Jingyong Tang; Guoping He; Li Dong; Liang Fang

A new one-step smoothing newton method for second-order cone programming

Applications of Mathematics, Vol. 57 (2012), No. 4, 311–331

Persistent URL: http://dml.cz/dmlcz/142901

A NEW ONE-STEP SMOOTHING NEWTON METHOD FOR SECOND-ORDER CONE PROGRAMMING*

Jingyong Tang, Shanghai and Xinyang, Guoping He, Qingdao,
Li Dong, Xinyang, Liang Fang, Tai'an

(Received November 3, 2010)

Abstract. In this paper, we present a new one-step smoothing Newton method for solving the second-order cone programming (SOCP). Based on a new smoothing function of the well-known Fischer-Burmeister function, the SOCP is approximated by a family of parameterized smooth equations. Our algorithm solves only one system of linear equations and performs only one Armijo-type line search at each iteration. It can start from an arbitrary initial point and does not require the iterative points to be in the sets of strictly feasible solutions. Without requiring strict complementarity at the SOCP solution, the proposed algorithm is shown to be globally and locally quadratically convergent under suitable assumptions. Numerical experiments demonstrate the feasibility and efficiency of our algorithm.

Keywords: second-order cone programming, smoothing Newton method, global convergence, quadratic convergence

MSC 2010: 90C25, 90C30, 49M37, 65K05, 65Y20

1. Introduction

Second-order cone programming (SOCP) problem is a convex programming problem in which a linear function is minimized over the intersection of an affine space with the Cartesian product of second-order (or Lorentz or ice-cream) cones. A typical second-order cone (SOC) in $\mathbb{R}^l$ has the form

$$\mathcal{L}^l := \{x = (x_1; \bar{x}) \in \mathbb{R} \times \mathbb{R}^{l-1} \colon x_1 \geqslant \|\bar{x}\|\},$$

* This work was supported by National Natural Science Foundation of China (10971122), Natural Science Foundation of Shandong Province (Y2008A01), Specialized Research Foundation for the Doctoral Program of Higher Education (20093718110005) and Project of Shandong Province Higher Educational Science and Technology Program (J10LA51).

where $\|\cdot\|$ refers to the standard Euclidean norm, and x_1 is the first element of x and $\bar{x}$ is the vector containing the remaining elements of x. For simplicity, we use the semi-colon ";" to join vectors in a column. Thus, for instance, for column vectors x, y, and z we use $(x; y; z)$ to represent $(x^{\mathrm{T}}, y^{\mathrm{T}}, z^{\mathrm{T}})^{\mathrm{T}}$.

Let $\mathcal{L} \subset \mathbb{R}^n$ be the Cartesian product of several second-order cones, i.e.,

$$\mathcal{L} := \mathcal{L}^1 \times \mathcal{L}^2 \times \ldots \times \mathcal{L}^r$$

with $\mathcal{L}^i \subset \mathbb{R}^{n_i}$ for each i, $i = 1, 2, \ldots, r$, and $n = \sum_{i=1}^{r} n_i$. In this paper we consider SOCP in the standard format

$$\text{(P)} \qquad \min\{c^{\mathrm{T}}x\colon\ Ax = b,\ x \in \mathcal{L}\} \tag{1}$$

and the dual problem of (P) is given by

$$\text{(D)} \qquad \max\{b^{\mathrm{T}}y\colon\ A^{\mathrm{T}}y + s = c,\ s \in \mathcal{L}\} \tag{2}$$

where $A \in \mathbb{R}^{m\times n}$, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. Without loss of generality, we assume that $r = 1$ and $n_1 = n$ in the subsequent analysis, since our analysis can be easily extended to the general case.

The sets of strictly feasible solutions of (1) and (2) are

$$\begin{aligned} \mathcal{F}^0(\mathrm{P}) &= \{x\colon\ Ax = b,\ x \in \mathcal{L}^0\}, \\ \mathcal{F}^0(\mathrm{D}) &= \{(y, s)\colon\ A^{\mathrm{T}}y + s = c,\ s \in \mathcal{L}^0\}, \end{aligned}$$

respectively, where

$$\mathcal{L}^0 := \{x = (x_1; \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}\colon\ x_1 > \|\bar{x}\|,\ x_1 > 0\}.$$

Throughout the paper, we make the following assumptions:

Assumption 1.1. Both (1) and (2) are strictly feasible, i.e., $\mathcal{F}^0(\mathrm{P}) \times \mathcal{F}^0(\mathrm{D}) \neq \emptyset$.

Assumption 1.2. A has full row rank.

Under Assumption 1.1, it can be shown that both (1) and (2) have optimal solutions and their optimal values coincide [1].

SOCP problem includes the linear programming problem, the convex quadratic programming problem, and the quadratically constrained convex quadratic programming problem as special cases [1]. In recent years the SOCP problem has received considerable attention from researchers for its wide range of applications in many fields, such as engineering, optimal control and design, machine learning, robust programming and combinatorial programming and so on (see, e.g., [6], [13], [18], [23], [25]). Many researchers have studied interior-point methods (IPMs) for solving SOCP and achieved plentiful and beautiful results (see, e.g., [2], [7], [8], [12], [14], [17], [19]).

Recently, smoothing Newton methods have attracted a lot of attention partially due to their superior numerical performances (see, e.g., [3], [4], [9]–[11], [15], [16], [20], [21], [24], [28]). However, the available smoothing methods are mostly for solving the complementarity problems (see, e.g., [3], [4], [9]–[11], [15], [20], [21], [24]). The results of smoothing methods for solving the SOCP are rather rare. Moreover, the global convergence and locally superlinear (or quadratic) convergence of some algorithms depend on the assumptions of uniform nonsingularity and strict complementarity (see, e.g., [3], [20]). Without the uniform nonsingularity assumption, the algorithm given in [28] usually needs to solve two linear systems of equations and to perform at least two line searches per iteration. Lastly, Qi, Sun, and Zhou [21] proposed a class of new smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities under a nonsingularity assumption. The method in [21] was shown to be locally superlinearly/quadratically convergent without strict complementarity.

Motivated by these ideas, in this paper we present a new one-step smoothing Newton method for solving the SOCP based on a new smoothing function. Without requiring strict complementarity at the SOCP solution, the proposed algorithm is proved to be globally and locally quadratically convergent. Our algorithm has the following nice properties:

(i) it is well-defined and any accumulation point of the iteration sequence is a solution to the SOCP;
(ii) it can start from an arbitrary initial point and does not require the iterative points to be in the sets of strictly feasible solutions;
(iii) it solves only one linear system of equations and performs only one Armijo-type line search per iteration;
(iv) if an accumulation point of the iteration sequence satisfies a nonsingularity assumption, then the whole iteration sequence converges to the accumulation point globally and locally quadratically without strict complementarity;
(v) it has an effective numerical performance in many situations.

Some notation is used throughout the paper as follows. $\mathbb{R}^n$, $\mathbb{R}^n_+$ and $\mathbb{R}^n_{++}$ denote the set of vectors with n components, the set of nonnegative vectors and the set of positive vectors, respectively. The product $\mathbb{R}^n \times \mathbb{R}^m$ is identified with $\mathbb{R}^{n+m}$, I represents the identity matrix with suitable dimension, and $\|\cdot\|$ denotes the 2-norm of the vector x defined by $\|x\| = \sqrt{x^{\mathrm{T}}x}$. For any $\alpha, \beta > 0$, $\alpha = O(\beta)$ or $\alpha = o(\beta)$

means that α/β is uniformly bounded or, respectively, tends to zero as $\beta \to 0$. For any $x, y \in \mathbb{R}^n$, we write $x \succeq_{\mathcal{L}} y$ or $x \succ_{\mathcal{L}} y$ if $x - y \in \mathcal{L}$ or, respectively, $x - y \in \mathcal{L}^0$. For any square matrix $A \in \mathbb{R}^{n\times n}$, we write $A \succeq 0$ or $A \succ 0$ if the symmetric part of A is positive semi-definite or, respectively, positive definite.

The rest of this paper is organized as follows. In Section 2, we introduce some preliminaries which are used in the subsequent sections. Based on the Fischer-Burmeister function, a new smoothing function and its properties are given in Section 3. In Section 4, we present a one-step smoothing Newton method for solving the SOCP and state some preliminary results. The global convergence and locally quadratic convergence of the algorithm are investigated in Section 5. Numerical results are reported in Section 6. Finally, some conclusions are summarized in the last section.

2. Preliminaries

In this section, we briefly recall some algebraic properties of the SOC $\mathcal{L}$ and its associated Euclidean Jordan algebra. Our main sources for this section are [1], [8].

For any vectors $x = (x_1; \bar{x}), s = (s_1; \bar{s}) \in \mathbb{R}\times\mathbb{R}^{n-1}$, their Jordan product associated with the SOC $\mathcal{L}$ is defined by

$$x \circ s := (x^{\mathrm{T}} s; x_1\bar{s} + s_1\bar{x}).$$

One easily checks that this operator is commutative and $(\mathbb{R}^n, \circ)$ is a Euclidean Jordan algebra with the vector

$$\mathbf{e} := (1; 0; \ldots; 0)$$

as the identity element.

Spectral factorization is one of the basic and important concepts in Euclidean Jordan algebra. We have the following theorem [1] with respect to the SOC $\mathcal{L}$.

Theorem 2.1 (Spectral factorization). *For any vector $x = (x_1; \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$, its spectral factorization with respect to the* SOC $\mathcal{L}$ *is*

$$x = \lambda_1 u_1 + \lambda_2 u_2, \tag{3}$$

where λ_1, λ_2 are the spectral values given by

$$\lambda_i = x_1 + (-1)^i \|\bar{x}\|, \quad i = 1, 2, \tag{4}$$

and u_1, u_2 *are the spectral vectors of* x *given by*

$$(5) \qquad u_i = \begin{cases} \dfrac{1}{2}\Big(1, (-1)^i \dfrac{\bar{x}}{\|\bar{x}\|}\Big), & \bar{x} \neq 0, \\ \dfrac{1}{2}(1, (-1)^i \kappa), & \bar{x} = 0, \end{cases} \qquad i = 1, 2,$$

with any $\kappa \in \mathbb{R}^{n-1}$ *such that* $\|\kappa\| = 1$.

Using (4), for each $x = (x_1; \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$ we can define the following notions [1]:

square root: $x^{1/2} := \lambda_1^{1/2} u_1 + \lambda_2^{1/2} u_2$ for $x \in \mathcal{L}$;

inverse: $x^{-1} := \lambda_1^{-1} u_1 + \lambda_2^{-1} u_2$, if $x \in \mathcal{L}^0$; otherwise, x is singular;

square: $x^2 := \lambda_1^2 u_1 + \lambda_2^2 u_2$.

Indeed, one has $x^2 = x \circ x$ and $(x^{1/2})^2 = x$. If x^{-1} is defined, then $x \circ x^{-1} = \mathbf{e}$. Moreover, $x^2 \in \mathcal{L}$ for all $x \in \mathbb{R}^n$.

Given an element $x = (x_1; \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$, we define the symmetric matrix

$$(6) \qquad L_x := \begin{pmatrix} x_1 & \bar{x}^{\mathrm{T}} \\ \bar{x} & x_1 I \end{pmatrix},$$

where I represents the $n-1 \times n-1$ identity matrix. The matrix L_x can be viewed as a linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$ with the following properties.

Properties 2.2. For any $x = (x_1; \bar{x}), y = (y_1; \bar{y}) \in \mathbb{R} \times \mathbb{R}^{n-1}$, L_x has the following properties:

(a) $L_x y = x \circ y$ and $L_x + L_y = L_{x+y}$;

(b) $x \in \mathcal{L} \Leftrightarrow L_x \succeq 0$ and $x \in \mathcal{L}^0 \Leftrightarrow L_x \succ 0$;

(c) if $x \in \mathcal{L}^0$, then L_x is invertible and the inverse L_x^{-1} is given by

$$L_x^{-1} = \frac{1}{\det(x)} \begin{pmatrix} x_1 & -\bar{x}^{\mathrm{T}} \\ -\bar{x} & \dfrac{\det(x)}{x_1} I + \dfrac{\bar{x}\bar{x}^{\mathrm{T}}}{x_1} \end{pmatrix},$$

where $\det(x) := x_1^2 - \|\bar{x}\|^2$ denotes the determination of x.

Now we recall the concepts of semismoothness and smoothing function. Given a mapping $F\colon \mathbb{R}^n \to \mathbb{R}^m$, if F is locally Lipschitz continuous, then the set

$$\partial_B F(z) := \{V \in \mathbb{R}^{m \times n}\colon \exists \{z_k\} \subseteq D_F\colon z_k \to z,\ F'(z_k) \to V\}$$

is nonempty and is called the B-subdifferential of F at z, where $D_F \subset \mathbb{R}^n$ denotes the set of points at which F is differentiable. The convex hull $\partial F(z) := \mathrm{conv}(\partial_B F(z))$ is the generalized Jacobian of F at z in the sense of Clarke [5]. Semismoothness was originally introduced by Mifflin [16] for functionals. Convex functions, smooth functions, and piecewise linear functions are examples of semismooth functions. The composition of (strongly) semismooth functions is still a (strongly) semismooth function [16]. In [22], Qi and Sun extended the definition of semismooth functions to vector-valued functions.

Definition 2.3 ([22])**.** Suppose that $F\colon \mathbb{R}^n \to \mathbb{R}^m$ is locally Lipschitz continuous around $x \in \mathbb{R}^n$. Then F is said to be semismooth at x if F is directionally differentiable at x and for any $V \in \partial F(x+h)$ and $h \to 0$,

$$F(x+h) - F(x) - Vh = o(\|h\|);$$

F is said to be strongly semismooth at x if F is directionally differentiable at x and for any $V \in \partial F(x+h)$ and $h \to 0$,

$$F(x+h) - F(x) - Vh = O(\|h\|^2);$$

F is said to be (strongly) semismooth if it is (strongly) semismooth everywhere in $\mathbb{R}^n$.

The concept of a smoothing function of a nondifferentiable function was introduced by Hayashi, Yamashita, and Fukushima [10].

Definition 2.4 ([10])**.** For a nondifferentiable function $g\colon \mathbb{R}^n \to \mathbb{R}^m$, we consider a function $g_\mu\colon \mathbb{R}^n \to \mathbb{R}^m$ with a parameter $\mu > 0$ that has the following properties:
(i) g_μ is differentiable for any $\mu > 0$;
(ii) $\lim\limits_{\mu \downarrow 0} g_\mu(x) = g(x)$ for any $x \in \mathbb{R}^n$.

Such a function g_μ is called a smoothing function of g.

3. The new smoothing function and its properties

In this section, we present a new smoothing function and give its properties. Based on this new function, we reformulate the SOCP as a nonlinear system of equations.

In [9], it has been shown that the Fischer-Burmeister (FB) function $\varphi_{\mathrm{FB}}(x,s)\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ defined by

$$\varphi_{\mathrm{FB}}(x,s) = x + s - \sqrt{x^2+s^2} \tag{7}$$

possesses the following important property

$$\varphi_{\mathrm{FB}}(x,s) = 0 \Leftrightarrow x \circ s = 0, \quad x \succeq_{\mathcal{L}} 0,\ \ s \succeq_{\mathcal{L}} 0. \tag{8}$$

The Fischer-Burmeister function $\varphi_{\rm FB}(x,s)$ is globally Lipschitz continuous, continuously differentiable around any $(x,s)\in\mathbb{R}^n\times\mathbb{R}^n$ if $v_1\neq\|\overline{v}\|$, where $v:=x^2+s^2$, and strongly semismooth [26]. However, $\varphi_{\rm FB}$ is typically nonsmooth, because it is not differentiable at $(0;0)\in\mathbb{R}^n\times\mathbb{R}^n$, which limits its practical applications. Many smoothing functions based on $\varphi_{\rm FB}$ have been presented for solving the complementarity problems (see [9], [11] and the references therein).

In this paper, by smoothing the symmetric perturbed Fischer-Burmeister function, we obtain the new vector-valued function $\varphi(\mu,x,s)\colon\ \mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n\to\mathbb{R}^n$ defined by

$$
\begin{aligned}
\varphi(\mu,x,s) &= (\cos\mu+\sin\mu)(x+s) \\
&\qquad - \sqrt{(x\cos\mu+s\sin\mu)^2+(x\sin\mu+s\cos\mu)^2+2\mu^2\mathbf{e}}.
\end{aligned}
\tag{9}
$$

As we will show, the function $\varphi(\mu,x,s)$ has many good properties. These properties play an important role in the analysis of the convergence of our one-step smoothing Newton method. In the next theorem, we show that the function φ given by (9) is a smoothing function of $\varphi_{\rm FB}$.

Theorem 3.1. *Let $(\mu,x,s)\in\mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n$ and let $\varphi(\mu,x,s)$ be defined by* (9). *Then the following results hold.*

(I) *$\varphi(\mu,x,s)$ is globally Lipschitz continuous and strongly semismooth for any $\mu>0$. Moreover, $\varphi(\mu,x,s)$ is continuously differentiable at any $(\mu,x,s)\in\mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n$ with its Jacobian*

$$
\varphi'(\mu,x,s) = \begin{pmatrix} (\cos\mu-\sin\mu)(x+s) \\ \qquad - L_\omega^{-1}[L_{\omega_1}(s\cos\mu-x\sin\mu)+L_{\omega_2}(x\cos\mu-s\sin\mu)+2\mu\mathbf{e}] \\ (\cos\mu+\sin\mu)I-L_\omega^{-1}(L_{\omega_1}\cos\mu+L_{\omega_2}\sin\mu) \\ (\cos\mu+\sin\mu)I-L_\omega^{-1}(L_{\omega_1}\sin\mu+L_{\omega_2}\cos\mu) \end{pmatrix},
$$

where

$$\omega_1:=\omega_1(\mu,x,s)=x\cos\mu+s\sin\mu, \tag{10}$$

$$\omega_2:=\omega_2(\mu,x,s)=x\sin\mu+s\cos\mu, \tag{11}$$

$$\omega:=\omega(\mu,x,s)=\sqrt{\omega_1^2+\omega_2^2+2\mu^2\mathbf{e}}. \tag{12}$$

(II) *$\lim\limits_{\mu\downarrow0}\varphi(\mu,x,s)=\varphi_{\rm FB}(x,s)$ for any $(x,s)\in\mathbb{R}^n\times\mathbb{R}^n$. Thus, $\varphi(\mu,x,s)$ is a smoothing function of $\varphi_{\rm FB}(x,s)$.*

P r o o f. (I) By Theorem 3.2 in [26], it is not difficult to show that $\varphi(\mu, x, s)$ is globally Lipschitz continuous, strongly semismooth everywhere and continuously differentiable at any $(\mu, x, s) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$. Now we prove the Jacobian of $\varphi(\mu, x, s)$. By (9)–(12), we find

$$\varphi(\mu, x, s) = (\cos\mu + \sin\mu)(x + s) - \omega.$$

Thus

$$\varphi'_\mu(\mu, x, s) = (\cos\mu - \sin\mu)(x + s) - \omega'_\mu, \tag{13}$$

$$\varphi'_x(\mu, x, s) = (\cos\mu + \sin\mu)I - \omega'_x, \tag{14}$$

$$\varphi'_s(\mu, x, s) = (\cos\mu + \sin\mu)I - \omega'_s. \tag{15}$$

For any $(\mu, x, s) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$ we have $\omega \in \mathcal{L}^0$ and therefore L_ω is invertible. Due to the definition of ω, we get

$$\omega^2 = \omega_1^2 + \omega_2^2 + 2\mu^2\mathbf{e}.$$

By finding the derivative on both sides of the last relation, we obtain that

$$\omega'_\mu(\mu, x, s) = L_\omega^{-1}[L_{\omega_1}(s\cos\mu - x\sin\mu) + L_{\omega_2}(x\cos\mu - s\sin\mu) + 2\mu\mathbf{e}], \tag{16}$$

$$\omega'_x(\mu, x, s) = L_\omega^{-1}(L_{\omega_1}\cos\mu + L_{\omega_2}\sin\mu), \tag{17}$$

$$\omega'_s(\mu, x, s) = L_\omega^{-1}(L_{\omega_1}\sin\mu + L_{\omega_2}\cos\mu). \tag{18}$$

Then, from (13)–(18) we have the desired Jacobian formula.

Next we prove (II). For any $x = (x_1; \bar{x}), s = (s_1; \bar{s}) \in \mathbb{R} \times \mathbb{R}^{n-1}$, it follows from the spectral factorization of ω^2 that

$$\varphi(\mu, x, s) = (\cos\mu + \sin\mu)(x + s) - \big(\sqrt{\lambda_1(\mu)}u_1(\mu) + \sqrt{\lambda_2(\mu)}u_2(\mu)\big),$$

where

$$\lambda_i(\mu) = \|\omega_1\|^2 + \|\omega_2\|^2 + 2\mu^2 + 2(-1)^i\|v(\mu)\|, \quad i = 1, 2,$$

$$u_i(\mu) = \begin{cases} \dfrac{1}{2}\Big(1, (-1)^i\dfrac{v(\mu)}{\|v(\mu)\|}\Big), & v(\mu) \neq 0, \\ \dfrac{1}{2}(1, (-1)^i\kappa), & v(\mu) = 0, \end{cases} \quad i = 1, 2,$$

$$v(\mu) = (x_1\cos\mu + s_1\sin\mu)(\bar{x}\cos\mu + \bar{s}\sin\mu) + (x_1\sin\mu + s_1\cos\mu)(\bar{x}\sin\mu + \bar{s}\cos\mu)$$

with any $\kappa \in \mathbb{R}^{n-1}$ such that $\|\kappa\| = 1$. In a similar way, we can easily obtain that

$$\varphi_{\mathrm{FB}}(x, s) = x + s - \big(\sqrt{\lambda_1}u_1 + \sqrt{\lambda_2}u_2\big),$$

where

$$\lambda_i = \|x\|^2 + \|s\|^2 + 2(-1)^i \|v\|, \quad i = 1, 2,$$

$$u_i = \begin{cases} \frac{1}{2}\Big(1, (-1)^i \frac{v}{\|v\|}\Big), & v \neq 0, \\ \frac{1}{2}(1, (-1)^i \kappa), & v = 0, \end{cases} \qquad i = 1, 2,$$

$$v = x_1 \bar{x} + s_1 \bar{s}$$

with any $\kappa \in \mathbb{R}^{n-1}$ such that $\|\kappa\| = 1$. Without loss of generality, we choose the same $\kappa \in \mathbb{R}^{n-1}$ as in $u_i(\mu)$. It is easy to find that

$$\lim_{\mu \downarrow 0} v(\mu) = v, \quad \lim_{\mu \downarrow 0} \|\omega_1\|^2 = \|x\|^2, \quad \lim_{\mu \downarrow 0} \|\omega_2\|^2 = \|s\|^2.$$

Thus, we have

$$\lim_{\mu \downarrow 0} \lambda_i(\mu) = \lambda_i, \quad \lim_{\mu \downarrow 0} u_i(\mu) = u_i, \quad i = 1, 2,$$

which implies that $\lim_{\mu \downarrow 0} \varphi(\mu, x, s) = \varphi_{\mathrm{FB}}(x, s)$. Therefore, it follows from (I) and Definition 2.4 that $\varphi(\mu, x, s)$ is a smoothing function of $\varphi_{\mathrm{FB}}(x, s)$. This completes the proof. $\square$

Under Assumption 1.1, it is well known that an optimal solution to (1) and (2) has to satisfy the optimality conditions [1]

$$\begin{aligned} Ax &= b, \\ A^{\mathrm{T}} y + s &= c, \\ x \circ s = 0, \quad x, s &\in \mathcal{L}, \ y \in \mathbb{R}^m. \end{aligned} \tag{19}$$

Let $z := (\mu, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^m$. By using the smoothing function (9), we define the function $H(z)\colon \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{++} \times \mathbb{R}^m \times \mathbb{R}^n$ by

$$H(z) := \begin{pmatrix} \mathrm{e}^{\mu} - 1 \\ G(z) \end{pmatrix}, \tag{20}$$

where

$$G(z) := \begin{pmatrix} b - Ax \\ \varphi(\mu, x, c - A^{\mathrm{T}} y) \end{pmatrix}. \tag{21}$$

In view of (8) and Theorem 3.1, $z^* = (\mu^*, x^*, y^*)$ is a solution to the system $H(z) = 0$ if and only if $(x^*, y^*, c - A^{\mathrm{T}} y^*)$ satisfies the optimality conditions (19), which occurs if and only if $(x^*, y^*, c - A^{\mathrm{T}} y^*)$ is the optimal solution to (1) and (2). Then we can apply Newton's method to the nonlinear system of equations $H(z) = 0$.

The following theorem shows that the function $H(z)$ defined by (20) and (21) is Lipschitz continuous, strongly semismooth and continuously differentiable. Moreover, we also derive the computable formula for the Jacobian of the function $H(z)$ and give a condition for its Jacobian to be invertible.

Theorem 3.2. *Let $z := (\mu, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^m$ and $H\colon \mathbb{R}^{1+n+m} \to \mathbb{R}^{1+m+n}$ be defined by* (20) *and* (21)*. Then the following results hold.*

(i) *H is globally Lipschitz continuous, strongly semismooth everywhere on $\mathbb{R}^{1+n+m}$ and continuously differentiable at any $z := (\mu, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^m$ with its Jacobian*

$$H'(z) = \begin{pmatrix} \mathrm{e}^{\mu} & 0 & 0 \\ 0 & -A & 0 \\ B(z) & C(z) & -D(z)A^{\mathrm{T}} \end{pmatrix}, \tag{22}$$

where

$$\begin{aligned} B(z) &= (\cos\mu - \sin\mu)(x + c - A^{\mathrm{T}}y) - L_{\overline{\omega}}^{-1}[L_{\overline{\omega}_1}((c - A^{\mathrm{T}}y)\cos\mu - x\sin\mu) \\ &\qquad + L_{\overline{\omega}_2}(x\cos\mu - (c - A^{\mathrm{T}}y)\sin\mu) + 2\mu\mathbf{e}], \\ C(z) &= (\cos\mu + \sin\mu)I - L_{\overline{\omega}}^{-1}(L_{\overline{\omega}_1}\cos\mu + L_{\overline{\omega}_2}\sin\mu), \\ D(z) &= (\cos\mu + \sin\mu)I - L_{\overline{\omega}}^{-1}(L_{\overline{\omega}_1}\sin\mu + L_{\overline{\omega}_2}\cos\mu), \\ \overline{\omega}_1 &:= \overline{\omega}_1(\mu, x, c - A^{\mathrm{T}}y) = x\cos\mu + (c - A^{\mathrm{T}}y)\sin\mu, \\ \overline{\omega}_2 &:= \overline{\omega}_2(\mu, x, c - A^{\mathrm{T}}y) = x\sin\mu + (c - A^{\mathrm{T}}y)\cos\mu, \\ \overline{\omega} &:= \overline{\omega}(\mu, x, s) = \sqrt{\overline{\omega}_1^2 + \overline{\omega}_2^2 + 2\mu^2\mathbf{e}}. \end{aligned}$$

(ii) *If A has full row rank, then $H'(z)$ is invertible for any $\mu > 0$.*

Proof. By Theorem 3.1, it is not difficult to show that (i) holds. Now we prove (ii). Fix any $\mu > 0$ and let $\Delta z := (\Delta\mu, \Delta x, \Delta y) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$ be a vector in the null space of $H'(z)$. It is sufficient to prove that the linear system of equations

$$H'(z)\Delta z = 0 \tag{23}$$

has only the zero solution, i.e., $\Delta\mu = 0$, $\Delta x = 0$, and $\Delta y = 0$. By (22) and (23), we have

$$\Delta\mu = 0, \tag{24}$$

$$A\Delta x = 0, \tag{25}$$

$$C(z)\Delta x - D(z)A^{\mathrm{T}}\Delta y = 0. \tag{26}$$

Premultiplying (26) by $L_{\overline{\omega}}$ yields

$$L_{\overline{\omega}}C(z)\Delta x - L_{\overline{\omega}}D(z)A^{\mathrm{T}}\Delta y = 0. \tag{27}$$

From the definitions of $C(z)$ and $D(z)$, a simple calculation reveals that

$$L_{\overline{\omega}}C(z) = (\cos\mu + \sin\mu)L_{\overline{\omega}} - (L_{\overline{\omega}_1}\cos\mu + L_{\overline{\omega}_2}\sin\mu),$$
$$L_{\overline{\omega}}D(z) = (\cos\mu + \sin\mu)L_{\overline{\omega}} - (L_{\overline{\omega}_1}\sin\mu + L_{\overline{\omega}_2}\cos\mu).$$

Since

$$\overline{\omega}^2 - (\overline{\omega}_1^2 + \overline{\omega}_2)^2 = 2\mu^2\mathbf{e} \in \mathcal{L}^0,$$

Lemma 3.5 in [11] shows that

$$(\cos\mu + \sin\mu)L_{\overline{\omega}} - (L_{\overline{\omega}_1}\cos\mu + L_{\overline{\omega}_2}\sin\mu) \succ 0, \tag{28}$$
$$(\cos\mu + \sin\mu)L_{\overline{\omega}} - (L_{\overline{\omega}_1}\sin\mu + L_{\overline{\omega}_2}\cos\mu) \succ 0, \tag{29}$$
$$\begin{aligned}&[(\cos\mu + \sin\mu)L_{\overline{\omega}} - (L_{\overline{\omega}_1}\cos\mu + L_{\overline{\omega}_2}\sin\mu)]\\ &\quad\times[(\cos\mu + \sin\mu)L_{\overline{\omega}} - (L_{\overline{\omega}_1}\sin\mu + L_{\overline{\omega}_2}\cos\mu)] \succ 0.\end{aligned} \tag{30}$$

Thus, both $L_{\overline{\omega}}C(z)$ and $L_{\overline{\omega}}D(z)$ are positive definite and hence invertible. Premultiplying (27) by $\Delta x^{\mathrm{T}}(L_{\overline{\omega}}D(z))^{-1}$ and taking into account $A\Delta x = 0$, we have

$$\Delta x^{\mathrm{T}}(L_{\overline{\omega}}D(z))^{-1}(L_{\overline{\omega}}C(z))\Delta x = 0. \tag{31}$$

From (30) we obtain that the symmetric part of $(L_{\overline{\omega}}C(z))(L_{\overline{\omega}}D(z))$ is positive definite. Denoting $\overline{\Delta x} = (L_{\overline{\omega}}D(z))^{-1}\Delta x$, we have

$$\Delta x^{\mathrm{T}}(L_{\overline{\omega}}D(z))^{-1}(L_{\overline{\omega}}C(z))\Delta x = \overline{\Delta x}^{\mathrm{T}}(L_{\overline{\omega}}C(z))(L_{\overline{\omega}}D(z))\overline{\Delta x} \geqslant 0.$$

Then, it follows from (31) that $\overline{\Delta x} = 0$, which gives $\Delta x = 0$. Since A has full row rank, (26) implies $\Delta y = 0$. Thus, the null space of $H'(z)$ comprises only the origin, and hence $H'(z)$ is invertible. The proof is completed. $\square$

4. Algorithm description

Based on the smoothing function (9) introduced in the previous section, we propose a one-step smoothing Newton method for solving the SOCP as defined by (1) and (2). Under suitable assumptions, we show the well-definedness of our algorithm.

For any $z := (\mu, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^m$, we denote $\Psi(z) = \|G(z)\|^2$ and

$$f(z) := \|H(z)\|^2 = (\mathrm{e}^{\mu} - 1)^2 + \Psi(z). \tag{32}$$

Now we give a formal description of our algorithm.

A l g o r i t h m 4.1 (A one-step smoothing Newton method for SOCP).

Step 0: Choose an accuracy parameter $\varepsilon > 0$. Choose constants $\delta \in (0,1)$, $\sigma \in (0,1/2)$ and $\mu_0 \in \mathbb{R}_{++}$ and let $\overline{z} := (\mu_0, 0, 0)$. Let $(x_0, y_0) \in \mathbb{R}^n \times \mathbb{R}^m$ be an arbitrary initial point. Let $z_0 := (\mu_0, x_0, y_0)$ and $\eta = \|H(z_0)\| + 1$. Choose $\gamma \in (0,1)$ such that $\mu_0 \eta \gamma < 1$. Set $k := 0$.

Step 1: If $\|H(z_k)\| \leqslant \varepsilon$, then stop. Else, let

$$\beta_k := \beta(z_k) = \mathrm{e}^{\mu_k} \gamma \min\{1, f(z_k)\}. \tag{33}$$

Step 2: Compute $\Delta z_k := (\Delta \mu_k, \Delta x_k, \Delta y_k) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$ by

$$H(z_k) + H'(z_k)\Delta z_k = \beta_k \overline{z}. \tag{34}$$

Step 3: Let l_k be the smallest nonnegative integer l such that

$$f(z_k + \delta^l \Delta z_k) \leqslant [1 - 2\sigma(1 - \mu_0 \eta \gamma)\delta^l] f(z_k). \tag{35}$$

Let $\alpha_k := \delta^{l_k}$.

Step 4: Set $z_{k+1} := z_k + \alpha_k \Delta z_k$ and $k := k+1$. Go to Step 1.

Note that Algorithm 4.1 solves only one system of linear equations and performs only one Armijo-type line search at each iteration. Moreover, from Step 3 and Step 4, it is easy to see that the sequence $\{f(z_k)\}$ is monotonically decreasing, and hence, the sequence $\{\|H(z_k)\|\}$ is monotonically decreasing. In order to show the well-definedness of the algorithm, we need the following two lemmas.

Lemma 4.1 ([11]). *For any $\mu > 0$, one has*

$$-\mu \leqslant \frac{1 - \mathrm{e}^{\mu}}{\mathrm{e}^{\mu}} \leqslant -\mu \mathrm{e}^{-\mu}.$$

Lemma 4.2. *For any $z_k = (\mu_k, x_k, y_k) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^m$ generated by Algorithm* 4.1, *one has,*

$$\mathrm{e}^{\mu_k} - 1 \leqslant \sqrt{f(z_k)}, \quad \textit{and} \quad \mathrm{e}^{\mu_k} \leqslant \eta.$$

P r o o f. From (32) it is easy to find that $\mathrm{e}^{\mu_k} - 1 \leqslant \sqrt{f(z_k)}$. Thus, we have

$$\mathrm{e}^{\mu_k} \leqslant \sqrt{f(z_k)} + 1 = \|H(z_k)\| + 1.$$

Since the sequence $\{\|H(z_k)\|\}$ is monotonically decreasing, i.e., $\|H(z_{k+1})\| \leqslant \|H(z_k)\|$ for all $k \geqslant 0$, we have

$$\mathrm{e}^{\mu_k} \leqslant \|H(z_k)\| + 1 \leqslant \|H(z_0)\| + 1 = \eta.$$

This completes the proof. □

The following theorem shows that Algorithm 4.1 is well-defined and generates an infinite sequence with some good features.

Theorem 4.3. *Suppose that A has full row rank. Then Algorithm* 4.1 *is well-defined and generates an infinite sequence $\{z_k := (\mu_k, x_k, y_k)\}$ with $\mu_k > 0$ and $z_k \in \Omega$ for all $k \geqslant 0$, where*

$$\Omega := \{z := (\mu, x, y) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^m \colon \ \mu \geqslant \mu_0 \gamma \min\{1, f(z)\}\}.$$

P r o o f. First, we prove that $\mu_k > 0$ and Step 2 is well-defined for all $k \geqslant 0$. The proof is done by induction. Suppose that $\mu_k > 0$ for some k, e.g., it is satisfied for $k = 0$. Since A has full row rank, Theorem 3.2 shows that the matrix $H'(z_k)$ is non-singular. Thus, the system of equations (34) is solvable. Let $\Delta z_k := (\Delta\mu_k, \Delta x_k, \Delta y_k) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$ be the unique solution to (34). Then

$$\Delta\mu_k = \frac{1 - \mathrm{e}^{\mu_k}}{\mathrm{e}^{\mu_k}} + \frac{\beta_k \mu_0}{\mathrm{e}^{\mu_k}}. \tag{36}$$

Since $\mu_k > 0$, from (33) and Lemma 4.1 we obtain that for any $\alpha \in (0, 1)$

$$\begin{aligned} \mu_k + \alpha\Delta\mu_k &= \mu_k + \alpha\Big(\frac{1 - \mathrm{e}^{\mu_k}}{\mathrm{e}^{\mu_k}} + \frac{\beta_k \mu_0}{\mathrm{e}^{\mu_k}}\Big) \\ &\geqslant (1 - \alpha)\mu_k + \alpha\gamma\mu_0 \min\{1, f(z_k)\} > 0, \end{aligned} \tag{37}$$

which implies that $\mu_{k+1} > 0$. Therefore, $\mu_k > 0$ for all $k \geqslant 0$ and Step 2 is well-defined at the kth iteration.

Next we prove that Step 3 is well-defined. For any $\alpha \in (0, 1)$, by the Taylor expansion and (36), we have

$$\begin{aligned} \mathrm{e}^{\mu_k + \alpha\Delta\mu_k} - 1 &= \mathrm{e}^{\mu_k}\mathrm{e}^{\alpha\Delta\mu_k} - 1 \\ &= \mathrm{e}^{\mu_k}[1 + \alpha\Delta\mu_k + O(\alpha^2)] - 1 \\ &= \mathrm{e}^{\mu_k} - 1 + \alpha\mathrm{e}^{\mu_k}\Delta\mu_k + O(\alpha^2) \\ &= (1 - \alpha)(\mathrm{e}^{\mu_k} - 1) + \alpha\beta_k\mu_0 + O(\alpha^2). \end{aligned}$$

Due to the definition of β_k, we have

$$\beta_k = \mathrm{e}^{\mu_k}\gamma \leqslant \mathrm{e}^{\mu_k}\gamma\sqrt{f(z_k)}, \quad \text{or} \quad \beta_k = \mathrm{e}^{\mu_k}\gamma f(z_k) \leqslant \mathrm{e}^{\mu_k}\gamma\sqrt{f(z_k)}.$$

Thus, from Lemma 4.2, we obtain that

$$\begin{aligned} (\mathrm{e}^{\mu_k + \alpha\Delta\mu_k} - 1)^2 &= (1 - \alpha)^2(\mathrm{e}^{\mu_k} - 1)^2 + 2\alpha(1 - \alpha)\mu_0\beta_k(\mathrm{e}^{\mu_k} - 1) \\ &\quad + \alpha^2\beta_k^2\mu_0^2 + O(\alpha^2) \\ &\leqslant (1 - 2\alpha)(\mathrm{e}^{\mu_k} - 1)^2 + 2\alpha\mu_0\gamma\sqrt{f(z_k)}\mathrm{e}^{\mu_k}(\mathrm{e}^{\mu_k} - 1) + O(\alpha^2) \\ &\leqslant (1 - 2\alpha)(\mathrm{e}^{\mu_k} - 1)^2 + 2\alpha\mu_0\eta\gamma f(z_k) + O(\alpha^2). \end{aligned} \tag{38}$$

On the other hand, by (34) we have

$$G(z_k) + G'(z_k)\Delta z_k = 0,$$

which yields

$$\Psi'(z_k)\Delta z_k = -2\|G(z_k)\|^2 = -2\Psi(z_k).$$

Denote

$$g(\alpha) := \Psi(z_k + \alpha\Delta z_k) - \Psi(z_k) - \alpha\Psi'(z_k)\Delta z_k.$$

Noting that $g(\alpha) = o(\alpha)$, we have

$$\begin{aligned} \|G(z_k + \alpha\Delta z_k)\|^2 &= \Psi(z_k + \alpha\Delta z_k) \\ &= \Psi(z_k) + \alpha\Psi'(z_k)\Delta z_k + g(\alpha) \\ &= \Psi(z_k) - 2\alpha\Psi(z_k) + o(\alpha) \\ &= (1 - 2\alpha)\Psi(z_k) + o(\alpha). \end{aligned} \tag{39}$$

Therefore,

$$\begin{aligned} f(z_k + \alpha\Delta z_k) &= \|H(z_k + \alpha\Delta z_k)\|^2 \\ &= (\mathrm{e}^{\mu_k + \alpha\Delta\mu_k} - 1)^2 + \|G(z_k + \alpha\Delta z_k)\|^2 \\ &\leqslant (1 - 2\alpha)(\mathrm{e}^{\mu_k} - 1)^2 + 2\alpha\mu_0\eta\gamma f(z_k) + (1 - 2\alpha)\Psi(z_k) + o(\alpha) \\ &= (1 - 2\alpha)f(z_k) + 2\alpha\mu_0\eta\gamma f(z_k) + o(\alpha) \\ &= [1 - 2(1 - \mu_0\eta\gamma)\alpha]f(z_k) + o(\alpha). \end{aligned} \tag{40}$$

Since $\mu_0\eta\gamma < 1$, there exists a constant $\overline{\alpha} \in (0, 1)$ such that

$$f(z_k + \alpha\Delta z_k) \leqslant [1 - 2\sigma(1 - \mu_0\eta\gamma)\alpha]f(z_k)$$

holds for any $\alpha \in (0, \overline{\alpha}]$ and $\sigma \in (0, 1/2)$. This demonstrates that Step 3 is well-defined at the kth iteration.

Finally, we prove $z_k \in \Omega$ for all $k \geqslant 0$ by induction on k. Obviously,

$$\mu_0 \geqslant \mu_0\gamma \min\{1, f(z_0)\}$$

which gives $z_0 \in \Omega$. Suppose that $z_k \in \Omega$, i.e., $\mu_k \geqslant \mu_0\gamma\min\{1, f(z_k)\}$. Then by (33) and (36), also using Lemma 4.1, we have

$$\begin{aligned}
\mu_{k+1} &= \mu_k + \alpha_k\Delta\mu_k \\
&= \mu_k + \alpha_k\Big(\frac{1-\mathrm{e}^{\mu_k}}{\mathrm{e}^{\mu_k}} + \frac{\beta_k\mu_0}{\mathrm{e}^{\mu_k}}\Big) \\
&\geqslant \mu_k + \alpha_k\Big(-\mu_k + \frac{\mathrm{e}^{\mu_k}\gamma\min\{1, f(z_k)\}\mu_0}{\mathrm{e}^{\mu_k}}\Big) \\
&\geqslant (1-\alpha_k)\mu_k + \alpha_k\mu_0\gamma\min\{1, f(z_k)\} \\
&\geqslant (1-\alpha_k)\mu_0\gamma\min\{1, f(z_k)\} + \alpha_k\mu_0\gamma\min\{1, f(z_k)\} \\
&= \mu_0\gamma\min\{1, f(z_k)\} \\
&\geqslant \mu_0\gamma\min\{1, f(z_{k+1})\},
\end{aligned}$$

where the last inequality follows from the fact that $f(z_{k+1}) \leqslant f(z_k)$. The proof is completed. □

5. Convergence analysis

In this section we show that any accumulation point of the iteration sequence $\{z_k := (\mu_k, x_k, y_k)\}$ is a solution to the system $H(z) = 0$. If the accumulation point z^* satisfies a nonsingularity assumption, then the iteration sequence converges to z^* locally quadratically without strict complementarity. First, we establish its global convergence.

Theorem 5.1 (Global convergence)**.** *Suppose that A has full row rank and that $\{z_k\}$ is the iteration sequence generated by Algorithm* 4.1*. Then any accumulation point $z^* = (\mu^*, x^*, y^*)$ of $\{z_k\}$ is a solution to $H(z) = 0$.*

Proof. Without loss of generality, we assume that z^* is the limit point of the sequence $\{z_k\}$ as $k \to \infty$. Since $\{f(z_k)\}$ is monotonically decreasing and bounded from below by zero, it follows from the continuity of $f(\cdot)$ that $\{f(z_k)\}$ converges to a non-negative number $f(z^*)$. If $f(z^*) = 0$, i.e., $H(z^*) = 0$, we obtain the desired result. Suppose that $f(z^*) > 0$. Since $z^* \in \Omega$, we have

$$\mu^* \geqslant \mu_0\gamma\min\{1, f(z^*)\} > 0$$

by Theorem 4.3. It follows from Theorem 3.2 that $H'(z^*)$ exists and is invertible. Hence, there exists a closed neighborhood $N(z^*)$ of z^* such that for any $z \in N(z^*)$

we have $\mu \in \mathbb{R}_{++}$ and $H'(z)$ is invertible. Then, for any $z \in N(z^*)$, let $\Delta z := (\Delta\mu, \Delta x, \Delta y) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m$ be the unique solution to the system of equations

$$H(z) + H'(z)\Delta z = \beta(z)\overline{z}.$$

Denote

$$g_z(\alpha) := \Psi(z + \alpha\Delta z) - \Psi(z) - \alpha\Psi'(z)\Delta z.$$

Then, for any $z \in N(z^*)$, we have $\lim\limits_{\alpha\to 0} |g_z(\alpha)|/\alpha = 0$. Similarly to the proof of Theorem 4.3, for any $\alpha \in (0, 1)$ and $z \in N(z^*)$, we have

$$\begin{gathered}
\mu + \alpha\Delta\mu > 0, \\
(\mathrm{e}^{\mu+\alpha\Delta\mu} - 1)^2 \leqslant (1 - 2\alpha)(\mathrm{e}^{\mu} - 1)^2 + 2\alpha\mu_0\eta\gamma f(z) + O(\alpha^2), \\
\|G(z + \alpha\Delta z)\|^2 = (1 - 2\alpha)\Psi(z) + o(\alpha).
\end{gathered}$$

Thus

$$f(z + \alpha\Delta z) = (\mathrm{e}^{\mu+\alpha\Delta\mu} - 1)^2 + \|G(z + \alpha\Delta z)\|^2 \leqslant [1 - 2(1 - \mu_0\eta\gamma)\alpha]f(z) + o(\alpha).$$

Hence, we can find a positive number $\overline{\alpha} \in (0, 1]$ such that

$$f(z + \alpha\Delta z) \leqslant [1 - 2\sigma(1 - \mu_0\eta\gamma)\alpha]f(z)$$

holds for any $\alpha \in (0, \overline{\alpha}]$, $\sigma \in (0, 1/2)$, and $z \in N(z^*)$. Therefore, for all sufficiently large k, there exists a nonnegative integer $\bar{l}$ such that $\delta^{\bar{l}} \in (0, \overline{\alpha}]$ and

$$f(z_k + \delta^{\bar{l}}\Delta z_k) \leqslant [1 - 2\sigma(1 - \mu_0\eta\gamma)\delta^{\bar{l}}]f(z_k).$$

For all sufficiently large k, since $\alpha_k = \delta^{l_k} \geqslant \delta^{\bar{l}}$, it follows from Step 3 and Step 4 in Algorithm 4.1 that

$$f(z_{k+1}) \leqslant [1 - 2\sigma(1 - \mu_0\eta\gamma)\delta^{l_k}]f(z_k) \leqslant [1 - 2\sigma(1 - \mu_0\eta\gamma)\delta^{\bar{l}}]f(z_k),$$

which implies that $f(z_{k+1}) \leqslant Cf(z_k)$, where $C = 1-2\sigma(1-\mu_0\eta\gamma)\delta^{\bar{l}} < 1$ is a constant and thus $\{f(z_k)\} \to 0$ as $k \to \infty$. This contradicts the fact that the sequence $\{f(z_k)\}$ converges to $f(z^*) > 0$. The proof is completed. $\square$

Next, we give the rate of convergence for Algorithm 4.1. To establish the locally quadratic convergence of our smoothing Newton method, we assume that z^* satisfies the nonsingularity condition but may not satisfy the strict complementarity.

Theorem 5.2 (Local convergence)**.** *Suppose that A has full row rank and that $z^* = (\mu^*, x^*, y^*)$ is an accumulation point of the iteration sequence $\{z_k\}$ generated by Algorithm 4.1. If all $V \in \partial H(z^*)$ are nonsingular, then we have*

(i) $z_{k+1} = z_k + \Delta z_k$ *for all* z_k *sufficiently close to* z^*.

(ii) *The whole sequence* $\{z_k\}$ *converges to* z^* *quadratically, i.e.,*

$$\|z_{k+1} - z^*\| = O(\|z_k - z^*\|^2);$$

moreover,

$$\mu_{k+1} = O(\mu_k^2).$$

Proof. By using Theorem 3.1 and Theorem 3.2, we can prove the theorem similarly to Theorem 8 in [21]. For brevity, we omit the details here. □

6. Numerical experiments

In order to evaluate the efficiency of Algorithm 4.1, we have conducted some numerical experiments. All experiments were performed on a personal computer with 2.0 GB memory and Intel(R) Pentium(R) Dual-Core CPU 2.93 GHz×2. The operating system was Windows XP (SP3) and the computer codes were all written in Matlab 7.0.1.

Throughout the experiments, we used $\|H(z_k)\| \leqslant \varepsilon$ as the stopping criterion. The starting points were chosen to be $x_0 = \mathbf{e} \in \mathbb{R}^n$, $y_0 = 0 \in \mathbb{R}^m$, and the parameters used in the algorithm were chosen as follows:

$$\mu_0 = 0.01, \quad \sigma = 0.25, \quad \delta = 0.75, \quad \gamma = \frac{1}{1 + \|H(z_0)\|}.$$

First, we test the following problems with random data which are representative to some extent.

Problem $\mathcal{P}$. We consider the SOCP problem (1) with a single SOC. Its data are given as follows:

$$B = \begin{pmatrix} 100 & 2 & & & \\ -2 & 100 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & -2 & 100 & 2 \\ & & & -2 & 100 \end{pmatrix} \in \mathbb{R}^{m\times m}, \quad A = [B\,\mathbf{randn}(m, n-m)],$$

$$b = 10\mathbf{e}_m + 4\,\mathbf{rand}(m,1) - 2\,\mathbf{ones}(m,1) \quad c = 10\mathbf{e}_n + 4\,\mathbf{rand}(n,1) - 2\,\mathbf{ones}(n,1),$$

where $\mathbf{ones}(m,1) = (1,\dots,1)^{\mathrm{T}} \in \mathbb{R}^m$ and $\mathbf{e}_n := (1;0) \in \mathbb{R}\times\mathbb{R}^{n-1}$.

Numerical results for Algorithm 4.1 on problem $\mathcal{P}$ are displayed in Tab. 1, in which **IT** denotes the number of iterations and **CPU** denotes the CPU time in seconds needed for obtaining the optimal solution satisfying the stopping rule. Let **FV** and **MU** denote the values of $\|H(z_k)\|$ and μ_k at the final iterate, respectively. The results in Tab. 1 indicate that our smoothing method is efficient and can deal with sparse SOCP problems.

m	n	IT	CPU	FV	MU
5	10	6	0.006	2.6947×10^{-7}	2.1580×10^{-10}
10	20	7	0.009	1.5946×10^{-8}	1.4718×10^{-11}
20	40	7	0.014	2.3326×10^{-8}	1.8568×10^{-11}
25	50	7	0.025	2.6041×10^{-8}	2.4911×10^{-11}
30	60	8	0.042	2.0317×10^{-9}	1.7866×10^{-12}
80	120	11	0.174	9.1693×10^{-8}	6.1451×10^{-11}
150	200	15	0.262	4.2141×10^{-9}	6.2541×10^{-11}

Table 1. Numerical results for Algorithm 4.1 on problem $\mathcal{P}$ with $\varepsilon = 10^{-6}$.

Next, we randomly generate test problems with size n $(= 2m)$ and $r = 1$. In detail, we generate a random matrix $A \in \mathbb{R}^{m\times n}$ with full row rank and random vectors $x \in \mathcal{L}^0$, $s \in \mathcal{L}^0$, $y \in \mathbb{R}^m$, and then let $b := Ax$, $c := A^{\mathrm{T}}y + s$. Thus the generated problems (1) and (2) have optimal solutions and their optimal values coincide, because they have strictly feasible points. The random problems of each case are generated 10times, and the tested results are listed in Tabs. 2–3, where **LIT** denotes the largest value of the iterative numbers; **AIT** denotes the average value of the iterative numbers; **LCPU** denotes the largest value of the CPU time in seconds; **ACPU** denotes the average value of the CPU time in seconds; **LFV** and **LMU** denote the largest values of $\|H(z_k)\|$ and μ_k, and **SFV** and **SMU** denote the smallest values of $\|H(z_k)\|$ and μ_k, and **AFV** denotes the average values of $\|H(z_k)\|$ when the algorithm terminates within ten testings. From Tab. 2, we see that our algorithm can solve all the test problems and can deal with large-scale SOCP problems. The algorithm can find a solution point meeting the desired accuracy in very few iterations and in short CPU time. Moreover, we may observe that the number of iterations obviously does not change, but the CPU time grows with the problem size. To compare Tab. 2 and Tab. 3, we can find that the number of iterations and the CPU time have slightly changed when the stopping criterion becomes smaller. In addition, we also did our numerical experiments with different μ_0, σ, δ and it is shown that there are slight changes in results for other values of μ_0, σ, δ.

Finally, for given sizes m and n, we randomly generate 6 test problems. For comparison purpose, we also use SDPT3 [27] to solve the same problems. The results are listed in Tab. 4 which indicates that Algorithm 4.1 performs very well. We also

m	n	LIT	AIT	LCPU	ACPU	LFV	SFV	LMU	SMU
50	100	6	5.8	0.08	0.05	3.772×10^{-7}	3.529×10^{-10}	1.321×10^{-8}	9.135×10^{-12}
100	200	6	6.0	0.25	0.22	3.941×10^{-7}	1.096×10^{-9}	1.834×10^{-8}	4.363×10^{-11}
150	300	6	6.0	1.15	1.12	8.141×10^{-7}	8.055×10^{-10}	3.144×10^{-8}	4.202×10^{-11}
200	400	6	6.0	2.60	2.55	2.115×10^{-7}	4.059×10^{-9}	1.213×10^{-8}	1.764×10^{-10}
250	500	7	6.1	5.65	4.91	8.577×10^{-7}	6.127×10^{-13}	6.203×10^{-8}	1.097×10^{-14}
300	600	7	6.2	9.38	8.17	8.690×10^{-7}	9.887×10^{-13}	6.965×10^{-8}	1.662×10^{-14}
350	700	7	6.3	15.08	13.58	9.903×10^{-7}	5.573×10^{-13}	7.960×10^{-8}	4.844×10^{-13}
400	800	7	6.3	21.24	18.86	8.749×10^{-7}	1.037×10^{-12}	7.701×10^{-8}	1.289×10^{-14}
450	900	7	6.5	29.68	28.32	9.890×10^{-7}	1.274×10^{-12}	9.481×10^{-8}	1.393×10^{-14}
500	1000	7	6.8	41.21	39.19	9.981×10^{-7}	1.462×10^{-12}	1.034×10^{-8}	1.697×10^{-14}

Table 2. Numerical results for Algorithm 4.1 with $\varepsilon = 10^{-6}$.

m	n	$\varepsilon = 10^{-8}$			$\varepsilon = 10^{-12}$		
		AIT	ACPU	AFV	AIT	ACPU	AFV
100	200	6.8	0.388	5.7858×10^{-10}	7.0	0.409	9.6677×10^{-14}
200	400	7.0	2.947	3.3171×10^{-13}	7.0	3.008	3.3171×10^{-13}
300	600	7.0	9.761	7.3236×10^{-13}	7.2	10.913	5.7595×10^{-13}
400	800	7.0	20.675	1.5158×10^{-12}	7.6	22.603	8.2433×10^{-13}
500	1000	7.0	40.429	2.5215×10^{-12}	8.0	44.568	8.9905×10^{-13}

Table 3. Numerical results for Algorithm 4.1 with different stopping criterion.

m	n	Algorithm 4.1			SDPT3		
		IT	CPU	FV	IT	CPU	FV
50	50	5	0.07	3.8172×10^{-11}	6	0.09	2.5967×10^{-10}
50	100	5	0.08	1.5764×10^{-12}	7	0.14	3.2475×10^{-11}
80	80	6	0.12	2.2513×10^{-11}	8	0.17	1.8143×10^{-9}
80	150	7	0.10	7.1125×10^{-10}	8	0.19	3.6571×10^{-8}
100	200	6	0.16	3.1569×10^{-9}	9	0.23	4.3627×10^{-7}
150	150	8	0.20	5.3411×10^{-10}	9	0.26	2.7124×10^{-7}

Table 4. Comparison of Algorithm 4.1 and SDPT3 on SOCPs with $\varepsilon = 10^{-6}$.

obtained similar results for other random examples. Therefore, Algorithm 4.1 may be of practical interest.

7. Conclusions

In this paper, we introduce a new smoothing function of the well-known Fischer-Burmeister function. Based on this new function, we present a one-step smoothing Newton method for solving second-order cone programming. The proposed algorithm solves only one system of linear equations and performs only one line search at each iteration. This algorithm can start from an arbitrary point and is quadratically convergent under a mild assumption. We also report some preliminary computational experiments. The preliminary numerical results demonstrate that the proposed algorithm is promising.

Acknowledgement. The authors would like to thank the referees for their valuable comments and suggestions, which have considerably improved the paper.

References

[1] *F. Alizadeh, D. Goldfarb*: Second-order cone programming. Math. Program. *95* (2003), 3–51.
[2] *Y. Q. Bai, G. Q. Wang, C. Roos*: Primal-dual interior-point algorithms for second-order cone optimization based on kernel functions. Nonlinear Anal., Theory Methods Appl. *70* (2009), 3584–3602.
[3] *B. Chen, N. Xiu*: A global linear and local quadratic non-interior continuation method for nonlinear complementarity problems based on Chen-Mangasarian smoothing functions. SIAM J. Optim. *9* (1999), 605–623.
[4] *J. S. Chen, P. Tseng*: An unconstrained smooth minimization reformulation of the second-order cone complementarity problem. Math. Program., Ser B *104* (2005), 293–327.
[5] *F. H. Clarke*: Optimization and Nonsmooth Analysis. Wiley & Sons, New York, 1983, Reprinted by SIAM, Philadelphia, 1990.
[6] *R. Debnath, M. Muramatsu, H. Takahashi*: An efficient support vector machine learning method with second-order cone programming for large-scale problems. Appl. Intel. *23* (2005), 219–239.
[7] *J. Faraut, A. Korányi*: Analysis on Symmetric Cones. Clarendon Press, Oxford, 1994.
[8] *L. Faybusovich*: Euclidean Jordan algebras and interior-point algorithms. Positivity *1* (1997), 331–357.
[9] *M. Fukushima, Z.-Q. Luo, P. Tseng*: Smoothing functions for second-order-cone complementarity problems. SIAM J. Optim. *12* (2002), 436–460.
[10] *S. Hayashi, N. Yamashita, M. Fukushima*: A combined smoothing and regularization method for monotone second-order cone complementarity problems. SIAM J. Optim. *15* (2005), 593–615.
[11] *H. Jiang*: Smoothed Fischer-Burmeister equation methods for the complementarity problem. Technical Report. Department of Mathematics, The University of Melbourne, Parille, Victoria, Australia, June 1997.
[12] *Y.-J. Kuo, H. D. Mittelmann*: Interior point methods for second-order cone programming and OR applications. Comput. Optim. Appl. *28* (2004), 255–285.
[13] *M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret*: Applications of second-order cone programming. Linear Algebra Appl. *284* (1998), 193–228.
[14] *R. D. C. Monteiro, T. Tsuchiya*: Polynomial convergence of primal-dual algorithms for the second order program based the MZ-family of directions. Math. Program. *88* (2000), 61–83.
[15] *C. Ma, X. Chen*: The convergence of a one-step smoothing Newton method for P_0-NCP based on a new smoothing NCP-function. J. Comput. Appl. Math. *216* (2008), 1–13.
[16] *R. Mifflin*: Semismooth and semiconvex functions in constrained optimization. SIAM J. Control Optim. *15* (1977), 959–972.
[17] *Y. E. Nesterov, M. J. Todd*: Primal-dual interior-point methods for self-scaled cones. SIAM J. Optim. *8* (1998), 324–364.

[18] *X. Peng, I. King*: Robust BMPM training based on second-order cone programming and its application in medical diagnosis. Neural Networks *21* (2008), 450–457.
[19] *J. Peng, C. Roos, T. Terlaky*: A new class of polynomial primal-dual interior-point methods for second-order cone optimization based on self-regular proximities. SIAM J. Optim. *13* (2002), 179–203.
[20] *L. Qi, D. Sun*: Improving the convergence of non-interior point algorithms for nonlinear complementarity problems. Math. Comput. *69* (2000), 283–304.
[21] *L. Qi, D. Sun, G. Zhou*: A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities. Math. Program. *87* (2000), 1–35.
[22] *L. Qi, J. Sun*: A nonsmooth version of Newton's method. Math. Program. *58* (1993), 353–367.
[23] *P. K. Shivaswamy, C. Bhattacharyya, A. J. Smola*: Second order cone programming approaches for handling missing and uncertain data. J. Mach. Learn. Res. *7* (2006), 1283–1314.
[24] *S. Hayashi, N. Yamashita, M. Fukushima*: A combined smoothing and regularization method for monotone second-order cone complementarity problems. SIAM J. Optim. *15* (2005), 593–615.
[25] *T. Sasakawa, T. Tsuchiya*: Optimal magnetic shield design with second-order cone programming. SIAM J. Sci. Comput. *24* (2003), 1930–1950.
[26] *D. Sun, J. Sun*: Strong semismoothness of the Fischer-Burmeister SDC and SOC complementarity functions. Math. Program., Ser A. *103* (2005), 575–581.
[27] *K. C. Toh, R. H. Tütüncü, M. J. Todd*: SDPT3 Version 3.02—A MATLAB software for semidefinite-quadratic-linear programming. `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`.
[28] *P. Tseng*: Error bounds and superlinear convergence analysis of some Newton-type methods in optimization. Nonlinear Optimization and Related Topics (G. Di Pillo, F. Giannessi, eds.). Kluwer Academic Publishers, Dordrecht; Appl. Optim. *36* (2000), 445–462.

Authors' addresses: *J. Tang*, Department of Mathematics, Shanghai Jiaotong University, 200240 Shanghai, P. R. China, and College of Mathematics and Information Science, Xinyang Normal University, 464000 Xinyang, P. R. China, e-mail: `tangjingyong926@163.com`; *G. He*, College of Information Science and Engineering, Shandong University of Science and Technology, 266510 Qingdao, P. R. China, e-mail: `hegp@263.net`; *L. Dong* (corresponding author), College of Mathematics and Information Science, Xinyang Normal University, 464000 Xinyang, P. R. China, e-mail: `citycity926@163.com`; *L. Fang*, College of Mathematics and Systems Science, Taishan University, 271012 Tai'an, P. R. China, e-mail: `fangliang3@163.com`.