A logarithmic barrier interior-point method based on majorant functions for second-order cone programming

Baha Alzalg[†*]

[†]*Department of Mathematics, The University of Jordan, Amman 11942, Jordan*
Submission date: June 1, 2017

Abstract We present a logarithmic barrier interior-point method for solving a second-order cone programming problem. Newton's method is used to compute the descent direction, and majorant functions are used as an efficient alternative to line search methods to determine the displacement step along the direction. The efficiency of our method is shown by presenting numerical experiments.

Keywords Second order cone programming, interior-point methods, logarithmic barrier methods, majorant functions, Jordan algebras
AMS Classification 90C25, 90C46, 90C51, 17A15

1 Introduction

Interior-point methods [1–15] are one of the efficient methods developed to solve linear and nonlinear programming problems. Conic programming problems are a generalization of linear programming problems that include second-order cone programming problems and semidefinite programming problems as interesting special cases. Several interior-point methods have been developed for solving second-order cone programming (SOCP) problems. These interior-point methods can be classified into three main types: Smoothing-type methods [16–20], path-following methods [21, 22], and barrier/penalty methods [24, 25]. This paper presents an algorithm which based on the last type of interior-point methods for solving SOCP problems.

In this paper, we particularly propose a logarithmic barrier interior-point method for solving SOCP problems. In fact, the main difficulty to be anticipated in establishing an iteration in such a method will come from the determination and computation of the displacement step. Various approaches are developed to overcome this difficulty. It is known [1, 2] that the computation of the displacement step is expensive specifically while using line search methods. Crouzeix and Merikhi [2] proposed efficient and less expensive procedures in semidefinite programming not only to avoid line search methods, but also to accelerate the algorithm's convergence. Mennichea and Benterki [1] have utilized this idea for linear programming problems. The purpose of this paper is to exploit this idea for SOCP problems.

We introduce some notations used to write the primal and dual forms of the SOCP problem. We use "," for adjoining scalars, vectors and matrices in a row, and use ";" for adjoining them in a column. For example, a vector $\boldsymbol{u} \in \mathbb{R}^n$ can be written as $\boldsymbol{u} = (u_1; u_2; \ldots; u_n)$. For each vector $\boldsymbol{v} \in \mathbb{R}^n$

*Email: `b.alzalg@ju.edu.jo`
The work of the author was supported in part by the Deanship of Scientific Research at the University of Jordan.

whose first entry is indexed with 0, we write $\widetilde{v}$ for the subvector consisting of entries 1 through $n-1$; therefore $v = (v_0; \widetilde{v}) \in \mathbb{R} \times \mathbb{R}^{n-1}$. We let $\mathcal{E}^n$ denote the n^{th}-dimensional real vector space $\mathbb{R} \times \mathbb{R}^{n-1}$ whose elements v are indexed from 0. The n^{th}-dimensional *second-order cone* is defined as

$$\mathcal{E}_+^n := \{ v \in \mathcal{E}^n : v_0 \geq \|\widetilde{v}\| \},$$

where $\|\cdot\|$ denotes the Euclidean norm. We define the *determinant* of $w \in \mathcal{E}^n$ as $\det(w) := w_0^2 - \|\overline{w}\|^2$. Note that $w \in \mathcal{E}_+^n$ ($w \in \text{int}\, \mathcal{E}_+^n := \{ v \in \mathcal{E}^n : v_0 > \|\widetilde{v}\| \}$) if and only if $\det(w) \geq 0$ ($\det(w) > 0$).

Let $r \geq 1$ be an integer. For each $i = 1, 2, \ldots, r$, let m, n, n_i be positive integers such that $n = \sum_{i=1}^r n_i$. Let also x, c and z be vectors in $\mathbb{R}^n$, y and b be vectors in $\mathbb{R}^m$, and A be a matrix in $\mathbb{R}^{m \times n}$ such that they are all conformally partitioned as

$$\begin{array}{lcllcl}
x & := & (x_1; x_2; \ldots; x_r), & \text{where } x_i & \in & \mathcal{E}^{n_i}; \\
s & := & (s_1; s_2; \ldots; s_r), & s_i & \in & \mathcal{E}^{n_i}; \\
c & := & (c_1; c_2; \ldots; c_r), & c_i & \in & \mathcal{E}^{n_i}; \\
A & := & (A_1, A_2, \ldots, A_r), & A_i & \in & \mathbb{R}^{m_i \times n_i},
\end{array}$$

for $i = 1, 2, \ldots, r$. The *SOCP problem and its dual* in multi-block structures are defined as

$$(P_i) \quad \begin{array}{ll} \max & c_1^\mathsf{T} x_1 + \cdots + c_r^\mathsf{T} x_r \\ \text{s.t.} & A_1 x_1 + \cdots + A_r x_r = b, \\ & x_i \in \mathcal{E}_+^{n_i}, \; i = 1, \ldots, r; \end{array} \qquad (D_i) \quad \begin{array}{ll} \min & b^\mathsf{T} y \\ \text{s.t.} & A_i{}^\mathsf{T} y - s_i = c_i, \; i = 1, \ldots, r, \\ & s_i \in \mathcal{E}_+^{n_i}, \; i = 1, \ldots, r. \end{array}$$

The pair (P_i, D_i) can be compactly rewritten as

$$(P) \quad \begin{array}{ll} \max & c^\mathsf{T} x \\ \text{s.t.} & Ax = b, \\ & x \in \mathcal{E}_{r+}^n; \end{array} \qquad (D) \quad \begin{array}{ll} \min & b^\mathsf{T} y \\ \text{s.t.} & A^\mathsf{T} y - s = c, \\ & s \in \mathcal{E}_{r+}^n, \end{array}$$

where $\mathcal{E}_{r+}^n$ denotes the Cartesian product of r second-order cones. More specifically, $\mathcal{E}_{r+}^n := \mathcal{E}_+^{n_1} \times \mathcal{E}_+^{n_2} \times \cdots \times \mathcal{E}_+^{n_r}$. Hence $\text{int}\, \mathcal{E}_{r+}^n := \text{int}\, \mathcal{E}_+^{n_1} \times \text{int}\, \mathcal{E}_+^{n_2} \times \cdots \times \text{int}\, \mathcal{E}_+^{n_r}$. Similarly, we define $\mathcal{E}_r^n$ as $\mathcal{E}_r^n := \mathcal{E}^{n_1} \times \mathcal{E}^{n_2} \times \cdots \times \mathcal{E}^{n_r}$.

Now, we make assumptions about the primal-dual pair (P, D). First, we define the following feasibility sets:

$$\begin{array}{lcllcl}
\mathcal{F}_P & := & \{ x \in \mathcal{E}_r^n : Ax = b, x \in \mathcal{E}_{r+}^n \}, & \mathcal{F}_P^\circ & := & \{ x \in \mathcal{E}_r^n : Ax = b, x \in \text{int}\, \mathcal{E}_{r+}^n \}; \\
\mathcal{F}_D & := & \left\{ y \in \mathbb{R}^m : A^\mathsf{T} y - c \in \mathcal{E}_{r+}^n \right\}, & \mathcal{F}_D^\circ & := & \left\{ y \in \mathbb{R}^m : A^\mathsf{T} y - c \in \text{int}\, \mathcal{E}_{r+}^n \right\}.
\end{array}$$

Assumption 1.1. The m rows of the matrix A are linearly independent.

Assumption 1.2. The sets $\mathcal{F}_P^\circ$ and $\mathcal{F}_D^\circ$ are nonempty.

Let $\mu > 0$ be a barrier parameter and $f_\mu : \mathbb{R}^m \longrightarrow]-\infty, +\infty]$ be a barrier function defined as

$$f_\mu(y) := \begin{cases} b^\mathsf{T} y - \mu \displaystyle\sum_{i=1}^r \ln \det \left(A_i^\mathsf{T} y_i - c_i \right) + 2r\mu \ln \mu, & \text{if } \; y \in \mathcal{F}_D^\circ; \\ +\infty, & \text{if } \; y \in \mathbb{R}^m / \mathcal{F}_D^\circ. \end{cases}$$

Then solving problem (D) is equivalent to solving the perturbed unconstrained optimization problems

$$\left(D_\mu \right) \quad \begin{array}{ll} \min & f_\mu(y) \\ \text{s.t.} & y \in \mathbb{R}^m. \end{array}$$

The focus of this paper is on solving Problem (D_μ). This paper is organized as follows. In Section 2, we review the Jordan algebraic structure of the second-order cone and give some preliminary results. In Section 3, after considering the existence and uniqueness of the optimal solution of Problem (D_μ), we show the convergence of Problem (D_μ) to Problem (D) in the sense that the optimal solution of (D_μ) approaches the optimal solution of (D) as $\mu \to 0$. In Section 4, we propose an interior-point algorithm for solving Problem (D_μ). Newton's method is applied to compute the descent direction by solving the nonlinear system resulted from the optimality conditions associated with (D_μ). As an effective and less expensive alternative to line search methods, the so-called majorant functions are used to determine the displacement step along the descent direction. In Section 5, we prove the performance of our method and show its efficiency by presenting numerical experiments. Section 6 contains some concluding remarks.

2 Background and preliminary results

This section provides the necessary background for the upcoming development. In Subsection 2.1, we review the basics of the Jordan algebra associated with the second-order cone. We refer the reader to [21, 26] for more details of this algebra. In Subsection 2.2, we review some statistical inequalities.

2.1 The Jordan algebraic structure of the second-order cone

It is known that the space $\mathcal{E}^n$ under the bilinear map $\circ : \mathcal{E}^n \times \mathcal{E}^n \longrightarrow \mathcal{E}^n$ defined as

$$\boldsymbol{x} \circ \boldsymbol{y} := \begin{bmatrix} \boldsymbol{x}^\mathsf{T}\boldsymbol{y} \\ x_0\bar{\boldsymbol{y}} + y_0\bar{\boldsymbol{x}} \end{bmatrix}$$

forms a Euclidean Jordan algebra (see [26] for definition) equipped with the standard inner product $\langle \boldsymbol{x}, \boldsymbol{y}\rangle := \boldsymbol{x}^\mathsf{T}\boldsymbol{y}$.

The *spectral decomposition* of $\boldsymbol{x}$ in $\mathcal{E}^n$ is a decomposition of $\boldsymbol{x}$ into *eigenvectors* ($\boldsymbol{c}_1(\boldsymbol{x})$ and $\boldsymbol{c}_2(\boldsymbol{x})$) together with *eigenvalues* ($\lambda_1(\boldsymbol{x})$ and $\lambda_2(\boldsymbol{x})$). Such a spectral decomposition can be obtained as follows:

$$\boldsymbol{x} = \underbrace{(x_0 + \|\tilde{\boldsymbol{x}}\|)}_{\lambda_1(\boldsymbol{x})} \underbrace{\left(\frac{1}{2}\right)\left(1; \frac{\tilde{\boldsymbol{x}}}{\|\tilde{\boldsymbol{x}}\|}\right)}_{\boldsymbol{c}_1(\boldsymbol{x})} + \underbrace{(x_0 - \|\tilde{\boldsymbol{x}}\|)}_{\lambda_2(\boldsymbol{x})} \underbrace{\left(\frac{1}{2}\right)\left(1; -\frac{\tilde{\boldsymbol{x}}}{\|\tilde{\boldsymbol{x}}\|}\right)}_{\boldsymbol{c}_2(\boldsymbol{x})}.$$

We define the *trace* and *determinant* of $\boldsymbol{x}$ as

$$\operatorname{trace}(\boldsymbol{x}) := \lambda_1(\boldsymbol{x}) + \lambda_2(\boldsymbol{x}) = 2x_0 \quad \text{and} \quad \det(\boldsymbol{x}) := \lambda_1(\boldsymbol{x})\lambda_2(\boldsymbol{x}) = x_0^2 - \|\tilde{\boldsymbol{x}}\|^2.$$

Note that $\boldsymbol{c}_1(\boldsymbol{x})$ and $\boldsymbol{c}_2(\boldsymbol{x})$ satisfy the properties $\boldsymbol{c}_1(\boldsymbol{x}) \circ \boldsymbol{c}_2(\boldsymbol{x}) = \boldsymbol{0}, \boldsymbol{c}_1^2(\boldsymbol{x}) = \boldsymbol{c}_1(\boldsymbol{x}), \boldsymbol{c}_2^2(\boldsymbol{x}) = \boldsymbol{c}_2(\boldsymbol{x})$, and $\boldsymbol{c}_1(\boldsymbol{x}) + \boldsymbol{c}_2(\boldsymbol{x}) = \boldsymbol{e}_n$, where $\boldsymbol{e}_n := (1; \boldsymbol{0}) \in \mathcal{E}^n$ is the *identity vector* of $\mathcal{E}^n$.

Any continuous function $f : \mathbb{R} \longrightarrow \mathbb{R}$ is defined on $\mathcal{E}^n$ as

$$f(\boldsymbol{x}) := f(\lambda_1(\boldsymbol{x}))\boldsymbol{c}_1(\boldsymbol{x}) + f(\lambda_2(\boldsymbol{x}))\boldsymbol{c}_2(\boldsymbol{x}).$$

In particular, $\boldsymbol{x}^n$ for $n \geq 2$, which is defined recursively as $\boldsymbol{x}^n := \boldsymbol{x} \circ \boldsymbol{x}^{n-1}$, can be redefined as

$$\boldsymbol{x}^n := \lambda_1^n(\boldsymbol{x})\boldsymbol{c}_1(\boldsymbol{x}) + \lambda_2^n(\boldsymbol{x})\boldsymbol{c}_2(\boldsymbol{x}).$$

Observe that $x^{-1} \circ x = e_n$. The vector x is called *invertible* if x^{-1} is defined, and *noninvertible* otherwise. Note that every positive definite element is invertible and its inverse is also positive definite.

The *Frobenius norm* of $x \in \mathcal{E}^n$ is defined as

$$\|x\|_F := \sqrt{\lambda_1^2(x) + \lambda_2^2(x)} = 2\|x\|.$$

Associated with each vector $x \in \mathcal{E}^n$, we define the *arrow-shaped matrix* $\text{Arw}(x) \in \mathbb{R}^{n\times n}$ to be such that $x \circ y = \text{Arw}(x)y$ for every $y \in \mathcal{E}^n$. This yields

$$\text{Arw}(x) := \begin{bmatrix} x_0 & \tilde{x}^\mathsf{T} \\ \tilde{x} & x_0 I \end{bmatrix}.$$

Note that $x \in \mathcal{E}^n_+$ ($x \in \text{int } \mathcal{E}^n_+$) if and only if $\text{Arw}(x)$ is positive semidefinite ($\text{Arw}(x)$ is positive definite). The *quadratic representation* of $x \in \mathcal{E}^n$, $Q_x : \mathcal{E}^n \longrightarrow \mathcal{E}^n$, is defined as

$$Q_x := 2\text{Arw}^2(x) - \text{Arw}(x^2) = \begin{bmatrix} \|x\|^2 & 2x_0\tilde{x}^\mathsf{T} \\ 2x_0\tilde{x} & \det(x)I + 2\tilde{x}\tilde{x}^\mathsf{T} \end{bmatrix}.$$

The following lemma gives expressions for the gradient and Hessian of the barrier function. The first statement in item 1 is taken from [26, Proposition III.4.2], and the first statement in item 2 is taken from [26, Proposition II.3.3]. The second statement in each item is obtained by applying chain rule.

Lemma 2.1. *Let x and u be two vectors of $\mathcal{E}^n$, and $y = y(x)$ be a function of x in $\mathcal{E}^n$. Then*

(i) $\nabla_x \ln\det x = x^{-1}$, *or* $\nabla_x(\ln\det x)[u] = (x^{-1})^\mathsf{T} u$, *provided that* $\det x$ *is positive (so x is invertible). More generally,* $\nabla_x \ln\det y = (\nabla_x y)^\mathsf{T} y^{-1}$, *provided that* $\det y$ *is positive.*

(ii) $\nabla_x x^{-1} = -Q(x^{-1})$, *or* $\nabla_x x^{-1}[u] = -Q(x^{-1})u$, *provided that x is invertible, and hence we have* $\nabla^2_{xx} \ln\det x = \nabla_x x^{-1} = -Q(x^{-1})$. *More generally,* $\nabla_x y^{-1} = -Q(y^{-1})\nabla_x y$ *provided that y is invertible.*

All the above notions are also used in the block sense as one can see in the following definition.

Definition 2.1. Let $x = (x_1; x_2; \ldots; x_r)$, $y = (y_1; y_2; \ldots; y_r)$, and $x_i, y_i \in \mathcal{E}^{n_i}$ for $i = 1, 2, \ldots, r$. Then

(i) $x \circ y := (x_1 \circ y_1; x_2 \circ y_2; \ldots; x_r \circ y_r)$.

(ii) $x^\mathsf{T} y := {x_1}^\mathsf{T} y_1 + {x_2}^\mathsf{T} y_2 + \cdots + {x_r}^\mathsf{T} y_r$.

(iii) $\text{Arw}(x) := \text{Arw}(x_1) \oplus \text{Arw}(x_2) \oplus \cdots \oplus \text{Arw}(x_r)$.

(iv) $Q_x := Q_{x_1} \oplus Q_{x_2} \oplus \cdots \oplus Q_{x_r}$.

(v) $e := (e_{n_1}; e_{n_2}; \ldots; e_{n_r})$ is the identity vector of $\mathcal{E}^n_r$.

(vi) $f(x) := (f(x_1); f(x_2); \ldots; f(x_r))$. In particular, $x^{-1} := ({x_1}^{-1}; {x_2}^{-1}; \ldots; {x_r}^{-1})$.

(vii) $\|x\|_F^2 := \|x_1\|_F^2 + \|x_2\|_F^2 + \cdots + \|x_r\|_F^2$.

Note that x has $2r$ eigenvalues (including multiplicities) comprised of the union of the eigenvalues of each x_i for $i = 1, 2, \ldots, r$.

2.2 Preliminary inequalities

Let $x_1, x_2, \dots, x_n \in \mathbb{R}$ be a sample of size n, then its *mean* $\bar{x}$ and its *standard deviation* $\sigma_{\boldsymbol{x}}$ are respectively defined as

$$\bar{x} := \frac{1}{n}\sum_{k=1}^{n} x_k \quad \text{and} \quad \sigma_{\boldsymbol{x}}^2 := \frac{1}{n}\sum_{k=1}^{n} x_k^2 - \bar{x}^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \bar{x})^2.$$

We have the following proposition.

Proposition 2.1. *Assume that $\boldsymbol{x} \in \mathbb{R}^n$, then we have*

$$\bar{x} - \sigma_{\boldsymbol{x}}\sqrt{n-1} \le \min_{1\le k\le n} x_k \le \bar{x} - \frac{\sigma_{\boldsymbol{x}}}{\sqrt{n-1}}, \quad \textit{and} \quad \bar{x} + \frac{\sigma_{\boldsymbol{x}}}{\sqrt{n-1}} \le \max_{1\le k\le n} x_k \le \bar{x} + \sigma_{\boldsymbol{x}}\sqrt{n-1}.$$

In particular, if $x_k > 0$ for all $k = 1, 2, \dots, n$, then we also have

$$n\ln\left(\bar{x} - \sigma_{\boldsymbol{x}}\sqrt{n-1}\right) \le A \le \sum_{k=1}^{n}\ln(x_k) \le B \le n\ln(\bar{x}), \quad \textit{where}$$

$$A = (n-1)\ln\left(\bar{x} + \frac{\sigma_{\boldsymbol{x}}}{\sqrt{n-1}}\right) + \ln\left(\bar{x} - \sigma_{\boldsymbol{x}}\sqrt{n-1}\right), \quad \textit{and} \quad B = (n-1)\ln\left(\bar{x} - \frac{\sigma_{\boldsymbol{x}}}{\sqrt{n-1}}\right) + \ln\left(\bar{x} + \sigma_{\boldsymbol{x}}\sqrt{n-1}\right).$$

The first statement in Proposition 2.1 is due to [27] and the second statement is due to [2, Theorem 5].

3 The theoretical aspects of Problem (D_μ)

In this section, we show that Problem (P_μ) has a unique optimal solution and that this optimal solution converges to the optimal solution of Problem (P) when μ goes to 0. First, we have the following definition.

Definition 3.1. Let g be a real-valued function defined on a metric space Y and $\alpha \ge 0$. Then

(i) The set $C_\alpha(g) := \{\boldsymbol{y} \in Y : g(\boldsymbol{y}) \le \alpha\}$ is called the *α-level set* of g.

(ii) The function g is called *inf-compact* if the level sets $C_\alpha(g)$ are compact for all $\alpha > 0$.

(iii) The *recession function* of g is the function $(g)_\infty : Y \longrightarrow \mathbb{R}$ defined by

$$(g)_\infty(\Delta\boldsymbol{y}) := \lim_{t\to+\infty}\frac{g(\boldsymbol{y} + t\Delta\boldsymbol{y}) - g(\boldsymbol{y})}{t}.$$

(iv) The *recession cone* of g is the 0-level set of the recession function of g, denoted by $C_0\left((g)_\infty\right)$.

Proving that Problem (D_μ) has an optimal solution is equivalent to proving that f_μ is inf-compact, this in turns equivalent to showing that the following recession cone

$$C_0\left(\left(f_\mu\right)_\infty\right) := \left\{\boldsymbol{d} \in \mathbb{R}^m : \left(f_\mu\right)_\infty(\boldsymbol{d}) \le 0\right\}$$

is reduced to the origin. That is, $(f_\mu)_\infty(\boldsymbol{d}) \le 0$ implies that $\boldsymbol{d} = \boldsymbol{0}$.

From Problem (D), we can write $\boldsymbol{s}(\boldsymbol{y}) := A^{\mathsf{T}}\boldsymbol{y} - \boldsymbol{c}$. Using the notations introduced in Section 2, the function $f_{\mu}(\cdot)$ can be rewritten as

$$f_{\mu}(\boldsymbol{y}) := \begin{cases} \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} - \mu \ln\det \boldsymbol{s}(\boldsymbol{y}) + 2r\mu\ln\mu, & \text{if } \ \boldsymbol{s}(\boldsymbol{y}) \in \operatorname{int}\mathcal{E}^{n}_{r+}; \\ +\infty, & \text{if not.} \end{cases}$$

Let $\boldsymbol{h}(\boldsymbol{d}) := A^{\mathsf{T}}\boldsymbol{d} \in \operatorname{int}\mathcal{E}^{n}_{r+}$ and $\boldsymbol{\omega}(\boldsymbol{d}) := \boldsymbol{s}^{-1/2} \circ \left(\boldsymbol{h}(\boldsymbol{d}) \circ \boldsymbol{s}^{-1/2}\right)$ for $\boldsymbol{s}(\boldsymbol{y}) \in \operatorname{int}\mathcal{E}^{n}_{r+}$. Noting that $\boldsymbol{s}(\boldsymbol{y} + t\boldsymbol{d}) = \boldsymbol{s}(\boldsymbol{y}) + t\boldsymbol{h}(\boldsymbol{d})$, we have

$$\begin{aligned}
\left(f_{\mu}\right)_{\infty}(\boldsymbol{d}) &= \lim_{t\to+\infty} \frac{f_{\mu}(\boldsymbol{y}+t\boldsymbol{d}) - f_{\mu}(\boldsymbol{y})}{t} \\
&= \lim_{t\to+\infty} \frac{\boldsymbol{b}^{\mathsf{T}}(\boldsymbol{y}+t\boldsymbol{d}) - \mu\ln\det\boldsymbol{s}(\boldsymbol{y}+t\boldsymbol{d}) - \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} + \mu\ln\det\boldsymbol{s}(\boldsymbol{y})}{t} \\
&= \boldsymbol{b}^{\mathsf{T}}\boldsymbol{d} - \mu\lim_{t\to+\infty} \frac{\ln\det\boldsymbol{s}(\boldsymbol{y}+t\boldsymbol{d}) - \ln\det\boldsymbol{s}(\boldsymbol{y})}{t} \\
&= \boldsymbol{b}^{\mathsf{T}}\boldsymbol{d} - \mu\lim_{t\to+\infty} \frac{\ln\det\left(\boldsymbol{s}(\boldsymbol{y}) + t\boldsymbol{h}(\boldsymbol{d})\right) - \ln\det\boldsymbol{s}(\boldsymbol{y})}{t} \\
&= \boldsymbol{b}^{\mathsf{T}}\boldsymbol{d} - \mu\lim_{t\to+\infty} \frac{\ln\det\left(\boldsymbol{s}^{1/2}(\boldsymbol{y}) \circ \left((\boldsymbol{e} + t\boldsymbol{\omega}(\boldsymbol{d})) \circ \boldsymbol{s}^{1/2}(\boldsymbol{y})\right)\right) - \ln\det\boldsymbol{s}(\boldsymbol{y})}{t} \\
&= \boldsymbol{b}^{\mathsf{T}}\boldsymbol{d} - \mu\lim_{t\to+\infty} \frac{\ln\left(\det\boldsymbol{s}(\boldsymbol{y})\det\left(\boldsymbol{e} + t\boldsymbol{\omega}(\boldsymbol{d})\right)\right) - \ln\det\boldsymbol{s}(\boldsymbol{y})}{t} \\
&= \boldsymbol{b}^{\mathsf{T}}\boldsymbol{d} - \mu\lim_{t\to+\infty} \frac{\ln\det\left(\boldsymbol{e} + t\boldsymbol{\omega}(\boldsymbol{d})\right)}{t} \\
&= \boldsymbol{b}^{\mathsf{T}}\boldsymbol{d} - \mu\lim_{t\to+\infty} \sum_{i=1}^{2r} \frac{\ln\left(1 + t\lambda_i(\boldsymbol{\omega})\right)}{t} \\
&= \boldsymbol{b}^{\mathsf{T}}\boldsymbol{d}.
\end{aligned}$$

Hence, requiring the inequality $(f_{\mu})_{\infty}(\boldsymbol{d}) \leq 0$ to hold is equivalent to requiring that $\boldsymbol{b}^{\mathsf{T}}\boldsymbol{d} \leq 0$ and $\boldsymbol{h}(\boldsymbol{d}) \in \operatorname{int}\mathcal{E}^{n}_{r+}$ to hold. This leads us to consider the following lemma.

Lemma 3.1. *Let $\boldsymbol{d} \in \mathbb{R}^m$ be such that $\boldsymbol{b}^{\mathsf{T}}\boldsymbol{d} \leq 0$ and $\boldsymbol{h}(\boldsymbol{d}) = A^{\mathsf{T}}\boldsymbol{d} \in \operatorname{int}\mathcal{E}^{n}_{r+}$, then $\boldsymbol{d} = \boldsymbol{0}$.*

Proof. Suppose in the contrary that $\boldsymbol{d} \neq \boldsymbol{0}$. Based on Assumption 1.1, there exists $\boldsymbol{x} \in \operatorname{int}\mathcal{E}^{n}_{r+}$ such that $A\boldsymbol{x} = \boldsymbol{b}$. Let $\boldsymbol{h} := \boldsymbol{h}(\boldsymbol{d}) = A^{\mathsf{T}}\boldsymbol{d}$. It immediately follows that

$$\boldsymbol{b}^{\mathsf{T}}\boldsymbol{d} = \boldsymbol{x}^{\mathsf{T}}A^{\mathsf{T}}\boldsymbol{d} = \sum_{i=1}^{r} \boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{h}_i = \sum_{i=1}^{r}\left(x_{i0}h_{i0} + \widetilde{\boldsymbol{x}_i}^{\mathsf{T}}\widetilde{\boldsymbol{h}_i}\right) > \sum_{i=1}^{r}\left(\left\|\widetilde{\boldsymbol{x}_i}\right\|\left\|\widetilde{\boldsymbol{h}_i}\right\| + \widetilde{\boldsymbol{x}_i}^{\mathsf{T}}\widetilde{\boldsymbol{h}_i}\right) \geq \sum_{i=1}^{r}\left(\left|\widetilde{\boldsymbol{x}_i}\right|\left|\widetilde{\boldsymbol{h}_i}\right| + \widetilde{\boldsymbol{x}_i}^{\mathsf{T}}\widetilde{\boldsymbol{h}_i}\right) \geq 0,$$

where the strict inequality follows from the fact that $\boldsymbol{x}_i, \boldsymbol{h}_i \in \operatorname{int}\mathcal{E}^{n_i}_{+}$ for $i = 1, 2, \ldots, r$, and the first nonstrict inequality follows from Hölder's inequality. This obviously contradicts the hypothesis of the lemma on the nonpositivity of $\boldsymbol{b}^{\mathsf{T}}\boldsymbol{d}$. The proof is complete. □

From Lemma 3.1, we conclude that the function f_{μ} is inf-compact, which in turns concludes that Problem (D_{μ}) has an optimal solution.

Note that $\nabla_{\boldsymbol{y}}\boldsymbol{s}(\boldsymbol{y}) = \nabla_{\boldsymbol{y}}\left(A^{\mathsf{T}}\boldsymbol{y} - \boldsymbol{c}\right) = A^{\mathsf{T}}$. Then, using Lemma 2.1, we have

$$\nabla_{\boldsymbol{y}} f_{\mu}(\boldsymbol{y}) = \boldsymbol{b} - \mu\left(\nabla_{\boldsymbol{y}}\boldsymbol{s}(\boldsymbol{y})\right)^{\mathsf{T}}\boldsymbol{s}^{-1}(\boldsymbol{y}) = \boldsymbol{b} - \mu A\,\boldsymbol{s}^{-1}(\boldsymbol{y}),$$

and hence

$$H := \nabla^2_{\boldsymbol{y}\boldsymbol{y}} f_{\mu}(\boldsymbol{y}) = -\mu A\,\nabla_{\boldsymbol{y}}\boldsymbol{s}^{-1}(\boldsymbol{y}) = \mu A\,Q\left(\boldsymbol{s}^{-1}(\boldsymbol{y})\right)\nabla_{\boldsymbol{y}}\boldsymbol{s}(\boldsymbol{y}) = \mu A\,Q\left(\boldsymbol{s}^{-1}(\boldsymbol{y})\right)A^{\mathsf{T}}. \tag{1}$$

Since $s^{-1}(y) \in \text{int } (\mathcal{E}_+^n)$, the matrix $Q\left(s^{-1}(y)\right)$ is positive definite. By (1) and Assumption 1.1, the matrix $H = \nabla^2_{yy} f_\mu(y)$ is also positive definite. This implies that Problem (D_μ) is strictly convex and therefore concludes that it has a unique optimal solution.

Now, we show that Problem (D_μ) converges to Problem (D) as $\mu \longrightarrow 0$. We have the following lemma.

Lemma 3.2. *For $\mu > 0$, let Problem (D_μ) have y_μ as an optimal solution, then Problem (D) has $y^\star := \lim_{\mu\to 0} y_\mu$ as an optimal solution.*

Proof. Let $y \in \mathcal{F}_D^\circ$ be arbitrary and $\mu > 0$ be given. Let also $f(y) := b^\mathsf{T} y$ and define

$$\phi(y,\mu) := f_\mu(y) = \begin{cases} f(y) - \mu \ln\det s(y) + 2r\mu\ln\mu, & \text{if } s(y) \in \text{int } \mathcal{E}_{r+}^n; \\ +\infty, & \text{if not.} \end{cases}$$

Since the function $\phi(\cdot,\cdot)$ is differentiable at the point (y_μ, μ), we have

$$\begin{aligned} f(y) &= \phi(y, 0) \\ &\geq \phi(y_\mu,\mu) + \left(y - y_\mu\right)^\mathsf{T} \nabla_y \phi(y_\mu,\mu) + (0-\mu)\frac{\partial}{\partial_\mu}\phi(y_\mu,\mu) \\ &= \phi(y_\mu,\mu) - \mu\,\frac{\partial}{\partial_\mu}\phi(y_\mu,\mu) \\ &= f(y_\mu) - \mu\ln\det s(y_\mu) + 2r\mu\ln\mu - \mu\left(-\ln\det s(y_\mu) + 2r\ln\mu + 2r\right) \\ &= f(y_\mu) - 2\mu r, \end{aligned}$$

where the second equality follows from the fact that $\nabla_y \phi(y_\mu,\mu) = \nabla_y f_\mu(y_\mu) = \mathbf{0}$ because the point (y_μ,μ) is optimal. Since $y \in \mathcal{F}_D^\circ$ was arbitrary, we then have

$$f(y_\mu) - 2\mu r \leq \min_{y\in\mathcal{F}_D^\circ} f(y) \leq f(y_\mu).$$

Letting μ approach zero and $y^\star := \lim_{\mu\to 0} y_\mu$, we get

$$f\left(y^\star\right) = f\left(\lim_{\mu\to 0} y_\mu\right) = \lim_{\mu\to 0} f(y_\mu) = \min_{y\in\mathcal{F}_D^\circ} f(y).$$

The result is established. □

4 The numerical aspects of Problem (D_μ)

The focus of this section is on the numerical solution of Problem (D_μ). Note that y_μ is an optimal solution of Problem (D_μ) if it satisfies the equation

$$\nabla_y f_\mu(y_\mu) = \mathbf{0}. \tag{2}$$

We solve (2) by using the Newton's approach, which in turn means finding a vector $y_\mu^{(k)} + d^{(k)}$ at each iteration satisfying the system

$$H^{(k)} d^{(k)} = -\nabla_y f_\mu(y_\mu). \tag{3}$$

Since $H^{(k)} = \nabla^2_{yy} f_\mu(y_\mu)$ is positive definite, the most desirable methods for solving (3) are the Cholesky methods and the conjugate gradient methods.

We introduce a displacement step t_k satisfying the strict second-order cone constraint

$$A^{\top}\left(\boldsymbol{y}_{\mu}^{(k)}+t^{(k)}\boldsymbol{d}^{(k)}\right)-\boldsymbol{c}\in \operatorname{int}\mathcal{E}_{r+}^{n}.$$

This guarantees that all the iterate $\boldsymbol{y}_{\mu}^{(k)}+\boldsymbol{d}^{(k)}$ remains strictly feasible, which in turns guarantees the convergence of the algorithm to an optimal solution $\boldsymbol{y}^{\star}$ of (D_{μ}).

A prototype algorithm for solving Problem (D_{μ}) is formally stated in Algorithm 4.1. For the sake of simplicity we drop the index μ from $\boldsymbol{y}_{\mu}$ and $\boldsymbol{y}_{\mu}^{(k)}$, and write $\boldsymbol{y}$ instead of $\boldsymbol{y}_{\mu}$ and $\boldsymbol{y}^{(k)}$ instead of $\boldsymbol{y}_{\mu}^{(k)}$.

Algorithm 4.1. Prototype algorithm for solving Problem (D_{μ})

Begin algorithm
1: Initialize $k=0$, $\boldsymbol{y}^{(0)}$, $\boldsymbol{d}^{(0)}$, ϵ, $k=0$
Ensure $\boldsymbol{y}^{(0)}$ is a strictly feasible solution of (D), $\boldsymbol{d}^{(0)}\in\mathbb{R}^{m}$
2: **While** $\left|\boldsymbol{b}^{\top}\boldsymbol{d}^{(k)}\right|>\epsilon$ **do**
3: Solve the system $H^{(k)}\boldsymbol{d}^{(k)}=-\nabla f_{\mu}(\boldsymbol{y}^{(k)})$
4: Compute the displacement step $t^{(k)}$
5: Set $\boldsymbol{y}^{(k+1)}=\boldsymbol{y}^{(k)}+t^{(k)}\boldsymbol{d}^{(k)}$
6: Set $k=k+1$
7: **End while**
End algorithm

The most known methods used to compute the optimal displacement step $t^{(k)}$ are the line search methods, which require minimizing the unidimensional function

$$\phi(t):=\min_{t>0} f_{\mu}(\boldsymbol{y}+t\boldsymbol{d}).$$

It was reported [1,2] that the line search methods are computationally expensive, and even are inapplicable in the case of semidefinite programing. This difficulty can be overcome by exploiting a simple and much less expensive method than line search methods. Such a method firstly proposed by Crouzeix and Merikhi [2] for semidefinite programming, and then utilized by Mennichea and Benterki [1] for linear programming. In this method, the function $\theta(t)$ defined as

$$\theta(t):=\frac{1}{\mu}(f_{\mu}(\boldsymbol{y}+t\boldsymbol{d})-f_{\mu}(\boldsymbol{y}))$$

is approached by the simple majorant function producing the optimal displacement step t_k at each iteration k. The following remark tells us what exactly keeps the function $\theta(t)$ well-defined.

Remark 4.1. *It is necessary that the point $\boldsymbol{y}+t\boldsymbol{d}$ still in $\mathcal{F}_{D}^{\circ}$ for all $\boldsymbol{y}\in\mathcal{F}_{D}^{\circ}$ to keep function $\theta(t)$ well defined. This in turns requires finding $\hat{t}>0$ such that $\boldsymbol{y}+t\boldsymbol{d}\in\mathcal{F}_{D}^{\circ}$ for any $t\in[0,\hat{t}]$.*

Recall that we defined $\boldsymbol{\omega}$ as $\boldsymbol{\omega}:=\boldsymbol{\omega}(\boldsymbol{d})=\boldsymbol{s}^{-1/2}\circ\left(\boldsymbol{h}(\boldsymbol{d})\circ\boldsymbol{s}^{-1/2}\right)$ where $\boldsymbol{s}(\boldsymbol{y})=A^{\top}\boldsymbol{y}-\boldsymbol{c}\in\operatorname{int}\mathcal{E}_{r+}^{n}$, and $\boldsymbol{h}(\boldsymbol{d})=A^{\top}\boldsymbol{d}\in\operatorname{int}\mathcal{E}_{r+}^{n}$. Let $\lambda_1(\boldsymbol{\omega}),\lambda_2(\boldsymbol{\omega}),\ldots,\lambda_{2r}(\boldsymbol{\omega})$ be the $2r$ eigenvalues of $\boldsymbol{\omega}$. We have the following lemma.

Lemma 4.1. *Let* $\hat{t} = \sup\{t : 1 + t\lambda_i(\boldsymbol{\omega}) > 0, i = 1, 2, \ldots, 2r\}$*. For all* $t \in [0, \hat{t}]$*, the following function* $\theta(t)$ *is well defined*

$$\theta(t) = \sum_{i=1}^{2r} \left(t \left(\lambda_i(\boldsymbol{\omega}) - \lambda_i^2(\boldsymbol{\omega}) \right) - \ln\left(1 + t\lambda_i(\boldsymbol{\omega})\right) \right), \ t \in [0, \hat{t}[.$$

Proof. Note that $\boldsymbol{s}(\boldsymbol{y} + t\boldsymbol{d}) = \boldsymbol{s}(\boldsymbol{y}) + t\boldsymbol{h}(\boldsymbol{d})$. We then have

$$\begin{aligned}
\theta(t) &= \frac{1}{\mu}\left(f_\mu(\boldsymbol{y} + t\boldsymbol{d}) - f_\mu(\boldsymbol{y}) \right) \\
&= \frac{1}{\mu}\left(\boldsymbol{b}^\top(\boldsymbol{y} + t\boldsymbol{d}) - \mu \ln\det \boldsymbol{s}(\boldsymbol{y} + t\boldsymbol{d}) - \boldsymbol{b}^\top\boldsymbol{y} + \mu\ln\det\boldsymbol{s}(\boldsymbol{y}) \right) \\
&= \frac{1}{\mu}\left(t\boldsymbol{b}^\top\boldsymbol{d} - \mu\ln\det\boldsymbol{s}(\boldsymbol{y} + t\boldsymbol{d}) + \mu\ln\det\boldsymbol{s}(\boldsymbol{y}) \right) \\
&= \frac{1}{\mu}\left(t\boldsymbol{b}^\top\boldsymbol{d} - \mu\ln\det\left(\boldsymbol{s}(\boldsymbol{y}) + t\boldsymbol{h}(\boldsymbol{d})\right) + \mu\ln\det\boldsymbol{s}(\boldsymbol{y}) \right) \\
&= \frac{1}{\mu}\left(t\boldsymbol{b}^\top\boldsymbol{d} - \mu\ln\det\left(\boldsymbol{s}^{1/2}(\boldsymbol{y}) \circ \left((\boldsymbol{e} + t\boldsymbol{\omega}(\boldsymbol{d})) \circ \boldsymbol{s}^{1/2}(\boldsymbol{y}) \right)\right) + \mu\ln\det\boldsymbol{s}(\boldsymbol{y}) \right) \\
&= \frac{1}{\mu}\left(t\boldsymbol{b}^\top\boldsymbol{d} - \mu\ln\left(\det\boldsymbol{s}(\boldsymbol{y})\det\left(\boldsymbol{e} + t\boldsymbol{\omega}(\boldsymbol{d})\right)\right) + \mu\ln\det\boldsymbol{s}(\boldsymbol{y}) \right) \\
&= \frac{1}{\mu}\left(t\boldsymbol{b}^\top\boldsymbol{d} - \mu\ln\det\left(\boldsymbol{e} + t\boldsymbol{\omega}(\boldsymbol{d})\right) \right) \\
&= \frac{t}{\mu}\boldsymbol{b}^\top\boldsymbol{d} - \sum_{i=1}^{2r}\ln\left(1 + t\lambda_i(\boldsymbol{\omega})\right).
\end{aligned} \tag{4}$$

Since $\nabla_{\boldsymbol{y}} f_\mu(\boldsymbol{y}) = \boldsymbol{b} - \mu A\, \boldsymbol{s}^{-1}(\boldsymbol{y})$, we have

$$\boldsymbol{d}^\top\boldsymbol{b} = \boldsymbol{d}^\top\nabla_{\boldsymbol{y}} f_\mu(\boldsymbol{y}) + \mu\boldsymbol{d}^\top A\, \boldsymbol{s}^{-1}(\boldsymbol{y}). \tag{5}$$

Due to the fact that the direction $\boldsymbol{d}$ satisfies $\nabla^2_{\boldsymbol{yy}} f_\mu(\boldsymbol{y})\boldsymbol{d} = -\nabla_{\boldsymbol{y}} f_\mu(\boldsymbol{y})$, we have that

$$\boldsymbol{d}^\top\nabla^2_{\boldsymbol{yy}} f_\mu(\boldsymbol{y})\boldsymbol{d} = -\boldsymbol{d}^\top\nabla_{\boldsymbol{y}} f_\mu(\boldsymbol{y}). \tag{6}$$

Substituting (6) into (5), we get

$$\begin{aligned}
\boldsymbol{d}^\top\boldsymbol{b} &= -\boldsymbol{d}^\top\nabla^2_{\boldsymbol{yy}} f_\mu(\boldsymbol{y})\boldsymbol{d} + \mu\boldsymbol{d}^\top A\boldsymbol{s}^{-1}(\boldsymbol{y}) \\
&= -\boldsymbol{d}^\top\nabla^2_{\boldsymbol{yy}} f_\mu(\boldsymbol{y})\boldsymbol{d} + \mu(\boldsymbol{h}(\boldsymbol{d}))^\top\boldsymbol{s}^{-1}(\boldsymbol{y}) \\
&= -\boldsymbol{d}^\top\nabla^2_{\boldsymbol{yy}} f_\mu(\boldsymbol{y})\boldsymbol{d} + \frac{\mu}{2}\operatorname{trace}\left(\boldsymbol{h}(\boldsymbol{d}) \circ \boldsymbol{s}^{-1}(\boldsymbol{y})\right) \\
&= -\boldsymbol{d}^\top\nabla^2_{\boldsymbol{yy}} f_\mu(\boldsymbol{y})\boldsymbol{d} + \frac{\mu}{2}\operatorname{trace}\left(\boldsymbol{s}^{-1/2}(\boldsymbol{y}) \circ \left(\boldsymbol{h}(\boldsymbol{d}) \circ \boldsymbol{s}^{-1/2}(\boldsymbol{y})\right)\right) \\
&= -\boldsymbol{d}^\top\nabla^2_{\boldsymbol{yy}} f_\mu(\boldsymbol{y})\boldsymbol{d} + \frac{\mu}{2}\operatorname{trace}\left(\boldsymbol{\omega}(\boldsymbol{d})\right) \\
&= -\boldsymbol{d}^\top\nabla^2_{\boldsymbol{yy}} f_\mu(\boldsymbol{y})\boldsymbol{d} + \frac{\mu}{2}\sum_{i=1}^{2r}\lambda_i\left(\boldsymbol{\omega}\right).
\end{aligned} \tag{7}$$

From (1), we have

$$
\begin{aligned}
\boldsymbol{d}^{\mathsf{T}}\nabla^2_{\boldsymbol{yy}}f_\mu(\boldsymbol{y})\boldsymbol{d} &= \mu\boldsymbol{d}^{\mathsf{T}}A\,Q\left(\boldsymbol{s}^{-1}(\boldsymbol{y})\right)A^{\mathsf{T}}\boldsymbol{d}\\
&= \mu(\boldsymbol{h}(\boldsymbol{d}))^{\mathsf{T}}Q\left(\boldsymbol{s}^{-1}(\boldsymbol{y})\right)\boldsymbol{h}(\boldsymbol{d})\\
&= \frac{\mu}{2}\mathrm{trace}\left(\boldsymbol{h}(\boldsymbol{d})\circ Q\left(\boldsymbol{s}^{-1}(\boldsymbol{y})\right)\boldsymbol{h}(\boldsymbol{d})\right)\\
&= \frac{\mu}{2}\mathrm{trace}\left(\boldsymbol{h}^2(\boldsymbol{d})\circ Q\left(\boldsymbol{s}^{-1}(\boldsymbol{y})\right)\boldsymbol{e}\right)\\
&= \frac{\mu}{2}\mathrm{trace}\left(\boldsymbol{h}^2(\boldsymbol{d})\circ \boldsymbol{s}^{-2}(\boldsymbol{y})\right)\\
&= \frac{\mu}{2}\mathrm{trace}\left(\boldsymbol{\omega}^2(\boldsymbol{d})\right)\\
&= \frac{\mu}{2}\sum_{i=1}^{2r}\lambda_i^2\left(\boldsymbol{\omega}\right).
\end{aligned} \tag{8}
$$

Substituting (8) into (7), we get

$$
\boldsymbol{d}^{\mathsf{T}}\boldsymbol{b} = \frac{\mu}{2}\sum_{i=1}^{2r}\left(\lambda_i\left(\boldsymbol{\omega}\right)-\lambda_i^2\left(\boldsymbol{\omega}\right)\right). \tag{9}
$$

Substituting (9) into (4), we obtain

$$
\theta(t) = \frac{t}{2}\sum_{i=1}^{2r}\left(\lambda_i\left(\boldsymbol{\omega}\right)-\lambda_i^2\left(\boldsymbol{\omega}\right)\right) - \sum_{i=1}^{2r}\ln\left(1+t\lambda_i\left(\boldsymbol{\omega}\right)\right),\ \text{for } t\in[0,\hat{t}[,
$$

with $\hat{t} = \sup\left\{t : 1 + t\lambda_i\left(\boldsymbol{\omega}\right), i = 1, 2, \ldots, 2r\right\}$. The proof is complete. □

A lower bound $\hat{t}_1$ of $\hat{t}$ is based on Proposition 2.1 and is given by

$$
\hat{t}_1 := \sup\{t : 1 + t\beta_1 > 0\}\ \text{ with }\ \beta_1 := \bar{\lambda}(\boldsymbol{\omega}) - \sigma_{\lambda(\boldsymbol{\omega})}\sqrt{2r-1}, \tag{10}
$$

where, as defined in Section 2, $\bar{\lambda}(\boldsymbol{\omega}) := \frac{1}{2r}\sum_{i=1}^{2r}\lambda_i(\boldsymbol{\omega})$ and $\sigma^2_{\lambda(\boldsymbol{\omega})} := \frac{1}{2r}\sum_{i=1}^{2r}\lambda_i^2(\boldsymbol{\omega}) - \bar{\lambda}^2(\boldsymbol{\omega})$.

Another bound $\hat{t}_2$ is based on the inequality $|\lambda_i(\boldsymbol{\omega})| \leq \|\boldsymbol{\omega}\|_F$ for $i = 1, 2, \ldots, 2r$, and is given by

$$
\hat{t}_2 := \sup\{t : 1 + t\beta_2 > 0\}\ \text{ with }\ \beta_2 := -\|\boldsymbol{\omega}\|_F. \tag{11}
$$

Now, we look for a majorant function $\hat{\theta}$ of the function θ on $[0, \hat{t}_i[$, for $i = 1, 2, \ldots, 2r$, which can be used as an upper approximation of θ. Such an upper approximation may be more efficient to manipulate than θ. The function $\hat{\theta}$ is chosen to be simple and close enough to θ and to satisfy the following properties

$$
\hat{\theta}(0) = 0\ \text{ and }\ \hat{\theta}''(0) = -\hat{\theta}'(0) = \|\boldsymbol{\omega}\|_F^2. \tag{12}
$$

As an analogue to the choices for the majorant function given in [2] for semidefinite programming and then in [1] for linear programming, we consider following three choices for the majorant function $\hat{\theta}$ for second-order cone programming:

Choice 4.1 (The first majorant function θ_0)**.** *We may define a majorant function* θ_0 *on* $[0, \hat{t}_0[$, *with* $\hat{t}_0 = \hat{t}_1$, *by*

$$
\theta_0(t) := \left(2\bar{\lambda}(\boldsymbol{\omega}) - \|\boldsymbol{\omega}\|_F^2\right)t - (2r-1)\ln\left(1 + t\left(\bar{\lambda}(\boldsymbol{\omega}) + \sigma_{\lambda(\boldsymbol{\omega})}\right)\right) - \ln\left(1 + t\left(\bar{\lambda}(\boldsymbol{\omega}) - \sigma_{\lambda(\boldsymbol{\omega})}\right)\right).
$$

Choice 4.2 (The second majorant function θ_1)**.** *We may define a majorant function θ_1 on $[0, \hat{t}_1[$ by*

$$\theta_1(t) := \left(2r\bar{\lambda}(\omega) - \|\omega\|_F^2\right) t - \frac{\|\omega\|_F^2}{\left(\bar{\lambda}(\omega) - \sigma_{\lambda(\omega)}\sqrt{2r-1}\right)^2} \ln\left(1 + t\left(\bar{\lambda}(\omega) - \sigma_{\lambda(\omega)}\sqrt{2r-1}\right)\right).$$

Choice 4.3 (The third majorant function θ_2)**.** *We may define a majorant function θ_2 on $[0, \hat{t}_2[$ by*

$$\theta_2(t) := -\left(\|\omega\|_F - \|\omega\|_F^2\right) t - \ln\left(1 - t\|\omega\|_F\right).$$

Choice 4.1 is obtained from the second statement of Proposition 2.1 by setting $x_i = 1 + t\lambda_i(\omega)$ for $i = 1, 2, \ldots, 2r$, then $\bar{x} = 1 + t\bar{\lambda}(\omega)$ and $\sigma_x = t\sigma_{\lambda(\omega)}$. Note that the majorant function in Choice 4.1 can be alternatively written as

$$\theta_0(t) := \gamma_0 t - (2r-1)\ln(1 + t\alpha_0) - \ln(1 + t\beta_0),$$

where

$$\gamma_0 := 2r\bar{\lambda}(\omega) - \|\omega\|_F^2, \quad \alpha_0 := \bar{\lambda}(\omega) + \frac{\sigma_{\lambda(\omega)}}{\sqrt{2r-1}} \quad \text{and} \quad \beta_0 := \bar{\lambda}(\omega) - \sigma_{\lambda(\omega)}\sqrt{2r-1} = \beta_1.$$

One can see that the majorant function $\theta_0(t)$ is definite and convex on $[0, \hat{t}_0[$. One can also see that this majorant function satisfies the properties in (12).

Choices 4.2 and 4.3 are obtained in such a way that they have the following type

$$\tilde{\theta}(t) = \tilde{\gamma} t - \tilde{\delta}\ln(1 + \tilde{\beta} t), \quad t \in [0, \tilde{t}[\;\text{ with }\; \tilde{t} = \sup\left\{t : 1 + t\tilde{\beta} > 0\right\}.$$

For Choice 4.2, we take $\tilde{\beta} = \beta_0 = \beta_1$ and $\tilde{\gamma} = \gamma_0 = \gamma_1$, and we choose $\tilde{\delta} = \delta_1 > 1$ in such a way that we have $\|\omega\|_F^2 = \tilde{\delta}\tilde{\beta}^2$. For Choice 4.3, we take $\tilde{\beta} = \beta_2$, and we choose $\tilde{\gamma} = \gamma_2$ and $\tilde{\delta} = \delta_2$ in such a way that we have $\|\omega\|_F^2 = \tilde{\delta}\tilde{\beta}^2 = \tilde{\delta}\tilde{\beta} - \tilde{\gamma}$.

We can see that the majorant functions $\theta_1(t)$ and $\theta_2(t)$ are definite and convex on $[0, \hat{t}_1[$ and $[0, \hat{t}_2[$, respectively. We can also see that these two majorant functions satisfy the properties in (12). The following theorem compares the functions $\theta, \theta_0, \theta_1$ and θ_2.

Theorem 4.1. *Let $t^\star = \min\left\{\hat{t}_1, \hat{t}_2\right\}$, then we have $\theta(t) \le \theta_0(t) \le \theta_1(t) \le \theta_2(t) \le +\infty, \;\; \forall t \in [0, t^\star[$.*

Proof. Let $t \in [0, t^\star[$. We first show that $\theta(t) \le \theta_0(t)$. Let $x_1, x_2 > 0$. Using the second statement of Proposition 2.1, we have

$$\sum_{i=1}^{2r} \ln(x_i) \ge \ln\left(\bar{x} + \sigma_x\right) + \ln\left(\bar{x} - \sigma_x\right).$$

This implies that

$$\sum_{i=1}^{2r} \ln(x_i) + t\|\omega\|_F^2 \ge \ln\left(\bar{x} + \sigma_x\right) + \ln\left(\bar{x} - \sigma_x\right) + t\|\omega\|_F^2,$$

which in turn implies that

$$2t\bar{\lambda}(\omega) - \left(\sum_{i=1}^{2r} \ln(x_i) + t\|\omega\|_F^2\right) \le 2t\bar{\lambda}(\omega) - \ln\left(\bar{x} - \sigma_x\right) - \ln\left(\bar{x} - \sigma_x\right) - t\|\omega\|_F^2.$$

Taking $x_i = 1 + t\lambda_i(\boldsymbol{\omega})$ for $i = 1, 2, \ldots, 2r$, hence $\bar{\boldsymbol{x}} = 1 + t\bar{\lambda}(\boldsymbol{\omega})$ and $\sigma_{\boldsymbol{x}} = t\sigma_{\lambda(\boldsymbol{\omega})}$, we get

$$2t\bar{\lambda}(\boldsymbol{\omega}) - \left(\sum_{i=1}^{2r} \ln(1 + t\lambda_i(\boldsymbol{\omega})) + t\|\boldsymbol{\omega}\|_F^2 \right) \leq 2t\bar{\lambda}(\boldsymbol{\omega}) - \ln\left(1 + t\bar{\lambda}(\boldsymbol{\omega}) - t\sigma_{\lambda(\boldsymbol{\omega})}\right) - \ln\left(1 + t\bar{\lambda}(\boldsymbol{\omega}) - t\sigma_{\lambda(\boldsymbol{\omega})}\right) - t\|\boldsymbol{\omega}\|_F^2.$$

Note that the left-hand side of the above inequality is nothing but the function $\theta(t)$ and the right-hand side of the above inequality is nothing but the function $\theta_0(t)$. This means we have shown that $\theta(t) \leq \theta_0(t)$ on $[0, t^\star[$.

Now, we prove that $\theta_0(t) \leq \theta_1(t)$. Consider the function

$$f(t) := \theta_1(t) - \theta_0(t) = \gamma_1 t - \delta_1 \ln(1 + \beta_1 t) - \gamma_0 t + (2r - 1)\ln(1 + \alpha_0 t) + \ln(1 + \beta_0 t)$$

for $t \in [0, t^\star[$. Since $\beta_0 = \beta_1, \tilde{\delta}\beta_0^2 - \beta_0^2 = (2r-1)\alpha_0^2$ and $\alpha_0 \geq \beta_0$, we have

$$f''(t) = \frac{\tilde{\delta}\beta_0^2 - \beta_0^2}{(1 + t\beta_0)^2} - \frac{(2r-1)\alpha_0^2}{(1 + t\alpha_0)^2} = \frac{(2r-1)\alpha_0^2}{(1 + t\beta_0)^2} - \frac{(2r-1)\alpha_0^2}{(1 + t\alpha_0)^2} \geq 0.$$

Because $f''(t) \geq 0$ and $f'(0) = 0$, we have $f'(t) \geq 0$ for all $t \in [0, t^\star[$. Because $f'(t) \geq 0$ and $f(0) = 0$, we also have $f(t) \geq 0$ for all $t \in [0, t^\star[$. Therefore, $\theta_0(t) \leq \theta_1(t)$ on $[0, t^\star[$.

Next, we prove that $\theta_1(t) \leq \theta_2(t)$. Similarly, consider the function

$$g(t) := \theta_2(t) - \theta_1(t) = \gamma_2 t - \delta_2 \ln(1 + \beta_2 t) - \gamma_1 t + \delta_1 \ln(1 + \beta_1 t)$$

for $t \in [0, t^\star[$. Since $\|\boldsymbol{\omega}\|_F^2 = \delta_1\beta_1^2$, $\beta_2 = -\|\boldsymbol{\omega}\|_F$, and $\beta_1 \geq \beta_2$, we have

$$g''(t) = \frac{\beta_2^2}{(1 + t\beta_2)^2} - \frac{\delta\beta_1^2}{(1 + t\beta_1)^2} = \|\boldsymbol{\omega}\|_F^2 \left(\frac{1}{(1 + t\beta_2)^2} - \frac{1}{(1 + t\beta_1)^2} \right) \geq 0.$$

Because $g''(t) \geq 0$ and $g(0) = g'(0) = 0$, we have $g(t) \geq 0$ for all $t \in [0, t^\star[$. Therefore, $\theta_1(t) \leq \theta_2(t)$ on $[0, t^\star[$. The proof is complete. □

We conclude that the function $\theta_0(t)$ reaches its minimum when

$$\theta_0'(t) = \gamma_0 - \frac{(2r-1)\alpha_0}{1 + \alpha_0 t} - \frac{\beta_0}{1 + \beta_0 t} = 0,$$

which is reached at the point

$$t_0 = b - \sqrt{b^2 - c}, \quad \text{where} \quad b = \frac{1}{2}\left(\frac{2r}{\gamma_0} - \frac{1}{\alpha_0} - \frac{1}{\beta_0} \right) \quad \text{and} \quad c = -\frac{\|\boldsymbol{\omega}\|_F^2}{\alpha_0\beta_0\gamma_0}.$$

We can also conclude that the functions $\theta_j(t)$, for $j = 1, 2$, reach their minima when

$$\theta_j'(t) = \gamma_j - \frac{\delta_j\beta_j}{1 + \beta_j t} = 0,$$

which are reached at the points $t_j = \delta_j/\gamma_j - 1/\beta_j$, for $j = 1, 2$.

5 Numerical experiments

In this section, we present some numerical results to demonstrate the performance of our methods. Our numerical experiments are carried out on a PC with Intel(R) Dual CPU at 2.20 GHz and 2 GB of physical memory. The PC runs MATLAB Version: 7.4.0.287 (R2007a) on Windows XP Enterprise 32-bit operating system. We denote by

"S_0" the first strategy which uses the majorant function θ_0,
"S_1" the second strategy which uses the majorant function θ_1,
"S_2" the third strategy which uses the majorant function θ_2,
"Iter" the number of iterations taken to obtain the optimal solution, and
"CPU(s)" the time (in seconds) required to obtain the optimal solution.

The following two examples are taken from the literature, see for example [3,28]. We point out that the matrices used in these examples have full row rank.

Example 5.1 (Problem with a single second-order cone ($r = 1$))**.** *Let n and m be positive integers such that $n = 2m$. We consider the SOCP problem*

$$\begin{array}{ll} \min & \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} \\ \text{s.t.} & A^{\mathsf{T}}\boldsymbol{y} - \boldsymbol{c} \in \mathcal{E}^n_+, \end{array}$$

where $\boldsymbol{b} = \overbrace{(2;2;\ldots,2)}^{m \text{ times}} - 10\boldsymbol{e}_m - 4\,\boldsymbol{rand}(m,1)$, $\boldsymbol{c} = \overbrace{(2;2;\ldots,2)}^{n \text{ times}} - 10\boldsymbol{e}_n - 4\,\boldsymbol{rand}(n,1)$, *and*

$$A = \left[\begin{array}{cccccc|c} -100 & -2 & & & & & \\ 2 & -100 & -2 & & & & \\ & \ddots & \ddots & \ddots & & & Randn(m, n\text{-}m) \\ & & 2 & -100 & -2 & & \\ & & & 2 & -100 & & \end{array}\right] \in \mathbb{R}^{m\times n}.$$

We take $\boldsymbol{y}^{(0)} = \boldsymbol{0} \in \mathbb{R}^m$ as our initial strictly feasible point. We also take $\mu = 0.3$ and $\epsilon = 10^{-6}$. The numerical results related to this example are displayed in Table 5.1.

Problem size	S_0		S_1		S_2	
(m, n)	*Iter*	*CPU(s)*	*Iter*	*CPU(s)*	*Iter*	*CPU(s)*
(5, 10)	5	0.00610	5	0.00610	11	0.00815
(10, 20)	6	0.00672	6	0.00672	13	0.00854
(20, 40)	6	0.00700	6	0.00700	12	0.00831
(25, 50)	7	0.00915	7	0.00887	14	0.01095
(30, 60)	9	0.01416	10	0.02156	34	0.02919
(80, 160)	17	0.15601	19	0.37543	58	0.60872

Table 5.1. *The numerical results of Example* 5.1.

The numerical results in Table 5.1 show the difference in performance between the strategy S_2 on one hand and the strategies S_0 and S_1 on the other hand. It is clear that S_2 does not even compete with S_0 and S_1. Therefore, in the next example, we continue only with the strategies S_0 and S_1.

Example 5.2 (Problem with multiple second-order cones ($r > 1$)). *Let n, m and r be positive integers such that $n = 2m$ and r is an even number dividing n. We consider the SOCP problem*

$$\begin{array}{ll} \min & \boldsymbol{b}^{\mathsf{T}}\boldsymbol{y} \\ \text{s.t.} & A^{\mathsf{T}}\boldsymbol{y} - \boldsymbol{c} \in \underbrace{\mathcal{E}_{+}^{n/r} \times \cdots \times \mathcal{E}_{+}^{n/r}}_{r \text{ times}}, \end{array}$$

where $\boldsymbol{b} = \overbrace{(2; 2; \ldots, 2)}^{m \text{ times}}$, $\boldsymbol{c} = \overbrace{(\boldsymbol{e}_{n/r}; \ldots; \boldsymbol{e}_{n/r})}^{r \text{ times}} \in \overbrace{\mathcal{E}^{n/r} \times \cdots \times \mathcal{E}^{n/r}}^{r \text{ times}}$ *and* $A = [I_m \mid Randn(m, n-m)] \in \mathbb{R}^{m \times n}$.

We take $\boldsymbol{y}^{(0)} = \overbrace{(1.5\, \boldsymbol{e}_{n/r}; \ldots; 1.5\, \boldsymbol{e}_{n/r})}^{r/2 \text{ times}}$ *as our initial strictly feasible point. We also take* $\mu = 0.4$ *and* $\epsilon = 10^{-6}$. *The optimal solution is* $\boldsymbol{y}^{\star} = \underbrace{(\boldsymbol{e}_{n/r}; \ldots; \boldsymbol{e}_{n/r})}_{r/2 \text{ times}}$. *Our numerical results are displayed in Table 5.2.*

Problem size		S_0		S_1	
(m, n)	r	*Iter*	*CPU(s)*	*Iter*	*CPU(s)*
(100, 200)	10	27	0.69765	33	4.58421
(200, 400)	20	29	47.6790	34	54.5614
(300, 600)	30	30	173.456	36	211.671
(360, 720)	36	34	217.205	39	268.000
(400, 800)	40	35	329.764	42	388.742
(460, 920)	46	38	391.585	47	614.536

Table 5.2. *The numerical results of Example* 5.2.

Looking at the time of computation and number of iterations in Table 5.2, we conclude that the first strategy S_0 is the best strategy to obtain the optimal solution. Note that our conclusion exactly matches the same conclusions obtained from the numerical results of [1, Section 4] for linear programming and that of [2, Section 8] for semidefinite programming. However, this is not surprising because of the remarkable agreement of these numerical results and the comparison result in Theorem 4.1 between the majorant functions.

6 Conclusions

In this paper, we have presented a logarithmic barrier interior-point method for solving the second-order cone programming problem. We have proven the existence and uniqueness of the optimal solution of the corresponding perturbed problem and have verified its convergence to the optimal solution of the original problem when the barrier parameter approaches zero. Newton's method has been applied to find a new iterative point by calculating a sufficient descent direction. Due to the high computational cost, we have avoided using several methods, such as the line search methods, to calculate the displacement step. Alternatively, a new approach based on majorant functions has been proposed to accomplish this task. Our numerical results show the efficiency of the proposed approach. We have used three strategies to perform our approach based on which majorant function we selected. Regardless the strategy we undertake, our algorithm converges to the optimal solution. The numerical results show that the first strategy is the best route to the most desirable end after looking at the computed time and number of iterations.

Finally, by using the machinery of Euclidean Jordan algebras, we point out the results of this paper can be extended word-by-word to general symmetric cone programming.

References

[1] L. Mennichea, D. Benterki, A logarithmic barrier approach for linear programming, J. Comput. Appl. Math. 312 (2017) 267–275.

[2] J.P. Crouzeix, B. Merikhi, A logarithm barrier method for semidefinite programming, RAIRO Oper. Res. 42 (2008) 123–139.

[3] D. Benterki, J.P. Crouzeix, B. Merikhi, A numerical feasible interior point method for linear semidefinite programs, RAIRO Oper. Res. 41 (2007) 49–59.

[4] I. Touil, D. Benterki, A. Yassine, A feasible primal-dual interior point method for linear semidefinite programming, J. Comput. Appl. Math. 312 (2017) 216–230.

[5] S. Wright, Primal-Dual Interior Point Methods, SIAM, Philadelphia, 1997.

[6] D. Hertog, Interior-point Approach to Linear Quadratic and Convex Programming, Kluwer Academic Publishers, Dordrecht, 1994.

[7] Y.E. Nestrov, A. Nemiroveskii, Interior Point Polynomial Algorithms in Convex Programming, SIAM, 1994.

[8] C. Roos, T. Terlaky, J.P. Vial, Theory and Algorithms for Linear Optimization: An interior-point Approach, W. John & Sons, 1997.

[9] Y. Ye, Interior Point Algorithms: Theory and Analysis, in: Wiley-Interscience Series in Discrete Mathematics Optimization, John wiley & Sons, New York, 1997.

[10] Z. Kebbiche, A. Keraghel, A. Yassine, Extension of a projective interior point method for linearly constrained convex programming, Appl. Math. Comput. 193 (2007) 553–559.

[11] R. Naseri, A. Valinejad, An extended variant of Karmarkars interior point algorithm, Appl. Math. Comput. 184 (2007) 737–742.

[12] M. Achache, A weighted path-following method for the linear complementarity problem, Studia Univ. Babes-Bolyai Ser. Inform. 48 (2004) 61–73.

[13] Z. Darvay, A weighted-path-following method for linear optimization, Studia Univ. Babes-Bolyai Ser. Inform. 47 (2002) 3–12.

[14] F. Alizadeh, J.-P. Haberly, M.-L. Overton, Primal-dual interior-point methods for semidefinite programming, convergence rates, stability and numerical results. SIAM J. Optim. 8 (1998) 746–768.

[15] D. Benterki, J.-P. Crouzeix, B. Merikhi, A numerical implementation of an interior point method for semi-definite programming. Pesquisa Operacional 23 (2003) 49–59.

[16] J. Tang, G. He, L. Dong, L. Fang, J. Zhou, A globally and quadratically convergent smoothing Newton method for solving second-order cone optimization. Appl. Math. Model. 39 (2015) 2180–2193.

[17] L. Dong, J. Tang, J. Zhou, A smoothing Newton algorithm for solving the monotone second-order cone complementarity problems. J. Appl. Math. Comput. 40 (2012) 45–61.

[18] X. Chi, S. Liu, A new one-step smoothing Newton method for second-order cone programming. J. Comput. Appl. Math. 223 (2009) 114–123.

[19] J. Tang, G. He, L. Dong, L. Fang, A smoothing Newton method for second-order cone optimization based on a new smoothing function. Appl. Math. Comput. 218 (2011) 1317–1329.

[20] L. Fang, A smoothing-type Newton method for second-order cone programming problems based on a new smooth function. J. Appl. Math. Comput. 34 (2010) 147–161.

[21] F. Alizadeh, D. Goldfarb, Second-order cone programming. Math. Program. Ser. B 95 (2003) 3–51.

[22] Y.-J. Hans, D. Mittelmann, Interior point methods for second-order cone programming and OR applications. Comput. Optim. Appl. 28 (2004) 255–285.

[23] B. Kheirfam, A corrector-predictor path-following method for second-order cone optimization. Int. J. Computer Math. 93 (2016) 2064–2078.

[24] Z. Hao, Z. Wan, X. Chi, J. Chen, A power penalty method for second-order cone nonlinear complementarity problems. J. Comput. Appl. Math. 290 (2015) 136–149.

[25] A. Auslender, An exact penalty method for nonconvex problems covering, in particular, nonlinear programming, semidefinite programming, and second-order cone programming. SIAM J. Optim. 25 (2015) 1732–1759.

[26] J. Faraut, A. Korányi, Analysis on Symmetric Cones, Oxford University Press, Oxford, UK, 1994.

[27] H. Wolkowicz, G.-P.-H. Styan, Bounds for eigenvalues using traces. Linear Algebra Appl. 29 (1980) 471–506.

[28] J. Tang, G. He, L. Dong, L. Fang, A new one-step smoothing newton method for second-order cone programming. Applications of Mathematics, 57 (2012) 311–331.