-vectors.

## 2 Logarithmic barrier function

We use the following logarithmic barrier function for $C$:

$$\phi(u) = \sum_{k=1}^{K} \phi_k(u_k), \qquad \phi_k(u) = \begin{cases} -\sum_{j=1}^{p} \log u_j & C_k = \mathbf{R}_+^p \\ -(1/2)\log(u_0^2 - u_1^T u_1) & C_k = \mathcal{Q}_p \\ -\log\det \mathbf{mat}(u) & C_k = \mathcal{S}_p. \end{cases}$$

Note that $\phi(tu) = \phi(u) - m\log t$ for $t > 0$ where

$$m = m_1 + \cdots + m_K, \qquad m_k = \begin{cases} p & C_k = \mathbf{R}_+^p \\ 1 & C_k = \mathcal{Q}_p \\ p & C_k = \mathcal{S}_p. \end{cases} \tag{4}$$

We refer to $m$ as the *degree* of the cone $C$.

### 2.1 Gradient

We write the gradients of $\phi$ and $\phi_k$ at $u$ as $g(u) = \nabla\phi(u)$ and $g_k(u_k) = \nabla\phi_k(u_k)$:

$$g_k(u_k) = \begin{cases} -\mathbf{diag}(u_k)^{-1}\mathbf{1} & C_k = \mathbf{R}_+^p \\ -(u_k^T J u_k)^{-1} J u_k & C_k = \mathcal{Q}_p \\ -\mathbf{vec}(\mathbf{mat}(u_k)^{-1}) & C_k = \mathcal{S}_p \end{cases}$$

where $\mathbf{1}$ is a $p$-vector of ones and

$$J = \begin{bmatrix} 1 & 0 \\ 0 & -I_{p-1} \end{bmatrix}.$$

It can be verified that $g(u) \prec 0$ and $u^T g(u) = -m$ for $u \succ 0$.

### 2.2 Hessian

The Hessians of $\phi$ and $\phi_k$ at $u$ are denoted $H(u) = \nabla^2\phi(u)$ and $H_k(u_k) = \nabla^2\phi_k(u_k)$. The Hessian for $C_k = \mathbf{R}_+^p$ is $H_k(u_k) = \mathbf{diag}(u_k)^{-2}$. The Hessian for $C_k = \mathcal{Q}_p$ is

$$H_k(u_k) = \frac{1}{(u_k^T J u_k)^2}\left(2Ju_k u_k^T J - (u_k^T J u_k)J\right), \qquad H_k(u_k)^{-1} = 2u_k u_k^T - (u_k^T J u_k)J.$$

For future reference, we give the symmetric square roots:

$$\begin{aligned} H_k(u_k)^{1/2} &= \frac{1}{u_k^T J u_k}\begin{bmatrix} u_{k0} & -u_{k1}^T \\ -u_{k1} & \left(u_{k0} + (u_k^T J u_k)^{1/2}\right)^{-1} u_{k1}u_{k1}^T + (u_k^T J u_k)^{1/2} I \end{bmatrix} \\ H_k(u_k)^{-1/2} &= \begin{bmatrix} u_{k0} & u_{k1}^T \\ u_{k1} & \left(u_{k0} + (u_k^T J u_k)^{1/2}\right)^{-1} u_{k1}u_{k1}^T + (u_k^T J u_k)^{1/2} I \end{bmatrix}. \end{aligned}$$

For $C_k = \mathcal{S}_p$, the Hessian is defined by

$$H_k(u)v = \mathbf{vec}\left(\mathbf{mat}(u_k)^{-1}\,\mathbf{mat}(v)\,\mathbf{mat}(u_k)^{-1}\right).$$

### 2.3 Self-scaled property

The following property is known as the *self-scaled* property of the barrier function [NT97, Tun98, NT98]. Suppose $w \succ 0$. Then $H(w)u \succ 0$ for all $u \succ 0$ and

$$\phi(H(w)u) = \phi(u) - 2\phi(w). \tag{5}$$

This identity is straightforward to derive in the case of the nonnegative orthant and the positive semidefinite cone, so we prove it only for the second-order cone $C_k = \mathcal{Q}_p$. Define $v_k = H_k(w_k)u_k$. Then

$$\begin{aligned} v_k &= \frac{1}{(w_k^T J w_k)^2}\left(2(w_k^T J u_k) J w_k - (w_k^T J w_k) J u_k\right) \\ v_k^T J v_k &= \frac{1}{(w_k J w_k)^4}\left(2(w_k^T J u_k) J w_k - (w_k^T J w_k) J u_k\right)^T \left(2(w_k^T J u_k) w_k - (w_k^T J w_k) u_k\right) \\ &= \frac{1}{(w_k^T J w_k)^4}\left(4(w_k^T J u_k)^2 (w_k^T J w_k) - 4(w_k^T J u_k)^2 (w_k^T J w_k) + (w_k^T J w_k)^2 (u_k^T J u_k)\right) \\ &= \frac{u_k^T J u_k}{(w_k^T J w_k)^2}. \end{aligned}$$

Hence $\phi_k(v_k) = -(1/2)\log(v_k^T J v_k) = -(1/2)\log(u_k^T J u_k) - \log(w_k^T J w_k)$.

By taking the first and second derivatives with respect to $u$ of each side of (5) we see that

$$H(w) g\left(H(w)u\right) = g(u), \qquad H(w) H\left(H(w)u\right) H(w) = H(u)$$

for all $u, w \succ 0$. Equivalently,

$$H(w) g(v) = g(H(w)^{-1} v), \qquad H(w) H(v) H(w) = H(H(w)^{-1} v) \tag{6}$$

for all $v, w \succ 0$. Thus, $H(H(w)^{-1}v)^{1/2}$ is the symmetric square root of $H(w)H(v)H(w)$.

# 3 Central path

The central path for (3) is defined as the family of points $(s, x, y, z)$ that satisfy

$$\begin{bmatrix} 0 \\ 0 \\ s \end{bmatrix} + \begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -c \\ b \\ h \end{bmatrix}, \qquad (s, z) \succ 0, \qquad z = -\mu g(s) \tag{7}$$

for some $\mu > 0$. Primal-dual algorithms are based on an equivalent definition of the central path in which the primal and dual variables appear symmetrically. The symmetric parametrization is obtained by writing $z = -\mu g(s)$ as $s \circ z = \mu\, \mathbf{e}$, where the $k$th component of the product $s \circ z$ is defined as $(s \circ z)_k = s_k \circ z_k$ with

$$u \circ v = \begin{cases} (u_1 v_1, \ldots, u_p v_p) & C_k = \mathbf{R}_+^p \\ (u^T v, u_0 v_1 + v_0 u_1) & C_k = \mathcal{Q}_p \\ (1/2)\, \mathbf{vec}(\mathbf{mat}(u)\, \mathbf{mat}(v) + \mathbf{mat}(v)\, \mathbf{mat}(u)) & C_k = \mathcal{S}_p, \end{cases}$$

and $\mathbf{e}$ is the vector $\mathbf{e} = (\mathbf{e}_1, \ldots, \mathbf{e}_K)$,

$$\mathbf{e}_k = \begin{cases} (1, 1, \ldots, 1) & C_k = \mathbf{R}^p_+ \\ (1, 0, \ldots, 0) & C_k = \mathcal{Q}_p \\ \mathbf{vec}(I_p) & C_k = \mathcal{S}_p. \end{cases}$$

Note that $\mathbf{e}^T(z \circ s) = z^T s$ and $\mathbf{e}^T\mathbf{e} = m$.

Replacing the condition $z = -\mu g(s)$ in (7) by the symmetric expression $s \circ z = \mu\,\mathbf{e}$ gives

$$\begin{bmatrix} 0 \\ 0 \\ s \end{bmatrix} + \begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -c \\ b \\ h \end{bmatrix}, \qquad (s, z) \succ 0, \qquad z \circ s = \mu\,\mathbf{e}. \tag{8}$$

For future reference, we give some useful properties of certain powers associated with the $\circ$ product. The inverse, square, and square root of $u$ are defined by the relations $u^{-1} \circ u = \mathbf{e}$, $u^2 = u \circ u$, $u^{1/2} \circ u^{1/2} = u$, and can be computed componentwise for each subvector in $u = (u_1, \ldots, u_k)$. If $C_k = \mathbf{R}^p_+$ the operations are the componentwise vector operations on $u_k$. If $C_k = \mathcal{S}_p$ they are given by the matrix inverse, square, and symmetric square root. If $C_k = \mathcal{Q}_p$, we have

$$u_k^{-1} = \frac{1}{u_k^T J u_k} J u_k, \qquad u_k^2 = \begin{bmatrix} u_k^T u_k \\ 2 u_{k0} u_{k1} \end{bmatrix}, \qquad u_k^{1/2} = \frac{1}{\sqrt{2(u_{k0} + \sqrt{u_k^T J u_k})}} \begin{bmatrix} u_{k0} + \sqrt{u_k^T J u_k} \\ u_{k1} \end{bmatrix}.$$

Note that

$$(u_k^{-1})^T J u_k^{-1} = (u_k^T J u_k)^{-1}, \qquad (u_k^{1/2})^T J u_k^{1/2} = (u_k^T J u_k)^{1/2}.$$

The following general properties of the logarithmic barrier are also useful.

$$H(u)^{-1} = H(u^{-1}), \qquad H(u)^{1/2} = H(u^{1/2}), \tag{9}$$

$$H(u)u = u^{-1}, \qquad H(u^{1/2})u = \mathbf{e}, \qquad (H(u)v)^{-1} = H(u)^{-1}v^{-1}. \tag{10}$$

## 4 Nesterov-Todd scaling

A primal-dual scaling $W$ is a linear transformation

$$\tilde{s} = W^{-T} s, \qquad \tilde{z} = W z$$

that leaves the cone and the central path invariant, *i.e.*,

$$s \succ 0 \iff \tilde{s} \succ 0, \qquad z \succ 0 \iff \tilde{z} \succ 0, \qquad s \circ z = \mu\,\mathbf{e} \iff \tilde{s} \circ \tilde{z} = \mu\,\mathbf{e}.$$

If $W$ is a scaling we can write the central path equations (8) equivalently as

$$\begin{bmatrix} 0 \\ 0 \\ s \end{bmatrix} + \begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -c \\ b \\ h \end{bmatrix}, \qquad (s, z) \succ 0, \qquad (Wz) \circ (W^{-T}s) = \mu\,\mathbf{e}. \tag{11}$$

The self-scaled property of the logarithmic barrier function (section 2.3) provides a method for constructing primal-dual scalings that are symmetric: the Hessian of the barrier at any strictly positive point is a primal-dual scaling. To see this, we first note that the self-scaled property implies that multiplications with the Hessian and inverse Hessian leave the interior of the cone invariant. Second, if $W$ is the Hessian of the barrier at some point, then, from (6), $z = -\mu g(s)$ is equivalent to $Wz = -\mu g(W^{-1}s)$, *i.e.*, $(Wz) \circ (W^{-1}s) = \mu \mathbf{e}$. In general, however, one can also consider other, non-symmetric, scaling matrices.

Interior-point algorithms are based on linearizing the central path equations (11), using a primal-dual scaling that changes at each iteration depending on the values of the current iterates $\hat{s}$, $\hat{z}$. The *Nesterov-Todd scaling* at $\hat{s}$, $\hat{z}$ is derived from the unique scaling point $w$ that satisfies

$$H(w)\hat{s} = \hat{z}$$

[NT97, NT98]. A general expression for the scaling point is

$$w = H(\hat{s}^{-1/2}) \left( H(\hat{s}^{-1/2})\hat{z} \right)^{-1/2} = H(\hat{z}^{1/2}) \left( H(\hat{z}^{-1/2})\hat{s} \right)^{1/2}. \tag{12}$$

To see this, define $u = H(\hat{s}^{-1/2})\hat{z}$. From (6) and (9)–(10),

$$\begin{aligned} H(w)\hat{s} &= H(\hat{s}^{1/2})H(u^{-1/2})H(\hat{s}^{1/2})\hat{s} \\ &= H(\hat{s}^{1/2})H(u^{-1/2})\mathbf{e} \\ &= H(\hat{s}^{1/2})u \\ &= \hat{z}. \end{aligned}$$

A Nesterov-Todd scaling is obtained by factoring $H(w)$ as $H(w)^{-1} = W^T W$, where $W$ is a scaling matrix. Hence $W\hat{z} = W^{-T}\hat{s}$ and we will denote this vector as $\lambda$:

$$\lambda = W^{-T}\hat{s} = W\hat{z}.$$

One possible factorization is the symmetric square root: $W = H(w)^{-1/2}$. This is a scaling matrix, because from (9), $H(w)^{-1/2} = H(w^{-1/2})$ and we have seen that the Hessian of the barrier at any strictly positive point is a scaling matrix. Note that there may exist more than one suitable factorization $H(w)^{-1} = W^T W$. For example, for the second-order and semidefinite cones, one can choose the symmetric scaling $W = H(w)^{-1/2}$, or a nonsymmetric scaling.

## 4.1 Nonnegative orthant

**Scaling** Any positive diagonal matrix $W_k$ can be used as a scaling for $C_k = \mathbf{R}^p_+$.

**Nesterov-Todd scaling point** The Nesterov-Todd scaling point at $\hat{s}_k$, $\hat{z}_k$ is

$$w_k = \hat{s}_k^{1/2} \circ \hat{z}_k^{-1/2}.$$

**Nesterov-Todd scaling** The Nesterov-Todd scaling is

$$W_k = \mathbf{diag}(w_k) = \mathbf{diag}(\hat{s}_k^{1/2} \circ \hat{z}_k^{-1/2}).$$

The scaled variable $\lambda_k = W_k^{-1}\hat{s}_k = W_k\hat{z}_k$ is

$$\lambda_k = \hat{z}_k^{1/2} \circ \hat{s}_k^{1/2}.$$

### 4.2 Second-order cone

**Scaling** Any matrix $W_k$ that satisfies

$$W_k J W_k^T = \beta^2 J \tag{13}$$

where $\beta \neq 0$, can be used as scaling matrix for $C_k = \mathcal{Q}_p$. (The matrix $(1/\beta)W_k$ is sometimes called a *hypernormal matrix* [RS88].) Note that $W_k$ is necessarily nonsingular, and

$$W_k^T J W_k = J W_k^{-1} (W_k J W_k^T) J W_k = \beta^2 J. \tag{14}$$

Examples of symmetric scaling matrices are Hessians $H_k(u)$ or inverse Hessians $H_k(u)^{-1}$ of the second-order cone barrier (in this case $\beta = 1/(u^T J u)$, resp., $\beta = u^T J u$). The matrix

$$\frac{1}{u^T J u} H_k^{-1}(u) = \frac{2}{u^T J u} u u^T - J$$

is also called a *hyperbolic Householder matrix* [RS88]. A product of scaling matrices (for example, $W_k J$), is also a (generally nonsymmetric) scaling matrix.

We now verify that if $W_k$ satisfies (13), then the second-order cone and the central path are preserved under multiplication with $W_k$. Let $\mathbf{e}_k$ be the first unit vector and $v = W_k^T \mathbf{e}_k = (v_0, v_1)$ the first row of $W_k$. This is a nonnegative vector since, from (13),

$$v^T J v = (W_k^T \mathbf{e}_k)^T J (W_k^T \mathbf{e}_k) = \mathbf{e}_k^T W_k J W_k^T \mathbf{e}_k = \beta^2 \geq 0.$$

Suppose $x = (x_0, x_1) \in \mathcal{Q}_p$ and $\tilde{x} = W_k x$. Then

$$\tilde{x}_0 = v^T x = v_0 x_0 + v_1^T x_1 \geq v_0 x_0 - \|v_1\|_2 \|x_1\|_2 \geq 0$$

by the Cauchy-Schwarz inequality, and

$$\tilde{x}^T J \tilde{x} = x^T W_k^T J W_k x = \beta^2 x^T J x \geq 0.$$

Conversely, using (14) we see that if $\tilde{x}$ is in the second-order cone, then $x = W_k^{-1}\tilde{x}$ is also in the second-order cone. A similar argument shows that multiplications with $W_k^T$ and $W_k^{-T}$ preserve the second-order cone. Furthermore, if $z_k$ and $s_k$ are on the central path, *i.e.*,

$$z_k = -\mu g_k(s_k) = \frac{\mu}{s_k^T J s_k} J s_k,$$

then $\tilde{z}_k = W_k z_k$, $\tilde{s}_k = W_k^{-T} s_k$ are on the transformed central path, with the same parameter $\mu$:

$$\tilde{z}_k = \frac{\mu}{s_k^T J s_k} W_k J W_k^T \tilde{s}_k = \frac{\mu}{\tilde{s}_k^T J \tilde{s}_k} J \tilde{s}_k = -\mu g_k(\tilde{s}_k).$$

**Nesterov-Todd scaling point** The Nesterov-Todd scaling point $w_k$ is uniquely defined by

$$H(w_k)^{-1} \hat{z}_k = \hat{s}_k.$$

Let $\overline{z}_k$ and $\overline{s}_k$ be the normalized vectors

$$\overline{z}_k = \frac{1}{(\hat{z}_k^T J \hat{z}_k)^{1/2}} \hat{z}_k, \qquad \overline{s}_k = \frac{1}{(\hat{s}_k^T J \hat{s}_k)^{1/2}} \hat{s}_k,$$

and define

$$\gamma = \left(\frac{1+\overline{z}_k^T\overline{s}_k}{2}\right)^{1/2}, \qquad \overline{w}_k = \frac{1}{2\gamma}\left(\overline{s}_k + J\overline{z}_k\right). \tag{15}$$

We have

$$\overline{w}_k^T J\overline{w}_k = 1, \qquad \overline{w}_k^T\overline{z}_k = \overline{w}_k^T J\overline{s}_k = \gamma.$$

From this it is easy to see that

$$\left(2\overline{w}_k\overline{w}_k^T - J\right)\overline{z}_k = \overline{s}_k, \qquad \left(2J\overline{w}_k\overline{w}_k^T J - J\right)\overline{s}_k = \overline{z}_k.$$

In other words the hyperbolic Householder transformation defined by $\overline{w}_k$ maps $\bar{z}_k$ to $\bar{s}_k$.

In terms of the unnormalized variables, this means that if we define

$$w_k^T J w_k = \left(\frac{\hat{s}_k^T J\hat{s}_k}{\hat{z}_k^T J\hat{z}_k}\right)^{1/2}, \qquad w_k = \left(w_k^T J w_k\right)^{1/2}\overline{w}_k,$$

then

$$H(w_k)^{-1}\hat{z}_k = \left(2w_kw_k^T - (w_k^T J w_k)J\right)\hat{z}_k = (w_k^T J w_k)\left(2\overline{w}_k\overline{w}_k^T - J\right)\hat{z}_k = \hat{s}_k.$$

**Symmetric Nesterov-Todd scaling** Let $\overline{w}_k$ be as in (15), and define

$$v_k = \overline{w}_k^{1/2} = \frac{1}{(2(\overline{w}_{k0}+1))^{1/2}}(\overline{w}_k + \mathbf{e}_k).$$

We have $v_k^T J v_k = 1$, so the matrices

$$\overline{W}_k = 2v_kv_k^T - J, \qquad \overline{W}_k^{-1} = 2Jv_kv_k^T J - J$$

are hyperbolic Householder matrices. More explicitly, written in terms of $\overline{w}_k$,

$$\overline{W}_k = \begin{bmatrix} \overline{w}_{k0} & \overline{w}_{k1}^T \\ \overline{w}_{k1} & I + (\overline{w}_{k0}+1)^{-1}\overline{w}_{k1}\overline{w}_{k1}^T \end{bmatrix}, \qquad \overline{W}_k^{-1} = \begin{bmatrix} \overline{w}_{k0} & -\overline{w}_{k1}^T \\ -\overline{w}_{k1} & I + (\overline{w}_{k0}+1)^{-1}\overline{w}_{k1}\overline{w}_{k1}^T \end{bmatrix}.$$

$\overline{W}_k$ is the Householder transformation that maps $J\overline{w}_k$ to $\mathbf{e}_k$, and therefore

$$\overline{W}_k(2J\overline{w}_k\overline{w}_k^T J - J)\overline{W}_k = (2\mathbf{e}_k\mathbf{e}_k^T - J) = I.$$

In other words, $\overline{W}_k = \left(2\overline{w}_k\overline{w}_k^T - J\right)^{1/2}$. In terms of the unnormalized variables, $H(w_k)^{-1} = W_k^T W_k$ where

$$W_k = (w_k^T J w_k)^{1/2}\overline{W}_k = \left(\frac{\hat{s}_k^T J\hat{s}_k}{\hat{z}_k^T J\hat{z}_k}\right)^{1/4}\overline{W}_k.$$

To find expressions for the scaled variables, we define

$$\overline{\lambda}_k = \overline{W}_k\overline{z}_k = \overline{W}_k^{-1}\overline{s}_k = J\overline{W}_k J\overline{s}_k.$$

We have $\overline{\lambda}_k^T J\overline{\lambda}_k = 1$ and

$$\overline{\lambda}_{k0} = \gamma, \qquad \overline{\lambda}_k - J\overline{\lambda}_k = \overline{W}_k(\overline{z}_k - J\overline{s}_k).$$

The last expression provides a way to evaluate $\overline{\lambda}_k$ directly from $\hat{s}_k$ and $\hat{z}_k$:

$$\begin{aligned}\overline{\lambda}_{k1} &= \frac{1}{2}\left(\overline{W}_k(\overline{z}_k - J\overline{s}_k)\right)_1 \\ &= \frac{1}{2}\left(\overline{z}_{k1} + \overline{s}_{k1} + \frac{\overline{z}_{k0} - \overline{s}_{k0}}{\overline{w}_{k0}+1}\overline{w}_{k1}\right) \\ &= \frac{1}{\overline{s}_{k0} + \overline{z}_{k0} + 2\gamma}\left((\gamma + \overline{z}_{k0})\overline{s}_{k1} + (\gamma + \overline{s}_{k0})\overline{z}_{k1}\right).\end{aligned}$$

The unnormalized scaled variable is

$$\lambda_k = W_k \hat{z}_k = W_k^{-1}\hat{s}_k = \left((\hat{s}_k^T J \hat{s}_k)(\hat{z}_k^T J \hat{z}_k)\right)^{1/4} \overline{\lambda}_k.$$

### 4.3 Semidefinite cone

**Scaling** Any nonsingular congruence transformation can be used as a scaling for $C_k \in \mathcal{S}_p$:

$$W_k v = \mathbf{vec}(R^T \mathbf{mat}(v) R), \qquad W_k^{-T} u = \mathbf{vec}(R^{-1} \mathbf{mat}(u) R^{-T}).$$

**Nesterov-Todd scaling point** The scaling point at $\hat{s}_k$, $\hat{z}_k$ is the symmetric matrix for which

$$\mathbf{mat}(w_k)\hat{Z}_k \mathbf{mat}(w_k) = \hat{S}_k$$

where $\hat{S}_k = \mathbf{mat}(\hat{s}_k)$ and $\hat{Z}_k = \mathbf{mat}(\hat{z}_k)$. From (12),

$$w_k = \mathbf{vec}\left(\hat{S}_k^{1/2}\left(\hat{S}_k^{1/2}\hat{Z}_k\hat{S}_k^{1/2}\right)^{-1/2}\hat{S}_k^{1/2}\right).$$

**Nonsymmetric Nesterov-Todd scaling** The scaling point $w_k$ can be computed in factored form $w_k = \mathbf{vec}(R_k R_k^T)$, where $R_k$ diagonalizes $\mathbf{mat}(\hat{z}_k)$ and $\mathbf{mat}(\hat{s}_k)$:

$$R_k^T \mathbf{mat}(\hat{z}_k) R_k = R_k^{-1} \mathbf{mat}(\hat{s}_k) R_k^{-T} = \mathbf{mat}(\lambda_k),$$

with $\mathbf{mat}(\lambda_k)$ diagonal. Note that $W_k^{-T}\hat{s}_k = W_k \hat{z}_k = \lambda_k$ and $\lambda_k^T \lambda_k = \hat{s}_k^T \hat{z}_k$.

The scaling matrix $R_k$ can be computed as follows. We first compute Cholesky factorizations

$$S_k = \mathbf{mat}(\hat{s}_k) = L_1 L_1^T, \qquad Z_k = \mathbf{mat}(\hat{z}_k) = L_2 L_2^T.$$

Next, we compute the SVD

$$L_2^T L_1 = U \Lambda_k V^T$$

and take $\lambda_k = \mathbf{vec}(\mathbf{diag}(\Lambda_k))$. Finally, we form

$$R_k = L_1 V \Lambda_k^{-1/2} = L_2^{-T} U \Lambda_k^{1/2}.$$

It can be verified that $R_k^T S_k^{-1} R_k = \Lambda_k^{-1}$ and $R_k^T Z_k R_k = \Lambda_k$ and that the inverse of $R_k$ is given by $R_k^{-1} = \Lambda_k^{1/2} V^T L_1^{-1} = \Lambda_k^{-1/2} U^T L_2^T$.

### 4.4 Compositions of scaling matrices

If $V$ is a scaling matrix, then

$$V^T H(w)^{-1} V = H(V^T w)^{-1}. \tag{16}$$

This is easy to see for the nonnegative orthant and the semidefinite cone. To verify the property for $C_k = \mathcal{Q}_p$, assume $V_k^T J V_k = V_k J V_k^T = \beta^2 J$. Then

$$\begin{aligned} V_k^T H_k(w_k)^{-1} V_k &= V_k^T (2 w_k w_k^T V_k - (w_k^T J w_k) J) V_k \\ &= V_k^T 2 w_k w_k^T V - (w_k^T J w_k) \beta^2 J \\ &= 2 V_k^T w_k w_k^T V_k - (w_k^T V_k J V_k^T w_k) J \\ &= H_k(V_k^T w_k)^{-1}. \end{aligned}$$

## 5 Path-following algorithm for cone QPs

The algorithm implemented in `coneqp` is based on linearizing the central path equations (11), obtained from (8) after applying a scaling with a matrix $W$.

### 5.1 Outline

We denote the current iterates by $(\hat{s}, \hat{x}, \hat{y}, \hat{z})$. We start at initial values $(\hat{s}, \hat{x}, \hat{y}, \hat{z}) = (s_0, x_0, y_0, z_0)$, where $s_0 \succ 0$, $z_0 \succ 0$. We also compute the Nesterov-Todd scaling $W$ at $\hat{s}$, $\hat{z}$, and the scaled variable $\lambda := W^{-T}\hat{s} = W\hat{z}$.

1. *Evaluate residuals, gap, and stopping criteria.* Compute

$$\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \hat{s} \end{bmatrix} + \begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} + \begin{bmatrix} c \\ -b \\ -h \end{bmatrix} \tag{17}$$

   and

$$\hat{\mu} = \frac{\hat{s}^T \hat{z}}{m} = \frac{\lambda^T \lambda}{m}.$$

   Terminate if $(s, x, y, z) = (\hat{s}, \hat{x}, \hat{y}, \hat{z})$ satisfies (approximately) the optimality conditions

$$\begin{bmatrix} 0 \\ 0 \\ s \end{bmatrix} = \begin{bmatrix} P & A^T & G^T \\ -A & 0 & 0 \\ -G & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} c \\ b \\ h \end{bmatrix}, \qquad (s, z) \succeq 0, \qquad z^T s = 0.$$

2. *Affine direction.* Solve the linear equations

$$\begin{bmatrix} 0 \\ 0 \\ \Delta s_{\mathrm{a}} \end{bmatrix} + \begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\mathrm{a}} \\ \Delta y_{\mathrm{a}} \\ \Delta z_{\mathrm{a}} \end{bmatrix} = - \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \tag{18a}$$

$$\lambda \circ \left( W \Delta z_{\mathrm{a}} + W^{-T} \Delta s_{\mathrm{a}} \right) = -\lambda \circ \lambda. \tag{18b}$$

3. *Step size and centering parameter.* Compute

$$\begin{aligned}
\alpha &= \sup\{\alpha \in [0,1] \mid (\hat{s},\hat{z}) + \alpha(\Delta s_{\rm a}, \Delta z_{\rm a}) \succeq 0\} \\
&= \sup\left\{\alpha \in [0,1] \mid (\lambda,\lambda) + \alpha(W^{-T}\Delta s_{\rm a}, W\Delta z_{\rm a}) \succeq 0\right\} \\
\rho &= \frac{(\hat{s}+\alpha\Delta s_{\rm a})^T(\hat{z}+\alpha\Delta z_{\rm a})^T}{\hat{s}^T\hat{z}} \\
&= 1-\alpha+\alpha^2\frac{(W^{-T}\Delta s_{\rm a})^T(W\Delta z_{\rm a})}{\lambda^T\lambda} \\
\sigma &= \max\{0,\min\{1,\rho\}\}^3.
\end{aligned}$$

4. *Combined direction.* Solve the linear equation

$$\begin{bmatrix} 0 \\ 0 \\ \Delta s \end{bmatrix} + \begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = -(1-\eta)\begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \tag{19a}$$

$$\lambda \circ \left(W\Delta z + W^{-T}\Delta s\right) = -\lambda\circ\lambda - \gamma(W^{-T}\Delta s_{\rm a})\circ(W\Delta z_{\rm a}) + \sigma\hat{\mu}\mathbf{e}. \tag{19b}$$

Common choices for $\eta$ are $\eta = 0$ and $\eta = \sigma$. The current implementation uses $\eta = 0$. The parameter $\gamma$ is 1 or 0, depending on whether or not a Mehrotra correction is used. The default value is $\gamma = 1$.

5. *Update iterates and scaling matrices.*

$$(\hat{s},\hat{x},\hat{y},\hat{z}) := (\hat{s},\hat{x},\hat{y},\hat{z}) + \alpha(\Delta s, \Delta x, \Delta y, \Delta z)$$

where

$$\alpha = \sup\left\{\alpha\in[0,1] \;\middle|\; (\lambda,\lambda) + \frac{\alpha}{0.99}(W^{-T}\Delta s, W\Delta z) \succeq 0\right\}.$$

Compute the scaling matrix $W$ for $\hat{s}$, $\hat{z}$, and the scaled variable $\lambda := W^{-T}\hat{s} = W\hat{z}$.

## 5.2 Discussion

The equations (18) are obtained by substituting $(s,x,y,z) = (\hat{s},\hat{x},\hat{y},\hat{z}) + (\Delta s_{\rm a}, \Delta x_{\rm a}, \Delta y_{\rm a}, \Delta z_{\rm a})$ in the two equations in (11) with $\mu = 0$, and setting the second order terms in

$$(W(\hat{z}+\Delta z_{\rm a}))\circ(W^{-T}(\hat{s}+\Delta s_{\rm a})) = (\lambda + W\Delta z_{\rm a})\circ(\lambda + W^{-T}\Delta s_{\rm a}) = 0$$

equal to zero. If $\eta = \gamma = 0$ the equations (19) are obtained in the same way by linearizing (11) with $\mu = \sigma\hat{\mu}$. Nonzero values of $\eta$ can be justified by writing (19a) as

$$\begin{bmatrix} 0 \\ 0 \\ \hat{s}+\Delta s \end{bmatrix} + \begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & 0 \end{bmatrix} \begin{bmatrix} x+\Delta x \\ y+\Delta y \\ z+\Delta z \end{bmatrix} + \begin{bmatrix} c \\ -b \\ -h \end{bmatrix} = \eta \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix}.$$

This shows that taking a unit step in the direction $(\Delta s, \Delta x, \Delta y, \Delta z)$ decreases the residual by a fraction $\eta$. If we choose $\gamma = 1$, we approximate the second order terms in

$$(W(\hat{z}+\Delta z))\circ(W^{-T}(\hat{s}+\Delta s)) = \sigma\hat{\mu}\,\mathbf{e},$$

as

$$(W^{-T}\Delta s)\circ(W\Delta z) \approx (W^{-T}\Delta s_{\rm a})\circ(W\Delta z_{\rm a}).$$

The second term on the righthand side of (19b) is known as the Mehrotra correction [Meh92, Wri97].

### 5.3 Initialization

If primal and dual starting points $\hat{x}$, $\hat{s}$, $\hat{y}$, $\hat{z}$ are not specified by the user, they are selected as follows. We solve the linear equation

$$\begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & -I \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -c \\ b \\ h \end{bmatrix}, \tag{20}$$

and take $\hat{x} = x$, $\hat{y} = y$. The equation (20) gives the optimality conditions for the pair of primal and dual problems

$$\begin{array}{ll} \text{minimize} & (1/2)x^TPx + c^Tx + (1/2)\|s\|_2^2 \\ \text{subject to} & Gx + s = h \\ & Ax = b \end{array}$$

and

$$\begin{array}{ll} \text{maximize} & -(1/2)w^TPw - h^Tz - b^Ty - (1/2)\|z\|_2^2 \\ \text{subject to} & Pw + G^Tz + A^Ty + c = 0. \end{array}$$

The initial value of $\hat{s}$ is computed from the residual $h - Gx = -z$, as

$$\hat{s} = \begin{cases} -z & \alpha_{\rm p} < 0 \\ -z + (1 + \alpha_{\rm p})\mathbf{e} & \text{otherwise} \end{cases}$$

where $\alpha_{\rm p} = \inf\{\alpha \mid -z + \alpha\,\mathbf{e} \succeq 0\}$. The initial value of $z$ is

$$\hat{z} = \begin{cases} z & \alpha_{\rm d} < 0 \\ z + (1 + \alpha_{\rm d})\mathbf{e} & \text{otherwise,} \end{cases}$$

where $\alpha_{\rm d} = \inf\{\alpha \mid z + \alpha\,\mathbf{e} \succeq 0\}$.

### 5.4 Newton equations

The most expensive computation in each iteration of the algorithm is the solution of the linear equations in steps 2 and 4. These equations differ only in the righthand side and are of the form

$$\begin{bmatrix} 0 \\ 0 \\ \Delta s \end{bmatrix} + \begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \tag{21a}$$

$$\lambda \circ \left( W\Delta z + W^{-T}\Delta s \right) = d_s. \tag{21b}$$

We refer to the equations as Newton equations because they can be interpreted as linearizations of the central path conditions. In this section we describe how CVXOPT reduces the Newton equations to a smaller $3 \times 3$ block equation (KKT system). Later, in section 10, we explain how the $3 \times 3$ block equation is solved.

Eliminating $\Delta s$ from (44b) gives

$$\begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & -W^TW \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \\ d_z - W^T(\lambda \diamond d_s) \end{bmatrix} \tag{22a}$$

$$\Delta s = W^T \left(\lambda \diamond d_s - W\Delta z\right). \tag{22b}$$

Here $u \diamond v$ denotes the inverse of $u \circ v$ taken as a linear function of $v$, *i.e.*, $u \circ (u \diamond v) = v$ for all $v$. For $C_k = \mathbf{R}^p_+$, $\lambda_k \diamond v = \mathbf{diag}(\lambda_k)^{-1} v$. For $C_k = \mathcal{Q}_p$,

$$\begin{aligned}
\lambda_k \diamond v &= \begin{bmatrix} \lambda_{k0} & \lambda_{k1}^T \\ \lambda_{k1} & \lambda_{k0} I \end{bmatrix}^{-1} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} \\
&= \frac{1}{\lambda_{k0}^2 - \lambda_{k1}^T \lambda_{k1}} \begin{bmatrix} \lambda_{k0} & -\lambda_{k1}^T \\ -\lambda_{k1} & \lambda_{k0}^{-1}\left((\lambda_{k0}^2 - \lambda_{k1}^T\lambda_{k1})I + \lambda_{k1}\lambda_{k1}^T\right) \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}
\end{aligned}$$

If $C_k = \mathcal{S}_p$, with $\Lambda = \mathbf{mat}(\lambda_k)$, $\lambda_k \diamond v$ is the solution of

$$\frac{1}{2}(\Lambda\, \mathbf{mat}(x) + \mathbf{mat}(x)\Lambda) = \mathbf{mat}(v),$$

*i.e.*, $\lambda_k \diamond v = \mathbf{vec}(\mathbf{mat}(v) \odot \Gamma)$ where $\Gamma_{ij} = 2/(\Lambda_{ii} + \Lambda_{jj})$ and $\odot$ denotes the Hadamard (element-wise) matrix product.

Note that $u = W^T(\lambda \diamond d_s)$ is the solution of $\hat{z} \circ u = d_s$. Hence, the solution of (22) depends only on the product $W^T W$ and not on the scaling $W$ itself as the righthand side of (22a) may suggest. Note also that for the affine scaling Newton equation (step 2), the righthand side of (22a) simplifies to

$$d_z - W^T(\lambda \diamond d_s) = -r_z + W^T\lambda = -r_z + \hat{s}.$$

# 6 Self-dual embedding of cone LPs

The `conelp` algorithm is based on a self-dual reformulation of the cone LPs [YTM94, dKRT97]. In this section we first describe a homogeneous embedding, and explain how it can be used to detect primal and dual infeasibility. We then give a slightly larger extended embedding that has the advantage of being strictly feasible and define the central path for the embedded problem.

## 6.1 Homogeneous self-dual embedding

The primal and dual cone LPs can be embedded in a self-dual cone LP

$$\begin{array}{ll}
\text{minimize} & 0 \\
\text{subject to} & \begin{bmatrix} 0 \\ 0 \\ s \\ \kappa \end{bmatrix} = \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \tau \end{bmatrix} \\
& (s, \kappa, z, \tau) \succeq 0.
\end{array} \tag{23}$$

This problem is always feasible, since $(s, \kappa, x, y, z, \tau) = 0$ is a feasible point. Moreover any feasible point is optimal. We also note that the equality constraint implies that

$$s^T z + \kappa\tau = \begin{bmatrix} x \\ y \\ z \\ \tau \end{bmatrix}^T \begin{bmatrix} 0 \\ 0 \\ s \\ \kappa \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \tau \end{bmatrix}^T \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \tau \end{bmatrix} = 0$$

at all feasible points. In particular, this shows that there are no strictly feasible points.

Now suppose $(s, \kappa, x, y, z, \tau)$ is a solution of (23) with $\kappa + \tau > 0$.

- If $\tau > 0$, $\kappa = 0$, we can divide $x$, $y$, $z$ by $\tau$ to obtain a solution of the Karush-Kuhn-Tucker (KKT) conditions for (1),

$$\begin{bmatrix} 0 \\ 0 \\ s \end{bmatrix} = \begin{bmatrix} 0 & A^T & G^T \\ -A & 0 & 0 \\ -G & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} c \\ b \\ h \end{bmatrix} = 0 \qquad (s, z) \succeq 0, \qquad z^T s = 0. \tag{24}$$

- If $\tau = 0$, $\kappa > 0$, then $h^T z + b^T y + c^T x < 0$, so we must have $h^T z + b^T y < 0$ or $c^T x < 0$ or both. If $h^T z + b^T y < 0$, this provides a proof of primal infeasibility, since

$$G^T z + A^T y = 0, \qquad z \succeq 0, \qquad h^T z + b^T y < 0. \tag{25}$$

  If $c^T x < 0$, this provides a proof of dual infeasibility, since

$$Gx + s = 0, \qquad Ax = 0, \qquad s \succeq 0, \qquad c^T x < 0. \tag{26}$$

If $\tau = \kappa = 0$, no conclusion can be made about (1).

### 6.2 Extended self-dual embedding

As an extension, we can define another self-dual cone LP

$$\begin{array}{ll} \text{minimize} & (m+1)\theta \\ \text{subject to} & \begin{bmatrix} 0 \\ 0 \\ s \\ \kappa \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & A^T & G^T & c & q_x \\ -A & 0 & 0 & b & q_y \\ -G & 0 & 0 & h & q_z \\ -c^T & -b^T & -h^T & 0 & q_\tau \\ -q_x^T & -q_y^T & -q_z^T & -q_\tau & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \tau \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ m+1 \end{bmatrix} \\ & (s, \kappa, z, \tau) \succeq 0. \end{array} \tag{27}$$

Here $m$ is the degree of the cone, defined in (4), and

$$\begin{bmatrix} q_x \\ q_y \\ q_z \\ q_\tau \end{bmatrix} = \frac{m+1}{s_0^T z_0 + 1} \left( \begin{bmatrix} 0 \\ 0 \\ s_0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} \right) \tag{28}$$

where $x_0$, $s_0$, $y_0$, $z_0$ can be chosen arbitrarily with $(s_0, z_0) \succ 0$. This LCP is always strictly feasible and

$$(s, \kappa, x, y, z, \tau, \theta) = (s_0, 1, x_0, y_0, z_0, 1, \frac{s_0^T z_0 + 1}{m+1})$$

is a strictly feasible point. By taking the inner product of both sides of the equality constraint in (27) with $(x, y, z, \tau, \theta)$ we see that the constraint implies that

$$\theta = \frac{s^T z + \kappa\tau}{m+1}, \tag{29}$$

so $\theta \geq 0$ for all feasible points.

It is easily verified that (27) is self-dual, *i.e.*, its dual problem is formally the same (if we change the objective to a maximization). Therefore, at optimum the solution must satisfy a complementarity condition with itself, and we can write the optimality conditions for (27) as

$$\begin{bmatrix} 0 \\ 0 \\ s \\ \kappa \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & A^T & G^T & c & q_x \\ -A & 0 & 0 & b & q_y \\ -G & 0 & 0 & h & q_z \\ -c^T & -b^T & -h^T & 0 & q_\tau \\ -q_x^T & -q_y^T & -q_z^T & -q_\tau & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \tau \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ m+1 \end{bmatrix}, \tag{30a}$$

$$(s, \kappa, z, \tau) \succeq 0, \qquad z^T s + \kappa\tau = 0. \tag{30b}$$

Combined with (29), this implies that at the optimum $\theta = 0$ and the extended embedding reduces to the homogeneous embedding. If $(s, \kappa, x, y, z, \tau, \theta)$ is an optimal solution with $\kappa + \tau > 0$, we can therefore extract from it an optimal solution of (1), or a proof of primal or dual infeasibility.

## 6.3 Central path of the embedded problem

The central path of (27) is defined as the solution of

$$\begin{bmatrix} 0 \\ 0 \\ s \\ \kappa \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & A^T & G^T & c & q_x \\ -A & 0 & 0 & b & q_y \\ -G & 0 & 0 & h & q_z \\ -c^T & -b^T & -h^T & 0 & q_\tau \\ -q_x^T & -q_y^T & -q_z^T & -q_\tau & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \tau \\ \theta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ m+1 \end{bmatrix}, \tag{31a}$$

$$(s, \kappa, z, \tau) \succ 0, \qquad z = -\mu g(s), \qquad \tau = \mu/\kappa \tag{31b}$$

where $\mu$ is a nonnegative parameter. It follows from the equalities in (31b) and the property $s^T g(s) = -m$ that $\mu = (s^T z + \kappa\tau)/(m+1)$. By taking the inner product with $(x, y, z, \tau, \theta)$ on both sides of the equality (31) we also see that $\theta = \mu$ at points on the central path. We can therefore parametrize the central path more simply as

$$\begin{bmatrix} 0 \\ 0 \\ s \\ \kappa \end{bmatrix} = \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \tau \end{bmatrix} + \mu \begin{bmatrix} q_x \\ q_y \\ q_z \\ q_\tau \end{bmatrix} \tag{32a}$$

$$(s, \kappa, z, \tau) \succ 0, \qquad z = -\mu g(s), \qquad \tau = \mu/\kappa. \tag{32b}$$

The last equality in (31) was dropped because it is redundant: by taking the inner product of both sides of (32a) with $(0, 0, z, \tau)$, we get

$$z^T s + \tau\kappa = \mu(q_x^T x + q_y^T y + q_z^T z + q_\tau \tau),$$

and hence the last equation in (31). We will use (32) to parametrize the central path. Alternatively, we can interpret (32) as a nonstandard definition of the central path for the homogeneous embedding (23).

As in section 3 we replace the condition $z = -\mu g(s)$ by the symmetric relation $s \circ z = \mu \mathbf{e}$. This gives

$$\begin{bmatrix} 0 \\ 0 \\ s \\ \kappa \end{bmatrix} = \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \tau \end{bmatrix} + \mu \begin{bmatrix} q_x \\ q_y \\ q_z \\ q_\tau \end{bmatrix} \tag{33a}$$

$$(s, \kappa, z, \tau) \succ 0, \qquad z \circ s = \mu \mathbf{e}, \qquad \kappa\tau = \mu \tag{33b}$$

as a symmetric equivalent of the central path equations (32).

# 7 Path-following algorithm for cone LPs

The algorithm computes search directions by linearizing the central path equations (33) around the current iterate $(\hat{s}, \hat{\kappa}, \hat{x}, \hat{y}, \hat{z}, \hat{\tau})$, after applying a scaling with a matrix $W$:

$$\begin{bmatrix} 0 \\ 0 \\ s \\ \kappa \end{bmatrix} = \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \tau \end{bmatrix} + \mu \begin{bmatrix} q_x \\ q_y \\ q_z \\ q_\tau \end{bmatrix} \tag{34a}$$

$$(W^{-T}s) \circ (Wz) = \mu \mathbf{e}, \quad \kappa\tau = \mu. \tag{34b}$$

## 7.1 Outline

We start at initial values $(\hat{s}, \hat{\kappa}, \hat{x}, \hat{y}, \hat{z}, \hat{\tau}) = (s_0, 1, x_0, y_0, z_0, 1)$, where $s_0 \succ 0$, $z_0 \succ 0$, and define $(q_x, q_y, q_z, q_\tau)$ as in (28). We also compute the Nesterov-Todd scaling $W$ at $\hat{s}$, $\hat{z}$, and the scaled variable $\lambda := W^{-T}\hat{s} = W\hat{z}$.

1. *Evaluate residuals, gap, and stopping criteria.* Compute

$$\begin{bmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \hat{s} \\ \hat{\kappa} \end{bmatrix} - \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{\tau} \end{bmatrix} \tag{35}$$

   and

$$\hat{\mu} = \frac{\hat{s}^T\hat{z} + \hat{\kappa}\hat{\tau}}{m+1} = \frac{\lambda^T\lambda + \hat{\kappa}\hat{\tau}}{m+1}.$$

   Terminate if $(s, x, y, z) = (\hat{s}/\hat{\tau}, \hat{x}/\hat{\tau}, \hat{y}/\hat{\tau}, \hat{z}/\hat{\tau})$ satisfies (approximately) the optimality conditions (24), or $(\hat{z}, \hat{y})$ is an (approximate) certificate of primal infeasibility (25), or $(\hat{s}, \hat{x})$ is an (approximate) certificate of dual infeasibility (26).

2. *Affine direction.* Solve the linear equations

$$\begin{bmatrix} 0 \\ 0 \\ \Delta s_{\mathrm{a}} \\ \Delta \kappa_{\mathrm{a}} \end{bmatrix} - \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\mathrm{a}} \\ \Delta y_{\mathrm{a}} \\ \Delta z_{\mathrm{a}} \\ \Delta \tau_{\mathrm{a}} \end{bmatrix} = - \begin{bmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{bmatrix} \tag{36a}$$

$$\lambda \circ \left(W\Delta z_{\mathrm{a}} + W^{-T}\Delta s_{\mathrm{a}}\right) = -\lambda \circ \lambda, \qquad \hat{\kappa}\Delta\tau_{\mathrm{a}} + \hat{\tau}\Delta\kappa_{\mathrm{a}} = -\hat{\kappa}\hat{\tau}. \tag{36b}$$

3. *Step size and centering parameter.* Compute

$$\begin{aligned}
\alpha &= \sup\{\alpha \in [0,1] \mid (\hat{s}, \hat{\kappa}, \hat{z}, \hat{\tau}) + \alpha(\Delta s_{\mathrm{a}}, \Delta\kappa_{\mathrm{a}}, \Delta z_{\mathrm{a}}, \Delta\tau_{\mathrm{a}}) \succeq 0\} \\
&= \sup\left\{\alpha \in [0,1] \mid (\lambda, \hat{\kappa}, \lambda, \hat{\tau}) + \alpha(W^{-T}\Delta s_{\mathrm{a}}, \Delta\kappa_{\mathrm{a}}, W\Delta z_{\mathrm{a}}, \Delta\tau_{\mathrm{a}}) \succeq 0\right\} \\
\sigma &= \left(\frac{(\hat{s} + \alpha\Delta s_{\mathrm{a}})^T(\hat{z} + \alpha\Delta z_{\mathrm{a}})^T + (\hat{\kappa} + \alpha\Delta\kappa_{\mathrm{a}})(\hat{\tau} + \alpha\Delta\tau_{\mathrm{a}})}{\hat{s}^T\hat{z} + \hat{\kappa}\hat{\tau}}\right)^3 \\
&= (1-\alpha)^3. \qquad (37)
\end{aligned}$$

4. *Combined direction.* Solve the linear equation

$$\begin{bmatrix} 0 \\ 0 \\ \Delta s \\ \Delta\kappa \end{bmatrix} - \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{bmatrix} = -(1-\sigma) \begin{bmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{bmatrix} \qquad (38a)$$

$$\lambda \circ \left(W\Delta z + W^{-T}\Delta s\right) = -\lambda \circ \lambda - (W^{-T}\Delta s_{\mathrm{a}}) \circ (W\Delta z_{\mathrm{a}}) + \sigma\hat{\mu}\mathbf{e}, \qquad (38b)$$

$$\hat{\kappa}\Delta\tau + \hat{\tau}\Delta\kappa = -\hat{\kappa}\hat{\tau} - \Delta\kappa_{\mathrm{a}}\Delta\tau_{\mathrm{a}} + \sigma\hat{\mu}. \qquad (38c)$$

5. *Update iterates and scaling matrices.*

$$(\hat{s}, \hat{\kappa}, \hat{x}, \hat{y}, \hat{z}, \hat{\tau}) := (\hat{s}, \hat{\kappa}, \hat{x}, \hat{y}, \hat{z}, \hat{\tau}) + \alpha(\Delta s, \Delta\kappa, \Delta x, \Delta y, \Delta z, \Delta\tau)$$

where

$$\alpha = \sup\left\{\alpha \in [0,1] \;\middle|\; (\lambda, \hat{\kappa}, \lambda, \hat{\tau}) + \frac{\alpha}{0.99}(W^{-T}\Delta s, \Delta\kappa, W\Delta z, \Delta\tau) \succeq 0\right\}.$$

Compute the scaling matrix $W$ for $\hat{s}$, $\hat{z}$, and the scaled variable $\lambda := W^{-T}\hat{s} = W\hat{z}$.

### 7.2 Discussion

We discuss steps 2–4 in more detail. We first derive some useful properties of the affine scaling direction computed in step 2. The first equation in (36b) is equivalent to

$$\hat{s} \circ \Delta z_{\mathrm{a}} + \hat{z} \circ \Delta s_{\mathrm{a}} = -\hat{s} \circ \hat{z}$$

in unscaled coordinates. Taking the inner product with $\mathbf{e}$ on both sides gives

$$\hat{s}^T\Delta z_{\mathrm{a}} + \hat{z}^T\Delta s_{\mathrm{a}} = -\hat{s}^T\hat{z}, \qquad \hat{\kappa}\Delta\tau_{\mathrm{a}} + \hat{\tau}\Delta\kappa_{\mathrm{a}} = -\hat{\kappa}\hat{\tau}. \qquad (39)$$

Furthermore,

$$\begin{aligned}
\Delta z_{\rm a}^T \Delta s_{\rm a} + \Delta\tau_{\rm a}\Delta\kappa_{\rm a} &= -\begin{bmatrix} \Delta x_{\rm a} \\ \Delta y_{\rm a} \\ \Delta z_{\rm a} \\ \Delta\tau_{\rm a} \end{bmatrix}^T \begin{bmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{bmatrix} \\
&= -\begin{bmatrix} \Delta x_{\rm a} \\ \Delta y_{\rm a} \\ \Delta z_{\rm a} \\ \Delta\tau_{\rm a} \end{bmatrix}^T \left( \begin{bmatrix} 0 \\ 0 \\ \hat{s} \\ \hat{\kappa} \end{bmatrix} - \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{\tau} \end{bmatrix} \right) \\
&= -\hat{s}^T \Delta z_{\rm a} - \hat{\kappa}\Delta\tau_{\rm a} - \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{\tau} \end{bmatrix}^T \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\rm a} \\ \Delta y_{\rm a} \\ \Delta z_{\rm a} \\ \Delta\tau_{\rm a} \end{bmatrix} \\
&= -\hat{s}^T \Delta z_{\rm a} - \hat{\kappa}\Delta\tau_{\rm a} - \hat{z}^T\Delta s_{\rm a} - \hat{\tau}\Delta\kappa_{\rm a} + \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{\tau} \end{bmatrix}^T \begin{bmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{bmatrix} \\
&= -\hat{s}^T \Delta z_{\rm a} - \hat{\kappa}\Delta\tau_{\rm a} - \hat{z}^T\Delta s_{\rm a} - \hat{\tau}\Delta\kappa_{\rm a} + \hat{s}^T\hat{z} + \hat{\kappa}\hat{\tau} \\
&= 0. \qquad (40)
\end{aligned}$$

Lines 1 and 4 follow from (36a) and the skew-symmetry of the coefficient matrix. Line 6 follows from (39). Line 5 follows from the skew-symmetry of the coefficient matrix in the definition of the residuals (35):

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{\tau} \end{bmatrix}^T \begin{bmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{bmatrix} = \hat{s}^T\hat{z} + \hat{\kappa}\hat{\tau} = (m+1)\hat{\mu}. \qquad (41)$$

The simple expression for $\sigma$ in (37) follows by plugging in (39) and (40) in the definition.

The combined direction computed in step 4 has similar properties. From (38b) and (38c) we see that

$$\begin{aligned}
\hat{s}^T\Delta z + \hat{\kappa}\Delta\tau + \hat{z}^T\Delta s + \hat{\tau}\Delta\kappa &= -\hat{s}^T\hat{z} - \hat{\kappa}\hat{\tau} - \Delta s_{\rm a}^T\Delta z_{\rm a} - \Delta\kappa_{\rm a}\Delta\tau_{\rm a} + \sigma\hat{\mu}(m+1) \\
&= -(1-\sigma)(\hat{s}^T\hat{z} + \hat{\kappa}\hat{\tau}) \qquad (42)
\end{aligned}$$

and

$$
\begin{aligned}
\Delta z^T \Delta s + \Delta\tau\Delta\kappa &= -(1-\sigma) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{bmatrix}^T \begin{bmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{bmatrix} \\
&= -(1-\sigma) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{bmatrix}^T \left( \begin{bmatrix} 0 \\ 0 \\ \hat{s} \\ \hat{\kappa} \end{bmatrix} - \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{\tau} \end{bmatrix} \right) \\
&= -(1-\sigma) \left( \hat{s}^T \Delta z + \hat{\kappa}\Delta\tau + \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{\tau} \end{bmatrix}^T \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{bmatrix} \right) \\
&= -(1-\sigma) \left( \hat{s}^T \Delta z + \hat{\kappa}\Delta\tau + \hat{z}^T \Delta s + \hat{\tau}\Delta\kappa + (1-\sigma) \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \\ \hat{\tau} \end{bmatrix}^T \begin{bmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{bmatrix} \right) \\
&= -(1-\sigma) \left( \hat{s}^T \Delta z + \hat{\kappa}\Delta\tau + \hat{z}^T \Delta s + \hat{\tau}\Delta\kappa + (1-\sigma)(\hat{s}^T \hat{z} + \hat{\kappa}\hat{\tau}) \right) \\
&= 0. \qquad (43)
\end{aligned}
$$

Next we show by induction that

$$(r_x, r_y, r_z, r_\tau) = \hat{\mu}(q_x, q_y, q_z, q_\tau)$$

at the beginning of each iteration. In the first iteration, this is true by definition of $(q_x, q_y, q_z, q_\tau)$. Suppose it is satisfied by the current iterates. Then

$$
\begin{aligned}
(r_x^+, r_y^+, r_z^+, r_\tau^+) &= (1-\alpha(1-\sigma))(r_x, r_y, r_z, r_\tau) \\
&= (1-\alpha(1-\sigma))\hat{\mu}(q_x, q_y, q_z, q_\tau) \\
&= \hat{\mu}^+(q_x, q_y, q_z, q_\tau),
\end{aligned}
$$

because, from (42) and (43), $\hat{\mu}^+ = (1-\alpha(1-\sigma))\hat{\mu}$. Using this property, we can write (38a) as

$$
\begin{bmatrix} 0 \\ 0 \\ \Delta s \\ \Delta\kappa \end{bmatrix} - \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{bmatrix} = \sigma\hat{\mu} \begin{bmatrix} q_x \\ q_y \\ q_z \\ q_\tau \end{bmatrix} - \begin{bmatrix} r_x \\ r_y \\ r_z \\ r_\tau \end{bmatrix}.
$$

This shows that (38a) can be interpreted as the central path equation (34a) with $(s, \kappa, x, y, z, \tau)$ replaced by $(\hat{s} + \Delta s, \hat{\kappa} + \Delta\kappa, \hat{x} + \Delta x, \hat{y} + \Delta y, \hat{z} + \Delta z, \hat{\tau} + \Delta\tau)$, and $\mu = \sigma\hat{\mu}$. Making the same substitution in the nonlinear central path equations (34b) gives

$$(W^{-T}(\hat{z} + \Delta z)) \circ (W(\hat{s} + \Delta s)) = \sigma\hat{\mu}\,\mathbf{e}, \qquad (\hat{\kappa} + \Delta\kappa)(\hat{\tau} + \Delta\tau) = \sigma\hat{\mu}.$$

Expanding the products, using $W^{-T}\hat{z} = W\hat{s} = \lambda$, and using the Mehrotra correction to approximate the second-order terms as

$$(W^{-T}\Delta s) \circ (W\Delta z) \approx (W^{-T}\Delta s_{\mathrm{a}}) \circ (W\Delta z_{\mathrm{a}}), \qquad \Delta\tau\Delta\kappa \approx \Delta\kappa_{\mathrm{a}}\Delta\tau_{\mathrm{a}}$$

gives (38b) and (38c).

In summary, we see that in step 4 a search direction is computed by linearizing the central path equations (34) around the current iterates with $\mu = \sigma\hat{\mu}$. Step 2 is the linearization for $\mu = 0$. Step 3 is a heuristic for choosing $\sigma$, based on the result for $\mu = 0$.

### 7.3 Initialization

If primal and dual starting points $\hat{x}$, $\hat{s}$, $\hat{y}$, $\hat{z}$ are not specified by the user, they are selected as follows. The initial primal variable $\hat{x}$ is the solution of the constrained least-squares problem

$$\begin{array}{ll} \text{minimize} & \|Gx - h\|_2^2 \\ \text{subject to} & Ax = b. \end{array}$$

The initial value of $\hat{s}$ is computed from the residual $\tilde{s} = G\hat{x} - h$, as

$$\hat{s} = \begin{cases} \tilde{s} & \alpha_{\mathrm{p}} < 0 \\ \tilde{s} + (1 + \alpha_{\mathrm{p}})\mathbf{e} & \text{otherwise} \end{cases}$$

where $\alpha_{\mathrm{p}} = \inf\{\alpha \mid \tilde{s} + \alpha\,\mathbf{e} \succeq 0\}$. The values $\hat{x}$, $\tilde{s}$ can be computed by solving the linear equation

$$\begin{bmatrix} 0 & A^T & G^T \\ A & 0 & 0 \\ G & 0 & -I \end{bmatrix} \begin{bmatrix} \hat{x} \\ y \\ -\tilde{s} \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ h \end{bmatrix}.$$

The initial dual variables $\hat{y}$, $\hat{z}$ are computed by solving a least-norm problem

$$\begin{array}{ll} \text{minimize} & \|z\|_2^2 \\ \text{subject to} & G^T z + A^T y + c = 0. \end{array}$$

If the solution is $\hat{y}$, $\tilde{z}$, then we use $\hat{y}$ as initial value of $y$, and

$$\hat{z} = \begin{cases} \tilde{z} & \alpha_{\mathrm{d}} < 0 \\ \tilde{z} + (1 + \alpha_{\mathrm{d}})\mathbf{e} & \text{otherwise,} \end{cases}$$

where $\alpha_{\mathrm{d}} = \inf\{\alpha \mid \tilde{z} + \alpha\,\mathbf{e} \succeq 0\}$, as the initial value of $z$. The least-norm problem is equivalent to the the linear equation

$$\begin{bmatrix} 0 & A^T & G^T \\ A & 0 & 0 \\ G & 0 & -I \end{bmatrix} \begin{bmatrix} x \\ \hat{y} \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} -c \\ 0 \\ 0 \end{bmatrix}.$$

### 7.4 Newton equations

The two linear equations in steps 2 and 4 differ only in the righthand side and are of the form

$$\begin{bmatrix} 0 \\ 0 \\ \Delta s \\ \Delta\kappa \end{bmatrix} - \begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & 0 & h \\ -c^T & -b^T & -h^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{bmatrix} = - \begin{bmatrix} d_x \\ d_y \\ d_z \\ d_\tau \end{bmatrix} \tag{44a}$$

$$\lambda \circ \left( W\Delta z + W^{-T}\Delta s \right) = -d_s, \qquad \hat\kappa \Delta\tau + \hat\tau \Delta\kappa = -d_\kappa. \tag{44b}$$

Eliminating $\Delta s$ and $\Delta\kappa$ from (44b) gives

$$\begin{bmatrix} 0 & A^T & G^T & c \\ -A & 0 & 0 & b \\ -G & 0 & W^TW & h \\ -c^T & -b^T & -h^T & \hat\kappa/\hat\tau \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta\tau \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \\ d_z - W^T(\lambda \diamond d_s) \\ d_\tau - d_\kappa/\hat\tau \end{bmatrix} \tag{45a}$$

$$\Delta s = -W^T \left( \lambda \diamond d_s + W\Delta z \right), \qquad \Delta\kappa = -(d_\kappa + \hat\kappa\Delta\tau)/\hat\tau. \tag{45b}$$

As in section 5.4, $u \diamond v$ denotes the inverse of $u \circ v$ taken as a linear function of $v$, *i.e.*, $W^T(\lambda \diamond d_s)$ is the solution fo $z \circ u = d_s$. For the affine scaling Newton equation (step 2), the righthand side of (45a) simplifies to

$$d_z - W^T(\lambda \diamond d_s) = r_z - W^T\lambda = r_z - \hat s, \qquad d_\tau - d_\kappa/\hat\tau = r_\tau - \hat\kappa.$$

To solve the $4 \times 4$ block system (45) we solve two KKT systems

$$\begin{bmatrix} 0 & A^T & G^T \\ -A & 0 & 0 \\ -G & 0 & W^TW \end{bmatrix} \begin{bmatrix} x^{(1)} \\ y^{(1)} \\ z^{(1)} \end{bmatrix} = - \begin{bmatrix} c \\ b \\ h \end{bmatrix} \tag{46}$$

and

$$\begin{bmatrix} 0 & A^T & G^T \\ -A & 0 & 0 \\ -G & 0 & W^TW \end{bmatrix} \begin{bmatrix} x^{(2)} \\ y^{(2)} \\ z^{(2)} \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \\ d_z - W^T(\lambda \diamond d_s) \end{bmatrix}, \tag{47}$$

and make a linear combination with

$$\begin{aligned} \Delta\tau &= \frac{d_\tau - d_\kappa/\hat\tau + c^Tx^{(2)} + b^Ty^{(2)} + h^Tz^{(2)}}{\hat\kappa/\hat\tau - c^Tx^{(1)} - b^Ty^{(1)} - h^Tz^{(1)}} \\ &= \frac{d_\tau - d_\kappa/\hat\tau + c^Tx^{(2)} + b^Ty^{(2)} + h^Tz^{(2)}}{\hat\kappa/\hat\tau + \|Wz^{(1)}\|_2^2} \end{aligned}$$

to get

$$x = x^{(2)} + \Delta\tau\, x^{(1)}, \qquad y = y^{(2)} + \Delta\tau\, y^{(1)}, \qquad z = z^{(2)} + \Delta\tau\, z^{(1)}.$$

In summary, the main computation in one iteration of the algorithm is the solution of three equations with the same coefficient matrix (equation (46) and equation (47) for the two different righthand sides in steps 2 and 4).

## 8 Step length computation

Steps 3 and 5 of the algorithms require the computation of the maximum $\alpha$ such that

$$\lambda + \alpha\Delta\tilde{s} \succeq 0, \qquad \lambda + \alpha\Delta\tilde{z} \succeq 0,$$

where $\Delta\tilde{s} = W^{-T}\Delta s$ and $\Delta\tilde{z} = W\Delta z$. To facilitate the calculation we compute, for each cone,

$$\rho_k = H(\lambda_k)^{1/2}\Delta\tilde{s}_k, \qquad \sigma_k = H(\lambda_k)^{1/2}\Delta\tilde{z}_k.$$

The maximum step size is then $\alpha = \min_k \alpha_k$ where

$$\alpha_k = \sup\left\{\alpha \mid \mathbf{e}_k + \alpha\rho_k \succeq 0, \mathbf{e}_k + \alpha\sigma_k \succeq 0\right\}.$$

(This follows from $H(\lambda^{1/2})\lambda = \mathbf{e}$; see (10).)

### 8.1 Nonnegative orthant

For the nonnegative orthant $C_k = \mathbf{R}^p_+$,

$$\rho_k = \lambda_k^{-1} \circ \Delta\tilde{s}_k, \qquad \sigma_k = \lambda_k^{-1} \circ \Delta\tilde{z}_k, \qquad \alpha_k = \max\left\{0,\ -\min_i \rho_{ki},\ -\min_i \sigma_{ki}\right\}^{-1}.$$

### 8.2 Second-order cone

For the second-order cone $C_k = \mathcal{Q}_p$,

$$\begin{aligned}
\rho_k &= \frac{1}{(\lambda_k^T J \lambda_k)^{1/2}} \begin{bmatrix} \overline{\lambda}_k^T J \Delta\tilde{s}_k \\ \Delta\tilde{s}_{k1} - (\overline{\lambda}_k^T J \Delta\tilde{s}_k + \Delta\tilde{s}_{k0})(\overline{\lambda}_{k0}+1)^{-1}\overline{\lambda}_{k1} \end{bmatrix} \\
\sigma_k &= \frac{1}{(\lambda_k^T J \lambda_k)^{1/2}} \begin{bmatrix} \overline{\lambda}_k^T J \Delta\tilde{z}_k \\ \Delta\tilde{z}_{k1} - (\overline{\lambda}_k^T J \Delta\tilde{z}_k + \Delta\tilde{z}_{k0})(\overline{\lambda}_{k0}+1)^{-1}\overline{\lambda}_{k1} \end{bmatrix}
\end{aligned}$$

where $\overline{\lambda}_k = \lambda_k/(\lambda_k^T J \lambda_k)^{1/2}$. The maximum step size is

$$\alpha_k = \max\left\{0,\ \ \|\rho_{k1}\|_2 - \rho_{k0},\ \|\sigma_{k1}\|_2 - \sigma_{k0}\right\}^{-1}.$$

### 8.3 Semidefinite cone

For the semidefinite cone $C_k = \mathbf{S}^p_+$,

$$\rho_k = \mathbf{vec}(\Lambda_k^{-1/2}\Delta\tilde{S}_k\Lambda_k^{-1/2}), \qquad \sigma_k = \mathbf{vec}(\Lambda_k^{-1/2}\Delta\tilde{Z}_k\Lambda_k^{-1/2}),$$

where $\Lambda_k = \mathbf{mat}(\lambda_k)$, $\Delta\tilde{S}_k = \mathbf{mat}(\Delta\tilde{s}_k)$, $\Delta\tilde{Z}_k = \mathbf{mat}(\Delta\tilde{z}_k)$. We determine $\alpha_k$ by taking two eigenvalue decompositions

$$\mathbf{mat}(\rho_k) = Q_s\,\mathbf{diag}(\gamma_s)Q_s^T, \qquad \mathbf{mat}(\sigma_k) = Q_z\,\mathbf{diag}(\gamma_z)Q_z^T.$$

The maximum step size is

$$\alpha_k = \max\left\{0, -\min_i \gamma_{si}, -\min_i \gamma_{zi}\right\}^{-1}.$$

# 9 Updating the scaling matrix

At the end of each iteration, we update the scaling point, scaling matrix, and scaled variables. The current scaling point $w$ and scaling $W$ satisfy

$$H(w)\hat{s} = \hat{z}, \qquad W\hat{z} = W^{-T}\hat{s} = \lambda.$$

We need to compute a scaling point $w^+$ and scaling $W^+$ such that

$$H(w^+)(\hat{s} + \alpha\Delta s) = \hat{z} + \alpha\Delta z, \qquad W^+(\hat{z} + \alpha\Delta z) = (W^+)^{-T}(\hat{s} + \alpha\Delta s) = \lambda^+.$$

This can be achieved as follows. We first compute the scaling point $q$ for the scaled coordinates:

$$H(q)\tilde{s}^+ = \tilde{z}^+, \qquad \tilde{s}^+ = \lambda + \alpha\Delta\tilde{s}, \qquad \tilde{z}^+ = \lambda + \alpha\Delta\tilde{z}.$$

The new scaling point is $w^+ = W^T q$. This follows from (16):

$$\begin{aligned} H(W^T q)^{-1}(\hat{z} + \alpha\Delta z) &= W^T H(q)^{-1} W(\hat{z} + \alpha\Delta z) \\ &= W^T H(q)^{-1}(\lambda + \alpha\Delta\tilde{z}) \\ &= W^T(\lambda + \alpha\Delta\tilde{s}) \\ &= \hat{s} + \alpha\Delta s. \end{aligned}$$

## 9.1 Nonnegative orthant

If $C_k = \mathbf{R}^p_+$, the update is straightforward:

$$\begin{aligned} w_k^+ &= (\lambda_k + \alpha\Delta\tilde{s}_k)^{1/2} \circ (\lambda_k + \alpha\Delta\tilde{z}_k)^{-1/2} \circ w_k \\ \lambda_k^+ &= (\lambda_k + \alpha\Delta\tilde{z}_k)^{1/2} \circ (\lambda_k + \alpha\Delta\tilde{s}_k)^{1/2}. \end{aligned}$$

## 9.2 Second-order cone

**Updated NT scaling point** If $C_k = \mathcal{Q}_p$, we compute the scaling point $q_k$ for the scaled variables, which satisfies

$$H_k(q_k)^{-1}\tilde{s}_k^+ = \tilde{z}_k^+, \tag{48}$$

as in section 4.2: $q_k = (q_k^T J q_k)^{1/2}\bar{q}_k$ where

$$q_k^T J q_k = \left(\frac{(\tilde{s}_k^+)^T J \tilde{s}_k^+}{(\tilde{z}_k^+)^T J \tilde{z}_k^+}\right)^{1/2}, \qquad \bar{q}_k = \frac{1}{2\gamma^+}\left(\bar{\tilde{s}}_k^+ + J\bar{\tilde{z}}_k^+\right), \qquad \gamma^+ = \left(\frac{1 + (\bar{\tilde{z}}_k^+)^T \bar{\tilde{s}}_k^+}{2}\right)^{1/2}$$

and $\bar{\tilde{z}}_k^+$ and $\bar{\tilde{s}}_k^+$ are the normalized scaled variables (in the current scaling)

$$\bar{\tilde{z}}_k^+ = \frac{1}{\left((\tilde{z}_k^+)^T J \tilde{z}_k^+\right)^{1/2}}\tilde{z}_k^+, \qquad \bar{\tilde{s}}_k^+ = \frac{1}{\left((\tilde{s}_k^+)^T J \tilde{s}_k^+\right)^{1/2}}\tilde{s}_k^+.$$

Note that

$$\bar{q}_k^T J \bar{q}_k = 1, \qquad \bar{q}_k^T \bar{\tilde{z}}_k^+ = \bar{q}_k^T J \bar{\tilde{s}}_k^+ = \gamma^+.$$

The new scaling point then follows as

$$w_k^+ = W_k^T q_k = \left((w_k^T J w_k)(q_k^T J q_k)\right)^{1/2}\left(2v_k v_k^T - J\right)\bar{q}_k.$$

**Updated scaling matrix** It follows that the parameters of the new scaling matrix

$$W_k^+ = \left((w_k^+)^T J w_k^+\right)^{1/2} \overline{W}_k^+, \qquad \overline{W}_k^+ = 2v_k^+(v_k^+)^T - J,$$

can be determined as follows.

1. The new scaling factor is

$$(w_k^+)^T J w_k^+ = (w_k^T J w_k)(q_k^T J q_k), \qquad q_k^T J q_k = \left(\frac{(\widetilde{s}_k^+)^T J \widetilde{s}_k^+}{(\widetilde{z}_k^+)^T J \widetilde{z}_k^+}\right)^{1/2}.$$

2. The unitary vector $v_k^+$, which defines the updated Householder transformation, is

$$v_k^+ := (\overline{w}_k^+)^{1/2} = \frac{1}{(2(\overline{w}_{k0}^+ + 1))^{1/2}}(\overline{w}_k^+ + \mathbf{e}_k)$$

with

$$\overline{w}_k^+ = (2v_k v_k^T - J)\bar{q}_k, \qquad \overline{q}_k = \frac{1}{2\gamma^+}\left(\bar{\tilde{s}}_k^+ + J\bar{\tilde{z}}_k^+\right), \qquad \gamma^+ = \left(\frac{1 + (\bar{\tilde{z}}_k^+)^T \bar{\tilde{s}}_k^+}{2}\right)^{1/2}.$$

**Updated scaled variable** The updated scaled variable

$$\lambda_k^+ = \left((\lambda_k^+)^T J (\lambda_k^+)\right)^{1/2} \overline{\lambda}_k^+$$

can be computed from the updated scaled variables $\tilde{s}_k^+$, $\tilde{z}_k^+$ as follows. The norm is easy to compute:

$$\begin{aligned}
(\lambda_k^+)^T J \lambda_k^+ &= \left((s_k^+)^T J s_k^+\right)^{1/2} \left((z_k^+)^T J z_k^+\right)^{1/2} \\
&= \left((\tilde{s}_k^+)^T W_k J W_k \tilde{s}_k^+\right)^{1/2} \left((\tilde{z}_k^+)^T W_k^{-1} J W_k^{-1} \tilde{z}_k^+\right)^{1/2} \\
&= \left((\tilde{s}_k^+)^T J \tilde{s}_k^+\right)^{1/2} \left((\tilde{z}_k^+)^T J \tilde{z}_k^+\right)^{1/2}.
\end{aligned}$$

The normalized vector $\overline{\lambda}_k^+$ is defined as

$$\overline{\lambda}_k^+ = \overline{W}_k^+ \overline{z}_k^+ = (\overline{W}_k^+)^{-1} \overline{s}_k^+ = J \overline{W}_k^+ J \overline{s}_k^+.$$

Its first component is

$$\overline{\lambda}_{k0}^+ = (\overline{w}_k^+)^T \overline{z}_k^+ = \overline{q}_k^T \overline{W}_k \overline{z}_k^+ = \overline{q}_k^T \bar{\tilde{z}}_k^+ = \gamma^+.$$

The rest follows from

$$(I - J)\overline{\lambda}_k^+ = \overline{W}_k^+ \left(\overline{z}_k^+ - J\overline{s}_k^+\right) = \overline{W}_k^+ \overline{W}_k^{-1} \left(\bar{\tilde{z}}_k^+ - J\overline{\tilde{s}_k}^+\right) = \overline{W}_k^+ J \overline{W}_k \left(J\bar{\tilde{z}}_k^+ - \bar{\tilde{s}}_k^+\right).$$

Define $u_k = \overline{\overline{s}}_k^+ - J\overline{\overline{z}}_k^+$. Using the fact that $\overline{q}_k^T J u_k = 0$, we get

$$
\begin{aligned}
2\overline{\lambda}_{k1}^+ &= -\left(\left(\frac{1}{\overline{w}_{k0}^+ + 1}(\overline{w}_k^+ + \mathbf{e}_k)(\overline{w}_k^+ + \mathbf{e}_k)^T - J\right) J\overline{W}_k u_k\right)_1 \\
&= -\frac{1}{\overline{w}_{k0}^+ + 1}\overline{w}_{k1}^+(\overline{W}_k\overline{q}_k + \mathbf{e}_k)^T J\overline{W}_k u_k + (\overline{W}_k u_k)_1 \\
&= -\frac{1}{\overline{w}_{k0}^+ + 1}\overline{w}_{k1}^+(\overline{q}_k^T J u_k + \mathbf{e}_k^T\overline{W}_k u_k) + (\overline{W}_k u_k)_1 \\
&= -\frac{\overline{w}_k^T u_k}{\overline{w}_{k0}^+ + 1}\overline{w}_{k1}^+ + (\overline{W}_k u_k)_1 \\
&= \left(\overline{W}_k\left(-\frac{\overline{w}_k^T u_k}{\overline{w}_{k0}^+ + 1}\overline{q}_k + u_k\right)\right)_1 \\
&= \left(\overline{W}_k\left(-\frac{2v_{k0}(v_k^T u_k) - u_{k0}}{\overline{w}_{k0}^+ + 1}\overline{q}_k + u_k\right)\right)_1 \\
\overline{\lambda}_{k1}^+ &= \left(\overline{W}_k\left(-\frac{v_{k0}(v_k^T u_k) - u_{k0}/2}{\overline{w}_{k0}^+ + 1}\overline{q}_k + \frac{1}{2}u_k\right)\right)_1 \\
&= \left(\overline{W}_k\left(\frac{1 - d/\gamma^+}{2}\overline{\overline{s}}_k - \frac{1 + d/\gamma^+}{2}J\overline{\overline{z}}_k\right)\right)_1
\end{aligned}
$$

where

$$
d = \frac{v_{k0}(v_k^T u_k) - u_{k0}/2}{\overline{w}_{k0}^+ + 1} = \frac{v_{k0}(v_k^T u_k) - u_{k0}/2}{2v_{k0}(v_k^T\overline{q}_k) - \overline{q}_{k0} + 1}.
$$

## 9.3 Semidefinite cone

If $C_k = \mathcal{S}_p$, we use the eigenvalue decompositions

$$
\Lambda_k^{-1/2}\Delta\tilde{S}_k\Lambda_k^{-1/2} = Q_s\,\mathbf{diag}(\gamma_s)Q_s^T, \qquad \Lambda_k^{-1/2}\Delta\tilde{Z}_k\Lambda_k^{-1/2} = Q_z\,\mathbf{diag}(\gamma_z)Q_z^T,
$$

where $\Lambda_k = \mathbf{mat}(\lambda_k)$, $\Delta\tilde{S}_k = \mathbf{mat}(\Delta\tilde{s}_k)$, $\Delta\tilde{Z}_k = \mathbf{mat}(\Delta\tilde{z}_k)$, to factor the new iterates in the old scaling coordinates as

$$
R_k^{-1}S_k^+R_k^{-T} = \Lambda_k + \alpha\Delta\tilde{S}_k = L_1L_1^T, \qquad R_k^T Z_k^+ R_k = \Lambda_k + \alpha\Delta\tilde{Z}_k = L_2L_2^T,
$$

with $L_1 = \Lambda_k^{1/2}Q_s(I + \alpha\,\mathbf{diag}(\gamma_s))^{1/2}$, $L_2 = \Lambda_k^{1/2}Q_z(I + \alpha\,\mathbf{diag}(\gamma_z))^{1/2}$. We then take an SVD

$$
L_2^T L_1 = U\Lambda_k^+ V^T.
$$

The scaling matrix that satisfies

$$
(R_k^+)^{-1}S_k^+(R_k^+)^{-T} = (R_k^+)^T Z_k^+ R_k^+ = \Lambda_k^+
$$

is given by

$$
R_k^+ = R_k L_1 V(\Lambda_k^+)^{-1/2} = R_k L_2^{-T} U(\Lambda_k^+)^{1/2}.
$$

Its inverse is

$$
(R_k^+)^{-1} = (\Lambda_k^+)^{1/2}V^T L_1^{-1}R_k^{-1} = (\Lambda_k^+)^{-1/2}U^T L_2^T R_k^{-1}.
$$

## 10 Linear equation solvers

Each iteration of the interior-point methods requires the solution of a small number (2 or 3) of linear equations

$$\begin{bmatrix} P & A^T & G^T \\ A & 0 & 0 \\ G & 0 & -W^TW \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \tag{49}$$

(with $P = 0$ for cone LPs). We refer to an equation of this form as a Karush-Kuhn-Tucker (KKT) system. In addition, if the problem includes second-order cone or semidefinite constraints, one step of iterative refinement is applied when solving (21) or (44). This increases the number of KKT systems solved per iteration by two. In this section we describe the default methods for solving the KKT system (49). These solvers do not exploit problem structure except, to a limited extent, sparsity. (However, CVXOPT allows the user to provide a 'custom' solver that exploits problem structure in the KKT equations of a particular cone program.)

### 10.1 Cholesky factorization

The equation (49) can be reduced to

$$\begin{bmatrix} P + G^TW^{-1}W^{-T}G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b_x + G^TW^{-1}W^{-T}b_z \\ b_y \end{bmatrix}. \tag{50}$$

From $x$, $y$ the solution $z$ follows as $Wz = W^{-T}(Gx - b_z)$.

If $P + G^TW^{-1}W^{-T}G$ is nonsingular, we can solve (50) via a Cholesky factorization $P + G^TW^{-1}W^{-T}G = LL^T$. We solve

$$AL^{-T}L^{-1}A^Ty = AL^{-T}L^{-1}\left(b_x + G^TW^{-1}W^{-T}b_z\right) - b_y,$$

using a Cholesky factorization of $AL^{-T}L^{-1}A^T$ to obtain $y$, and then

$$LL^Tx = b_x + G^TW^{-1}W^{-T}b_z - A^Ty$$

to obtain $x$.

If $P + G^TW^{-1}W^{-T}TG$ is singular, we first write (50) as

$$\begin{bmatrix} P + G^TW^{-1}W^{-T}G + A^TA & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b_x + G^TW^{-1}W^{-T}b_z + A^Tb_y \\ b_y \end{bmatrix}$$

We compute the Cholesky factorization $P + G^TW^{-1}W^{-T}G + A^TA = LL^T$, and solve

$$AL^{-T}L^{-1}A^Ty = AL^{-T}L^{-1}\left(b_x + G^TW^{-1}W^{-T}b_z + A^Tb_y\right) - b_y,$$

using a Cholesky factorization of $AL^{-T}L^{-1}A^T$ to obtain $y$, and then

$$LL^Tx = b_x + G^TW^{-1}W^{-T}b_z + A^T(b_y - y)$$

to obtain $x$.

The Cholesky factorization method is the default KKT equation solver for linear programs and quadratic programs (*i.e.*, cone LPs and cone QPs without second-order cone or semidefinite constraints). The CHOLMOD sparse Cholesky factorization algorithms are used for factoring sparse matrices and the LAPACK algorithm for factoring dense matrices. No attempts are made to separate $G$ and $A$ in dense and sparse submatrices and to exploit such structure. (For large sparse problems replacing the equalities by two inequalities $As \preceq b$, $Ax \succeq b$ may therefore be faster.)

## 10.2 Two QR factorizations

This method is the default method for cone LPs with second-order cone or semidefinite constraints. We write the KKT system (with $P = 0$) as

$$\begin{bmatrix} 0 & A^T & \tilde{G}^T \\ A & 0 & 0 \\ \tilde{G} & 0 & -I \end{bmatrix} \begin{bmatrix} x \\ y \\ Wz \end{bmatrix} = \begin{bmatrix} b_x \\ b_y \\ W^{-T} b_z \end{bmatrix} \tag{51}$$

where $\tilde{G} = W^{-T} G$. To solve this we use two QR factorizations, of $A^T$ and $\tilde{G} Q_2$,

$$A^T = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \qquad \tilde{G} Q_2 = Q_3 R_3. \tag{52}$$

The solution $x$, $y$, $Wz$ is computed in the following steps:

$$\begin{aligned} w &= W^{-T} b_z - \tilde{G} Q_1 R_1^{-T} b_y \\ u &= R_3^{-T} Q_2^T b_x + Q_3^T w \\ Wz &= Q_3 u - w \\ y &= R_1^{-1} \left( Q_1^T b_x - Q_1^T \tilde{G}^T (Wz) \right) \\ x &= Q_1 R_1^{-T} b_y + Q_2 R_3^{-1} u. \end{aligned}$$

To verify this, we first use the QR factorization of $A^T$ to write (51) as

$$\begin{bmatrix} 0 & 0 & R_1 & Q_1^T \tilde{G}^T \\ 0 & 0 & 0 & Q_2^T \tilde{G}^T \\ R_1^T & 0 & 0 & 0 \\ \tilde{G} Q_1 & \tilde{G} Q_2 & 0 & -I \end{bmatrix} \begin{bmatrix} Q_1^T x \\ Q_2^T x \\ y \\ Wz \end{bmatrix} = \begin{bmatrix} Q_1^T b_x \\ Q_2^T b_x \\ b_y \\ W^{-T} b_z \end{bmatrix}. \tag{53}$$

From the third equation, we have $Q_1^T x = R_1^{-T} b_y$. The three remaining equations in the variables $Q_2^T x$, $y$, $Wz$ are

$$\begin{aligned} R_1 y &= Q_1^T b_x - Q_1^T \tilde{G}^T (Wz) \\ Q_2^T \tilde{G}^T (Wz) &= Q_2^T b_x \\ Wz &= \tilde{G} Q_1 (Q_1^T x) + \tilde{G} Q_2 (Q_2^T x) - W^{-T} b_z \\ &= Q_3 R_3 Q_2^T x - w \end{aligned}$$

if we define $w = W^{-T} b_z - \tilde{G} Q_1 R_1^{-T} b_y$. Multiplying the last equation on the left with $R_3^T Q_3^T = Q_2^T \tilde{G}^T$ and using the second equation gives an equation in $Q_2^T x$:

$$R_3^T R_3 (Q_2^T x) = Q_2^T b_x + R_3^T Q_3^T w.$$

The LAPACK dense QR factorization routines are used for the factorizations (52), so no sparsity in $A$ or $G$ is exploited.

### 10.3 QR factorization and Cholesky factorization

The third method is the default method for cone QPs with second-order or semidefinite constraints. We write the KKT system as

$$\begin{bmatrix} P & A^T & \tilde{G}^T \\ A & 0 & 0 \\ \tilde{G} & 0 & -I \end{bmatrix} \begin{bmatrix} x \\ y \\ Wz \end{bmatrix} = \begin{bmatrix} b_x \\ b_y \\ W^{-T} b_z \end{bmatrix} \tag{54}$$

with $\tilde{G} = W^{-T} G$. We use a QR factorization of $A^T$ to eliminate the equality constraints and a Cholesky factorization of size $n - p$ to solve the remaining problem

$$A^T = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}, \qquad Q_2^T (P + \tilde{G}^T \tilde{G}) Q_2 = LL^T. \tag{55}$$

We can use the QR factorization to write (54) as

$$\begin{bmatrix} Q_1^T P Q_1 & Q_1^T P Q_2 & R_1 & Q_1^T \tilde{G}^T \\ Q_2^T P Q_1 & Q_2^T P Q_2 & 0 & Q_2^T \tilde{G}^T \\ R_1^T & 0 & 0 & 0 \\ \tilde{G} Q_1 & \tilde{G} Q_2 & 0 & -I \end{bmatrix} \begin{bmatrix} Q_1^T x \\ Q_2^T x \\ y \\ Wz \end{bmatrix} = \begin{bmatrix} Q_1^T b_x \\ Q_2^T b_x \\ b_y \\ W^{-T} b_z \end{bmatrix}. \tag{56}$$

Eliminating $Wz$ gives

$$\begin{bmatrix} Q_1^T (P + \tilde{G}^T \tilde{G}) Q_1 & Q_1^T (P + \tilde{G}^T \tilde{G}) Q_2 & R_1 \\ Q_2^T (P + \tilde{G}^T \tilde{G}) Q_1 & Q_2^T (P + \tilde{G}^T \tilde{G}) Q_2 & 0 \\ R_1^T & 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1^T x \\ Q_2^T x \\ y \end{bmatrix} = \begin{bmatrix} Q_1^T (b_x + \tilde{G}^T W^{-T} b_z) \\ Q_2^T (b_x + \tilde{G}^T W^{-T} b_z) \\ b_y \end{bmatrix}.$$

From the third equation, $Q_1^T x = R_1^{-T} b_y$. From the second equation and the Cholesky factorization of $Q_2^T (P + \tilde{G}^T \tilde{G}) Q_2$ we can solve for $Q_2^T x$. From the first equation we solve for $y$.

The LAPACK routines are used for the QR and Cholesky factorizations (55).

## References


[AG03] F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming Series B*, 95:3–51, 2003.

[ART03] E. D. Andersen, C. Roos, and T. Terlaky. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(2):249–277, 2003.

[dKRT97] E. de Klerk, C. Roos, and T. Terlaky. Initialization in semidefinite programming via a self-dual skew-symmetric embedding. *Operations Research Letters*, 20(5):213–221, 1997.

[Meh92] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, November 1992.

[NT97] Yu. E. Nesterov and M. J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22(1):1–42, 1997.

[NT98] Yu. E. Nesterov and M. J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization*, 8(2):324–364, May 1998.

[RS88] C. M. Rader and A. O. Steinhardt. Hyperbolic Householder transforms. *SIAM Journal on Matrix Analysis and Applications*, 9(2):269–290, 1988.

[Stu00] J. F. Sturm. Similarity and other spectral relations for symmetric cones. *Linear Algebra and Its Applications*, 312:135–154, 2000.

[Stu02] J. F. Sturm. Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. *Optimization Methods and Software*, 17(6):1105–1154, 2002.

[Stu03] J. F. Sturm. Avoiding numerical cancellation in the interior point method for solving semidefinite programs. *Mathematical Programming Series B*, 95:219–247, 2003.

[Tsu99] T. Tsuchiya. A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming. *Optimization Methods and Software*, 11-12:141–182, 1999.

[TTT03] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming Series B*, 95:189–217, 2003.

[Tun98] L. Tunçel. Primal-dual symmetry and scale invariance of interior-point algorithms for convex optimization. *Mathematics of Operations Research*, 23(3):708–718, 1998.

[Wri97] S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, 1997.

[YTM94] Y. Ye, M. J. Todd, and S. Mizuno. An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research*, 19(1):53–67, 1994.