Applied Numerical Mathematics 135 (2019) 206–227

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Unified smoothing functions for absolute value equation associated with second-order cone

Chieu Thanh Nguyen [a], B. Saheya [b,1], Yu-Lin Chang [a], Jein-Shan Chen [a,*,2]

[a] Department of Mathematics, National Taiwan Normal University, Taipei 11677, Taiwan
[b] College of Mathematical Science, Inner Mongolia Normal University, Hohhot 010022, Inner Mongolia, PR China

ARTICLE INFO

Article history:
Received 19 February 2018
Received in revised form 12 July 2018
Accepted 30 August 2018
Available online 3 September 2018

Keywords:
Second-order cone
Absolute value equations
Smoothing Newton algorithm

ABSTRACT

In this paper, we explore a unified way to construct smoothing functions for solving the absolute value equation associated with second-order cone (SOCAVE). Numerical comparisons are presented, which illustrate what kinds of smoothing functions work well along with the smoothing Newton algorithm. In particular, the numerical experiments show that the well known loss function widely used in engineering community is the worst one among the constructed smoothing functions, which indicates that the other proposed smoothing functions can be employed for solving engineering problems.

1. Introduction

Recently, the paper [36] investigates a family of smoothing functions along with a smoothing-type algorithm to tackle the absolute value equation associated with second-order cone (SOCAVE) and shows the efficiency of such approach. Motivated by this article, we continue to ask two natural questions. (i) Whether there are other suitable smoothing functions that can be employed for solving the SOCAVE? (ii) Is there a unified way to construct smoothing functions for solving the SOCAVE? In this paper, we provide affirmative answers for these two queries. In order to smoothly convey the story of how we figure out the answers, we begin with recalling where the SOCAVE comes from.

The standard absolute value equation (AVE) is in the form of

$$Ax + B|x| = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n}$, $B \neq 0$, and $b \in \mathbb{R}^n$. Here $|x|$ means the componentwise absolute value of vector $x \in \mathbb{R}^n$. When $B = -I$, where I is the identity matrix, the AVE (1) reduces to the special form:

$$Ax - |x| = b.$$

It is known that the AVE (1) was first introduced by Rohn in [40], but was termed by Mangasarian [34]. During the past decade, there has been many researchers paying attention to this equation, for example, Caccetta, Qu and Zhou [1], Hu and

* Corresponding author.
E-mail addresses: thanhchieu90@gmail.com (C.T. Nguyen), saheya@imnu.edu.cn (B. Saheya), ylchang@math.ntnu.edu.tw (Y.-L. Chang), jschen@math.ntnu.edu.tw (J.-S. Chen).

[1] The author's work is supported by Natural Science Foundation of Inner Mongolia (Award Number: 2017MS0125).
[2] The author's work is supported by Ministry of Science and Technology, Taiwan.

https://doi.org/10.1016/j.apnum.2018.08.019
0168-9274/

Huang [12], Jiang and Zhang [19], Ketabchi and Moosaei [20], Mangasarian [26–33], Mangasarian and Meyer [34], Prokopyev [37], and Rohn [42].

We elaborate more about the developments of the AVE. Mangasarian and Meyer [34] show that the AVE (1) is equivalent to the bilinear program, the generalized LCP (linear complementarity problem), and to the standard LCP provided 1 is not an eigenvalue of A. With these equivalent reformulations, they also show that the AVE (1) is NP-hard in its general form and provide existence results. Prokopyev [37] further improves the above equivalence which indicates that the AVE (1) can be equivalently recast as LCP without any assumption on A and B, and also provides a relationship with mixed integer programming. In general, if solvable, the AVE (1) can have either unique solution or multiple (e.g., exponentially many) solutions. Indeed, various sufficiency conditions on solvability and non-solvability of the AVE (1) with unique and multiple solutions are discussed in [34,37,41]. Some variants of the AVE, like the absolute value equation associated with second-order cone and the absolute value programs, are investigated in [14] and [45], respectively.

Recently, another type of absolute value equation, a natural extension of the standard AVE (1), is considered [14,35,36]. More specifically the following absolute value equation associated with second-order cones, abbreviated as SOCAVE, is studied:

$$Ax + B|x| = b, \tag{2}$$

where $A, B \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ are the same as those in (1); $|x|$ denotes the absolute value of x coming from the square root of the Jordan product "$\circ$" of x and x. What is the difference between the standard AVE (1) and the SOCAVE (2)? Their mathematical formats look the same. In fact, the main difference is that $|x|$ in the standard AVE (1) means the componentwise $|x_i|$ of each $x_i \in \mathbb{R}$, i.e., $|x| = (|x_1|, |x_2|, \cdots, |x_n|)^T \in \mathbb{R}^n$; however, $|x|$ in the SOCAVE (2) denotes the vector satisfying $\sqrt{x^2} := \sqrt{x \circ x}$ associated with second-order cone under Jordan product. To understand its meaning, we need to introduce the definition of second-order cone (SOC). The second-order cone in $\mathbb{R}^n$ $(n \geq 1)$, also called the Lorentz cone, is defined as

$$\mathcal{K}^n := \left\{(x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1} \mid \|x_2\| \leq x_1\right\},$$

where $\|\cdot\|$ denotes the Euclidean norm. If $n = 1$, then $\mathcal{K}^n$ is the set of nonnegative reals $\mathbb{R}_+$. In general, a general second-order cone $\mathcal{K}$ could be the Cartesian product of SOCs, i.e.,

$$\mathcal{K} := \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r}.$$

For simplicity, we focus on the single SOC $\mathcal{K}^n$ because all the analysis can be carried over to the setting of Cartesian product. The SOC is a special case of symmetric cones and can be analyzed under Jordan product, see [9]. In particular, for any two vectors $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$ and $y = (y_1, y_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, the *Jordan product* of x and y associated with $\mathcal{K}^n$ is defined as

$$x \circ y := \begin{bmatrix} x^T y \\ y_1 x_2 + x_1 y_2 \end{bmatrix}.$$

The Jordan product, unlike scalar or matrix multiplication, is not associative, which is a main source of complication in the analysis of optimization problems involved SOC, see [5,6,10] and references therein for more details. The identity element under this Jordan product is $e = (1, 0, ..., 0)^T \in \mathbb{R}^n$. With these definitions, x^2 means the Jordan product of x with itself, i.e., $x^2 := x \circ x$; and $\sqrt{x}$ with $x \in \mathcal{K}^n$ denotes the unique vector such that $\sqrt{x} \circ \sqrt{x} = x$. In other words, the vector $|x|$ in the SOCAVE (2) is computed by

$$|x| := \sqrt{x \circ x}.$$

As remarked in the literature, the significance of the AVE (1) arises from the fact that the AVE is capable of formulating many optimization problems such as linear programs, quadratic programs, bimatrix games, and so on. Likewise, the SOCAVE (2) plays a similar role in various optimization problems involving second-order cones. There has been many numerical methods proposed for solving the standard AVE (1) and the SOCAVE (2). Please refer to [36] for a quick review. Basically, we follow the smoothing Newton algorithm employed in [36] to deal with the SOCAVE (2). This kind of algorithm has been a powerful tool for solving many other optimization problems, including symmetric cone complementarity problems [21,23,24], the system of inequalities under the order induced by symmetric cone [17,25,46], and so on. It is also employed for the standard AVE (1) in [18,43]. The new upshot of this paper lies on discovering more suitable smoothing functions and exploring a unified way to construct smoothing functions. Of course, the numerical performance among different smoothing functions are compared. These are totally new to the literature and are the main contribution of this paper.

To close this section, we recall some basic concepts and background materials regarding the second-order cone, which will be used in the subsequent analysis. More details can be found in [5,6,9,10,14]. First, we recall the expression of the *spectral decomposition* of x with respect to SOC. For $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, the spectral decomposition of x with respect to SOC is given by

$$x = \lambda_1(x) u_x^{(1)} + \lambda_2(x) u_x^{(2)}, \tag{3}$$

where $\lambda_i(x) = x_1 + (-1)^i \|x_2\|$ for $i = 1, 2$ and

$$u_x^{(i)} = \begin{cases} \frac{1}{2}\left(1, (-1)^i \frac{x_2^T}{\|x_2\|}\right)^T & \text{if} \quad \|x_2\| \neq 0, \\ \frac{1}{2}\left(1, (-1)^i \omega^T\right)^T & \text{if} \quad \|x_2\| = 0, \end{cases} \tag{4}$$

with $\omega \in \mathbb{R}^{n-1}$ being any vector satisfying $\|\omega\| = 1$. The two scalars $\lambda_1(x)$ and $\lambda_2(x)$ are called spectral values of x; while the two vectors $u_x^{(1)}$ and $u_x^{(2)}$ are called the spectral vectors of x. Moreover, it is obvious that the spectral decomposition of $x \in \mathbb{R}^n$ is unique if $x_2 \neq 0$. It is known that the spectral values and spectral vectors posses the following properties:

(i) $u_x^{(1)} \circ u_x^{(2)} = 0$ and $u_x^{(i)} \circ u_x^{(i)} = u_x^{(i)}$ for $i = 1, 2$;
(ii) $\|u_x^{(1)}\|^2 = \|u_x^{(2)}\|^2 = \frac{1}{2}$ and $\|x\|^2 = \frac{1}{2}(\lambda_1^2(x) + \lambda_2^2(x))$.

Next is the concept about the projection onto second-order cone. Let x_+ denote the projection of x onto $\mathcal{K}^n$, and x_- be the projection of $-x$ onto the dual cone $(\mathcal{K}^n)^*$ of $\mathcal{K}^n$, where the dual cone $(\mathcal{K}^n)^*$ is defined by $(\mathcal{K}^n)^* := \{y \in \mathbb{R}^n \mid \langle x, y\rangle \geq 0, \forall x \in \mathcal{K}^n\}$. In fact, the dual cone of $\mathcal{K}^n$ is itself, i.e., $(\mathcal{K}^n)^* = \mathcal{K}^n$. Due to the special structure of $\mathcal{K}^n$, the explicit formula of projection of $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$ onto $\mathcal{K}^n$ is obtained in [5,6,8–10] as below:

$$x_+ = \begin{cases} x & \text{if } x \in \mathcal{K}^n, \\ 0 & \text{if } x \in -\mathcal{K}^n, \\ u & \text{otherwise,} \end{cases} \quad \text{where} \quad u = \begin{bmatrix} \frac{x_1 + \|x_2\|}{2} \\ \left(\frac{x_1 + \|x_2\|}{2}\right) \frac{x_2}{\|x_2\|} \end{bmatrix}.$$

Similarly, the expression of x_- can be written out as

$$x_- = \begin{cases} 0 & \text{if } x \in \mathcal{K}^n, \\ -x & \text{if } x \in -\mathcal{K}^n, \\ w & \text{otherwise,} \end{cases} \quad \text{where} \quad w = \begin{bmatrix} -\frac{x_1 - \|x_2\|}{2} \\ \left(\frac{x_1 - \|x_2\|}{2}\right) \frac{x_2}{\|x_2\|} \end{bmatrix}.$$

It is easy to verify that $x = x_+ + x_-$ and

$$x_+ = (\lambda_1(x))_+ u_x^{(1)} + (\lambda_2(x))_+ u_x^{(2)} \quad x_- = (-\lambda_1(x))_+ u_x^{(1)} + (-\lambda_2(x))_+ u_x^{(2)},$$

where $(\alpha)_+ = \max\{0, \alpha\}$ for $\alpha \in \mathbb{R}$. As for the expression of $|x|$ associated with SOC. There is an alternative way via the so-called SOC-function to obtain the expression of $|x|$, which can be found in [2,3]. In any case, it comes out that

$$\begin{aligned} |x| &= \left[(\lambda_1(x))_+ + (-\lambda_1(x))_+\right] u_x^{(1)} + \left[(\lambda_2(x))_+ + (-\lambda_2(x))_+\right] u_x^{(2)} \\ &= \left|\lambda_1(x)\right| u_x^{(1)} + \left|\lambda_2(x)\right| u_x^{(2)}. \end{aligned}$$

2. Unified smoothing functions for SOCAVE

As mentioned in Section 1, we employ the smoothing Newton method for solving the SOCAVE (2), which needs a smoothing function to work with. Indeed, a family of smoothing functions was already considered in [36]. In this section, we look into what kinds of smoothing functions can be employed to work with the smoothing Newton algorithm for solving the SOCAVE (2).

Definition 2.1. A function $\phi : \mathbb{R}_{++} \times \mathbb{R} \to \mathbb{R}$ is called a smoothing function of $|t|$ if it satisfies the following:

(i) ϕ is continuously differentiable at $(\mu, t) \in \mathbb{R}_{++} \times \mathbb{R}$;
(ii) $\lim_{\mu \downarrow 0} \phi(\mu, t) = |t|$ for any $t \in \mathbb{R}$.

Given a smoothing function ϕ, we further define a vector-valued function $\Phi : \mathbb{R}_{++} \times \mathbb{R}^n \to \mathbb{R}^n$ as

$$\Phi(\mu, x) = \phi\left(\mu, \lambda_1(x)\right) u_x^{(1)} + \phi\left(\mu, \lambda_2(x)\right) u_x^{(2)} \tag{5}$$

where $\mu \in \mathbb{R}_{++}$ is a parameter, $\lambda_1(x)$, $\lambda_2(x)$ are the spectral values of x, and $u_x^{(1)}$, $u_x^{(2)}$ are the spectral vectors of x. Consequently, Φ is also smooth on $\mathbb{R}_{++} \times \mathbb{R}^n$. Moreover, it is easy to verify that

$$\lim_{\mu \to 0^+} \Phi(\mu, x) = |\lambda_1(x)|\, u_x^{(1)} + |\lambda_2(x)|\, u_x^{(2)} = |x|$$

which means each function $\Phi(\mu, x)$ serves as a smoothing function of $|x|$ associated with SOC. With this observation, for the SOCAVE (2), we further define the function $H(\mu, x) : \mathbb{R}_{++} \times \mathbb{R}^n \to \mathbb{R} \times \mathbb{R}^n$ by

$$H(\mu, x) = \begin{bmatrix} \mu \\ Ax + B\Phi(\mu, x) - b \end{bmatrix}, \quad \forall \mu \in \mathbb{R}_{++} \text{ and } x \in \mathbb{R}^n. \tag{6}$$

Proposition 2.1. *Suppose that $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$ has the spectral decomposition as in (3)–(4). Let $H : \mathbb{R}_{++} \times \mathbb{R}^n \to \mathbb{R}^n$ be defined as in (6). Then,*

(a) *$H(\mu, x) = 0$ if and only if x solves the SOCAVE (2);*
(b) *H is continuously differentiable at $(\mu, x) \in \mathbb{R}_{++} \times \mathbb{R}^n$ with the Jacobian matrix given by*

$$H'(\mu, x) = \begin{bmatrix} 1 & 0 \\ B\,\frac{\partial \Phi(\mu,x)}{\partial \mu} & A + B\,\frac{\partial \Phi(\mu,x)}{\partial x} \end{bmatrix} \tag{7}$$

where

$$\frac{\partial \Phi(\mu, x)}{\partial \mu} = \frac{\partial \phi(\mu, \lambda_1(x))}{\partial \mu} u_x^{(1)} + \frac{\partial \phi(\mu, \lambda_2(x))}{\partial \mu} u_x^{(2)},$$

$$\frac{\partial \Phi(\mu, x)}{\partial x} = \begin{cases} \frac{\partial \phi(\mu,x_1)}{\partial x_1} I & \text{if } x_2 = 0, \\ \begin{bmatrix} b & c\,\frac{x_2^T}{\|x_2\|} \\ c\,\frac{x_2}{\|x_2\|} & aI + (b-a)\,\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix} & \text{if } x_2 \neq 0, \end{cases}$$

with

$$\begin{aligned} a &= \frac{\phi(\mu, \lambda_2(x)) - \phi(\mu, \lambda_1(x))}{\lambda_2(x) - \lambda_1(x)}, \\ b &= \frac{1}{2}\left(\frac{\partial \phi(\mu, \lambda_2(x))}{\partial x_1} + \frac{\partial \phi(\mu, \lambda_1(x))}{\partial x_1}\right), \\ c &= \frac{1}{2}\left(\frac{\partial \phi(\mu, \lambda_2(x))}{\partial x_1} - \frac{\partial \phi(\mu, \lambda_1(x))}{\partial x_1}\right). \end{aligned} \tag{8}$$

Proof. (a) First, we observe that

$$\begin{aligned} H(\mu, x) = 0 &\Longleftrightarrow \mu = 0 \;\text{ and }\; Ax + B\Phi(\mu, x) - b = 0 \\ &\Longleftrightarrow Ax + B|x| - b = 0 \;\text{ and }\; \mu = 0. \end{aligned}$$

This indicates that x is a solution to the SOCAVE (2) if and only if (μ, x) is a solution to $H(\mu, x) = 0$.

(b) Since $\Phi(\mu, x)$ is continuously differentiable on $\mathbb{R}_{++} \times \mathbb{R}^n$, it is clear that $H(\mu, x)$ is continuously differentiable on $\mathbb{R}_{++} \times \mathbb{R}^n$. Thus, it remains to compute the Jacobian matrix of $H(\mu, x)$. Note that

$$\begin{aligned} \Phi(\mu, x) &= \phi(\mu, \lambda_1(x)) u_x^{(1)} + \phi(\mu, \lambda_2(x)) u_x^{(2)} \\ &= \begin{cases} \frac{1}{2}\begin{bmatrix} \phi(\mu, \lambda_1(x)) + \phi(\mu, \lambda_2(x)) \\ -\phi(\mu, \lambda_1(x))\frac{x_2^T}{\|x_2\|} + \phi(\mu, \lambda_2(x))\frac{x_2^T}{\|x_2\|} \end{bmatrix} & \text{if } x_2 \neq 0, \\ \frac{1}{2}\begin{bmatrix} \phi(\mu, \lambda_1(x)) + \phi(\mu, \lambda_2(x)) \\ -\phi(\mu, \lambda_1(x))\omega^T + \phi(\mu, \lambda_2(x))\omega^T \end{bmatrix} & \text{if } x_2 = 0 \end{cases} \\ &= \frac{1}{2}\begin{cases} \begin{bmatrix} \phi(\mu, \lambda_1(x)) + \phi(\mu, \lambda_2(x) \\ (-\phi(\mu, \lambda_1(x)) + \phi(\mu, \lambda_2(x)))\,\frac{\bar{x}_2}{\|x_2\|} \\ \vdots \\ (-\phi(\mu, \lambda_1(x)) + \phi(\mu, \lambda_2(x)))\,\frac{\bar{x}_n}{\|x_2\|} \end{bmatrix} & \text{if } x_2 \neq 0, \\ \begin{bmatrix} \phi(\mu, \lambda_1(x)) + \phi(\mu, \lambda_2(x)) \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \text{if } x_2 = 0, \end{cases} \end{aligned}$$

where $x_2 := (\bar{x}_2, \cdots, \bar{x}_n) \in \mathbb{R}^{n-1}$, $\omega = (\omega_2, \cdots, \omega_n) \in \mathbb{R}^{n-1}$. From chain rule, it is trivial that

$$\frac{\partial \Phi(\mu,x)}{\partial \mu} = \frac{\partial \phi(\mu,\lambda_1(x))}{\partial \mu} u_x^{(1)} + \frac{\partial \phi(\mu,\lambda_2(x))}{\partial \mu} u_x^{(2)}$$

In order to compute $\frac{\partial \Phi(\mu,x)}{\partial x}$, for simplicity, we denote

$$\Phi(\mu,x) := \frac{1}{2}\begin{bmatrix} \tau_1(\mu,x) \\ \tau_2(\mu,x) \\ \vdots \\ \tau_n(\mu,x) \end{bmatrix}.$$

To proceed, we discuss two cases.

(i) For $x_2 \neq 0$, we compute

$$\begin{aligned}
\frac{\partial \tau_1(\mu,x)}{\partial x_1} &= \frac{\partial \phi(\mu,\lambda_1(x))}{\partial x_1} + \frac{\partial \phi(\mu,\lambda_2(x))}{\partial x_1} \\
&= \frac{\partial \phi(\mu,\lambda_1(x))}{\partial \lambda_1(x)} \frac{\partial \lambda_1(x)}{\partial x_1} + \frac{\partial \phi(\mu,\lambda_2(x))}{\partial \lambda_2(x)} \frac{\partial \lambda_2(x)}{\partial x_1} \\
&= \frac{\partial \phi(\mu,\lambda_1(x))}{\partial \lambda_1(x)} + \frac{\partial \phi(\mu,\lambda_2(x))}{\partial \lambda_2(x)} := 2b
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial \tau_1(\mu,x)}{\partial \bar{x}_i} &= \frac{\partial \phi(\mu,\lambda_1(x))}{\partial \bar{x}_i} + \frac{\partial \phi(\mu,\lambda_2(x))}{\partial \bar{x}_i} \\
&= \frac{\partial \phi(\mu,\lambda_1(x))}{\partial \lambda_1(x)} \frac{\partial \lambda_1(x)}{\partial \bar{x}_i} + \frac{\partial \phi(\mu,\lambda_2(x))}{\partial \lambda_2(x)} \frac{\partial \lambda_2(x)}{\partial \bar{x}_i} \\
&= -\frac{\partial \phi(\mu,\lambda_1(x))}{\partial \lambda_1(x)} \frac{\bar{x}_i}{\|x_2\|} + \frac{\partial \phi(\mu,\lambda_2(x))}{\partial \lambda_2(x)} \frac{\bar{x}_i}{\|x_2\|} \\
&= \left(\frac{\partial \phi(\mu,\lambda_2(x))}{\partial \lambda_2(x)} - \frac{\partial \phi(\mu,\lambda_1(x))}{\partial \lambda_1(x)} \right) \frac{\bar{x}_i}{\|x_2\|} \\
&= \left(\frac{\partial \phi(\mu,\lambda_2(x))}{\partial x_1} - \frac{\partial \phi(\mu,\lambda_1(x))}{\partial x_1} \right) \frac{\bar{x}_i}{\|x_2\|} := 2c\, \frac{\bar{x}_i}{\|x_2\|}, \quad i = 2, \cdots, n.
\end{aligned}$$

Moreover,

$$\frac{\partial \tau_i(\mu,x)}{\partial x_1} = \left(\frac{\partial \phi(\mu,\lambda_2(x))}{\partial x_1} - \frac{\partial \phi(\mu,\lambda_1(x))}{\partial x_1} \right) \frac{\bar{x}_i}{\|x_2\|} = 2c\, \frac{\bar{x}_i}{\|x_2\|}, \quad i = 2, \cdots, n.$$

Similarly, we have

$$\begin{aligned}
\frac{\partial \tau_2(\mu,x)}{\partial \bar{x}_2} &= \left(\frac{\partial \phi(\mu,\lambda_2(x))}{\partial \bar{x}_2} - \frac{\partial \phi(\mu,\lambda_1(x))}{\partial \bar{x}_2} \right) \frac{\bar{x}_2}{\|x_2\|} + (\phi(\mu,\lambda_2(x)) - \phi(\mu,\lambda_1(x))) \frac{\partial \left(\frac{\bar{x}_2}{\|x_2\|} \right)}{\partial \bar{x}_2} \\
&= 2b\, \frac{\bar{x}_2 \cdot \bar{x}_2}{\|x_2\|^2} + (\phi(\mu,\lambda_2(x)) - \phi(\mu,\lambda_1(x))) \left(\frac{1}{\|x_2\|} - \frac{\bar{x}_2 \cdot \bar{x}_2}{\|x_2\|^3} \right) \\
&= 2a + 2(b-a)\, \frac{\bar{x}_2 \cdot \bar{x}_2}{\|x_2\|^2},
\end{aligned}$$

where a means $a := \dfrac{\phi(\mu,\lambda_2(x)) - \phi(\mu,\lambda_1(x))}{\lambda_2(x) - \lambda_1(x)}$. In general, mimicking the same derivation yields

$$\frac{\partial \tau_i(\mu,x)}{\partial \bar{x}_j} = \begin{cases} 2a + 2(b-a)\, \frac{\bar{x}_i \cdot \bar{x}_i}{\|x_2\|^2} & \text{if } i = j, \\ 2(b-a)\, \frac{\bar{x}_i \cdot \bar{x}_j}{\|x_2\|^2} & \text{if } i \neq j. \end{cases}$$

To sum up, we obtain

$$\frac{\partial \Phi(\mu,x)}{\partial x} = \begin{bmatrix} b & c\, \frac{x_2^T}{\|x_2\|} \\ c\, \frac{x_2}{\|x_2\|} & aI + (b-a)\, \frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix}$$

which is the desired result.

(ii) For $x_2 = 0$, it is clear to see

$$\frac{\partial \tau_1(\mu,x)}{\partial x_1} = 2\frac{\partial \phi(\mu,x_1)}{\partial x_1} \quad \text{and} \quad \frac{\partial \tau_1(\mu,x)}{\partial \bar{x}_i} = 0 \quad \text{for } i = 2,\cdots,n.$$

Since $\tau_i(\mu,x) = 0$ for $i = 2,\cdots,n$, it gives $\frac{\partial \tau_i(\mu,x)}{\partial x_1} = 0$. Moreover,

$$\begin{aligned}
\frac{\partial \tau_2(\mu,x)}{\partial \bar{x}_2} &= \lim_{\bar{x}_2\to 0} \frac{\tau_2(\mu,x_1,\bar{x}_2,0,\cdots,0) - \tau_2(\mu,x_1,0,\cdots,0)}{\bar{x}_2} \\
&= \lim_{\bar{x}_2\to 0} \frac{\phi(\mu,x_1+|\bar{x}_2|) - \phi(\mu,x_1-|\bar{x}_2|)}{\bar{x}_2}\frac{\bar{x}_2}{|\bar{x}_2|} \\
&= \lim_{\bar{x}_2\to 0} \frac{\phi(\mu,x_1+|\bar{x}_2|) - \phi(\mu,x_1-|\bar{x}_2|)}{|\bar{x}_2|} \\
&= \lim_{\bar{x}_2\to 0} \frac{\partial\phi(\mu,x_1+|\bar{x}_2|)}{\partial(|\bar{x}_2|)} - \frac{\partial\phi(\mu,x_1-|\bar{x}_2|)}{\partial(|\bar{x}_2|)} \quad \text{(as L'Hopital's rule)} \\
&= \lim_{\bar{x}_2\to 0} \frac{\partial\phi(\mu,x_1+|\bar{x}_2|)}{\partial(x_1+|\bar{x}_2|)} + \frac{\partial\phi(\mu,x_1-|\bar{x}_2|)}{\partial(x_1-|\bar{x}_2|)} \\
&= 2\frac{\partial \phi(\mu,x_1)}{\partial x_1}.
\end{aligned}$$

Thus, we obtain

$$\frac{\partial \tau_i(\mu,x)}{\partial \bar{x}_j} = \begin{cases} 2\frac{\partial \phi(\mu,x_1)}{\partial x_1} & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases}$$

which is equivalent to saying

$$\frac{\partial \Phi(\mu,x)}{\partial x} = \frac{\partial \phi(\mu,x_1)}{\partial x_1} I.$$

From all the above, we conclude that

$$\frac{\partial \Phi(\mu,x)}{\partial x} = \begin{cases} \frac{\partial \phi(\mu,x_1)}{\partial x_1} I & \text{if } x_2 = 0, \\ \begin{bmatrix} b & c\,\frac{x_2^T}{\|x_2\|} \\ c\,\frac{x_2}{\|x_2\|} & aI + (b-a)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix} & \text{if } x_2 \neq 0. \end{cases}$$

Thus, the proof is complete. □

Now, we are ready to answer the question about what kind of smoothing functions can be adopted in the smoothing type algorithm. Two technical lemmas are needed towards the answer.

Lemma 2.1. *Suppose that $M, N \in \mathbb{R}^{n\times n}$. Let $\sigma_{\min}(M)$ denote the minimum singular value of M, and $\sigma_{\max}(N)$ denote the maximum singular value of N. Then, the following hold.*

(a) $\sigma_{\min}(M) > \sigma_{\max}(N)$ *if and only if* $\sigma_{\min}(M^T M) > \sigma_{\max}(N^T N)$.
(b) *If* $\sigma_{\min}(M^T M) > \sigma_{\max}(N^T N)$*, then* $M^T M - N^T N$ *is positive definite.*

Proof. The proof is straightforward or can be found in usual textbook of matrix analysis, so we omit it here. □

Lemma 2.2. *Let $A, S \in \mathbb{R}^{n\times n}$ and A be symmetric. Suppose that the eigenvalues of A and SS^T are arranged in non-increasing order. Then, for each $k = 1, 2, \cdots, n$, there exists a nonnegative real number θ_k such that*

$$\lambda_{\min}(SS^T) \le \theta_k \le \lambda_{\max}(SS^T) \quad \text{and} \quad \lambda_k(SAS^T) = \theta_k \lambda_k(A).$$

Proof. Please see [11, Corollary 4.5.11] for a proof. □

We point out that the crucial key, which guarantees a smoothing function can be employed in the smoothing type algorithm, is the nonsingularity of the Jacobian matrix $H'(\mu,x))$ given in (7). As below, we provide under what condition the Jacobian matrix $H'(\mu,x))$ is nonsingular.

Theorem 2.1. *Consider a SOCAVE (2) with $\sigma_{\min}(A) > \sigma_{\max}(B)$. Let H be defined as in (6). Suppose that $\phi : \mathbb{R}_{++} \times \mathbb{R} \to \mathbb{R}$ is a smoothing function of $|t|$. If $-1 \le \frac{d}{dt}\phi(\mu, t) \le 1$ is satisfied, then the Jacobian matrix $H'(\mu, x)$ is nonsingular for any $\mu > 0$.*

Proof. From the expression of $H'(\mu, x)$ given as in (7), we know that $H'(\mu, x)$ is nonsingular if and only if the matrix $A + B\,\frac{\partial\Phi(\mu,x)}{\partial x}$ is nonsingular. Thus, it suffices to show that the matrix $A + B\,\frac{\partial\Phi(\mu,x)}{\partial x}$ is nonsingular under the conditions.

Suppose not, that is, there exists a vector $0 \ne v \in \mathbb{R}^n$ such that

$$\left[A + B\,\frac{\partial\Phi(\mu,x)}{\partial x}\right] v = 0$$

which implies that

$$v^T A^T A v = v^T \left[\frac{\partial\Phi(\mu,x)}{\partial x}\right]^T B^T B\,\frac{\partial\Phi(\mu,x)}{\partial x}\,v. \tag{9}$$

For convenience, we denote $C := \frac{\partial\Phi(\mu,x)}{\partial x}$. Then, it follows that $v^T A^T A v = v^T C^T B^T B C v$. Applying Lemma 2.2, there exists a constant $\hat{\theta}$ such that

$$\lambda_{\min}(C^T C) \le \hat{\theta} \le \lambda_{\max}(C^T C) \quad \text{and} \quad \lambda_{\max}(C^T B^T B C) = \hat{\theta}\lambda_{\max}(B^T B).$$

Note that if we can prove that

$$0 \le \lambda_{\min}(C^T C) \le \lambda_{\max}(C^T C) \le 1,$$

we will have $\lambda_{\max}(C^T B^T B C) \le \lambda_{\max}(B^T B)$. Then, by the assumption that the minimum singular value of A strictly exceeds the maximum singular value of B (i.e., $\sigma_{\min}(A) > \sigma_{\max}(B)$) and applying Lemma 2.1, we obtain $v^T A^T A v > v^T C^T B^T B C v$. This contradicts the identity (9), which shows the Jacobian matrix $H'(\mu, x)$ is nonsingular for $\mu > 0$.

Thus, in light of the above discussion, it suffices to claim $0 \le \lambda_{\min}(C^T C) \le \lambda_{\max}(C^T C) \le 1$. To this end, we discuss two cases.

Case 1: For $x_2 = 0$, we compute that $C = \frac{\partial\phi(\mu,x_1)}{\partial x_1} I$. Since $-1 \le \frac{\partial\phi(\mu,x_1)}{\partial x_1} \le 1$, it is clear that $0 \le \lambda(C^T C) \le 1$ for $\mu > 0$. Then, the claim is done.

Case 2: For $x_2 \ne 0$, using the fact that the matrix $M^T M$ is always positive semidefinite for any matrix $M \in \mathbb{R}^{m\times n}$, we see that the inequality $\lambda_{\min}(C^T C) \ge 0$ always holds. In order to prove $\lambda_{\max}(C^T C) \le 1$, we need to further argue that the matrix $I - C^T C$ is positive semidefinite. First, we write out

$$I - C^T C = \begin{bmatrix} 1 - b^2 - c^2 & -2bc\frac{x_2^T}{\|x_2\|} \\ -2bc\frac{x_2}{\|x_2\|} & (1-a^2)I + (a^2 - b^2 - c^2)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix}.$$

If $-1 < \frac{\partial\phi(\mu,\lambda_i(x))}{\partial x_1} < 1$, then we obtain

$$b^2 + c^2 = \frac{1}{2}\left[\left(\frac{\partial\phi(\mu,\lambda_1(x))}{\partial x_1}\right)^2 + \left(\frac{\partial\phi(\mu,\lambda_2(x))}{\partial x_1}\right)^2\right] < 1.$$

This indicates that $1 - b^2 - c^2 > 0$. By considering $[1 - b^2 - c^2]$ as an 1×1 matrix, this says $[1 - b^2 - c^2]$ is positive definite. Hence, its Schur complement can be computed as below:

$$\begin{aligned} &(1-a^2)I + (a^2 - b^2 - c^2)\frac{x_2 x_2^T}{\|x_2\|^2} - \frac{4b^2c^2}{1 - b^2 - c^2}\frac{x_2 x_2^T}{\|x_2\|^2} \\ &= (1-a^2)\left(I - \frac{x_2 x_2^T}{\|x_2\|^2}\right) + \left(1 - b^2 - c^2 - \frac{4b^2c^2}{1 - b^2 - c^2}\right)\frac{x_2 x_2^T}{\|x_2\|^2}. \end{aligned} \tag{10}$$

On the other hand, by the Mean Value Theorem, we have

$$\phi(\mu, \lambda_2(x)) - \phi(\mu, \lambda_1(x)) = \frac{\partial\phi(\mu,\xi)}{\partial\xi}(\lambda_2(x) - \lambda_1(x)),$$

where $\xi \in (\lambda_1(x), \lambda_2(x))$. To proceed, we need to further discuss two subcases.

(1) When $-1 < \frac{\partial\phi(\mu,\xi)}{\partial\xi} < 1$, we know $|\phi(\mu,\lambda_2(x)) - \phi(\mu,\lambda_1(x))| < |\lambda_2(x) - \lambda_1(x)|$. This together with (8) implies that $1 - a^2 > 0$ for any $\mu > 0$. In addition, for any $\mu > 0$, we observe that

$$\begin{aligned}
&(1-b^2-c^2)^2 - 4b^2c^2 \\
&= (1-(b-c)^2)(1-(b+c)^2) \\
&= \left[1 - \left(\frac{\partial\phi(\mu,\lambda_1(x))}{\partial x_1}\right)^2\right] \cdot \left[1 - \left(\frac{\partial\phi(\mu,\lambda_2(x))}{\partial x_1}\right)^2\right] > 0.
\end{aligned}$$

With all of these, we verify that the Schur complement of $[1 - b^2 - c^2]$ given as in (10) is a linear positive combination of the matrices $\left(I - \frac{x_2x_2^T}{\|x_2\|^2}\right)$ and $\frac{x_2x_2^T}{\|x_2\|^2}$, which yields that the Schur complement (10) of $[1 - b^2 - c^2]$ is positive semidefinite. Hence, the matrix $I - C^TC$ is also positive semidefinite, which is equivalent to saying $0 \le \lambda_{\min}(C^TC) \le \lambda_{\max}(C^TC) \le 1$.

(2) When $\frac{\partial\phi(\mu,\xi)}{\partial\xi} = \pm1$, we have

$$1 - a^2 = 0, \text{ and } (1-b^2-c^2)^2 - 4b^2c^2 > 0.$$

Since the matrix $\frac{x_2x_2^T}{\|x_2\|^2}$ is positive semidefinite, the matrix $I - C^TC$ is positive semidefinite. Hence, $0 \le \lambda_{\min}(C^TC) \le \lambda_{\max}(C^TC) \le 1$.

If either $\begin{cases} \frac{\partial\phi(\mu,\lambda_1(x))}{\partial x_1} = \pm1 \\ \frac{\partial\phi(\mu,\lambda_2(x))}{\partial x_1} = \pm1 \end{cases}$ or $\begin{cases} \frac{\partial\phi(\mu,\lambda_1(x))}{\partial x_1} = \pm1 \\ \frac{\partial\phi(\mu,\lambda_2(x))}{\partial x_1} = \mp1 \end{cases}$, then we have $b = \pm1, c = 0$ or $b = 0, c = \mp1$, which yields $b^2 + c^2 = 1$. Again, two subcases are needed.

(1) When $-1 < \frac{\partial\phi(\mu,\xi)}{\partial\xi} < 1$, we have $|\phi(\mu,\lambda_2(x)) - \phi(\mu,\lambda_1(x))| < |\lambda_2(x) - \lambda_1(x)|$. This implies that $1 - a^2 > 0$ for any $\mu > 0$. Therefore

$$I - C^TC = \begin{bmatrix} 0 & 0 \\ 0 & (1-a^2)\left(I - \frac{x_2x_2^T}{\|x_2\|^2}\right) \end{bmatrix}.$$

Since the matrix $I - \frac{x_2x_2^T}{\|x_2\|^2}$ is positive semidefinite, the matrix $I - C^TC$ is positive semidefinite. Hence, $0 \le \lambda_{\min}(C^TC) \le \lambda_{\max}(C^TC) \le 1$.

(2) When $\frac{\partial\phi(\mu,\xi)}{\partial\xi} = \pm1$, we have $I - C^TC = 0$, which leads to $\lambda(C^TC) = 1$.

From all the above, the proof is complete. □

We point out that the condition $\sigma_{\min}(A) > \sigma_{\max}(B)$ in Theorem 2.1 guarantees the unique solution according to [35, Theorem 4.1]. From Theorem 2.1, we realize that for a SOCAVE (2) with $\sigma_{\min}(A) > \sigma_{\max}(B)$, any smoothing function of $|t|$ with $-1 \le \frac{d}{dt}\phi(\mu,t) \le 1$ will be good for serving in the smoothing Newton algorithm when solving the above SOCAVE. With this, it is easy to find or construct smoothing functions of $|t|$ satisfying the above condition. One popular approach is a smoothing approximation via convolution for the absolute value function [4,22,38,44], which is described as below.

First, we construct a smoothing approximation for the plus function $(t)_+ = \max\{0, t\}$. Then, we consider the piecewise continuous function $d(t)$ with finite number of pieces, which is a density (kernel) function. In other words, it satisfies

$$d(t) \ge 0 \quad \text{and} \quad \int_{-\infty}^{+\infty} d(t)dt = 1.$$

With this $d(t)$, we further define $\hat{s}(t,\mu) := \frac{1}{\mu}d\left(\frac{t}{\mu}\right)$, where μ is a positive parameter. If $\int_{-\infty}^{+\infty} |t|\,d(t)dt < +\infty$, then a smoothing approximation for $(t)_+$ is formed. In particular,

$$\hat{p}(t,\mu) = \int_{-\infty}^{+\infty} (t-s)_+\hat{s}(s,\mu)ds = \int_{-\infty}^{t} (t-s)\hat{s}(s,\mu)ds \approx (t)_+.$$

The following are four well-known smoothing functions for the plus function [4,38]:

$$\hat{\phi}_1(\mu,t) = t + \mu \ln\left(1 + e^{-\frac{t}{\mu}}\right), \tag{11}$$

$$\hat{\phi}_2(\mu,t) = \begin{cases} t & \text{if} \quad t \geq \frac{\mu}{2}, \\ \frac{1}{2\mu}\left(t + \frac{\mu}{2}\right)^2 & \text{if} - \frac{\mu}{2} < t < \frac{\mu}{2}, \\ 0 & \text{if} \quad t \leq -\frac{\mu}{2}, \end{cases} \tag{12}$$

$$\hat{\phi}_3(\mu,t) = \frac{\sqrt{4\mu^2 + t^2} + t}{2}, \tag{13}$$

$$\hat{\phi}_4(\mu,t) = \begin{cases} t - \frac{\mu}{2} & \text{if} \quad t > \mu, \\ \frac{t^2}{2\mu} & \text{if} \quad 0 \leq t \leq \mu, \\ 0 & \text{if} \quad t < 0, \end{cases} \tag{14}$$

where the corresponding kernel functions are

$$d_1(t) = \frac{e^{-x}}{(1+e^{-x})^2},$$

$$d_2(t) = \begin{cases} 1 & \text{if } -\frac{1}{2} \leq x \leq \frac{1}{2}, \\ 0 & \text{otherwise}, \end{cases}$$

$$d_3(t) = \frac{2}{(x^2+4)^{\frac{3}{2}}},$$

$$d_4(t) = \begin{cases} 1 & \text{if } 0 \leq x \leq 1, \\ 0 & \text{otherwise}. \end{cases}$$

Next, in light of $|t| = (t)_+ + (-t)_-$, the smoothing function of $|t|$ via convolution can be written as

$$\hat{p}(|t|,\mu) = \hat{p}(t,\mu) + \hat{p}(-t,\mu) = \int_{-\infty}^{+\infty} |t-s|\,\hat{s}(s,\mu)ds.$$

Analogous to (11)–(14), we achieve the following smoothing functions for $|t|$:

$$\phi_1(\mu,t) = \mu\left[\ln\left(1 + e^{-\frac{t}{\mu}}\right) + \ln\left(1 + e^{\frac{t}{\mu}}\right)\right], \tag{15}$$

$$\phi_2(\mu,t) = \begin{cases} t & \text{if} \quad t \geq \frac{\mu}{2}, \\ \frac{t^2}{\mu} + \frac{\mu}{4} & \text{if } -\frac{\mu}{2} < t < \frac{\mu}{2}, \\ -t & \text{if} \quad t \leq -\frac{\mu}{2}, \end{cases} \tag{16}$$

$$\phi_3(\mu,t) = \sqrt{4\mu^2 + t^2}, \tag{17}$$

$$\phi_4(\mu,t) = \begin{cases} \frac{t^2}{2\mu} & \text{if } |t| \leq \mu, \\ |t| - \frac{\mu}{2} & \text{if } |t| > \mu. \end{cases} \tag{18}$$

If we take a Epanechnikov kernel function

$$K(t) = \begin{cases} \frac{3}{4}(1-t^2) & \text{if } |t| \leq 1, \\ 0 & \text{otherwise}, \end{cases}$$

then we obtain the following smoothing function for $|t|$:

$$\phi_5(\mu,t) = \begin{cases} t & \text{if} \quad t > \mu, \\ -\frac{t^4}{8\mu^3} + \frac{3t^2}{4\mu} + \frac{3\mu}{8} & \text{if} - \mu \leq t \leq \mu, \\ -t & \text{if} \quad t < \mu. \end{cases} \tag{19}$$

Moreover, taking a Gaussian kernel function $K(t) = \frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}$ for all $t \in \mathbb{R}$ yields

$$\hat{s}(t,\mu) := \frac{1}{\mu}K\left(\frac{t}{\mu}\right) = \frac{1}{\sqrt{2\pi\mu^2}}e^{-\frac{t^2}{2\mu^2}},$$

and it leads to the smoothing function [44] for $|t|$:

$$\phi_6(\mu, t) = t\,\mathrm{erf}\left(\frac{t}{\sqrt{2}\mu}\right) + \sqrt{\frac{2}{\pi}}\mu e^{-\frac{t^2}{2\mu^2}}, \tag{20}$$

where the error function is defined by

$$\mathrm{erf}(t) = \frac{2}{\sqrt{\pi}} \int_0^t e^{-u^2} du \quad \forall t \in \mathbb{R}.$$

In summary, we have constructed six smoothing functions from the above discussions. Can all the above functions serve as smoothing functions for solving SOCAVE? The answer is affirmative because it is not hard to verify that each ϕ_i possesses $-1 \le \frac{d}{dt}\phi_i(\mu, t) \le 1$. Thus, these six functions will be adopted for our numerical implementations. Accordingly, we need to define $\Phi_i(\mu, x)$ and $H_i(\mu, x)$ based on each ϕ_i. For subsequent needs, we only present the expression of each Jacobian matrix $H_i'(\mu, x)$ without detailed derivations.

Based on each ϕ_i, let $\Phi_i : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ for $i = 1, 2, \cdots, 6$ be similarly defined as in (5), i.e.

$$\Phi_i(\mu, x) = \phi_i\left(\mu, \lambda_1(x)\right) u_x^{(1)} + \phi_i\left(\mu, \lambda_2(x)\right) u_x^{(2)} \tag{21}$$

and $H_i : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ for $i = 1, 2, \cdots, 6$ be similarly defined as in (6), i.e.

$$H_i(\mu, x) = \begin{bmatrix} \mu \\ Ax + B\Phi_i(\mu, x) - b \end{bmatrix}, \quad \forall \mu \in \mathbb{R}_{++} \text{ and } x \in \mathbb{R}^n. \tag{22}$$

Then, each H_i is continuously differentiable on $\mathbb{R}_{++} \times \mathbb{R}^n$ with the Jacobian matrix given by

$$H_i'(\mu, x) = \begin{bmatrix} 1 & 0 \\ B\,\frac{\partial \Phi_i(\mu,x)}{\partial \mu} & A + B\,\frac{\partial \Phi_i(\mu,x)}{\partial x} \end{bmatrix} \tag{23}$$

for all $(\mu, x) \in \mathbb{R}_{++} \times \mathbb{R}^n$ with $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$. Moreover, the differentiation of each Φ_i is expressed as below.

(1) The Jacobian of Φ_1 is characterized as below.

$$\begin{aligned}
&\frac{\partial \Phi_1(\mu, x)}{\partial \mu} \\
&= \frac{\partial \phi_1(\mu, \lambda_1(x))}{\partial \mu} u_x^{(1)} + \frac{\partial \phi_1(\mu, \lambda_2(x))}{\partial \mu} u_x^{(2)} \\
&= \left[\frac{\phi_1(\mu, \lambda_1(x))}{\mu} + \frac{\lambda_1(x)}{\mu} \cdot \frac{1 - e^{\frac{\lambda_1(x)}{\mu}}}{1 + e^{\frac{\lambda_1(x)}{\mu}}}\right] u_x^{(1)} + \left[\frac{\phi_1(\mu, \lambda_2(x))}{\mu} + \frac{\lambda_2(x)}{\mu} \cdot \frac{1 - e^{\frac{\lambda_2(x)}{\mu}}}{1 + e^{\frac{\lambda_2(x)}{\mu}}}\right] u_x^{(2)},
\end{aligned}$$

$$\frac{\partial \Phi_1(\mu, x)}{\partial x} = \begin{cases} \dfrac{e^{\frac{x_1}{\mu}} - 1}{e^{\frac{x_1}{\mu}} + 1} I & \text{if } x_2 = 0, \\ \begin{bmatrix} b_1 & c_1 \frac{x_2^T}{\|x_2\|} \\ c_1 \frac{x_2}{\|x_2\|} & a_1 I + (b_1 - a_1)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix} & \text{if } x_2 \neq 0, \end{cases}$$

with

$$\begin{aligned}
a_1 &= \frac{\phi_1(\mu, \lambda_2(x)) - \phi_1(\mu, \lambda_1(x))}{\lambda_2(x) - \lambda_1(x)}, \\
b_1 &= \frac{1}{2}\left(\frac{e^{\frac{\lambda_1(x)}{\mu}} - 1}{e^{\frac{\lambda_1(x)}{\mu}} + 1} + \frac{e^{\frac{\lambda_2(x)}{\mu}} - 1}{e^{\frac{\lambda_2(x)}{\mu}} + 1}\right), \\
c_1 &= \frac{1}{2}\left(\frac{1 - e^{\frac{\lambda_1(x)}{\mu}}}{e^{\frac{\lambda_1(x)}{\mu}} + 1} + \frac{e^{\frac{\lambda_2(x)}{\mu}} - 1}{e^{\frac{\lambda_2(x)}{\mu}} + 1}\right).
\end{aligned}$$

(2) The Jacobian of Φ_2 is characterized as below.

$$\frac{\partial \Phi_2(\mu, x)}{\partial \mu} = \frac{\partial \phi_2(\mu, \lambda_1(x))}{\partial \mu} u_x^{(1)} + \frac{\partial \phi_2(\mu, \lambda_2(x))}{\partial \mu} u_x^{(2)}$$

with

$$\frac{\partial \phi_2(\mu,\lambda_i(x))}{\partial \mu} = \begin{cases} 0 & \text{if } \lambda_i(x) \geq \frac{\mu}{2}, \\ -\left(\frac{\lambda_i(x)}{\mu}\right)^2 + \frac{1}{4} & \text{if } -\frac{\mu}{2} < \lambda_i(x) < \frac{\mu}{2}, \\ 0 & \text{if } \lambda_i(x) \leq -\frac{\mu}{2}, \end{cases}$$

$$\frac{\partial \Phi_2(\mu,x)}{\partial x} = \begin{cases} dI & \text{if } x_2 = 0, \\ \begin{bmatrix} b_2 & c_2 \frac{x_2^T}{\|x_2\|} \\ c_2 \frac{x_2}{\|x_2\|} & a_2 I + (b_2 - a_2)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix} & \text{if } x_2 \neq 0, \end{cases}$$

with

$$a_2 = \frac{\phi_2(\mu,\lambda_2(x)) - \phi_2(\mu,\lambda_1(x))}{\lambda_2(x) - \lambda_1(x)},$$

$$b_2 = \begin{cases} 0 & \text{if } \lambda_2(x) \geq \frac{\mu}{2} > -\frac{\mu}{2} \geq \lambda_1(x), \\ 1 & \text{if } \lambda_2(x) > \lambda_1(x) \geq \frac{\mu}{2}, \\ \frac{\lambda_1(x)}{\mu} + \frac{1}{2} & \text{if } \lambda_2(x) \geq \frac{\mu}{2} > \lambda_1(x) > -\frac{\mu}{2}, \\ \frac{\lambda_1(x)+\lambda_2(x)}{\mu} & \text{if } \frac{\mu}{2} > \lambda_2(x) > \lambda_1(x) > -\frac{\mu}{2}, \\ \frac{\lambda_2(x)}{\mu} - \frac{1}{2} & \text{if } \frac{\mu}{2} > \lambda_2(x) > -\frac{\mu}{2} \geq \lambda_1(x), \\ -1 & \text{if } \lambda_1(x) < \lambda_2(x) \leq -\frac{\mu}{2}, \end{cases}$$

$$c_2 = \begin{cases} 1 & \text{if } \lambda_2(x) \geq \frac{\mu}{2} > -\frac{\mu}{2} \geq \lambda_1(x), \\ 0 & \text{if } \lambda_2(x) > \lambda_1(x) \geq \frac{\mu}{2}, \\ \frac{1}{2} - \frac{\lambda_1(x)}{\mu} & \text{if } \lambda_2(x) \geq \frac{\mu}{2} > \lambda_1(x) > -\frac{\mu}{2}, \\ \frac{\lambda_2(x)-\lambda_1(x)}{\mu} & \text{if } \frac{\mu}{2} > \lambda_2(x) > \lambda_1(x) > -\frac{\mu}{2}, \\ \frac{\lambda_2(x)}{\mu} + \frac{1}{2} & \text{if } \frac{\mu}{2} > \lambda_2(x) > -\frac{\mu}{2} \geq \lambda_1(x), \\ 0 & \text{if } \lambda_1(x) < \lambda_2(x) \leq -\frac{\mu}{2}, \end{cases}$$

$$d = \begin{cases} 1 & \text{if } x_1 \geq \frac{\mu}{2}, \\ \frac{2x_1}{\mu} & \text{if } -\frac{\mu}{2} < x_1 < \frac{\mu}{2}, \\ -1 & \text{if } x_1 \leq -\frac{\mu}{2}. \end{cases}$$

(3) The Jacobian of Φ_3 is characterized as below.

$$\frac{\partial \Phi_3(\mu,x)}{\partial \mu} = \frac{4\mu}{\sqrt{4\mu^2 + \lambda_1^2(x)}} u_x^{(1)} + \frac{4\mu}{\sqrt{4\mu^2 + \lambda_2^2(x)}} u_x^{(2)}$$

$$\frac{\partial \Phi_3(\mu,x)}{\partial x} = \begin{cases} \frac{x_1}{\sqrt{4\mu^2 + x_1^2}} I & \text{if } x_2 = 0, \\ \begin{bmatrix} b_3 & c_3 \frac{x_2^T}{\|x_2\|} \\ c_3 \frac{x_2}{\|x_2\|} & a_1 I + (b_1 - a_1)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix} & \text{if } x_2 \neq 0, \end{cases}$$

with

$$a_3 = \frac{\phi_3(\mu,\lambda_2(x)) - \phi_3(\mu,\lambda_1(x))}{\lambda_2(x) - \lambda_1(x)},$$

$$b_3 = \frac{1}{2}\left(\frac{\lambda_1(x)}{\sqrt{4\mu^2 + \lambda_1^2(x)}} + \frac{\lambda_2(x)}{\sqrt{4\mu^2 + \lambda_2^2(x)}}\right),$$

$$c_3 = \frac{1}{2}\left(\frac{-\lambda_1(x)}{\sqrt{4\mu^2 + \lambda_1^2(x)}} + \frac{\lambda_2(x)}{\sqrt{4\mu^2 + \lambda_2^2(x)}}\right).$$

(4) The Jacobian of Φ_4 is characterized as below.

$$\frac{\partial \Phi_4(\mu,x)}{\partial \mu} = \frac{\partial \phi_4(\mu,\lambda_1(x))}{\partial \mu} u_x^{(1)} + \frac{\partial \phi_4(\mu,\lambda_2(x))}{\partial \mu} u_x^{(2)}$$

with

$$\frac{\partial \phi_4(\mu,\lambda_i(x))}{\partial \mu} = \begin{cases} -\frac{1}{2} & \text{if } \lambda_i(x) > \mu, \\ -\frac{1}{2}\left(\frac{\lambda_i(x)}{\mu}\right)^2 & \text{if } -\mu \le \lambda_i(x) \le \mu, \\ -\frac{1}{2} & \text{if } \lambda_i(x) < -\mu, \end{cases}$$

$$\frac{\partial \Phi_4(\mu,x)}{\partial x} = \begin{cases} eI & \text{if } x_2 = 0, \\ \begin{bmatrix} b_4 & c_4 \frac{x_2^T}{\|x_2\|} \\ c_4 \frac{x_2}{\|x_2\|} & a_4 I + (b_4 - a_4)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix} & \text{if } x_2 \neq 0, \end{cases}$$

with

$$a_4 = \frac{\phi_4(\mu,\lambda_2(x)) - \phi_4(\mu,\lambda_1(x))}{\lambda_2(x) - \lambda_1(x)},$$

$$b_4 = \begin{cases} 0 & \text{if } \lambda_2(x) > \mu > -\mu > \lambda_1(x), \\ 1 & \text{if } \lambda_2(x) > \lambda_1(x) > \mu, \\ \frac{\lambda_1(x)}{2\mu} + \frac{1}{2} & \text{if } \lambda_2(x) > \mu \ge \lambda_1(x) \ge -\mu, \\ \frac{\lambda_1(x)+\lambda_2(x)}{2\mu} & \text{if } \mu \ge \lambda_2(x) > \lambda_1(x) \ge -\mu, \\ \frac{\lambda_2(x)}{2\mu} - \frac{1}{2} & \text{if } \mu \ge \lambda_2(x) \ge -\mu > \lambda_1(x), \\ -1 & \text{if } \lambda_1(x) < \lambda_2(x) < -\mu, \end{cases}$$

$$c_4 = \begin{cases} 1 & \text{if } \lambda_2(x) > \mu > -\mu > \lambda_1(x), \\ 0 & \text{if } \lambda_2(x) > \lambda_1(x) > \mu, \\ \frac{1}{2} - \frac{\lambda_1(x)}{2\mu} & \text{if } \lambda_2(x) > \mu \ge \lambda_1(x) \ge -\mu, \\ \frac{\lambda_2(x)-\lambda_1(x)}{2\mu} & \text{if } \mu \ge \lambda_2(x) > \lambda_1(x) \ge -\mu, \\ \frac{\lambda_2(x)}{2\mu} + \frac{1}{2} & \text{if } \mu \ge \lambda_2(x) \ge -\mu > \lambda_1(x), \\ 0 & \text{if } \lambda_1(x) < \lambda_2(x) < -\mu, \end{cases}$$

$$e = \begin{cases} 1 & \text{if } x_1 > \mu, \\ \frac{x_1}{\mu} & \text{if } -\mu \le x_1 \le \mu, \\ -1 & \text{if } x_1 < -\mu. \end{cases}$$

(5) The Jacobian of Φ_5 is characterized as below.

$$\frac{\partial \Phi_5(\mu,x)}{\partial \mu} = \frac{\partial \phi_5(\mu,\lambda_1(x))}{\partial \mu} u_x^{(1)} + \frac{\partial \phi_5(\mu,\lambda_2(x))}{\partial \mu} u_x^{(2)}$$

with

$$\frac{\partial \phi_5(\mu,\lambda_i(x))}{\partial \mu} = \begin{cases} 0 & \text{if } \lambda_i(x) > \mu, \\ \frac{3}{8}\left(\left(\frac{\lambda_i(x)}{\mu}\right)^2 - 1\right)^2 & \text{if } -\mu \le \lambda_i(x) \le \mu, \\ 0 & \text{if } \lambda_i(x) < -\mu, \end{cases}$$

$$\frac{\partial \Phi_5(\mu,x)}{\partial x} = \begin{cases} eI & \text{if } x_2 = 0, \\ \begin{bmatrix} b_5 & c_5 \frac{x_2^T}{\|x_2\|} \\ c_5 \frac{x_2}{\|x_2\|} & a_5 I + (b_5 - a_5)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix} & \text{if } x_2 \neq 0, \end{cases}$$

with

$$a_5 = \frac{\phi_5(\mu,\lambda_2(x)) - \phi_5(\mu,\lambda_1(x))}{\lambda_2(x) - \lambda_1(x)},$$

$$b_5 = \begin{cases} 0 & \text{if } \lambda_2(x) > \mu > -\mu > \lambda_1(x), \\ 1 & \text{if } \lambda_2(x) > \lambda_1(x) > \mu, \\ -\frac{1}{4}\left(\frac{\lambda_1(x)}{\mu}\right)^3 + \frac{3}{4}\frac{\lambda_1(x)}{\mu} + \frac{1}{2} & \text{if } \lambda_2(x) > \mu \ge \lambda_1(x) \ge -\mu, \\ -\frac{1}{4}\frac{\lambda_1^3(x)+\lambda_2^3(x)}{\mu^3} + \frac{3}{4}\frac{\lambda_1(x)+\lambda_2(x)}{\mu} & \text{if } \mu \ge \lambda_2(x) > \lambda_1(x) \ge -\mu, \\ -\frac{1}{4}\left(\frac{\lambda_2(x)}{\mu}\right)^3 + \frac{3}{4}\frac{\lambda_2(x)}{\mu} - \frac{1}{2} & \text{if } \mu \ge \lambda_2(x) \ge -\mu > \lambda_1(x), \\ -1 & \text{if } \lambda_1(x) < \lambda_2(x) < -\mu, \end{cases}$$

$$c_5 = \begin{cases} 1 & \text{if } \lambda_2(x) > \mu > -\mu > \lambda_1(x), \\ 0 & \text{if } \lambda_2(x) > \lambda_1(x) > \mu, \\ \frac{1}{2} + \frac{1}{4}\left(\frac{\lambda_1(x)}{\mu}\right)^3 - \frac{3}{4}\frac{\lambda_1(x)}{\mu} & \text{if } \lambda_2(x) > \mu \ge \lambda_1(x) \ge -\mu, \\ -\frac{1}{4}\frac{\lambda_2^3(x)-\lambda_1^3(x)}{\mu^3} + \frac{3}{4}\frac{\lambda_2(x)+\lambda_1(x)}{\mu} & \text{if } \mu \ge \lambda_2(x) > \lambda_1(x) \ge -\mu, \\ -\frac{1}{4}\left(\frac{\lambda_2(x)}{\mu}\right)^3 + \frac{3}{4}\frac{\lambda_2(x)}{\mu} + \frac{1}{2} & \text{if } \mu \ge \lambda_2(x) \ge -\mu > \lambda_1(x), \\ 0 & \text{if } \lambda_1(x) < \lambda_2(x) < -\mu, \end{cases}$$

$$e = \begin{cases} 1 & \text{if } x_1 > \mu, \\ -\frac{1}{2}\left(\frac{x_1}{\mu}\right)^3 + \frac{3}{2}\frac{x_1}{\mu} & \text{if } -\mu \le x_1 \le \mu, \\ -1 & \text{if } x_1 < -\mu. \end{cases}$$

(6) The Jacobian of Φ_6 is characterized as below.

$$\frac{\partial \Phi_6(\mu, x)}{\partial \mu} = \sqrt{\frac{2}{\pi}} e^{-\frac{\lambda_1^2(x)}{2\mu^2}} u_x^{(1)} + \sqrt{\frac{2}{\pi}} e^{-\frac{\lambda_2^2(x)}{2\mu^2}} u_x^{(2)}$$

$$\frac{\partial \Phi_6(\mu, x)}{\partial x} = \begin{cases} \operatorname{erf}\left(\frac{x_1}{\sqrt{2}\mu}\right) I & \text{if } x_2 = 0, \\ \begin{bmatrix} b_6 & c_6 \frac{x_2^T}{\|x_2\|} \\ c_6 \frac{x_2}{\|x_2\|} & a_6 I + (b_6 - a_6)\frac{x_2 x_2^T}{\|x_2\|^2} \end{bmatrix} & \text{if } x_2 \neq 0, \end{cases}$$

with

$$a_6 = \frac{\phi_6(\mu, \lambda_2(x)) - \phi_6(\mu, \lambda_1(x))}{\lambda_2(x) - \lambda_1(x)},$$

$$b_6 = \frac{1}{2}\left(\operatorname{erf}\left(\frac{\lambda_1(x)}{\sqrt{2}\mu}\right) + \operatorname{erf}\left(\frac{\lambda_2(x)}{\sqrt{2}\mu}\right)\right),$$

$$c_6 = \frac{1}{2}\left(\operatorname{erf}\left(\frac{\lambda_2(x)}{\sqrt{2}\mu}\right) - \operatorname{erf}\left(\frac{\lambda_1(x)}{\sqrt{2}\mu}\right)\right).$$

3. Smoothing Newton method

In this section, we study the smoothing Newton algorithm based on the smoothing function $\Phi_i(\mu, x)$ for $i \in \{1, 2, \dots, 6\}$ to solve the SOCAVE (2), and show its convergence properties. First, we present the generic framework of the smoothing Newton algorithm.

Algorithm 3.1 *(A smoothing Newton algorithm).*

Step 0 Choose $\delta \in (0, 1)$, $\sigma \in (0, 1)$, and $\mu_0 \in \mathbb{R}_{++}$, $x^0 \in \mathbb{R}^n$. Set $z^0 := (\mu_0, x^0)$, $e := (1, 0) \in \mathbb{R} \times \mathbb{R}^{n-1}$. Choose $\beta > 1$ satisfying $\min\{1, \|H_i(z^0)\|^2\} \le \beta\mu_0$. Set $k := 0$.
Step 1 If $\|H_i(z^k)\| = 0$, stop. Otherwise, set $\tau_k := \min\{1, \|H_i(z^k)\|\}$.
Step 2 Compute $\triangle z^k = (\triangle\mu_k, \triangle x^k) \in \mathbb{R} \times \mathbb{R}^n$ by

$$H_i(z^k) + H_i'(z^k)\triangle z^k = \frac{1}{\beta}\tau_k^2 e, \tag{24}$$

where $H_i'(z^k)$ denotes the Jacobian matrix of $H_i(z^k)$ at (μ_k, x^k) given by (7).
Step 3 Let α_k be the maximum of the values $1, \delta, \delta^2, \cdots$ such that

$$\|H_i(z^k + \alpha_k \triangle z^k)\| \le \left[1 - \sigma(1 - \frac{1}{\beta})\alpha_k\right] \|H_i(z^k)\|. \tag{25}$$

Step 4 Set $z^{k+1} := z^k + \alpha_k \triangle z^k$ and $k := k + 1$. Go to *Step* 1.

Theorem 2.1 indicates the Newton equation (24) in Algorithm 3.1 is solvable. It paves a way to show that the linear search (25) in Algorithm 3.1 is well-defined which is presented in Theorem 3.1 as below. More specifically, from [16, Lemma 3.1], we know that there exists an $\bar{\alpha} \in (0, 1]$ such that (25) holds for any $\alpha \in (0, \bar{\alpha}]$. This indicates that taking

$\alpha = \max\{1, \delta, \delta^2, \dots\}$, where $\delta \in (0, 1)$ in step 3, leads to (25) being well-defined. Indeed, the detailed arguments are very similar to those in [15,17,36], we only state it here and omit its proof.

Theorem 3.1. *Consider a SOCAVE (2) with $\sigma_{\min}(A) > \sigma_{\max}(B)$. Then, for $\triangle z \in \mathbb{R} \times \mathbb{R}^n$ given by (24), the linear search (25) is well-defined.*

Next, we discuss the convergence of Algorithm 3.1. To this end, we need the following results whose arguments are also similar to the ones in [17, Remark 2.1]. In particular, for Theorem 3.2(d), we provide a proof in light of the structure of each ϕ_i so that the readers can look into the analytic difference among them.

Theorem 3.2. *Consider a SOCAVE (2) with $\sigma_{\min}(A) > \sigma_{\max}(B)$. Let H_i be defined as in (22). Suppose that the sequence $\{z^k\}$ is generated by Algorithm 3.1. Then, the following results hold.*

(a) *The sequences $\{\|H_i(z^k)\|\}$ and $\{\tau_k\}$ are monotonically non-increasing.*
(b) *$\beta\mu_k \geq \tau_k^2$ for all k.*
(c) *The sequence $\{\mu_k\}$ is monotonically non-increasing and $\mu_k > 0$ for all k.*
(d) *The sequence $\{z^k\}$ is bounded.*

Proof. (a) From definition of the line search in (25) and $\tau_k := \min\{1, \|H_i(z^k)\|\}$, it is clear that $\{\|H_i(z^k)\|\}$ and $\{\tau_k\}$ are monotonically non-increasing.

(b) We prove this conclusion by induction. First, by Algorithm 3.1, it is clear that $\tau_0^2 \leq \beta\mu_0$ with τ_0, β and μ_0 chosen in Algorithm 3.1. Secondly, we suppose that $\tau_k^2 \leq \beta\mu_k$ for some k. Then, for $k+1$, we have

$$\begin{aligned}
\mu_{k+1} - \frac{\tau_{k+1}^2}{\beta} &= \mu_k + \alpha_k \triangle\mu_k - \frac{\tau_{k+1}^2}{\beta} \\
&= (1-\alpha_k)\mu_k + \alpha_k \frac{\tau_k^2}{\beta} - \frac{\tau_{k+1}^2}{\beta} \\
&\geq (1-\alpha_k)\frac{\tau_k^2}{\beta} + \alpha_k \frac{\tau_k^2}{\beta} - \frac{\tau_{k+1}^2}{\beta} \\
&\geq 0,
\end{aligned}$$

where the second equality holds due to the Newton equation (24), and the second inequality holds due to part (a). Hence, it follows that $\beta\mu_k \geq \tau_k^2$ for all k.

(c) From the iterative scheme $z^{k+1} = z^k + \alpha_k \Delta z^k$, we know $\mu_{k+1} = \mu_k + \alpha_k \triangle\mu_k$. By the Newton equations (24) and the line search as in (25) again, it follows that

$$\mu_{k+1} = (1-\alpha_k)\mu_k + \alpha_k \frac{\tau_k^2}{\beta} \geq (1-\alpha_k)\frac{\tau_k^2}{\beta} + \alpha_k \frac{\tau_k^2}{\beta} > 0$$

for all k. On the other hand, we have

$$\mu_{k+1} = (1-\alpha_k)\mu_k + \alpha_k \frac{\tau_k^2}{\beta} \leq (1-\alpha_k)\mu_k + \alpha_k\mu_k \leq \mu_k,$$

where the first inequality holds due to part (b). Hence, the sequence $\{\mu_k\}$ is monotonically non-increasing and $\mu_k > 0$ for all k.

(d) From part (a), we know the sequence $\{\|H_i(z^k)\|\}$ is bounded, which means there is a constant C such that $\|H_i(z^k)\| \leq C$. Thus,

$$\begin{aligned}
C &\geq \|H_i(z^k)\| \\
&\geq \left\| Ax^k + B\Phi_i(\mu_k, x^k) - b \right\| \\
&\geq \left\| Ax^k \right\| - \left\| B\Phi_i(\mu_k, x^k) \right\| - \|b\| \\
&= \sqrt{(x^k)^T A^T A x^k} - \sqrt{[\Phi_i(\mu_k, x^k)]^T B^T B \Phi_i(\mu_k, x^k)} - \|b\| \\
&\geq \sqrt{\lambda_{\min}(A^T A)}\|x^k\| - \sqrt{\lambda_{\max}(B^T B)\|\Phi_i(\mu_k, x^k)\|^2} - \|b\| \\
&= \sqrt{\lambda_{\min}(A^T A)} \left\|x^k\right\| - \sqrt{\lambda_{\max}(B^T B) \left\| \phi_i(\mu_k, \lambda_1(x^k))u_x^{(1)} + \phi_i(\mu_k, \lambda_2(x^k))u_x^{(2)} \right\|^2} - \|b\|
\end{aligned}$$

$$= \sqrt{\lambda_{\min}(A^TA)}\,\left\|x^k\right\| - \sqrt{\lambda_{\max}(B^TB)\left[\phi_i^2(\mu_k,\lambda_1(x^k))\|u_x^{(1)}\|^2 + \phi_i^2(\mu_k,\lambda_2(x^k))\|u_x^{(2)}\|^2\right]} - \|b\|$$

$$= \sqrt{\lambda_{\min}(A^TA)}\,\left\|x^k\right\| - \sqrt{\lambda_{\max}(B^TB)\frac{1}{2}\left[\phi_i^2(\mu_k,\lambda_1(x^k)) + \phi_i^2(\mu_k,\lambda_2(x^k))\right]} - \|b\|.$$

On the other hand, for $i = 1, 2, 3, 4, 5$, we see that

$$\phi_i^2(\mu_k,\lambda_1(x^k)) + \phi_i^2(\mu_k,\lambda_2(x^k)) = \sum_{j=1}^{2} f_i(\mu,\lambda_j(x^k)) + \lambda_1^2(x^k) + \lambda_2^2(x^k).$$

(i) For $i = 1$, we have $f_1(\mu,\lambda_j(x^k)) = 4\mu_k^2 \ln(e^{\frac{\lambda_j(x^k)}{\mu_k}} + 1)\ln(e^{-\frac{\lambda_j(x^k)}{\mu_k}} + 1)$. It is known that the function $g(t) = 4\ln(e^t + 1)\ln(e^{-t} + 1)$ is bounded for all $t \in \mathbb{R}$. It follows that there exists N_1 such that $\left|\sum_{j=1}^2 f_1(\mu,\lambda_j(x^k))\right| \le \mu_k^2 \mathrm{N}_1$.

(ii) For $i = 2, 4, 5$, it is easy to verify that there exist N_i such that $\left|\sum_{j=1}^2 f_i(\mu,\lambda_j(x^k))\right| \le \mu_k^2 \mathrm{N}_i$.

For $i = 3$, we have $\sum_{j=1}^2 f_i(\mu,\lambda_j(x^k)) = 8\mu_k^2 := \mu_k^2\mathrm{N}_3$, which yields

$$\begin{aligned}\phi_i^2(\mu_k,\lambda_1(x^k)) + \phi_i^2(\mu_k,\lambda_2(x^k)) &\le \mu_k^2\,\mathrm{N_i} + \lambda_1^2(x^k) + \lambda_2^2(x^k)\\ &= 2\left(\mu_k^2\,\frac{\mathrm{N_i}}{2} + \|x^k\|^2\right)\\ &\le 2\left(\mu_k\sqrt{\frac{\mathrm{N_i}}{2}} + \|x^k\|\right)^2.\end{aligned}$$

This together with $\|H_i(z^k)\| \le C$ implies that

$$\|x^k\| \le \frac{C + 2\sqrt{\lambda_{\max}(B^TB)}\,\mu_k\sqrt{\frac{\mathrm{N_i}}{2}} + \|b\|}{\sqrt{\lambda_{\min}(A^TA)} - \sqrt{\lambda_{\max}(B^TB)}}$$

holds for all k. Thus, the sequence $\{x^k\}$ is bounded.

(iv) For $i = 6$, we know that $|\mathrm{erf}(t)| \le 1$ and $0 < e^{-\frac{t^2}{2\mu^2}} \le 1$. Thus, it leads to

$$\begin{aligned}&\phi_6^2(\mu_k,\lambda_1(x^k)) + \phi_6^2(\mu_k,\lambda_2(x^k))\\ &\le \left(\lambda_1(x_k) + \sqrt{\frac{2}{\pi}}\mu_k\right)^2 + \left(\lambda_2(x_k) + \sqrt{\frac{2}{\pi}}\mu_k\right)^2\\ &\le 2\left(\mu_k\sqrt{\frac{2}{\pi}} + \|x^k\|\right)^2\end{aligned}$$

where the last inequality is due to $|\lambda_1(x_k)| + |\lambda_2(x_k)| \le 2\|x^k\|$. Then, it follows that

$$\|x^k\| \le \frac{C + 2\sqrt{\lambda_{\max}(B^TB)}\,\mu_k\sqrt{\frac{2}{\pi}} + \|b\|}{\sqrt{\lambda_{\min}(A^TA)} - \sqrt{\lambda_{\max}(B^TB)}}$$

holds for all k. Thus, the sequence $\{x^k\}$ is bounded.

From all the above, the proof is complete. □

Now, we shall show that any sequence $\{z_k\}$ is generated by Algorithm 3.1 convergent to a solution to the SOCAVE (2). In the next theorem we demonstrate that under our assumptions. The proof is essentially similar to a result [13,36, Theorem 4.1]. Hence, we omit the detailed proof and only present the convergence result.

Theorem 3.3. *Consider a SOCAVE (2) with $\sigma_{\min}(A) > \sigma_{\max}(B)$. Suppose that $\{z^k\}$ is generated by Algorithm 3.1. Then, any accumulation point of $\{z^k\}$ is a solution to the SOCAVE (2).*

Algorithm 3.1 possesses the local quadratic convergence rate. In fact, we can achieve it by similar arguments as those in [36,39, Theorem 8].

Theorem 3.4. *Consider a SOCAVE (2) with $\sigma_{\min}(A) > \sigma_{\max}(B)$. Let H_i be defined as in (6) and $z^\star$ be the unique solution to SOCAVE (2). Suppose that all $V \in \partial H_i(z^\star)$ are nonsingular. Then, the whole sequence $\{z^k\}$ converges to $z^\star$, and $\|z^{k+1} - z^\star\| = O(\|z^k - z^\star\|^2)$.*

4. Numerical results

In this section, we report some numerical results via five numerical examples to evaluate the efficiency of Algorithm 3.1. First, in our experiments, we set parameters as

$$\mu_0 = 0.1, \quad x_0 = \text{rand}(n, 1), \quad \delta = 0.5, \quad \sigma = 10^{-5} \quad \text{and} \quad \beta = \max(1, 1.01 * \tau_0^2/\mu).$$

We stop the iterations when $\|H(z_k)\| \le 10^{-6}$ or the number of iterations exceeds 100. All the experiments are done on a PC with Intel(R) CPU of 2.40 GHz and RAM of 4.00 GHz, and all the programming codes are written in Matlab and run in Matlab environment.

For each problem, we implement the smoothing Newton Algorithm 3.1 with six different smoothing functions $\phi_1(\mu, t)$, $\phi_2(\mu, t)$, $\phi_3(\mu, t)$, $\phi_4(\mu, t)$, $\phi_5(\mu, t)$, $\phi_6(\mu, t)$, respectively. Each problem is randomly generated 50 times and the average results are listed in Tables, where n denotes the size of problem, *itn* denotes the average number of iterations, *time* denotes the average value of the CPU time in seconds and *fails* means the number of failures.

Secondly, in order to compare the performance of smoothing function $\phi_i(\mu, t)$, for $i = 1, 2, 3, 4, 5, 6$ in the smoothing Newton Algorithm 3.1, we adopt the performance profile which is introduced in [7] as a means. In other words, we regard Algorithm 3.1 corresponding to a smoothing function $\phi_i(\mu, t)$, for $i = 1, 2, 3, 4, 5, 6$ as a solver, and assume that there are n_s solvers and n_p test problems from the test set $\mathcal{P}$ which is generated randomly. We are interested in using the iteration number and computing time as performance measure for Algorithm 3.1 with different $\phi_i(\mu, t)$. For each problem p and solver s, let

$$f_{p,s} = \text{iteration number (or computing time) required to solve problem } p \text{ by solver } s.$$

We employ the performance ratio

$$r_{p,s} := \frac{f_{p,s}}{\min\{f_{p,s} : s \in \mathcal{S}\}},$$

where $\mathcal{S}$ is the four solvers set. We assume that a parameter $r_{p,s} \le r_M$ for all p, s are chosen, and $r_{p,s} = r_M$ if and only if solver s does not solve problem p. In order to obtain an overall assessment for each solver, we define

$$\rho_s(\tau) := \frac{1}{n_p}\text{size}\{p \in \mathcal{P} : r_{p,s} \le \tau\},$$

which is called the performance profile of the number of iteration for solver s. Then, $\rho_s(\tau)$ is the probability for solver $s \in \mathcal{S}$ that a performance ratio $f_{p,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio. The performance profiles of each problem are depicted in Figs. 1–10.

Problem 4.1. Consider the SOCAVE (2) which is generated in the following way: first choose two random matrices $B, C \in \mathbb{R}^{n\times n}$ from a uniform distribution on $[-10, 10]$ for every element. We compute the maximal singular value σ_1 of B and the minimal singular value σ_2 of C, and let $\sigma := \min\{1, \sigma_2/\sigma_1\}$. Next, we divide C by σ multiplied by a random number in the interval $[0, 1]$, and the resulting matrix is denoted as A. Accordingly, the minimum singular values of A exceeds the maximal singular value of B. We choose randomly $b \in \mathbb{R}^n$ on $[0, 1]$ for every element. By Algorithm 3.1 in this paper, the resulting SOCAVE (2) is solvable. The initial point is chosen in the range $[0, 1]$ entry-wisely. Note that a similar way to construct the problem was given in [14].

Table 1 and Figs. 1–2 show that function $\phi_1(\mu, t)$ performs worst, whereas the difference among other functions is very slight. Fig. 1 demonstrates the performance profile of iteration numbers for Problem 4.1. The subplot in Fig. 1 is the zoomed plot for upper-left part of the Fig. 1. Fig. 2 shows the performance profile of computing time for Problem 4.1. The subplot in Fig. 2 is the zoomed plot for lower-left part of Fig. 2. From this figure, we can see that the performance of function $\phi_1(\mu, t)$ is also the worst one.

Problem 4.2. Consider the SOCAVE (2) which is generated in the following way: choose two random matrices $C, D \in \mathbb{R}^{n\times n}$ from a uniform distribution on $[-10, 10]$ for every element, and compute their singular value decompositions $C := U_1 S_1 V_1^T$ and $D := U_2 S_2 V_2^T$ with diagonal matrices S_1 and S_2; unitary matrices U_1, V_1, U_2 and V_2. Then, we choose randomly $b, c \in \mathbb{R}^n$ on $[0, 10]$ for every element. Next, we take $a \in \mathbb{R}^n$ by setting $a_i = c_i + 10$ for all $i \in \{1, \ldots, n\}$, so that $a \ge b$. Set $A := U_1 \text{Diag(a)} V_1^T$ and $B := U_2 \text{Diag(b)} V_2^T$, where Diag(x) denotes a diagonal matrix with its i-th diagonal element being x_i. The gap between the minimal singular value of A and the maximal singular value of B is limited and can be very small. We choose randomly $b \in \mathbb{R}^n$ in $[0, 10]$. The initial point is chosen in the range $[0, 1]$ entry-wisely.

For Problem 4.2, as depicted in Figs. 3–4 and Table 2, all the smoothing functions $\phi_i(\mu, t)$ for $i = 1, 2, 3, 4, 5, 6$ perform very well, without any discrepancy.

Table 1
Numerical results for Problem 4.1.

n	ϕ_1			ϕ_2			ϕ_3			ϕ_4			ϕ_5			ϕ_6		
	itn	time	fails	itn	time	fails	itn	time	fails	itn	time	fails	itn	time	fails	itn	time	fails
200	3.000	0.018	0	3.000	0.018	0	3.000	0.019	0	3.000	0.017	0	3.000	0.017	0	3.00	0.019	0
300	3.000	0.036	3	3.000	0.037	0	2.980	0.036	0	2.980	0.037	0	3.000	0.034	0	3.00	0.033	0
400	3.000	0.077	1	3.000	0.072	1	3.000	0.075	1	3.000	0.076	1	3.000	0.079	1	3.00	0.081	1
500	3.000	0.120	4	3.000	0.123	1	3.000	0.124	1	3.000	0.128	1	3.000	0.126	1	3.00	0.121	1
600	3.000	0.180	3	3.000	0.194	1	3.000	0.189	1	3.000	0.197	1	3.000	0.193	1	3.00	0.183	1
700	3.000	0.264	7	3.060	0.269	1	3.061	0.273	1	3.061	0.255	1	3.061	0.277	1	3.061	0.271	1
800	3.000	0.369	2	3.000	0.380	0	3.000	0.377	0	3.000	0.371	0	3.000	0.360	0	3.00	0.379	0
900	3.000	0.496	5	3.020	0.511	1	3.020	0.494	1	3.020	0.484	1	3.020	0.489	1	3.020	0.510	1
1000	3.000	0.643	5	3.080	0.681	1	3.041	0.644	1	3.122	0.678	1	3.082	0.665	1	3.082	0.687	1
1200	3.000	1.010	7	3.000	1.009	3	3.000	0.995	3	3.000	1.000	3	3.000	1.019	3	3.000	1.018	3
1500	3.000	1.726	13	3.570	2.097	4	3.205	1.795	6	3.500	2.003	4	3.565	2.080	4	3.565	2.102	4
2000	3.000	3.474	19	3.330	3.997	8	3.049	3.548	9	3.429	4.150	8	3.333	4.009	8	3.333	3.937	8

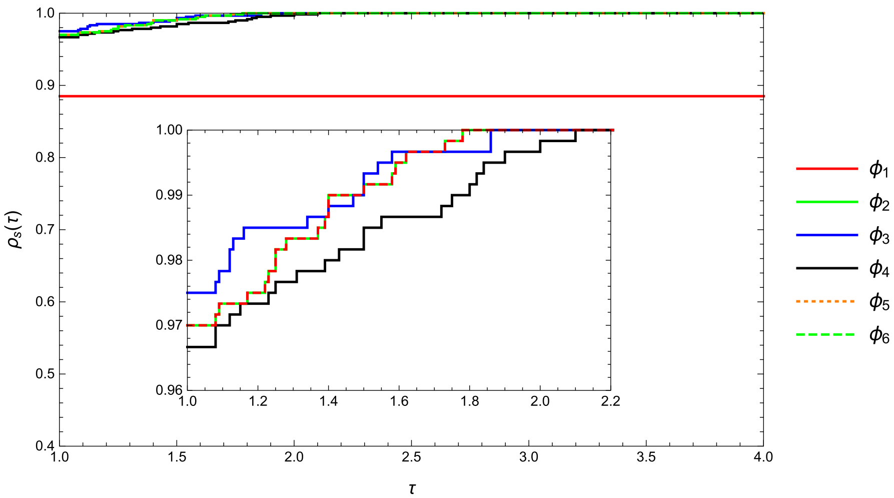

Fig. 1. Performance profile of iteration numbers of Problem 4.1. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

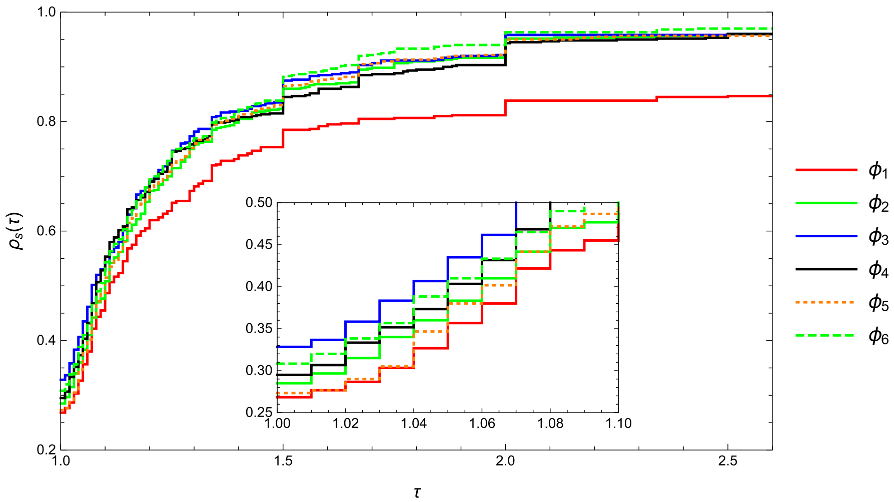

Fig. 2. Performance profile of computing time of Problem 4.1.

Problem 4.3. Consider the SOCAVE (2) which is generated in the following way: choose two random matrices $A, B \in \mathbb{R}^{n\times n}$ from a uniform distribution on $[-10, 10]$ for every element. In order to ensure that the SOCAVE (2) is solvable, we update the matrix A by the following: let $[USV] = svd(A)$. If $\min\{S(i,i)\} = 0$ for $i = 0, 1, \cdots, n$, we make $A = U(S + 0.01E)V$, and

Table 2
Numerical results for Problem 4.2.

n	ϕ_1			ϕ_2			ϕ_3			ϕ_4			ϕ_5			ϕ_6		
	itn	time	fails	itn	time	fails	itn	time	fails	itn	time	fails	itn	time	fails	itn	time	fails
200	4.560	0.027	0	4.560	0.024	0	4.560	0.027	0	4.560	0.024	0	4.560	0.026	0	4.560	0.027	0
300	4.660	0.057	0	4.660	0.053	0	4.660	0.056	0	4.660	0.055	0	4.660	0.054	0	4.660	0.058	0
400	4.780	0.129	0	4.780	0.129	0	4.780	0.134	0	4.780	0.135	0	4.780	0.129	0	4.780	0.134	0
500	4.800	0.210	0	4.800	0.213	0	4.800	0.222	0	4.800	0.221	0	4.800	0.217	0	4.800	0.206	0
600	4.820	0.316	0	4.820	0.319	0	4.820	0.338	0	4.820	0.339	0	4.820	0.334	0	4.820	0.317	0
700	4.900	0.470	0	4.900	0.471	0	4.900	0.495	0	4.900	0.485	0	4.900	0.483	0	4.900	0.454	0
800	4.980	0.693	0	4.980	0.677	0	4.980	0.708	0	4.980	0.707	0	4.980	0.687	0	4.980	0.650	0
900	4.980	0.919	0	4.980	0.899	0	4.980	0.947	0	4.980	0.917	0	4.980	0.912	0	4.980	0.867	0
1000	4.980	1.200	0	4.980	1.174	0	4.980	1.186	0	4.980	1.207	0	4.980	1.207	0	4.980	1.130	0
1200	4.960	1.922	0	4.960	1.856	0	4.960	1.837	0	4.960	1.852	0	4.960	1.894	0	4.960	1.765	0
1500	5.000	3.284	0	5.000	3.219	0	5.000	3.146	0	5.000	3.220	0	5.000	3.204	0	5.000	3.117	0
2000	5.000	6.914	0	5.000	6.609	0	5.000	6.783	0	5.000	6.767	0	5.000	6.712	0	5.000	6.649	0

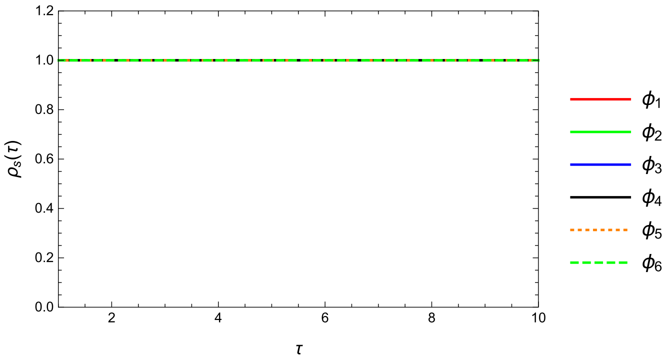

Fig. 3. Performance profile of iteration numbers of Problem 4.2.

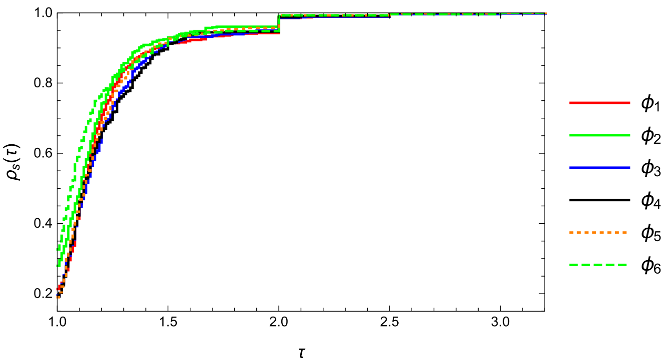

Fig. 4. Performance profile of computing time of Problem 4.2.

then $A = \frac{\lambda_{max}(B^T B)+0.01}{\lambda_{min}(A^T A)} A$. We choose randomly $b \in \mathbb{R}^n$ on $[0, 10]$ for every element. The initial point is chosen in the range $[0, 1]$ entry-wisely.

For Problem 4.3, from the Table 3 and Figs. 5–6, we see that $\phi_1(\mu, t)$ is obviously inferior to other functions. Moreover, in terms of the computing time, the function $\phi_4(\mu, t)$ is the best one, followed by $\phi_3(\mu, t)$. In summary, the function $\phi_1(\mu, t)$ is still the worst performer.

Problem 4.4. We consider the SOCAVE (2) which is generated the same as Problem 4.1. But, here the SOC is given by $\mathcal{K} := \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r}$, where $n_1 = \cdots = n_r = \frac{n}{r}$.

Figs. 7–8 show the performance profiles of Problem 4.4. Indeed, the performance profiles are similar to those for Problem 4.1 (only the cone structure is different). Again, the function $\phi_1(\mu, t)$ is still the worst performer and there is no significant difference among other five smoothing functions.

Table 3
Numerical results for Problem 4.3.

n	ϕ_1			ϕ_2			ϕ_3			ϕ_4			ϕ_5			ϕ_6		
	itn	time	fails	itn	time	fails	itn	time	fails	itn	time	fails	itn	time	fails	itn	time	fails
200	3.000	0.020	0	3.000	0.018	0	3.000	0.019	0	3.000	0.017	0	3.000	0.018	0	3.000	0.020	0
300	2.980	0.036	0	2.980	0.035	0	2.980	0.031	0	2.980	0.033	0	2.980	0.032	0	2.980	0.033	0
400	3.000	0.086	1	2.980	0.082	0	2.980	0.075	0	2.980	0.075	0	2.980	0.076	0	2.980	0.071	0
500	2.980	0.132	0	2.980	0.126	0	2.980	0.118	0	2.980	0.115	0	2.980	0.125	0	2.980	0.124	0
600	2.920	0.198	0	2.920	0.205	0	2.920	0.183	0	2.920	0.174	0	2.920	0.192	0	2.920	0.190	0
700	2.959	0.291	1	2.940	0.287	0	2.940	0.259	0	2.940	0.252	0	2.940	0.274	0	2.940	0.269	0
800	2.980	0.408	1	2.960	0.425	0	2.960	0.378	0	2.960	0.355	0	2.960	0.394	0	2.960	0.392	0
900	3.000	0.564	1	2.980	0.557	0	2.980	0.503	0	2.980	0.479	0	2.980	0.517	0	2.980	0.529	0
1000	2.956	0.697	5	2.880	0.676	0	2.880	0.636	0	2.880	0.602	0	2.880	0.646	0	2.880	0.651	0
1200	3.000	1.095	2	2.980	1.085	0	2.980	1.015	0	2.980	0.975	0	2.980	1.035	0	2.980	1.045	0
1500	2.977	1.927	6	2.900	1.728	0	2.900	1.655	0	2.900	1.627	0	2.900	1.664	0	2.900	1.676	0
2000	3.000	3.755	12	2.960	3.631	0	2.960	3.477	0	2.960	3.430	0	2.960	3.468	0	2.960	3.460	0

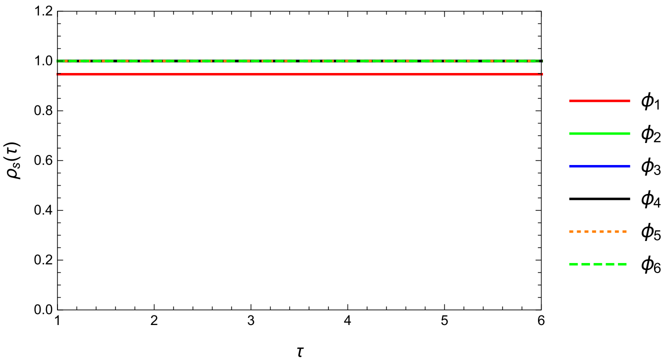

Fig. 5. Performance profile of iteration numbers of Problem 4.3.

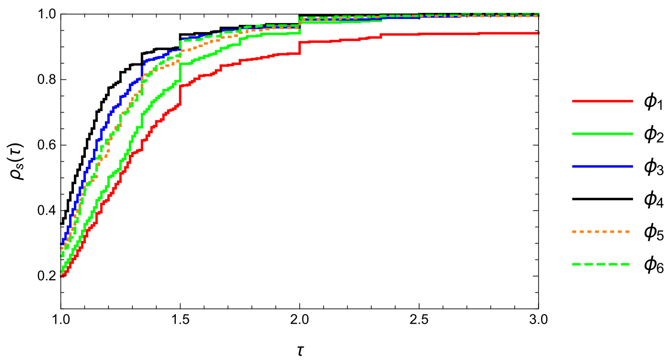

Fig. 6. Performance profile of computing time of Problem 4.3.

Problem 4.5. We consider the SOCAVE (2) which is generated the same as Problem 4.3. But, here the SOC is given by $\mathcal{K} := \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r}$, where $n_1 = \cdots = n_r = \frac{n}{r}$.

Figs. 9–10 show the performance profiles of Problem 4.5. They verify the poor performance of function $\phi_1(\mu, t)$ one more time.

In summary, the function $\phi_1(\mu, t)$ is not a good choice to work with the smoothing Newton algorithm. Note that $\phi_1(\mu, t)$ is related to loss function and widely used in engineering like machine learning. However, for the SOCAVE, the numerical performance of function $\phi_1(\mu, t)$ is always the worst one. This is a very interesting phenomenon and discovery. In other words, we may try to replace it by other smoothing functions for some appropriate algorithms towards real engineering problems. This will be our future investigations.

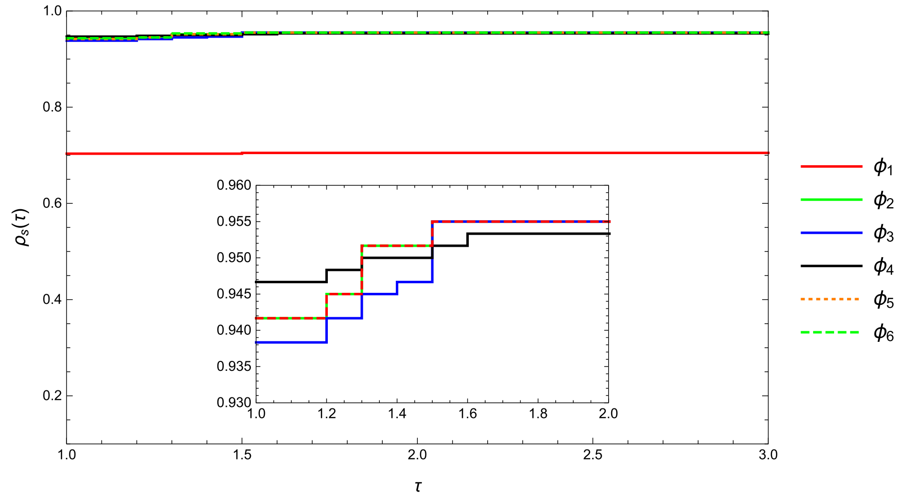

Fig. 7. Performance profile of iteration numbers of Problem 4.4.

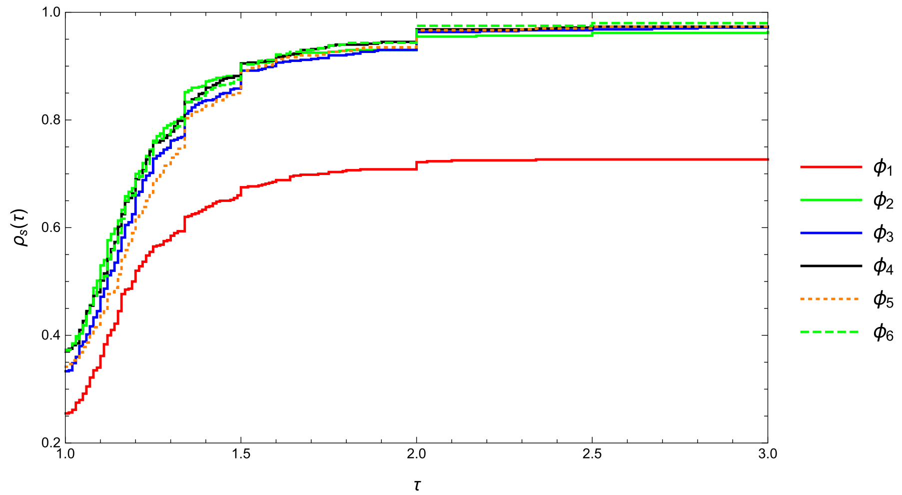

Fig. 8. Performance profile of computing time of Problem 4.4.

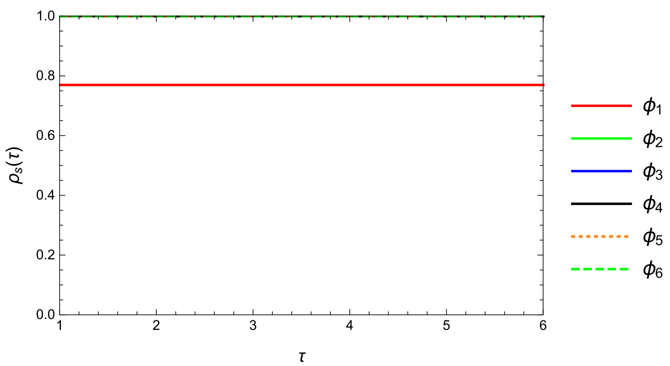

Fig. 9. Performance profile of iteration numbers of Problem 4.5.

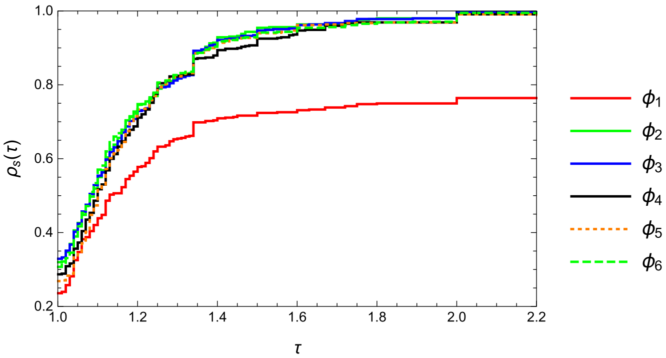

Fig. 10. Performance profile of computing time of Problem 4.5.

References

[1] L. Caccetta, B. Qu, G.-L. Zhou, A globally and quadratically convergent method for absolute value equations, Comput. Optim. Appl. 48 (2011) 45–58.
[2] J.-S. Chen, The convex and monotone functions associated with second-order cone, Optimization 55 (2006) 363–385.
[3] J.-S. Chen, X. Chen, P. Tseng, Analysis of nonsmooth vector-valued functions associated with second-order cones, Math. Program. 101 (2004) 95–117.
[4] C. Chen, O.L. Mangasarian, A class of smoothing functions for nonlinear and mixed complementarity problems, Comput. Optim. Appl. 5 (1996) 97–138.
[5] J.-S. Chen, S.-H. Pan, A survey on SOC complementarity functions and solution methods for SOCPs and SOCCPs, Pac. J. Optim. 8 (2012) 33–74.
[6] J.-S. Chen, P. Tseng, An unconstrained smooth minimization reformulation of second-order cone complementarity problem, Math. Program. 104 (2005) 293–327.
[7] E.D. Dolan, J.J. More, Benchmarking optimization software with performance profiles, Math. Program. 91 (2002) 201–213.
[8] F. Facchinei, J. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, Springer, New York, 2003.
[9] U. Faraut, A. Koranyi, Analysis on Symmetric Cones, Oxford Mathematical Monographs, Oxford University Press, New York, 1994.
[10] M. Fukushima, Z.-Q. Luo, P. Tseng, Smoothing functions for second-order cone complementarity problems, SIAM J. Optim. 12 (2002) 436–460.
[11] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.
[12] S.-L. Hu, Z.-H. Huang, A note on absolute value equations, Optim. Lett. 4 (2010) 417–424.
[13] S.-L. Hu, Z.-H. Huang, P. Wang, A nonmonotone smoothing Newton algorithm for solving nonlinear complementarity problems, Optim. Methods Softw. 24 (2009) 447–460.
[14] S.-L. Hu, Z.-H. Huang, Q. Zhang, A generalized Newton method for absolute value equations associated with second order cones, J. Comput. Appl. Math. 235 (2011) 1490–1501.
[15] Z.-H. Huang, Locating a maximally complementary solution of the monotone NCP by using non-interior-point smoothing algorithms, Math. Methods Oper. Res. 61 (2005) 41–55.
[16] Z.-H. Huang, J. Han, Z. Chen, A predictor–corrector smoothing Newton algorithm, based on a new smoothing function, for solving the nonlinear complementarity problem with a P_0 function, J. Optim. Theory Appl. 117 (2003) 39–68.
[17] Z.-H. Huang, Y. Zhang, W. Wu, A smoothing-type algorithm for solving system of inequalities, J. Comput. Appl. Math. 220 (2008) 355–363.
[18] X.-Q. Jiang, A smoothing Newton method for solving absolute value equations, Adv. Mater. Res. 765–767 (2013) 703–708.
[19] X.-Q. Jiang, Y. Zhang, A smoothing-type algorithm for absolute value equations, J. Ind. Manag. Optim. 9 (2013) 789–798.
[20] S. Ketabchi, H. Moosaei, Minimum norm solution to the absolute value equation in the convex case, J. Optim. Theory Appl. 154 (2012) 1080–1087.
[21] L.-C. Kong, J. Sun, N.-H. Xiu, A regularized smoothing Newton method for symmetric cone complementarity problems, SIAM J. Optim. 19 (2008) 1028–1047.
[22] J. Kreimer, R.Y. Rubinstein, Nondifferentiable optimization via smooth approximation: general analytical approach, Ann. Oper. Res. 39 (1992) 97–119.
[23] X.-H. Liu, W.-Z. Gu, Smoothing Newton algorithm based on a regularized one-parametric class of smoothing functions for generalized complementarity problems over symmetric cones, J. Ind. Manag. Optim. 6 (2010) 363–380.
[24] N. Lu, Z.-H. Huang, Convergence of a non-interior continuation algorithm for the monotone SCCP, Acta Math. Appl. Sin. Engl. Ser. 26 (2010) 543–556.
[25] N. Lu, Y. Zhang, A smoothing-type algorithm for solving inequalities under the order induced by a symmetric cone, J. Inequal. Appl. 2011 (2011).
[26] O.L. Mangasarian, Absolute value programming, Comput. Optim. Appl. 36 (2007) 43–53.
[27] O.L. Mangasarian, Absolute value equation solution via concave minimization, Optim. Lett. 1 (2007) 3–5.
[28] O.L. Mangasarian, A generalized Newton method for absolute value equations, Optim. Lett. 3 (2009) 101–108.
[29] O.L. Mangasarian, Knapsack feasibility as an absolute value equation solvable by successive linear programming, Optim. Lett. 3 (2009) 161–170.
[30] O.L. Mangasarian, Primal–dual bilinear programming solution of the absolute value equation, Optim. Lett. 6 (2012) 1527–1533.
[31] O.L. Mangasarian, Absolute value equation solution via dual complementarity, Optim. Lett. 7 (2013) 625–630.
[32] O.L. Mangasarian, Linear complementarity as absolute value equation solution, Optim. Lett. 8 (2014) 1529–1534.
[33] O.L. Mangasarian, Absolute value equation solution via linear programming, J. Optim. Theory Appl. 161 (2014) 870–876.
[34] O.L. Mangasarian, R.R. Meyer, Absolute value equation, Linear Algebra Appl. 419 (2006) 359–367.
[35] X.-H. Miao, W.-M. Hsu, J.-S. Chen, The solvabilities of three optimization problems associated with second-order cone, 2017, submitted for publication.
[36] X.-H. Miao, J.-T. Yang, B. Saheya, J.-S. Chen, A smoothing Newton method for absolute value equation associated with second-order cone, Appl. Numer. Math. 120 (October 2017) 82–96.
[37] O.A. Prokopyev, On equivalent reformulations for absolute value equations, Comput. Optim. Appl. 44 (2009) 363–372.
[38] L. Qi, D. Sun, Smoothing functions and smoothing Newton method for complementarity and variational inequality problems, J. Optim. Theory Appl. 113 (2002) 121–147.
[39] L. Qi, D. Sun, G.L. Zhou, A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequality problems, Math. Program. 87 (2000) 1–35.
[40] J. Rohn, A theorem of the alternative for the equation $Ax + B|x| = b$, Linear Multilinear Algebra 52 (2004) 421–426.
[41] J. Rohn, Solvability of systems of interval linear equations and inequalities, in: M. Fiedler, J. Nedoma, J. Ramik, J. Rohn, K. Zimmermann (Eds.), Linear Optimization Problems with Inexact Data, Springer, 2006, pp. 35–77.
[42] J. Rohn, An algorithm for solving the absolute value equation, Electron. J. Linear Algebra 18 (2009) 589–599.

[43] B. Saheya, C.-H. Yu, J.-S. Chen, Numerical comparisons based on four smoothing functions for absolute value equation, J. Appl. Math. Comput. 56 (2018) 131–149.
[44] S. Voronin, G. Ozkaya, D. Yoshida, Convolution based smooth approximations to the absolute value function with application to non-smooth regularization, arXiv:1408.6795v2 [math.NA], July 2015.
[45] S. Yamanaka, M. Fukushima, A branch and bound method for the absolute value programs, Optimization 63 (2014) 305–319.
[46] J.G. Zhu, H.W. Liu, X.L. Li, A regularized smoothing-type algorithm for solving a system of inequalities with a P_0-function, J. Comput. Appl. Math. 233 (2010) 2611–2619.