Jingyong Tang; Yuefen Chen

Smoothing functions and algorithm for nonsymmetric circular cone complementarity problems

Applications of Mathematics, Vol. 67 (2022), No. 2, 209–231

Persistent URL: http://dml.cz/dmlcz/149567

SMOOTHING FUNCTIONS AND ALGORITHM FOR NONSYMMETRIC CIRCULAR CONE COMPLEMENTARITY PROBLEMS

Jingyong Tang, Yuefen Chen, Xinyang

Received May 6, 2020. Published online December 7, 2021.

Abstract. There has been much interest in studying symmetric cone complementarity problems. In this paper, we study the circular cone complementarity problem (denoted by CCCP) which is a type of nonsymmetric cone complementarity problem. We first construct two smoothing functions for the CCCP and show that they are all coercive and strong semismooth. Then we propose a smoothing algorithm to solve the CCCP. The proposed algorithm generates an infinite sequence such that the value of the merit function converges to zero. Moreover, we show that the iteration sequence must be bounded if the solution set of the CCCP is nonempty and bounded. At last, we prove that the proposed algorithm has local superlinear or quadratical convergence under some assumptions which are much weaker than Jacobian nonsingularity assumption. Some numerical results are reported which demonstrate that our algorithm is very effective for solving CCCPs.

Keywords: circular cone complementarity problem; smoothing function; smoothing algorithm; superlinear/quadratical convergence

MSC 2020: 90C25, 90C30, 65K05

1. Introduction

The *second-order cone* in $\mathbb{R}^n$, also called the Lorentz cone, is defined as

$$\mathbb{K}^n = \{x = (x_1, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1};\ \|\bar{x}\| \leqslant x_1\},$$

where $\|\cdot\|$ denotes the Euclidean norm. The second-order cone $\mathbb{K}^n$ is a special symmetric cone and it is self-dual, i.e., $\mathbb{K}^n = (\mathbb{K}^n)^*$, where $(\mathbb{K}^n)^*$ is the dual cone of $\mathbb{K}^n$

This work was supported by the National Natural Science Foundation of China (11601466), Natural Science Foundation of Henan Province (202300410343) and the Training Plan of Young Key Teachers in Universities of Henan Province (2018GGJS096).

DOI: 10.21136/AM.2021.0129-20

defined by

$$(\mathbb{K}^n)^* = \{s \in \mathbb{R}^n;\ \langle s, x\rangle \geqslant 0\ \forall\, x \in \mathbb{K}^n\},$$

in which $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. In recent years, optimization problems with second-order cone constraints, such as second-order cone programming (SOCP) and second-order cone complementarity problem (SOCCP), have received considerable attention from researchers for its wide range of applications in many fields such as engineering, optimal control and design, machine learning, robust optimization and combinatorial optimization and so on (see [1], [6], [24]).

In this paper, we are interested in the *circular cone*, which is a type of nonsymmetric cone. The circular cone is a pointed closed convex cone having hyper-spherical sections orthogonal to its axis of revolution about which the cone is invariant to rotation. Let its half-aperture angle be θ with $\theta \in (0, \pi/2)$. Then the n-dimensional circular cone denoted by $\mathbb{C}^n_\theta$ can be expressed as

$$\begin{aligned}\mathbb{C}^n_\theta &= \{x = (x_1, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1};\ \|x\| \cos\theta \leqslant x_1\}\\ &= \{x = (x_1, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1};\ \|\bar{x}\| \leqslant x_1 \tan\theta\}.\end{aligned}$$

The circular cone $\mathbb{C}^n_\theta$ becomes the second-order cone $\mathbb{K}^n$ when $\theta = \pi/4$. Zhou and Chen [37] proved that the dual cone of $\mathbb{C}^n_\theta$ denoted by $(\mathbb{C}^n_\theta)^*$ can be expressed as

$$\begin{aligned}(\mathbb{C}^n_\theta)^* &= \{x = (x_1, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1};\ \|x\| \sin\theta \leqslant x_1\}\\ &= \{x = (x_1, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1};\ \|\bar{x}\| \leqslant x_1 \operatorname{ctan}\theta\}.\end{aligned}$$

Thus, when $\theta \neq \pi/4$, the circular cone $\mathbb{C}^n_\theta$ is not self-dual and hence it is not a symmetric cone. Many authors investigated theoretical properties of the circular cone. Alzalg [2] analyzed the algebraic structure of the circular cone. Zhou et al. [37], [38], [39] studied circular cone convexity, second order regularity of the circular cone and spectral factorization associated with the circular cone, and conducted variational analysis of circular cone programs. In [40], Zhou et al. studied the parabolic second-order directional derivative in the Hadamard sense of a vector-valued function associated with the circular cone.

Recently, many numerical algorithms have been studied for solving various optimization problems over the circular cone. Bai et al. [3], [4], Kheirfam and Wang [22] and Ma et al. [25] studied interior-point methods for circular cone programming. Chi et al. [13], [14] investigated nonmonotone smoothing Newton algorithms for circular cone programming. Chi et al. [12] proposed a regularized inexact smoothing Newton method for circular cone complementarity problem. Ke et al. [21] established a class of relaxation modulus-based matrix splitting iteration methods for circular cone nonlinear complementarity problems. Pirhaji et al. [29] studied a path following interior-point method for circular cone linear complementarity problems. Miao et al. [27] proposed a generalized Newton method for absolute value equations associated with circular cones. Lately, Miao et al. [26] constructed some complementarity functions and merit functions for the circular cone complementarity problem (CCCP), which finds $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ such that

$$\text{(CCCP) } x \in \mathbb{C}_\theta^n,\ y \in (\mathbb{C}_\theta^n)^*,\ \langle x, y\rangle = 0,\ y = F(x), \tag{1.1}$$

where $F\colon\ \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable function. When the half-aperture angle $\theta = \pi/4$, the CCCP reduces to the SOCCP. Thus, the CCCP can be viewed as the generalization of the SOCCP. Notice that when $\theta \neq \pi/4$, the circular cone $\mathbb{C}_\theta^n$ is not symmetric and in this case the CCCP is a type of nonsymmetric cone complementarity problem.

On the other hand, there has been much interest in smoothing-type algorithms for solving optimization problems over the circular cone and the second-order cone (e.g., [8], [9], [11], [10], [13], [14], [12], [16], [34], [33]). The main idea of this class of algorithms is to use a smoothing function to reformulate the problem concerned as a system of smooth nonlinear equations and then solve it by Newton's method. It is worth pointing out that to obtain local fast convergence rate, all these smoothing-type algorithms need the Jacobian nonsingularity assumption, which seems to be unnecessarily restrictive.

In this paper we aim to construct two smoothing functions and design a smoothing algorithm for the CCCP given in (1.1). Specifically, based on the relationship between the second-order cone and the circular cone, we introduce two smoothing functions for the CCCP and show that they are coercive and strong semismooth. By using these smoothing functions, we reformulate the CCCP as a system of smooth nonlinear equations and propose a smoothing algorithm to solve it. Different with existing smoothing-type algorithms, the proposed algorithm adopts a new line search technique, which is well-defined and easy to implement. We show that any accumulation point of the iteration sequence generated by this algorithm is a solution of the CCCP. Moreover, we prove that if the solution set of the CCCP is nonempty and bounded, then the generated iteration sequence must be bounded. At last, we prove that the proposed algorithm has local superlinear or quadratical convergence rate under some assumptions, which are much weaker than Jacobian nonsingularity assumption. We also give some numerical results which demonstrate that the proposed algorithm is very effective for solving CCCPs.

The paper is organized as follows. In the next section, we give some preliminary results. In Section 3, we introduce two smoothing functions for the CCCP. In Section 4, we reformulate the CCCP as a system of smooth nonlinear equations. In

Section 5, we propose a smoothing algorithm for solving the CCCP and show its well-definedness. The global and local superlinear/quadratical convergence of our algorithm are analyzed in Section 6. Numerical results are reported in Section 7. Some conclusions are given in Section 8.

Throughout the paper, $\mathbb{R}^n$ denotes the space of n-dimensional real column vectors and $\mathbb{R}^n_+$ (or $\mathbb{R}^n_{++}$) denotes the nonnegative (or positive) orthant in $\mathbb{R}^n$. For convenience, we write $(u_1^\top, \ldots, u_m^\top)^\top$ as $(u_1, \ldots, u_m)$ for any vectors $u_i \in \mathbb{R}^n$. Further I_m represents the $m \times m$ identity matrix and $\|\cdot\|$ denotes the Euclidean norm. For any $x, y \in \mathbb{R}^n$, we write $x \succeq_{\mathbb{K}^n} y$ (or $x \succ_{\mathbb{K}^n} y$) if $x - y \in \mathbb{K}^n$ (or $x - y \in \operatorname{int} \mathbb{K}^n$, where $\operatorname{int} \mathbb{K}^n$ denotes the interior of $\mathbb{K}^n$). For a given set $S \subset \mathbb{R}^n$, $\operatorname{conv}(S)$ denotes the convex hull of S. For any $a, b > 0$, $a = O(b)$ (or $a = o(b)$) means that a/b is uniformly bounded (or tends to zero) as $b \to 0$.

2. Preliminaries

For any vectors $x = (x_1, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$ and $y = (y_1, \overline{y}) \in \mathbb{R} \times \mathbb{R}^{n-1}$, in the setting of $\mathbb{K}^n$, the *Jordan product* of x and y is defined by

$$x \circ y = (x^\top y, x_1 \overline{y} + y_1 \bar{x}).$$

The Jordan product "$\circ$", unlike scalar or matrix multiplication, is not associative. The identity element under this product is $e := (1, 0, \ldots, 0)^\top \in \mathbb{R}^n$. For any x and y in $\mathbb{R}^n$, from [17], Proposition 2.1, we have

$$x \in \mathbb{K}^n,\ y \in \mathbb{K}^n,\ x \circ y = 0 \Leftrightarrow x \in \mathbb{K}^n,\ y \in \mathbb{K}^n,\ \langle x, y \rangle = 0.$$

For any $x = (x_1, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{n-1}$, its *spectral decomposition* with respect to $\mathbb{K}^n$ is

$$x = \lambda_1(x) c_1 + \lambda_2(x) c_2, \tag{2.1}$$

where $\lambda_1(x), \lambda_2(x)$ are spectral values and c_1, c_2 are spectral vectors, which are defined by

$$\lambda_i(x) = x_1 + (-1)^i \|\bar{x}\|, \quad c_i = \begin{cases} \dfrac{1}{2}\Big(1, (-1)^i \dfrac{\bar{x}}{\|\bar{x}\|}\Big), & \bar{x} \neq 0, \\ \dfrac{1}{2}(1, (-1)^i \omega), & \bar{x} = 0, \end{cases} \quad i = 1, 2 \tag{2.2}$$

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

For any $x = (x_1, \bar{x}) \in \mathbb{R}\times\mathbb{R}^{n-1}$ with its spectral decomposition given in (2.1)–(2.2), we define

$$x^2 := \lambda_1(x)^2 c_1 + \lambda_2(x)^2 c_2.$$

Moreover, if $x \in \mathbb{K}^n$, then $\lambda_2(x) \geqslant \lambda_1(x) \geqslant 0$ and we define

$$\sqrt{x} := \sqrt{\lambda_1(x)}c_1 + \sqrt{\lambda_2(x)}c_2.$$

It is easy to verify that $x^2 = x \circ x$ and $x = \sqrt{x} \circ \sqrt{x}$.

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

$$L_x := \begin{bmatrix} x_1 & \bar{x}^\top \\ \bar{x} & x_1 I_{n-1} \end{bmatrix}.$$

Here L_x can be viewed as a linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$ given by $L_x y = x \circ y$ for any $x, y \in \mathbb{R}^n$. Notice that L_x is positive semidefinite (or positive definite) if and only if $x \in \mathbb{K}^n$ (or $x \in \operatorname{int} \mathbb{K}^n$).

Now we introduce the definition of (strong) semismoothness. Let $H\colon \mathbb{R}^n \to \mathbb{R}^m$ be a locally Lipschitz continuous function. Then H is differentiable almost everywhere by Rademacher's theorem. Let $D_H \subseteq \mathbb{R}^n$ be the set of points at which H is differentiable.

The limiting Jacobian of H at x defined by

$$\partial_B H(x) := \Big\{V \in \mathbb{R}^{m\times n};\ V = \lim_{x^k \to x} H'(x^k),\ \{x^k\} \subseteq D_H\Big\}$$

is called the B-subdifferential of H at x. The Clarke's generalized Jacobian of H at x is $\partial H(x) := \operatorname{conv}(\partial_B H(x))$. We say H is directionally differentiable at x along the direction d if

$$H'(x; d) := \lim_{t\downarrow 0} \frac{H(x + td) - H(x)}{t}$$

exists, where $H'(x; d)$ is called the directional derivative of H at x along the direction d and H is directionally differentiable at x if H is directionally differentiable at x along any direction $d \neq 0$.

Definition 2.1. Let $H\colon \mathbb{R}^n \to \mathbb{R}^m$ be a locally Lipschitz continuous function around $x \in \mathbb{R}^n$. We say that H is semismooth at x if H is directionally differentiable at x and for any $y \to x$ and $V \in \partial H(y)$,

$$H(y) - H(x) - V(y - x) = o(\|y - x\|).$$

H is further said to be strongly semismooth at x if H is semismooth at x and for any $y \to x$ and $V \in \partial H(y)$,

$$H(y) - H(x) - V(y - x) = O(\|y - x\|^2).$$

3. Smoothing functions for the CCCP

For any $\theta \in (0, \pi/2)$ we denote $A := \begin{bmatrix} \tan\theta & 0 \\ 0 & I_{n-1} \end{bmatrix}$. Then A is positive definite with its inverse matrix $A^{-1} := \begin{bmatrix} \operatorname{ctan}\theta & 0 \\ 0 & I_{n-1} \end{bmatrix}$. Zhou and Chen characterized the relationship between the circular cone $\mathbb{C}^n_\theta$ and the second-order cone $\mathbb{K}^n$ as follows (see [37], Theorem 2.1):

$$\mathbb{C}^n_\theta = A^{-1}\mathbb{K}^n \quad \text{and} \quad A\mathbb{C}^n_\theta = \mathbb{K}^n. \tag{3.1}$$

Thus, by noticing $(\mathbb{C}^n_\theta)^* = \mathbb{C}^n_{\pi/2-\theta}$, for any $x, y \in \mathbb{R}^n$, we have

$$x \in \mathbb{C}^n_\theta \Leftrightarrow Ax \in \mathbb{K}^n, \quad y \in (\mathbb{C}^n_\theta)^* \Leftrightarrow A^{-1}y \in \mathbb{K}^n. \tag{3.2}$$

Based on relationship (3.2), we now construct two smoothing functions $\varphi_i\colon \mathbb{R}\times\mathbb{R}^n\times\mathbb{R}^n \to \mathbb{R}^n$ $(i = 1, 2)$ for the CCCP which are defined by

$$\varphi_1(\mu, x, y) = (1+\mu)(Ax + A^{-1}y) - \sqrt{(1-\mu)^2(Ax - A^{-1}y)^2 + 2\mu^2 e}, \tag{3.3}$$

$$\begin{aligned}\varphi_2(\mu, x, y) = {} & (1+\mu)(Ax + A^{-1}y) \\ & - \sqrt{(Ax + \mu A^{-1}y)^2 + (\mu Ax + A^{-1}y)^2 + 2\mu^2 e},\end{aligned} \tag{3.4}$$

where the square and square root of the vector are all defined under Jordan product "$\circ$" associated with the second-order cone $\mathbb{K}^n$ in Section 2. It is worth pointing out that the functions φ_1 and φ_2 are regularized versions of Chen-Harker-Kanzow-Smale (CHKS) smoothing function and Fischer-Burmeister (FB) smoothing function proposed in [26], which are denoted by

$$\begin{aligned}\varphi_{\rm CHKS}(\mu, x, y) &= Ax + A^{-1}y - \sqrt{(Ax - A^{-1}y)^2 + 2\mu^2 e}, \\ \varphi_{\rm FB}(\mu, x, y) &= Ax + A^{-1}y - \sqrt{(Ax)^2 + (A^{-1}y)^2 + 2\mu^2 e}.\end{aligned}$$

Moreover, when $\theta = \pi/4$, the functions φ_1 and φ_2 reduce to smoothing functions over the second-order cone studied in [11], [10].

Theorem 3.1. *The functions φ_i $(i = 1, 2)$ defined by* (3.3) *and* (3.4) *satisfy*

$$\varphi_i(0, x, y) = 0 \Leftrightarrow x \in \mathbb{C}^n_\theta,\ y \in (\mathbb{C}^n_\theta)^*,\ \langle x, y\rangle = 0. \tag{3.5}$$

P r o o f. From [17], Propositions 4.1 and 4.2 we have for any $i = 1, 2$,

$$\varphi_i(0, x, y) = 0 \Leftrightarrow Ax \in \mathbb{K}^n,\ A^{-1}y \in \mathbb{K}^n,\ \langle Ax, A^{-1}y\rangle = 0.$$

This together with (3.2) and $\langle Ax, A^{-1}y\rangle = \langle A^{-1}Ax, y\rangle = \langle x, y\rangle$ gives (3.5). $\square$

Theorem 3.2. *The functions φ_i $(i=1,2)$ defined by (3.3) and (3.4) are continuously differentiable at any $(\mu,x,y)\in\mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n$ with*

$$(\varphi_1)'_\mu = Ax + A^{-1}y - L_{w_1}^{-1}[-(1-\mu)(Ax-A^{-1}y)^2 + 2\mu e], \tag{3.6}$$

$$(\varphi_1)'_x = (1+\mu)A - (1-\mu)^2 L_{w_1}^{-1} L_{Ax-A^{-1}y} A, \tag{3.7}$$

$$(\varphi_1)'_y = (1+\mu)A^{-1} + (1-\mu)^2 L_{w_1}^{-1} L_{Ax-A^{-1}y} A^{-1}, \tag{3.8}$$

where $w_1 := \sqrt{(1-\mu)^2(Ax-A^{-1}y)^2 + 2\mu^2 e}$, *and*

$$(\varphi_2)'_\mu = Ax + A^{-1}y - L_{w_2}^{-1}(L_{Ax+\mu A^{-1}y}A^{-1}y + L_{\mu Ax+A^{-1}y}Ax + 2\mu e), \tag{3.9}$$

$$(\varphi_2)'_x = (1+\mu)A - L_{w_2}^{-1}(L_{Ax+\mu A^{-1}y} + \mu L_{\mu Ax+A^{-1}y})A, \tag{3.10}$$

$$(\varphi_2)'_y = (1+\mu)A^{-1} - L_{w_2}^{-1}(\mu L_{Ax+\mu A^{-1}y} + L_{\mu Ax+A^{-1}y})A^{-1}, \tag{3.11}$$

where $w_2 := \sqrt{(Ax+\mu A^{-1}y)^2 + (\mu Ax + A^{-1}y)^2 + 2\mu^2 e}$.

Proof. It is easy to see that $\varphi_i(\mu,x,y)$ $(i=1,2)$ are continuously differentiable at any $(\mu,x,y)\in\mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n$. For any $(\mu,x,y)\in\mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n$, from the definition of w_1 we have $w_1\in\operatorname{int}\mathbb{K}^n$ and therefore L_{w_1} is invertible. Since $w_1^2=(1-\mu)^2(Ax-A^{-1}y)^2+2\mu^2e$, it follows from [7], Lemma 3.1 that

$$\begin{aligned}
(w_1)'_\mu &= L_{w_1}^{-1}[-(1-\mu)(Ax-A^{-1}y)^2 + 2\mu e],\\
(w_1)'_x &= (1-\mu)^2 L_{w_1}^{-1} L_{Ax-A^{-1}y} A,\\
(w_1)'_y &= -(1-\mu)^2 L_{w_1}^{-1} L_{Ax-A^{-1}y} A^{-1}.
\end{aligned}$$

This gives (3.6)–(3.8). In the similar way, we can prove (3.9)–(3.11). □

The next theorems show that the smoothing functions φ_i $(i=1,2)$ defined by (3.3) and (3.4) are all coercive and strong semismooth. These properties play important roles in analyzing global and local convergence properties of smoothing-type algorithms.

Theorem 3.3. *Let φ_i $(i=1,2)$ be defined by (3.3) and (3.4) and $a_1,a_2\in\mathbb{R}_{++}$ with $a_1<a_2$. Let $\{(\mu_k,x^k,y^k)\}\subset\mathbb{R}_{++}\times\mathbb{R}^n\times\mathbb{R}^n$ be a sequence satisfying*

(c1) *$\mu_k\in[a_1,a_2]$, $\{(x^k,y^k)\}$ is unbounded; and*
(c2) *there is a bounded sequence $\{(p^k,q^k)\}$ such that $\{\langle x^k-p^k, y^k-q^k\rangle\}$ is bounded below.*

Then $\{\varphi_i(\mu_k,x^k,y^k)\}$ $(i=1,2)$ are unbounded.

P r o o f. Since A is positive definite for any $\theta \in (0, \pi/2)$ and

$$\langle Ax^k - Ap^k, A^{-1}y^k - A^{-1}q^k \rangle = \langle x^k - p^k, y^k - q^k \rangle,$$

by condition (c2), $\{(Ap^k, A^{-1}q^k)\}$ is bounded and $\langle Ax^k - Ap^k, A^{-1}y^k - A^{-1}q^k \rangle$ is bounded below. Note that by letting $X := Ax$ and $Y := A^{-1}y$, the functions φ_i $(i = 1, 2)$ can be written as

$$\begin{aligned}\varphi_1(\mu, x, y) &= \varphi_1(\mu, X, Y) := (1 + \mu)(X + Y) - \sqrt{(1 - \mu)^2 (X - Y)^2 + 2\mu^2 e},\\ \varphi_2(\mu, x, y) &= \varphi_2(\mu, X, Y) := (1 + \mu)(X + Y) - \sqrt{(X + \mu Y)^2 + (\mu X + Y)^2 + 2\mu^2 e}.\end{aligned}$$

Let $X^k := Ax^k$ and $Y^k := A^{-1}y^k$. Then the sequence $\{(\mu_k, X^k, Y^k)\}$ satisfies that $\mu_k \in [a_1, a_2]$ and $\{(X^k, Y^k)\}$ is unbounded, and there exists a bounded sequence $\{(P^k, Q^k)\}$, where $P^k := Ap^k$ and $Q^k := A^{-1}q^k$ such that $\{\langle X^k - P^k, Y^k - Q^k \rangle\}$ is bounded below. Hence, from [19], Theorem 4.1, $\varphi_1(\mu_k, X^k, Y^k)$ is unbounded and so is $\{\varphi_1(\mu_k, x^k, y^k)\}$, and from [12], Theorem 3.4, $\varphi_2(\mu_k, X^k, Y^k)$ is unbounded and so is $\{\varphi_2(\mu_k, x^k, y^k)\}$. $\square$

Theorem 3.4. *The functions* φ_i $(i = 1, 2)$ *defined by* (3.3) *and* (3.4) *are strongly semismooth on* $\mathbb{R}^{1+2n}$.

P r o o f. Since the composition of strongly semismooth functions is strongly semismooth, by [9], Theorem 4.2 and [32], Theorem 3.2, we have the desired results. $\square$

4. Smooth reformulation of the CCCP

Let $z := (\mu, x, y) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n$. By using the smoothing functions φ_i $(i = 1, 2)$ in (3.3) and (3.4), we define the function $H_i \colon \mathbb{R}^{1+2n} \to \mathbb{R}^{1+2n}$ as

$$H_i(z) := \begin{pmatrix} \mu \\ F(x) - y \\ \varphi_i(\mu, x, y) \end{pmatrix}, \quad i = 1, 2. \tag{4.1}$$

Then, by Theorem 3.1, we have

$$H_i(z) = 0 \ (i = 1, 2) \Leftrightarrow \mu = 0 \quad \text{and} \quad (x, y) \text{ is the solution of the CCCP.}$$

Thus, instead of solving the CCCP, one may apply some descent methods to solve the system of equations $H_i(z) = 0$ $(i = 1, 2)$ and make $\mu \to 0^+$ so that a solution of the CCCP can be found.

According to Theorem 3.2, the function $H_i(z)$ $(i=1,2)$ is continuously differentiable at any $z \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$ with its Jacobian

$$H_i'(z) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & F'(x) & -I_n \\ (\varphi_i(\mu,x,y))'_\mu & (\varphi_i(\mu,x,y))'_x & (\varphi_i(\mu,x,y))'_y \end{bmatrix}, \tag{4.2}$$

where $(\varphi_i(\mu,x,y))'_\mu$, $(\varphi_i(\mu,x,y))'_x$ and $(\varphi_i(\mu,x,y))'_y$ $(i=1,2)$ are given in equations (3.6)–(3.11), respectively. In smoothing-type algorithms, it is essential that the Jacobian $H_i'(z)$ $(i=1,2)$ is invertible, because the descent direction should be well-defined and unique to solve $H_i(z)=0$ $(i=1,2)$. For this purpose, in this paper we assume that the function F is monotone, i.e.,

$$\langle F(x)-F(y), x-y\rangle \geqslant 0 \quad \forall\,(x,y) \in \mathbb{R}^n \times \mathbb{R}^n. \tag{4.3}$$

This monotonic assumption is very standard and has been extensively used to design smoothing-type algorithms for the SOCCP (e.g., [8], [9], [18], [34], [33]). Miao et al. [26] also used the monotonicity of F to analyze properties of bounded level sets of merit functions for the CCCP.

Lemma 4.1 ([35], Lemma 2.6). *Let $a,b,u,v \in \mathbb{R}^n$ with $a \succ_{\mathbb{K}^n} 0$, $b \succ_{\mathbb{K}^n} 0$, $a\circ b \succ_{\mathbb{K}^n} 0$. If $\langle u,v\rangle \geqslant 0$ and $a\circ u + b\circ v = 0$, then $u=v=0$.*

Theorem 4.1. *If F is monotone, then the Jacobian $H_i'(z)$ $(i=1,2)$ defined by (4.2) is invertible at any $z \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$.*

Proof. We first show that $H_1'(z)$ is invertible at any $z \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$. For any $z \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$, let $\Delta z := (\Delta\mu, \Delta x, \Delta y) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n$ be any vector which satisfies $H_1'(z)\Delta z = 0$. It suffices to prove $\Delta z = 0$. By (4.2), we have from $H_1'(z)\Delta z = 0$ that

$$\Delta\mu = 0, \tag{4.4}$$

$$F'(x)\Delta x - \Delta y = 0, \tag{4.5}$$

$$(\varphi_1(\mu,x,y))'_\mu \Delta\mu + (\varphi_1(\mu,x,y))'_x \Delta x + (\varphi_1(\mu,x,y))'_y \Delta y = 0. \tag{4.6}$$

By (4.5) and the monotonicity of F, we have $\langle \Delta x, \Delta y\rangle = \langle \Delta x, (F'(x)\Delta x)\rangle \geqslant 0$, which gives

$$\begin{aligned} &\langle A\Delta x + \mu A^{-1}\Delta y, \mu A\Delta x + A^{-1}\Delta y\rangle \\ &\qquad = \mu(\|A\Delta x\|^2 + \|A^{-1}\Delta y\|^2) + (1+\mu^2)\langle \Delta x, \Delta y\rangle \geqslant 0. \end{aligned} \tag{4.7}$$

By (3.7), (3.8), (4.4) and (4.6), we have

$$(1+\mu)(A\Delta x + A^{-1}\Delta y) - L_{w_1}^{-1}[(1-\mu)^2(Ax - A^{-1}y)\circ(A\Delta x - A^{-1}\Delta y)] = 0,$$

where $w_1 := \sqrt{(1-\mu)^2(Ax - A^{-1}y)^2 + 2\mu^2 e}$, which yields

$$L_{w_1}((1+\mu)(A\Delta x + A^{-1}\Delta y)) - (1-\mu)^2(Ax - A^{-1}y)\circ(A\Delta x - A^{-1}\Delta y) = 0,$$

i.e.,

$$\begin{aligned}(4.8)\qquad [w_1 - (1-\mu)(Ax - A^{-1}y)]\circ(A\Delta x + \mu A^{-1}\Delta y)\\ + [w_1 + (1-\mu)(Ax - A^{-1}y)]\circ(\mu A\Delta x + A^{-1}\Delta y) = 0.\end{aligned}$$

By the definition of w_1 and $\mu > 0$, we have $w_1 \succ_{\mathbb{K}^n} 0$ and $w_1^2 \succ_{\mathbb{K}^n} (1-\mu)^2(Ax - A^{-1}y)^2$. Then from [17], Proposition 3.4, we have that

$$(4.9)\qquad w_1 - (1-\mu)(Ax - A^{-1}y) \succ_{\mathbb{K}^n} 0, \quad w_1 + (1-\mu)(Ax - A^{-1}y) \succ_{\mathbb{K}^n} 0.$$

Also notice that

$$(4.10)\qquad [w_1 - (1-\mu)(Ax - A^{-1}y)]\circ[w_1 + (1-\mu)(Ax - A^{-1}y)] = 2\mu^2 e \succ_{\mathbb{K}^n} 0.$$

Thus, by (4.7)–(4.10) and Lemma 4.1, we have $A\Delta x + \mu A^{-1}\Delta y = 0$ and $\mu A\Delta x + A^{-1}\Delta y = 0$. These two equalities imply that $A\Delta x = 0$ and $A^{-1}\Delta y = 0$. Since A is positive definite for any $\theta \in (0, \pi/2)$, we have $\Delta x = \Delta y = 0$. This proves that $H_1'(z)$ is invertible at any $z \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$. By a similar way, we can also prove that $H_2'(z)$ is invertible at any $z \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$. We complete the proof. □

Theorem 4.2. *Suppose that F is monotone. Then the function H_i $(i = 1, 2)$ defined by* (4.1) *is coercive in (x, y) for each $\mu > 0$, i.e.,*

$$\lim_{\|(x,y)\|\to\infty} \|H_i(\mu, x, y)\| = \infty.$$

P r o o f. By using Theorem 3.3, similarly as in the proof of [19], Lemma 5.3, we can prove the result. For brevity, we omit the details here. □

Theorem 4.3. *The function H_i $(i = 1, 2)$ defined by* (4.1) *is semismooth on $\mathbb{R}^{1+2n}$ and it is strongly semismooth on $\mathbb{R}^{1+2n}$ if F' is locally Lipschitz.*

P r o o f. The result holds by Theorem 3.4 and [30], Corollary 2.4. □

5. A smoothing algorithm

In the following, for simplicity, we only give our algorithm to solve $H_1(z)=0$. All results obtained still hold when we apply our algorithm to solve $H_2(z)=0$.

Algorithm 5.1 (A smoothing algorithm).

Step 1. Choose constants $\delta,\beta\in(0,1)$, $\Lambda_0>0$ and $\mu_0>0$. Choose $\gamma\in(0,1)$ such that $\gamma\leqslant\mu_0$. Choose $\xi\in(0,1)$ such that $\xi<1-\gamma$. Choose $(x^0,y^0)\in\mathbb{R}^n\times\mathbb{R}^n$ and let $z^0:=(\mu_0,x^0,y^0)$. Let $\varrho_0:=\gamma\min\{1,\|H_1(z^0)\|^2\}$. Let $p:=(1,0,\dots,0)\in\mathbb{R}^{1+2n}$. Set $k:=0$.

Step 2. If $\|H_1(z^k)\|=0$, then stop.

Step 3. Compute $\Delta z^k:=(\Delta\mu_k,\Delta x^k,\Delta y^k)$ by solving

$$H_1'(z^k)\Delta z^k=-H_1(z^k)+\varrho_k p. \tag{5.1}$$

Step 4. Set

$$U_k:=\|H_1(z^k)\|+\Lambda_k,\quad V_k:=\|H_1(z^k)\|+\beta\Lambda_k. \tag{5.2}$$

Let l_k be the smallest nonnegative integer l satisfying

$$\|H_1(z^k+\delta^l\Delta z^k)\|\leqslant\min\{(1-\xi\delta^l)U_k,V_k\}. \tag{5.3}$$

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

Step 5. Set $z^{k+1}:=z^k+\alpha_k\Delta z^k$. Set

$$\Lambda_{k+1}:=(1-\beta)\Lambda_k, \tag{5.4}$$

$$\varrho_{k+1}:=\gamma\min\{1,\|H_1(z^{k+1})\|^2,\varrho_k\}. \tag{5.5}$$

Set $k:=k+1$ and go back to Step 2.

Different with existing smoothing-type algorithms, in Step 4, Algorithm 5.1 adopts a new line search technique, which is easy to implement. It is easy to see that inequality (5.3) holds for all sufficiently large l, because when $l\to\infty$, the left-hand side of (5.3) tends to $\|H_1(z^k)\|$ but the right-hand side tends to $\min\{U_k,V_k\}=\|H_1(z^k)\|+\beta\Lambda_k$. Hence, the new line search technique is well-defined. As will be shown later, this new line search technique makes Algorithm 5.1 have encouraging convergent properties and practical computational performances.

Theorem 5.1. *If F is monotone, then Algorithm 5.1 is well-defined and generates a sequence $\{z^k=(\mu_k,x^k,y^k)\}$ with $\mu_k>0$ for all $k\geqslant 0$.*

P r o o f. Suppose that $z^k = (\mu_k, x^k, y^k) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$ for some k, e.g., it is satisfied for $k = 0$. From Theorem 4.1, $H_1'(z^k)$ is nonsingular and hence Step 3 is well-defined. Since inequality (5.3) holds for all sufficiently large l, Step 4 is also well-defined. Thus, we can get the $(k+1)$th iteration $z^{k+1} = z^k + \alpha_k \Delta z^k$ in Step 5. Moreover, by the first equation in (5.1), we have $\Delta\mu_k = -\mu_k + \varrho_k$. Since $\mu_k > 0$ and $\varrho_k > 0$, we have

$$\mu_{k+1} = \mu_k + \alpha_k \Delta\mu_k = (1-\alpha_k)\mu_k + \alpha_k \varrho_k > 0. \tag{5.6}$$

So, we can conclude that if $z^k = (\mu_k, x^k, y^k) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$ for some k, then $z^{k+1} = (\mu_{k+1}, x^{k+1}, y^{k+1})$ can be generated by Algorithm 5.1 with $z^{k+1} \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$. This together with $z^0 = (\mu_0, x^0, y^0) \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$ implies that Algorithm 5.1 is well-defined and generates a sequence $\{z^k = (\mu_k, x^k, y^k)\}$ with $\mu_k > 0$. The proof is completed. □

Lemma 5.1. *Suppose that F is monotone. Let $\{z^k\}$ be the sequence generated by Algorithm* 5.1. *Then there exist constants $\varrho^* \geqslant 0$ and $U^* \geqslant 0$ such that $\lim\limits_{k\to\infty} \varrho_k = \varrho^*$ and*

$$\lim_{k\to\infty} U_k = \lim_{k\to\infty} V_k = \lim_{k\to\infty} \|H_1(z^k)\| = U^*.$$

Moreover, $U^* = 0$ if $\varrho^* = 0$.

P r o o f. Since $\{\varrho_k\}$ is monotonically decreasing by its definition, there exists $\varrho^* \geqslant 0$ such that $\lim\limits_{k\to\infty} \varrho_k = \varrho^*$. From Steps 4 and 5 in Algorithm 5.1 it follows that for all $k \geqslant 0$,

$$\|H_1(z^{k+1})\| \leqslant \min\{(1-\xi\alpha_k)U_k, V_k\} \leqslant V_k = \|H_1(z^k)\| + \beta\Lambda_k.$$

By using this result, we have from (5.2) and (5.4) that for all $k \geqslant 0$,

$$\begin{aligned} U_{k+1} &= \|H_1(z^{k+1})\| + \Lambda_{k+1} = \|H_1(z^{k+1})\| + (1-\beta)\Lambda_k \\ &\leqslant \|H_1(z^k)\| + \beta\Lambda_k + (1-\beta)\Lambda_k = \|H_1(z^k)\| + \Lambda_k = U_k. \end{aligned} \tag{5.7}$$

This shows that $\{U_k\}$ is also monotonically decreasing and hence, it is convergent. So, there exists $U^* \geqslant 0$ such that $\lim\limits_{k\to\infty} U_k = U^*$. Moreover, by (5.4), we have $\Lambda_k = (1-\beta)^k \Lambda_0$ and hence $\lim\limits_{k\to\infty} \Lambda_k = 0$. Thus, from (5.2) it holds that

$$\lim_{k\to\infty} \|H_1(z^k)\| = \lim_{k\to\infty} (U_k - \Lambda_k) = U^*$$

and

$$\lim_{k\to\infty} V_k = \lim_{k\to\infty} (\|H_1(z^k)\| + \beta\Lambda_k) = U^*.$$

If $\varrho^* = 0$, then by the definition of ϱ_k, there exists a subsequence $\{z^{k_n}\}$ of $\{z^k\}$ such that $\lim\limits_{k_n\to\infty} \|H_1(z^{k_n})\| = 0$, which gives $U^* = 0$. The proof is completed. □

Lemma 5.2. *Suppose that F is monotone. Let $\{z^k = (\mu_k, x^k, y^k)\}$ be the sequence generated by Algorithm* 5.1. *Then $\mu_k \geqslant \varrho_k$ and $\mu_k \geqslant \mu_{k+1}$ for all $k \geqslant 0$. Moreover, $z^k \in L(U_0)$ for all $k \geqslant 0$, where*

$$L(U_0) := \{z \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n;\ \|H_1(z)\| \leqslant U_0\}.$$

Proof. Notice that if $\mu_k \geqslant \varrho_k$ for some k, then from (5.6) it holds that $\mu_{k+1} \geqslant (1-\alpha_k)\varrho_k + \alpha_k\varrho_k = \varrho_k \geqslant \varrho_{k+1}$ because $\{\varrho_k\}$ is monotonically decreasing. This together with $\mu_0 \geqslant \gamma \geqslant \varrho_0$ proves $\mu_k \geqslant \varrho_k$ for all $k \geqslant 0$. By this result, we can further obtain from (5.6) that $\mu_{k+1} \leqslant (1-\alpha_k)\mu_k + \alpha_k\mu_k = \mu_k$ for all $k \geqslant 0$. Moreover, by Theorem 5.1, (5.2) and (5.7), we have $z^k \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$ and $\|H_1(z^k)\| \leqslant U_k \leqslant U_0$ for all $k \geqslant 0$. This completes the proof. $\square$

6. Global convergence analysis

Theorem 6.1. *Suppose that F is monotone. Let $\{z^k = (\mu_k, x^k, y^k)\}$ be the sequence generated by Algorithm* 5.1. *Then*

$$\lim_{k\to\infty} \|H_1(z^k)\| = 0. \tag{6.1}$$

Proof. By Lemma 5.1, there exists $\varrho^* \geqslant 0$ and $U^* \geqslant 0$ such that

$$\lim_{k\to\infty} \varrho_k = \varrho^*, \quad \lim_{k\to\infty} U_k = U^*, \quad \lim_{k\to\infty} \|H_1(z^k)\| = U^*. \tag{6.2}$$

Now we assume $U^* > 0$. Then from Lemma 5.1 we have $\varrho^* > 0$. By Lemma 5.2, there exists $\mu^* \geqslant 0$ such that $\lim\limits_{k\to\infty} \mu_k = \mu^*$ and $\mu_0 \geqslant \mu_k \geqslant \mu^* \geqslant \varrho^* > 0$. Thus, if $\lim\limits_{k\to\infty} \|(x^k, y^k)\| = \infty$, then $\{\|H_1(z^k)\|\}$ must be unbounded by Theorem 4.2, which contradicts Lemma 5.2. Hence, $\{z^k = (\mu_k, x^k, y^k)\}$ is bounded and it has at least one accumulation point, denoted by $z^* := (\mu^*, x^*, y^*)$. Without loss of generality, we assume $\lim\limits_{k\to\infty} z^k = z^*$. From Steps 4 and 5 in Algorithm 5.1, for all sufficiently large k,

$$\|H_1(z^{k+1})\| \leqslant \min\{(1-\xi\alpha_k)U_k, V_k\} \leqslant (1-\xi\alpha_k)U_k. \tag{6.3}$$

Since $\lim\limits_{k\to\infty} \|H_1(z^k)\| = \lim\limits_{k\to\infty} U_k = U^* > 0$, by (6.3) we have $\lim\limits_{k\to\infty} \alpha_k = 0$. Let $\hat\alpha_k := \alpha_k/\delta$. Then $\lim\limits_{k\to\infty} \hat\alpha_k = 0$ and $\hat\alpha_k$ does not satisfy the line search criterion (5.3) for any sufficiently large k, i.e.,

$$\|H_1(z^k + \hat\alpha_k \Delta z^k)\| > \min\{(1-\xi\hat\alpha_k)U_k,\ V_k\}. \tag{6.4}$$

Since $\min\{\theta a, b\} \geqslant \theta \min\{a, b\}$ holds for any $a, b > 0$ and $\theta \in (0, 1)$, from (6.4), for any sufficiently large k,

$$\begin{aligned}\|H_1(z^k + \hat{\alpha}_k \Delta z^k)\| &> (1 - \xi\hat{\alpha}_k)\min\{U_k,\ V_k\} = (1 - \xi\hat{\alpha}_k)V_k \\ &= (1 - \xi\hat{\alpha}_k)(\|H_1(z^k)\| + \beta\Lambda_k) \geqslant (1 - \xi\hat{\alpha}_k)\|H_1(z^k)\|.\end{aligned}$$

It follows that for any sufficiently large k,

$$\frac{\|H_1(z^k + \hat{\alpha}_k \Delta z^k)\| - \|H_1(z^k)\|}{\hat{\alpha}_k} > -\xi\|H_1(z^k)\|,$$

i.e.,

$$\begin{aligned}&\frac{\|H_1(z^k + \hat{\alpha}_k \Delta z^k)\|^2 - \|H_1(z^k)\|^2}{\hat{\alpha}_k} \\ &\qquad > -\xi\|H_1(z^k)\|[\|H_1(z^k + \hat{\alpha}_k \Delta z^k)\| + \|H_1(z^k)\|].\end{aligned} \tag{6.5}$$

Since $z^* \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n$, $\|H_1(z)\|^2$ is continuously differentiable at z^*. Thus, by letting $k \to \infty$ in (6.5), also using (6.2), we have

$$H_1(z^*)^\top H_1'(z^*)\Delta z^* \geqslant -\xi(U^*)^2, \tag{6.6}$$

where Δz^* is the solution of $H_1'(z^*)\Delta z^* = -H_1(z^*) + \varrho^* p$. On the other hand, from (5.1) we have

$$\begin{aligned}H_1(z^k)^\top H_1'(z^k)\Delta z^k &= H_1(z^k)^\top[-H_1(z^k) + \varrho_k p] = -\|H_1(z^k)\|^2 + \mu_k\varrho_k \\ &\leqslant -(1 - \gamma)\|H_1(z^k)\|^2,\end{aligned} \tag{6.7}$$

where the inequality holds, because $\mu_k \leqslant \|H_1(z^k)\|$ and $\varrho_k \leqslant \gamma\min\{1, \|H_1(z^k)\|^2\} \leqslant \gamma\|H_1(z^k)\|$ for all $k \geqslant 0$. By letting $k \to \infty$ in (6.7), also using (6.2), we have

$$H_1(z^*)^\top H_1'(z^*)\Delta z^* \leqslant -(1 - \gamma)(U^*)^2. \tag{6.8}$$

By combining (6.6) and (6.8), we have $\xi(U^*)^2 \geqslant (1 - \gamma)(U^*)^2$, which together with $U^* > 0$ gives $\xi \geqslant 1 - \gamma$. This contradicts $\xi < 1 - \gamma$ in Step 1. Thus, $U^* = 0$ and by (6.2) we have (6.1). This completes the proof. $\square$

Theorem 6.2. *Suppose that F is monotone. Let $\{z^k\}$ be the sequence generated by Algorithm* 5.1. *Then any accumulation point of $\{z^k\}$ is a solution of $H_1(z) = 0$. Moreover, if the solution set of the CCCP is nonempty and bounded, then $\{z^k\}$ is bounded.*

P r o o f. The first result holds by (6.1) and a simple continuity discussion. By using Theorem 4.2 and Mountain Pass Theorem [28], Theorem 9.2.7, we can prove the second result similarly as [15], Theorem 5.4 and [33], Theorem 5.2. $\square$

7. Local superlinear/quadratical convergence

Let z^* be any accumulation point of the iteration sequence $\{z^k\}$ generated by Algorithm 5.1. To obtain local fast convergence, existing smoothing-type algorithms require the following *Jacobian nonsingularity assumption*:

$$\text{All } V \in \partial H_1(z^*) \text{ are nonsingular.}$$

In this section, we establish the local superlinear/quadratical convergence of Algorithm 5.1 under some assumptions, which are much weaker than Jacobian nonsingularity assumption.

Assumption 7.1. There exist a neighborhood $N(z^*, \varepsilon) := \{z \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n;\ \|z - z^*\| \leqslant \varepsilon\}$ and a constant $\xi > 0$ such that

$$\xi \|H_1(z)\| \geqslant \|z - z^*\| \quad \forall\, z \in N(z^*, \varepsilon). \tag{7.1}$$

Assumption 7.1 is a type of *local error bound condition*, which has been used to analyze local convergence properties of smoothing Levenberg-Marquardt algorithms (e.g., [20], [23]). The following lemma shows that Assumption 7.1 is weaker than Jacobian nonsingularity assumption.

Lemma 7.1. *If all $V \in \partial H_1(z^*)$ are nonsingular, then Assumption* 7.1 *holds.*

Proof. By [30], Proposition 3.1, there exist a neighborhood $N(z^*, \varepsilon_1)$ and a constant $c > 0$ such that for any $z \in N(z^*, \varepsilon_1)$ and $V \in \partial H_1(z)$, V is nonsingular and $\|V^{-1}\| \leqslant c$. Since H_1 is semismooth at z^*, there exists a neighborhood $N(z^*, \varepsilon_2)$ such that for any $z \in N(z^*, \varepsilon_2)$ and $V \in \partial H_1(z)$,

$$\|H_1(z) - H_1(z^*) - V(z - z^*)\| \leqslant \frac{1}{2c}\|z - z^*\|.$$

By Theorem 6.2, we have $H_1(z^*) = 0$. Let $\varepsilon := \min\{\varepsilon_1, \varepsilon_2\}$. Then for any $z \in N(z^*, \varepsilon)$ and $V \in \partial H_1(z)$ we have

$$\begin{aligned}
\|z - z^*\| &\leqslant \|z - z^*\| - \|V^{-1} H_1(z)\| + c\|H_1(z)\| \\
&\leqslant \|V^{-1} H_1(z) - (z^k - z^*)\| + c\|H_1(z^k)\| \\
&\leqslant \|V^{-1}\| \|H_1(z) - V(z^k - z^*)\| + c\|H_1(z)\| \\
&\leqslant \frac{1}{2}\|(z - z^*)\| + c\|H_1(z)\|,
\end{aligned}$$

and hence $\|z^k - z^*\| \leqslant 2c\|H_1(z)\|$. The proof is completed. $\square$

From Lemma 7.1, if Jacobian nonsingularity assumption holds at z^*, then Assumption 7.1 holds. However, the converse is not necessarily true. A simple counterexample is $H(z) = |z| = 0$, where $z \in \mathbb{R}$. It is easy to see that $\xi\|H(z)\| \geqslant |z - 0|$ for any $\xi \geqslant 1$ and $z \in \mathbb{R}$. But since $\partial_B H'(0) = \{1, -1\}$, we have $0 \in \partial H'(0)$, which implies that Jacobian nonsingularity assumption does not hold at $z^* = 0$. Hence, Assumption 7.1 is weaker than Jacobian nonsingularity assumption.

Assumption 7.2. There exist a neighborhood $N(z^*, \varepsilon) := \{z \in \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n;\ \|z - z^*\| \leqslant \varepsilon\}$ and constants $t \in [0, 1)$ and $C > 0$ such that

$$\|H_1'(z)^{-1}\| \leqslant \frac{C}{\|H_1(z)\|^t} \quad \forall\, z \in N(z^*, \varepsilon). \tag{7.2}$$

For Assumption 7.2, we have the following remarks.

(i) From Theorem 4.1, $H_1'(z)$ is invertible for any $z \in N(z^*, \varepsilon)$ and hence Assumption 7.2 is reasonable.

(ii) If inequality (7.2) holds for $t = 0$, then Assumption 7.2 reduces to that the set $\{\|H_1'(z)^{-1}\|;\ z \in N(z^*, \varepsilon)\}$ is bounded, which has been used in many literatures to prove the local fast convergence of their algorithms (e.g., [5], [36]).

(iii) In Theorem 6.1, we proved that $\lim\limits_{k\to\infty} \|H_1(z^k)\| = 0$. Hence, Assumption 7.2 allows $\|H_1'(z^k)^{-1}\| \to \infty$ as $k \to \infty$ when $t \in (0, 1)$. In what follows, we will show that our algorithm has local quadratical convergence if Assumption 7.2 holds for $t = 0$, and it still has local superlinear convergence if $\|H_1'(z^k)^{-1}\| \to \infty$ as $k \to \infty$ and Assumption 7.2 holds for some $t \in (0, 1)$.

(iv) From [30], Proposition 3.1, if Jacobian nonsingularity assumption holds at z^*, then $\{\|H_1'(z)^{-1}\|;\ z \in N(z^*, \varepsilon)\}$ is bounded and hence Assumption 7.2 holds. However, the converse is also not necessarily true. See the example $H(z) = |z| = 0$ again. Since $H'(z)^{-1} = \operatorname{sgn}(z)^{-1}$ for any $z \in D_H := \{z \neq 0\}$, when $z \to 0$, we have $\|H'(z)^{-1}\| = 1$. However, Jacobian nonsingularity assumption does not hold at $z^* = 0$. Hence, Assumption 7.2 is also weaker than Jacobian nonsingularity assumption.

Under Assumptions 7.1 and 7.2, we now give the local superlinear/quadratical convergence of Algorithm 5.1 as follows.

Theorem 7.1. *Suppose that F is monotone and F' is locally Lipschitz. Let z^* be any accumulation point of the sequence $\{z^k\}$ generated by Algorithm 5.1. If Assumptions 7.1 and 7.2 hold, then the whole sequence $\{z^k\}$ converges to z^* and*

$$\|z^{k+1} - z^*\| = O(\|z^k - z^*\|^{2-t}) \quad \textit{and} \quad \|H_1(z^{k+1})\| = O(\|H_1(z^k)\|^{2-t}), \quad t \in [0, 1).$$

Proof. By Theorems 6.1 and 6.2, we have $\lim\limits_{k\to\infty}\|H_1(z^k)\| = 0$ and $H_1(z^*) = 0$. Since H_1 is strongly semismooth at z^*, for all z^k sufficiently close to z^*,

$$\|H_1(z^k) - H_1(z^*) - H_1'(z^k)(z^k - z^*)\| = O(\|z^k - z^*\|^2). \tag{7.3}$$

Since H_1 is strongly semismooth at z^*, H_1 is locally Lipschitz continuous near z^*. Thus, for all z^k sufficiently close to z^*,

$$\|H_1(z^k)\| = \|H_1(z^k) - H_1(z^*)\| = O(\|z^k - z^*\|). \tag{7.4}$$

By (5.5) and (7.4), for all z^k sufficiently close to z^*,

$$\varrho_k \leqslant \gamma\|H_1(z^k)\|^2 = O(\|H_1(z^k)\|^2) = O(\|z^k - z^*\|^2). \tag{7.5}$$

Then by (5.1), (7.3), (7.5) and Assumptions 7.1 and 7.2, for all z^k sufficiently close to z^*,

$$\begin{aligned} \|z^k + \Delta z^k - z^*\| &= \|z^k + H_1'(z^k)^{-1}[-H_1(z^k) + \varrho_k p] - z^*\| \\ &\leqslant \|H_1'(z^k)^{-1}\|[\|H_1(z^k) - H_1(z^*) - H_1'(z^k)(z^k - z^*)\| + \varrho_k] \\ &\leqslant \frac{C}{\|H_1(z^k)\|^t} O(\|z^k - z^*\|^2) \\ &\leqslant \frac{C\xi^t}{\|z^k - z^*\|^t} O(\|z^k - z^*\|^2) \\ &= O(\|z^k - z^*\|^{2-t}). \end{aligned} \tag{7.6}$$

So, $z^k + \Delta z^k$ is sufficiently close to z^*. Moreover, from Assumption 7.1, for all z^k sufficiently close to z^*,

$$\|z^k - z^*\| = O(\|H_1(z^k)\|) = O(\|H_1(z^k) - H_1(z^*)\|). \tag{7.7}$$

Thus, for all z^k sufficiently close to z^*, it follows from (7.4), (7.6) and (7.7) that

$$\begin{aligned} \|H_1(z^k + \Delta z^k)\| &= O(\|z^k + \Delta z^k - z^*\|) = O(\|z^k - z^*\|^{2-t}) \\ &= O(\|H_1(z^k) - H_1(z^*)\|^{2-t}) = O(\|H_1(z^k)\|^{2-t}). \end{aligned} \tag{7.8}$$

By (7.8), for all z^k sufficiently close to z^*,

$$\|H_1(z^k + \Delta z^k)\| \leqslant (1 - \xi)\|H_1(z^k)\|,$$

which together with (5.2) implies that for all z^k sufficiently close to z^*,

$$\|H_1(z^k + \Delta z^k)\| \leqslant (1 - \xi)\min\{U_k, V_k\} \leqslant \min\{(1 - \xi)U_k, V_k\}.$$

This shows that for all z^k sufficiently close to z^*, $\alpha_k = 1$ is always accepted in Step 4 of Algorithm 5.1. Thus, for all z^k sufficiently close to z^*, $z^{k+1} = z^k + \Delta z^k$, which together with (7.6) and (7.8) proves the theorem. $\square$

8. Numerical experiments

In this section, we give some numerical results of Algorithm 5.1 for solving the following CCCP:

$$x \in \mathbb{C}_\theta,\ y \in \mathbb{C}_\theta^*,\ \langle x, y\rangle = 0,\ y = F(x), \tag{8.1}$$

where $\mathbb{C}_\theta \subset \mathbb{R}^n$ and $(\mathbb{C}_\theta)^* \subset \mathbb{R}^n$ are the Cartesian product of some $\mathbb{C}_\theta^m$ and $(\mathbb{C}_\theta^m)^*$, respectively, that is,

$$\mathbb{C}_\theta = \mathbb{C}_\theta^{n_1} \times \ldots \times \mathbb{C}_\theta^{n_r}, \quad \mathbb{C}_\theta^* = (\mathbb{C}_\theta^{n_1})^* \times \ldots \times (\mathbb{C}_\theta^{n_r})^* \tag{8.2}$$

with $r, n_1, \ldots, n_r \geqslant 1$ and $n = \sum\limits_{i=1}^{r} n_i$, $x = (x_1, \ldots, x_r) \in \mathbb{R}^n$ and $y = (y_1, \ldots, y_r) \in \mathbb{R}^n$ with $x_i, y_i \in \mathbb{R}^{n_i}$ and $F\colon\ \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable function.

All experiments are carried on a PC with CPU of Inter(R) Core(TM)i7-7700 CPU @ 3.60 GHz and RAM of 8.00GB. The program codes are written in MATLAB and run in MATLAB R2018a environment. The parameters used in Algorithm 5.1 are chosen as

$$\mu_0 = 10^{-3},\ \gamma = 10^{-4},\ \xi = 0.2,\ \delta = 0.8,\ \beta = 0.5,\ \Lambda_0 = 10.$$

Moreover, we use $\|H_i(z^k)\| \leqslant 10^{-6}$ $(i = 1, 2)$ as the stopping criterion.

Example 8.1. Consider the CCCP (8.1), where $\mathbb{C}_\theta = \mathbb{C}_\theta^3 \times \mathbb{C}_\theta^2$ and $F\colon\ \mathbb{R}^5 \to \mathbb{R}^5$ is given by

$$F(x) = \begin{pmatrix} 24(2x_1 - x_2)^3 + \exp(x_1 - x_3) - 4x_4 + x_5 \\ -12(2x_1 - x_2)^3 + 3(3x_2 + 5x_3)/\sqrt{1 + (3x_2 + 5x_3)^2} - 6x_4 - 7x_5 \\ -\exp(x_1 - x_3) + 5(3x_2 + 5x_3)/\sqrt{1 + (3x_2 + 5x_3)^2} - 3x_4 + 5x_5 \\ 4x_1 + 6x_2 + 3x_3 - 1 \\ -x_1 + 7x_2 - 5x_3 + 2 \end{pmatrix}.$$

Notice that F is monotone (see [18]). We apply Algorithm 5.1 to solve this example with $\theta = \pi/3, \pi/4, \pi/5, \pi/6$, respectively, by using $x^0 = y^0 = (1, \ldots, 1)^\top$ as the starting points. The obtained solutions are

$$\begin{aligned}
\Big(\theta = \frac{\pi}{3}\Big) x^* &\approx (0.16058, -0.07313, 0.26550, 0.53213, -0.24303)^\top, \\
\Big(\theta = \frac{\pi}{4}\Big) x^* &\approx (0.23240, -0.07308, 0.22061, 0.53390, -0.53390)^\top, \\
\Big(\theta = \frac{\pi}{5}\Big) x^* &\approx (0.25645, 0.00637, 0.18622, 0.61957, -0.45014)^\top, \\
\Big(\theta = \frac{\pi}{6}\Big) x^* &\approx (0.26412, 0.05190, 0.14339, 0.61623, -0.35578)^\top.
\end{aligned}$$

Tables 1 and 2 show a sequence of the first three components of x^k generated by Algorithm 5.1 with $H_i(z)$ $(i = 1, 2)$ for this example with $\theta = \pi/3$. From the values of $\|H_1(z^k)\|$ and $\|H_2(z^k)\|$, we can clearly see the local fast, at least superlinear, convergence of our algorithm.

k	x_1^k	x_2^k	x_2^k	$\|H_1(z^k)\|$
0	1.0000e + 00	1.0000e + 00	1.0000e + 00	3.3696e + 01
1	2.8956e − 01	−1.1600e − 01	1.7945e − 01	7.9149e + 00
2	2.0641e − 01	−8.8405e − 02	2.3495e − 01	2.0079e + 00
3	1.6896e − 01	−7.5926e − 02	2.5991e − 01	3.0480e − 01
4	1.6092e − 01	−7.3249e − 02	2.6527e − 01	1.2089e − 02
5	1.6058e − 01	−7.3134e − 02	2.6550e − 01	2.1646e − 05
6	1.6058e − 01	−7.3134e − 02	2.6550e − 01	6.9779e − 11
7	1.6058e − 01	−7.3134e − 02	2.6550e − 01	3.4164e − 15

Table 1. Numerical results of Algorithm 5.1 with $H_1(z)$ for Example 8.1.

k	x_1^k	x_2^k	x_2^k	$\|H_2(z^k)\|$
0	1.0000e + 00	1.0000e + 00	1.0000e + 00	3.3519e + 01
1	2.7284e − 01	−1.4614e − 01	1.5096e − 01	8.6764e + 00
2	2.0763e − 01	−9.1303e − 02	2.3215e − 01	2.0731e + 00
3	1.6978e − 01	−7.6230e − 02	2.5934e − 01	3.3478e − 01
4	1.6099e − 01	−7.3273e − 02	2.6522e − 01	1.4568e − 02
5	1.6058e − 01	−7.3134e − 02	2.6550e − 01	3.1413e − 05
6	1.6058e − 01	−7.3134e − 02	2.6550e − 01	1.4696e − 10
7	1.6058e − 01	−7.3134e − 02	2.6550e − 01	4.3620e − 15

Table 2. Numerical results of Algorithm 5.1 with $H_2(z)$ for Example 8.1.

Example 8.2. Consider the CCCP (8.1), where $\mathbb{C}_\theta = \mathbb{C}_\theta^{n_1} \times \ldots \times \mathbb{C}_\theta^{n_r}$ and $F(x) = Mx + q$ with $q \in \mathbb{R}^n$ and $M \in \mathbb{R}^{n\times n}$ being a positive semidefinite matrix.

We choose $q = \mathrm{rand}(n, 1)$ and generate the matrix $M \in \mathbb{R}^{n\times n}$ by the following procedure. We choose $N_i = \mathrm{rand}(n_i, n_i)$ for $i = 1, 2, \ldots, r$ and then let M be the block diagonal matrix with $N_1^\top N_1, \ldots, N_r^\top N_r$ as block diagonals, i.e., $M = \mathrm{diag}\{N_i^\top N_i\}_{i=1}^r$. In the experiments, we let $r = 4$ and $n_i = n/4$ for any $i = 1, 2, \ldots, r$. Moreover, we take $x^0 = (1, 0, \ldots, 0)^\top$ and $y^0 = (1, \ldots, 1)^\top$ as the starting point. First, we generate 10 problem instances for each size of the test problem with $\theta = \pi/4, \pi/5, \pi/6$, respectively. Numerical results are listed in Tables 3 and 4, where n denotes the size of test problems, **AIT** and **ACPU** denote the average value of iteration numbers and CPU time in seconds among the 10 testing, respectively. From

	$\theta = \frac{\pi}{4}$		$\theta = \frac{\pi}{5}$		$\theta = \frac{\pi}{6}$	
n	**AIT**	**ACPU**	**AIT**	**ACPU**	**AIT**	**ACPU**
500	6.0	0.50	7.0	0.56	7.3	0.61
1000	7.0	2.71	7.8	2.99	8.5	3.28
1500	7.1	8.49	8.1	9.10	8.8	9.95
2000	7.9	18.18	8.4	18.95	9.2	21.22
2500	8.0	30.96	9.0	35.22	9.7	37.52
3000	8.0	53.56	9.0	57.83	9.9	62.32

Table 3. Numerical results of Algorithm 5.1 with $H_1(z)$ for Example 8.2.

	$\theta = \frac{\pi}{4}$		$\theta = \frac{\pi}{5}$		$\theta = \frac{\pi}{6}$	
n	**AIT**	**ACPU**	**AIT**	**ACPU**	**AIT**	**ACPU**
500	7.0	0.74	7.6	0.73	8.1	0.79
1000	7.7	3.46	8.5	3.74	9.4	4.29
1500	8.0	10.38	9.0	12.03	10.0	12.82
2000	8.0	21.82	9.3	24.59	10.4	26.68
2500	8.7	41.57	10.0	45.40	10.9	48.33
3000	9.0	66.86	10.0	74.50	11.2	80.47

Table 4. Numerical results of Algorithm 5.1 with $H_2(z)$ for Example 8.2.

		Algorithm 5.1		Qi-Sun-Zhou's method	
	n	**AIT**	**ACPU**	**AIT**	**ACPU**
$H_1(z)$	500	5.3	0.43	6.7	0.48
	1000	5.9	2.27	7.0	2.61
	1500	6.0	6.52	7.7	8.46
	2000	6.0	13.50	8.0	17.72
	2500	6.0	23.53	8.0	30.45
	3000	6.7	42.85	8.0	49.43
$H_2(z)$	500	6.1	0.62	7.4	0.71
	1000	6.0	2.61	8.0	3.35
	1500	6.7	8.38	8.1	10.09
	2000	7.0	17.68	9.0	22.95
	2500	7.0	30.90	9.0	39.58
	3000	7.0	52.04	9.0	61.24

Table 5. Comparison of Algorithm 5.1 and Qi-Sun-Zhou's method.

Tables 3 and 4 we can see that our algorithm is very effective for solving CCCPs. Moreover, we observe that the performance of our algorithm based on the smoothing function φ_1 is better than that based on the smoothing function φ_2. This is an important new discovery and it is yet unknown whether similar phenomena happens in other different algorithms.

Next, we generate 10 problem instances for the test problem with $\theta = \pi/3$. For the purpose of comparison, we also apply the Qi-Sun-Zhou's smoothing Newton method [31] to solve these problem instances. Numerical results are listed in Table 5 from which we may find that our algorithm needs less iteration numbers and CPU time compared with Qi-Sun-Zhou's method. This is probably due to the nonmonotone line search adopted in our algorithm.

9. Conclusions

In this paper we investigated the CCCP, which is a type of nonsymmetric cone complementarity problem. We constructed two smoothing functions for the CCCP and proved that they have coerciveness and strong semismoothness. Moreover, we proposed a smoothing algorithm to solve the CCCP, which is designed based on a new line search technique. The proposed algorithm is tractable because it can start from an arbitrary point and it solves only one linear system of equations and performs only one line search at each iteration. Under suitable assumptions, we proved that the proposed algorithm has global convergence. Moreover, we established the local superlinear/quadratical convergence of the proposed algorithm under Assumptions 7.1 and 7.2, which are much weaker than Jacobian nonsingularity assumption. By the numerical results in Tables 1–5, we may find that our algorithm is very effective for solving CCCPs.

Acknowledgement. We are very grateful to the referees for their valuable comments on the paper, which have considerably improved the paper.

References

[1] *F. Alizadeh, D. Goldfarb*: Second-order cone programming. Math. Program. *95* (2003), 3–51. zbl MR doi
[2] *B. Alzalg*: The Jordan algebraic structure of the circular cone. Oper. Matrices *11* (2017), 1–21. zbl MR doi
[3] *Y. Bai, X. Gao, G. Wang*: Primal-dual interior-point algorithms for convex quadratic circular cone optimization. Numer. Algebra Control Optim. *5* (2015), 211–231. zbl MR doi
[4] *Y. Bai, P. Ma, J. Zhang*: A polynomial-time interior-point method for circular cone programming based on kernel functions. J. Ind. Manag. Optim. *12* (2016), 739–756. zbl MR doi
[5] *J. Burke, S. Xu*: A non-interior predictor-corrector path following algorithm for the monotone linear complementarity problem. Math. Program. *87* (2000), 113–130. zbl MR doi

[6] *J.-S. Chen, S. Pan*: A survey on SOC complementarity functions and solution methods for SOCPs and SOCCPs. Pac. J. Optim. *8* (2012), 33–74. zbl MR

[7] *J.-S. Chen, D. Sun, J. Sun*: The SC^1 property of the squared norm of the SOC Fischer-Burmeister function. Oper. Res. Lett. *36* (2008), 385–392. zbl MR doi

[8] *L. Chen, C. Ma*: A modified smoothing and regularized Newton method for monotone second-order cone complementarity problems. Comput. Math. Appl. *61* (2011), 1407–1418. zbl MR doi

[9] *X. D. Chen, D. Sun, J. Sun*: Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementarity problems. Comput. Optim. Appl. *25* (2003), 39–56. zbl MR doi

[10] *X. Chi, S. Liu*: A non-interior continuation method for second-order cone programming. Optimization *58* (2009), 965–979. zbl MR doi

[11] *X. Chi, S. Liu*: A one-step smoothing Newton method for second-order cone programming. J. Comput. Appl. Math. *223* (2009), 114–123. zbl MR doi

[12] *X. Chi, J. Tao, Z. Zhu, F. Duan*: A regularized inexact smoothing Newton method for circular cone complementarity problem. Pac. J. Optim. *13* (2017), 197–218. zbl MR

[13] *X. Chi, Z. Wan, Z. Zhu, L. Yuan*: A nonmonotone smoothing Newton method for circular cone programming. Optimization *65* (2016), 2227–2250. zbl MR doi

[14] *X. Chi, H. Wei, Z. Wan, Z. Zhu*: Smoothing Newton algorithm for the circular cone programming with a nonmonotone line search. Acta Math. Sci., Ser. B, Engl. Ed. *37* (2017), 1262–1280. zbl MR doi

[15] *F. Facchinei, C. Kanzow*: Beyond monotonicity in regularization methods for nonlinear complementarity problems. SIAM J. Control Optim. *37* (1999), 1150–1161. zbl MR doi

[16] *L. Fang, Z. Feng*: A smoothing Newton-type method for second-order cone programming problems based on a new smoothing Fischer-Burmeister function. Comput. Appl. Math. *30* (2011), 569–588. zbl MR doi

[17] *M. Fukushima, Z.-Q. Luo, P. Tseng*: Smoothing functions for second-order-cone complementarity problems. SIAM J. Optim. *12* (2001), 436–460. zbl MR doi

[18] *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. zbl MR doi

[19] *Z.-H. Huang, T. Ni*: Smoothing algorithms for complementarity problems over symmetric cones. Comput. Optim. Appl. *45* (2010), 557–579. zbl MR doi

[20] *P. Jin, C. Ling, H. Shen*: A smoothing Levenberg-Marquardt algorithm for semi-infinite programming. Comput. Optim. Appl. *60* (2015), 675–695. zbl MR doi

[21] *Y.-F. Ke, C.-F. Ma, H. Zhang*: The relaxation modulus-based matrix splitting iteration methods for circular cone nonlinear complementarity problems. Comput. Appl. Math. *37* (2018), 6795–6820. zbl MR doi

[22] *B. Kheirfam, G. Wang*: An infeasible full NT-step interior point method for circular optimization. Numer. Algebra Control Optim. *7* (2017), 171–184. zbl MR doi

[23] *W. Liu, C. Wang*: A smoothing Levenberg-Marquardt method for generalized semi-infinite programming. Comput. Appl. Math. *32* (2013), 89–105. zbl MR doi

[24] *M. S. Lobo, L. Vandenberghe, S. Boyd, H. Lebret*: Applications of second-order cone programming. Linear Algebra Appl. *284* (1998), 193–228. zbl MR doi

[25] *P. Ma, Y. Bai, J.-S. Chen*: A self-concordant interior point algorithm for nonsymmetric circular cone programming. J. Nonlinear Convex Anal. *17* (2016), 225–241. zbl MR

[26] *X.-H. Miao, S. Guo, N. Qi, J.-S. Chen*: Constructions of complementarity functions and merit functions for circular cone complementarity problem. Comput. Optim. Appl. *63* (2016), 495–522. zbl MR doi

[27] *X.-H. Miao, J. Yang, S. Hu*: A generalized Newton method for absolute value equations associated with circular cones. Appl. Math. Comput. *269* (2015), 155–168. zbl MR doi
[28] *R. S. Palais, C.-L. Terng*: Critical Point Theory and Submanifold Geometry. Lecture Notes in Mathematics 1353. Springer, Berlin, 1988. zbl MR doi
[29] *M. Pirhaji, M. Zangiabadi, H. Mansouri*: A path following interior-point method for linear complementarity problems over circular cones. Japan J. Ind. Appl. Math. *35* (2018), 1103–1121. zbl MR doi
[30] *L. Qi, J. Sun*: A nonsmooth version of Newton's method. Math. Program. *58* (1993), 353–367. zbl MR doi
[31] *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. zbl MR doi
[32] *D. Sun, J. Sun*: Strong semismoothness of Fischer-Burmeister SDC and SOC complementarity functions. Math. Program. *103* (2005), 575–581. zbl MR doi
[33] *J. Tang, L. Dong, J. Zhou, L. Sun*: A smoothing-type algorithm for the second-order cone complementarity problem with a new nonmonotone line search. Optimization *64* (2015), 1935–1955. zbl MR doi
[34] *J. Tang, G. He, L. Dong, L. Fang, J. Zhou*: A smoothing Newton method for the second-order cone complementarity problem. Appl. Math., Praha *58* (2013), 223–247. zbl MR doi
[35] *A. Yoshise*: Interior point trajectories and a homogeneous model for nonlinear complementarity problems over symmetric cones. SIAM J. Optim. *17* (2006), 1129–1153. zbl MR doi
[36] *J. Zhang, J. Chen*: A smoothing Levenberg-Marquardt type method for LCP. J. Comput. Math. *22* (2004), 735–752. zbl MR
[37] *J. Zhou, J.-S. Chen*: Properties of circular cone and spectral factorization associated with circular cone. J. Nonlinear Convex Anal. *14* (2013), 807–816. zbl MR
[38] *J. Zhou, J.-S. Chen, H.-F. Hung*: Circular cone convexity and some inequalities associated with circular cones. J. Inequal. Appl. *2013* (2013), Article ID 571, 17 pages. zbl MR doi
[39] *J. Zhou, J.-S. Chen, B. S. Mordukhovich*: Variational analysis of circular cone programs. Optimization *64* (2015), 113–147. zbl MR doi
[40] *J. Zhou, J. Tang, J.-S. Chen*: Parabolic second-order directional differentiability in the Hadamard sense of the vector-valued functions associated with circular cones. J. Optim. Theory Appl. *172* (2017), 802–823. zbl MR doi

Authors' address: *Jingyong Tang* (corresponding author), *Yuefen Chen*, College of Mathematics and Statistics, Xinyang Normal University, 464000 Xinyang, P. R. China, e-mail: `tangjy@xynu.edu.cn`, `yfchen@xynu.edu.cn`.