A COMBINED SMOOTHING AND REGULARIZATION METHOD FOR MONOTONE SECOND-ORDER CONE COMPLEMENTARITY PROBLEMS

SHUNSUKE HAYASHI, NOBUO YAMASHITA, AND MASAO FUKUSHIMA

Abstract. The Second-Order Cone Complementarity Problem (SOCCP) is a wide class of problems containing the Nonlinear Complementarity Problem (NCP) and the Second-Order Cone Programming Problem (SOCP). Recently, Fukushima, Luo and Tseng extended some merit functions and their smoothing functions for NCP to SOCCP. Moreover, they derived computable formulas for the Jacobians of the smoothing functions and gave conditions for the Jacobians to be invertible. In this paper, we propose a globally and quadratically convergent algorithm, which is based on smoothing and regularization methods, for solving monotone SOCCP. In particular, we study strong semismoothness and Jacobian consistency, which play an important role in establishing quadratic convergence of the algorithm. Furthermore, we examine effectiveness of the algorithm by means of numerical experiments.

Key words. second-order cone, complementarity problem, smoothing method, regularization method

AMS subject classifications. 90C33, 65K05

1. Introduction. The second-order cone complementarity problem (SOCCP) [11] is to find \((x, y, \zeta) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n\) such that

\[(1.1) \quad x \in \mathcal{K}, \; y \in \mathcal{K}, \; x^Ty = 0, \; F(x, y, \zeta) = 0,\]

where \(F: \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^n\) is a continuously differentiable mapping, and \(\mathcal{K} \subset \mathbb{R}^n\) is the Cartesian product of second-order cones, that is, \(\mathcal{K} = \mathcal{K}^{n_1} \times \mathcal{K}^{n_2} \times \cdots \times \mathcal{K}^{n_m}\) with \(n = n_1 + \cdots + n_m\), and the \(n_i\)-dimensional second-order cone \(\mathcal{K}^{n_i} \subset \mathbb{R}^{n_i}\) defined by

\[
\mathcal{K}^{n_i} := \left\{ (z_1, z_2) \in \mathbb{R} \times \mathbb{R}^{n_i - 1} \mid \|z_2\|_2 \leq z_1 \right\}.
\]

The SOCCP contains a wide class of problems such as the Nonlinear Complementarity Problem (NCP) and the Second-Order Cone Programming Problem (SOCP) [13]. For example, SOCCP with \(n_1 = n_2 = \cdots = n_m = 1\) and \(F(x, y, \zeta) = f(x) - y\) is NCP, and the KKT conditions for SOCP (possibly including nonlinear functions) reduce to SOCCP.

In this paper, we focus on the special SOCCP: Find \((x, y) \in \mathbb{R}^n \times \mathbb{R}^n\) such that

\[(1.2) \quad x \in \mathcal{K}, \; y \in \mathcal{K}, \; x^Ty = 0, \; y = f(x),\]

where \(f\) is a continuously differentiable mapping from \(\mathbb{R}^n\) to \(\mathbb{R}^n\). This SOCCP may seem rather restrictive. However, the KKT conditions for any SOCP with continuously differentiable functions can be written in the form of SOCCP (1.2). For more detail, see Section 6.

Recently, a number of methods for solving SOCCP have been proposed. For SOCP including affine functions, primal-dual interior-point methods [13, 15, 25] are
shown to be effective. For SOCCP, Fukushima, Luo and Tseng [11], Chen, Sun and Sun [5], and Chen, Chen and Tseng [2] studied smoothing and nonsmooth approaches. Especially, Fukushima, Luo and Tseng [11] showed that the min function and the Fischer-Burmeister function for the NCP can be extended to the SOCCP by means of Jordan algebra. Furthermore, they constructed smoothing functions for those functions and analyzed the properties of their Jacobians.

In this paper, we introduce not only smoothing methods but also regularization methods for SOCCPs. Smoothing methods have effectively been adopted to deal with nondifferentiable reformulations of complementarity problems [1, 3, 4, 12, 17, 19, 21, 26]. On the other hand, regularization methods have provided a fundamental tool to deal with ill-posed problems [7, 8, 17]. By combining these methods, we develop a hybrid algorithm for solving monotone SOCCPs. Moreover, we show conditions for the algorithm to be globally and quadratically convergent. Particularly, we highlight the two properties called strong semismoothness and Jacobian consistency, which play a crucial role in establishing quadratic convergence of the algorithm.

This paper is organized as follows. In Section 2, we review the spectral factorization associated with a second-order cone, which plays a key role in analyzing the properties of merit functions for the SOCCP. Moreover, we construct a merit function by means of the natural residual for the SOCCP. In Section 3, we introduce smoothing and regularization methods, which smoothen the natural residual and weaken the condition for the merit function to be level-bounded. In Section 4, we propose a prototype algorithm for solving the SOCCP and show that it has global convergence when applied to monotone SOCCPs. Moreover, by incorporating Newton’s method, we construct a concrete algorithm and establish quadratic convergence. In Section 5, we present some numerical results with the latter algorithm. In Section 6, we conclude the paper with some remarks.

Throughout this paper, we use the following notations: \( \| \cdot \| \) denotes the 2-norm defined by \( \| z \| = \sqrt{z^T z} \) for a vector \( z \) and \( \| M \| := \max_{z \in \mathbb{R}^n \setminus \{0\}} \| Mz \| / \| z \| \) for a matrix \( M \in \mathbb{R}^{n \times n} \). \( B(z, \varepsilon) \) denotes the open sphere with center \( z \) and radius \( \varepsilon \). For a given set \( S \), \( \text{cl} S \) denotes the closure of \( S \) and \( \text{co} S \) denotes the convex hull of \( S \). We often write \( z = (z_1, z_2) \) for \( (z_1, z_2^T)^T \).

2. Preliminaries.

2.1. Spectral factorization. We briefly review some properties of the spectral factorization with respect to a second-order cone, which will be used in the subsequent analysis. Spectral factorization is one of the basic concepts in Jordan algebra. For more detail, see [10, 11].

For any vector \( z = (z_1, z_2) \in \mathbb{R} \times \mathbb{R}^{n-1} \) (\( n \geq 2 \)), its spectral factorization with respect to the second-order cone \( K^n \) is defined as
\[
z = \lambda_1 u^1 + \lambda_2 u^2,
\]
where \( \lambda_1 \) and \( \lambda_2 \) are the spectral values given by
\[
\lambda_i = z_1 + (-1)^i \| z_2 \|, \quad i = 1, 2,
\]
and \( u^1 \) and \( u^2 \) are the spectral vectors given by
\[
u^i = \begin{cases} \frac{1}{2} \left( 1, (-1)^i \frac{z_2}{\| z_2 \|} \right) & (z_2 \neq 0), \\ \frac{1}{2} \left( 1, (-1)^i w \right) & (z_2 = 0), \end{cases} \quad i = 1, 2,
\]
with \( w \in \mathbb{R}^{n-1} \) such that \( \|w\| = 1 \). The spectral values and vectors have the following properties. For any \( z \in \mathbb{R}^n \), the inequality \( \lambda_1 \leq \lambda_2 \) holds, and \( \lambda_1 \geq 0 \) \( \iff \) \( z \in \mathcal{K}^n \).

Moreover, for any \( z \in \mathbb{R}^n \), we have \( \|u^i\| = 1/\sqrt{2} \) for \( i = 1, 2 \), and \( (u^1)^T u^2 = 0 \).

For any \( z \in \mathbb{R}^n \), let \( P_{\mathcal{K}^n}(z) \) denote the projection of \( z \) onto the second-order cone \( \mathcal{K}^n \), that is,

\[
P_{\mathcal{K}^n}(z) := \arg\min_{z' \in \mathcal{K}^n} \|z' - z\|.
\]

In particular, when \( n = 1 \), \( P_{\mathcal{K}^1}(\alpha) = \max\{0, \alpha\} \) for any \( \alpha \in \mathbb{R} \). For \( n \geq 2 \), the projection function \( P_{\mathcal{K}^n} \) can be calculated by using the spectral values and vectors of \( z \) [11], that is,

\[
P_{\mathcal{K}^n}(z) = \max\{0, \lambda_1\}u^1 + \max\{0, \lambda_2\}u^2,
\]

where \( \lambda_1 \) and \( \lambda_2 \) are the spectral values of \( z \) defined by (2.1), and \( u^1 \) and \( u^2 \) are the spectral vectors of \( z \) defined by (2.2).

**2.2. Merit function.** In this paper, we consider the unconstrained optimization reformulation of SOCCP (1.2):

\[
\text{Minimize } \Psi(x, y),
\]

where \( \Psi \) is a real-valued function on \( \mathbb{R}^n \times \mathbb{R}^n \). The objective function \( \Psi \) is called a merit function for SOCCP (1.2). In order to construct a merit function for SOCCP (1.2), it is convenient to introduce a function \( \Phi : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n \) satisfying

\[
\Phi(x, y) = 0 \iff x \in \mathcal{K}, \ y \in \mathcal{K}, \ x^T y = 0.
\]

By using such a function, we define \( H : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{2n} \) by

\[
H(x, y) := \left( \Phi(x, y) \ f(x) - y \right).
\]

It is obvious that SOCCP (1.2) is equivalent to the equation \( H(x, y) = 0 \). Moreover, we define function \( \Psi : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R} \) by

\[
\Psi(x, y) := \frac{1}{2}\|H(x, y)\|^2 = \frac{1}{2}\|\Phi(x, y)\|^2 + \frac{1}{2}\|f(x) - y\|^2.
\]

Then, it is easy to see that \( \Psi(x, y) \geq 0 \) for any \( (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \), and that \( \Psi(x, y) = 0 \) if and only if \( (x, y) \) is a solution of SOCCP (1.2). Therefore, the function \( \Phi \) defined by (2.6) can serve as a merit function for SOCCP (1.2).

Note that the complementarity condition on \( \mathcal{K} = \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_m} \) can be decomposed into complementarity conditions on each \( \mathcal{K}^{n_i} \), that is,

\[
x \in \mathcal{K}, \ y \in \mathcal{K}, \ x^T y = 0 \iff x^i \in \mathcal{K}^{n_i}, \ y^i \in \mathcal{K}^{n_i}, \ (x^i)^T y^i = 0 \quad (i = 1, \ldots, m),
\]

where \( x = (x^1, \ldots, x^m) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m} \) and \( y = (y^1, \ldots, y^m) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m} \).

This fact naturally leads us to construct a function \( \Phi \) satisfying (2.5) as

\[
\Phi(x, y) := \begin{pmatrix} \varphi^1(x^1, y^1) \\ \vdots \\ \varphi^m(x^m, y^m) \end{pmatrix},
\]
where \( \varphi^i : \mathbb{R}^{n_i} \times \mathbb{R}^{n_i} \to \mathbb{R}^{n_i} \) is a function satisfying

\[
\varphi^i(x^i, y^i) = 0 \iff x^i \in K^{n_i}, \ y^i \in K^{n_i}, (x^i)^T y^i = 0
\]

for each \( i = 1, \ldots, m \). Fukushima, Luo and Tseng [11] showed that (2.8) holds for the natural residual function \( \varphi^i_{\text{nr}} : \mathbb{R}^{n_i} \times \mathbb{R}^{n_i} \to \mathbb{R}^{n_i} \) defined by

\[
\varphi^i_{\text{nr}}(x^i, y^i) := x^i - P_{K^{n_i}}(x^i - y^i).
\]

Using this function, we define function \( \Phi_{\text{nr}} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \) by

\[
\Phi_{\text{nr}}(x, y) := \begin{pmatrix}
\varphi^1_{\text{nr}}(x^1, y^1) \\
\vdots \\
\varphi^m_{\text{nr}}(x^m, y^m)
\end{pmatrix}
\]

and \( H_{\text{nr}} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{2n} \) by

\[
H_{\text{nr}}(x, y) := \begin{pmatrix}
\Phi_{\text{nr}}(x, y) \\
f(x) - y
\end{pmatrix}.
\]

Then, we can construct a merit function \( \Psi_{\text{nr}} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \) for SOCCP (1.2) as

\[
\Psi_{\text{nr}}(x, y) := \frac{1}{2}\|H_{\text{nr}}(x, y)\|^2 = \frac{1}{2}\sum_{i=1}^{m} \|\varphi^i_{\text{nr}}(x^i, y^i)\|^2 + \frac{1}{2}\|f(x) - y\|^2.
\]

In what follows, we write \( \varphi_{\text{nr}} \) for \( \varphi^1_{\text{nr}} \) for simplicity of notation.

### 3. Smoothing and regularization.

In the previous section, we have constructed the merit function \( \Psi_{\text{nr}} \) from the natural residual \( \varphi_{\text{nr}} \). We can solve SOCCP (1.2) by minimizing \( \Psi_{\text{nr}} \) by an appropriate descent algorithm. However, the function \( \Psi_{\text{nr}} \) is not differentiable, and hence, methods that use the gradient of the function, such as the steepest descent method and Newton’s method, are not applicable. In order to get rid of this difficulty, we first introduce a smoothing method that solves a sequence of differentiable approximations to the original nondifferentiable problem.

To ensure global convergence of a descent method, the level-boundedness of the objective function plays an important role. If the function involved in the SOCCP is strongly monotone, then the merit function \( \Psi_{\text{nr}} \) is level-bounded. But the assumption of strong monotonicity is quite restrictive from a practical standpoint. To be amenable to a merely monotone problem, we propose to combine a regularization method with a smoothing method.

In the remainder of the paper, we assume \( K = K^n \). Then we can rewrite SOCCP (1.2) as follows: Find \( (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \) such that

\[
x \in K^n, \ y \in K^n, \ x^T y = 0, \ y = f(x).
\]

The assumption \( K = K^n \) is only for simplicity of presentation. In view of (2.7), the results obtained in the following can be extended to the general \( K \) in a straightforward manner (See Section 6). Under this assumption, \( \Phi_{\text{nr}}(x, y) \) is reduced to \( \varphi_{\text{nr}}(x, y) \), and hence, the vector-valued function \( H_{\text{nr}} \) and the merit function \( \Psi_{\text{nr}} \) are respectively rewritten as

\[
H_{\text{nr}}(x, y) = \begin{pmatrix}
\varphi_{\text{nr}}(x, y) \\
f(x) - y
\end{pmatrix},
\]

\[
\Psi_{\text{nr}}(x, y) = \frac{1}{2}\|H_{\text{nr}}(x, y)\|^2 = \frac{1}{2}\|\varphi_{\text{nr}}(x, y)\|^2 + \frac{1}{2}\|f(x) - y\|^2.
\]
3.1. Smoothing functions. In this subsection, we introduce a class of smoothing functions of the merit function $\Psi_{\text{nat}}$. For a non-differentiable function $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$, we consider a function $h_\mu : \mathbb{R}^n \rightarrow \mathbb{R}^m$ with a parameter $\mu > 0$ that has the following properties:

(a) $h_\mu$ is differentiable for any $\mu > 0$,
(b) $\lim_{\mu \downarrow 0} h_\mu(x) = h(x)$ for any $x \in \mathbb{R}^n$.

Such a function $h_\mu$ is called a smoothing function of $h$. Instead of handling the non-smooth equation $h(x) = 0$ directly, the smoothing method solves a family of smoothed subproblems $h_\mu(x) = 0$ for $\mu > 0$, and obtain a solution of the original problem by letting $\mu \downarrow 0$. Fukushima, Luo and Tseng [11] extended Chen and Mangasarian’s class [1] of smoothing functions for NCP to SOCCP, which may be regarded as a smoothing function of the natural residual $\varphi_{\text{nat}}$.

First we define a smoothing function of the projection function $P_{\nu}$ defined by (2.3). To this end, we consider a continuously differentiable convex function $\hat{g} : \mathbb{R} \rightarrow \mathbb{R}$ such that

$$
\lim_{\alpha \rightarrow -\infty} \hat{g}(\alpha) = 0, \quad \lim_{\alpha \rightarrow -\infty} (\hat{g}(\alpha) - \alpha) = 0, \quad 0 < \hat{g}'(\alpha) < 1.
$$

For example, $\hat{g}_1(\alpha) = (\sqrt{\alpha^4 + 4} + \alpha)/2$ and $\hat{g}_2(\alpha) = \ln(e^\alpha + 1)$ satisfy (3.3). By using $\hat{g}$, we define function $P_\mu : \mathbb{R}^n \rightarrow \mathbb{R}^n$ by

$$
P_\mu(z) := \mu \hat{g}(\lambda_1/\mu)u^1 + \mu \hat{g}(\lambda_2/\mu)u^2,
$$

where $\lambda_1$ and $\lambda_2$ are the spectral values of $z$ given by (2.1), and $u^1$ and $u^2$ are the spectral vectors of $z$ given by (2.2). Fukushima, Luo and Tseng [11] showed that the function $P_{1}$ defined by (3.4) is a smoothing function of $P_{\nu}$, by using the facts that $\lim_{\mu \downarrow 0} \mu \hat{g}(\lambda/\mu) = \max\{0, \lambda\}$ and that $\gamma_\mu(\lambda) := \mu \hat{g}(\lambda/\mu)$ is differentiable for any $\mu > 0$. Hence, from the definition (2.9) of $\varphi_{\text{nat}}$, the function $\varphi_{\mu} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ defined by

$$
\varphi_{\mu}(x, y) := x - P_{\mu}(x - y)
$$

becomes a smoothing function of $\varphi_{\text{nat}}$. In particular, by [11, Proposition 5.1], there exists a positive constant $\nu$ such that

$$
\|\varphi_{\mu}(x, y) - \varphi_{\text{nat}}(x, y)\| \leq \nu \mu
$$

for any $\mu > 0$ and $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$. Likewise, function $\Psi_{\mu} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ given by $\Psi_{\mu}(x, y) := (1/2)\|\varphi_{\mu}(x, y)\|^2 + (1/2)\|f(x) - y\|^2$ serves as a smoothing function of the merit function $\Psi_{\text{nat}}$. In the following, we denote $P_0(z) := P_{\nu}(z)$, $\varphi_0(x, y) := \varphi_{\text{nat}}(x, y)$ and $\Psi_0(x, y) := \Psi_{\text{nat}}(x, y)$.

3.2. Regularization method. Since $\Psi_{\mu}$ is differentiable, we may apply an appropriate gradient-based descent method to obtain a minimum $(x_\mu, y_\mu)$ of the function $\Psi_{\mu}$ for $\mu > 0$. In order that a sequence generated by such a descent method has an accumulation point, it is desirable that $\Psi_{\mu}$ is level-bounded, that is, the level sets $L_\alpha := \{ (x, y) \mid \Psi_{\mu}(x, y) \leq \alpha \}$ are bounded for all $\alpha \in \mathbb{R}$. Actually, by using (3.5) and the results for natural residual of variational inequality problems [16, 27], we can show that $\Psi_{\mu}$ is level-bounded for any $\mu \geq 0$ if $f$ is strongly monotone, that is, for some $\varepsilon > 0$,

$$
(x - y)^T(f(x) - f(y)) \geq \varepsilon \|x - y\|^2, \quad \forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n.
$$
However, the strong monotonicity is quite a severe condition. As a remedy for this inconvenience, we employ a regularization method.

Let the function $f_\varepsilon : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be defined by $f_\varepsilon(x) := f(x) + \varepsilon x$ with a positive parameter $\varepsilon$. The regularization method solves a sequence of SOCCPs involving $f_\varepsilon$ with $\varepsilon > 0$, so that a solution of the original SOCCP is obtained by taking the limit $\varepsilon \downarrow 0$. Define functions $H_{\mu, \varepsilon} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^2$ and $\Psi_{\mu, \varepsilon} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ by

\begin{align}
H_{\mu, \varepsilon}(x, y) &:= \begin{pmatrix} \varphi_\mu(x, y) \\ f_\varepsilon(x) - y \end{pmatrix}, \\
\Psi_{\mu, \varepsilon}(x, y) &:= \frac{1}{2} \| H_{\mu, \varepsilon}(x, y) \|^2 = \frac{1}{2} \| \varphi_\mu(x, y) \|^2 + \frac{1}{2} \| f_\varepsilon(x) - y \|^2.
\end{align}

If $f$ is monotone, that is, (3.6) holds for $\varepsilon = 0$, then $f_\varepsilon$ is strongly monotone for any $\varepsilon > 0$, and hence, the function $\Psi_{\mu, \varepsilon}$ is level-bounded for any $\mu \geq 0$ and $\varepsilon > 0$.

Finally, we give explicit expressions of functions $\nabla H_{\mu, \varepsilon}$ and $\nabla P_\mu$, which will be useful for the subsequent analysis.

**Remark 3.1.** From the definition of $H_{\mu, \varepsilon}$, $\varphi_\mu$, and $f_\varepsilon$, $\nabla H_{\mu, \varepsilon}(x, y)$ can be written as

\begin{equation}
\nabla H_{\mu, \varepsilon}(x, y) = \begin{pmatrix} I - \nabla P_\mu(x - y) & \nabla f_\varepsilon(x) + \varepsilon I \\ \nabla P_\mu(x - y) & -I \end{pmatrix},
\end{equation}

Let $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be defined by $g(z) := \tilde{g}(\lambda_1)u^1 + \tilde{g}(\lambda_2)u^2$, where $\lambda_1$ and $\lambda_2$ are the spectral values of $z$, and $u^1$ and $u^2$ are the spectral vectors of $z$. Then we have $\nabla P_\mu(z) = \nabla g(z/\mu)$ since $P_\mu(z) = \mu g(z/\mu)$ from (3.4). Therefore, by [11, Proposition 5.2], $\nabla P_\mu(z)$ is written as

\begin{equation}
\nabla P_\mu(z) = \begin{cases} \tilde{g}'(z_1/\mu)I & \text{if } z_2 = 0, \\
\begin{pmatrix} b_\mu & c_\mu z_2 \\ \frac{c_\mu z_2}{\| z_2 \|} & a_\mu I + \left( b_\mu - a_\mu \right) \frac{z_2 z_2^T}{\| z_2 \|^2} \end{pmatrix} & \text{if } z_2 \neq 0,
\end{cases}
\end{equation}

where

\begin{equation}
\begin{aligned}
a_\mu &= \frac{\tilde{g}(\lambda_2/\mu) - \tilde{g}(\lambda_1/\mu)}{\lambda_2/\mu - \lambda_1/\mu}, \\
b_\mu &= \frac{1}{2} \left( \tilde{g}'(\lambda_2/\mu) + \tilde{g}'(\lambda_1/\mu) \right), \\
c_\mu &= \frac{1}{2} \left( \tilde{g}'(\lambda_2/\mu) - \tilde{g}'(\lambda_1/\mu) \right).
\end{aligned}
\end{equation}

**4. Algorithm.** In this section, we present a globally and quadratically convergent algorithm for solving SOCCP (3.1). We first give a globally convergent prototype algorithm that utilizes smoothing and regularization techniques. Next, we study two properties called strong semismoothness and Jacobian consistency, which play an important role in establishing quadratic convergence of the algorithm. Finally, we construct a quadratically convergent algorithm by incorporating Newton’s method to the prototype algorithm.

**4.1. Prototype algorithm.** As we have stated in the previous section, for any $\mu > 0$ and $\varepsilon > 0$, the function $\Psi_{\mu, \varepsilon}$ defined by (3.8) is differentiable and level-bounded, provided $f$ is monotone. Therefore, by applying an appropriate descent
method, we may obtain a minimum \((x_{\mu,\varepsilon}, y_{\mu,\varepsilon})\) of the function \(\Psi_{\mu,\varepsilon}\). Moreover, letting \((\mu, \varepsilon)\) converge to \((0, 0)\), we can expect that \((x_{\mu,\varepsilon}, y_{\mu,\varepsilon})\) converges to a solution of the original SOCCP. However, in practice, it is usually impossible to compute an exact minimum of \(\Psi_{\mu,\varepsilon}\). So, we consider the following algorithm in which the function \(\Psi_{\mu,\varepsilon}\) is minimized only approximately at each iteration.

**Algorithm 1.**

1. **Step 0** Choose \((x^{(0)}, y^{(0)}) \in \mathbb{R}^n \times \mathbb{R}^n\), \(\mu_0 \in (0, \infty)\), \(\varepsilon_0 \in (0, \infty)\) and \(\alpha_0 \in (0, \infty)\). Set \(k := 0\).

2. **Step 1** Terminate the iteration if \(\Psi_{\text{min}}(x^{(k)}, y^{(k)}) = 0\).

3. **Step 2** Find a pair \((x^{(k+1)}, y^{(k+1)}) \in \mathbb{R}^n \times \mathbb{R}^n\) such that

   \[(4.1) \quad \Psi_{\mu_k, \varepsilon_k}(x^{(k+1)}, y^{(k+1)}) \leq \alpha_k.\]

4. **Step 3** Update parameters \(\mu_{k+1} \in (0, \mu_k), \varepsilon_{k+1} \in (0, \varepsilon_k)\) and \(\alpha_{k+1} \in (0, \alpha_k)\) so that they converge to 0 eventually. Set \(k := k + 1\). Go back to Step 1.

To obtain \((x^{(k+1)}, y^{(k+1)})\) in Step 2, we may use any suitable unconstrained minimization technique. These issues will be discussed in detail in Subsection 4.4.

In order for Algorithm 1 to be well-defined, there must exist a pair \((x^{(k+1)}, y^{(k+1)})\) satisfying (4.1) for any \(\alpha_k > 0\). For this purpose, it suffices to show that any stationary point of \(\Psi_{\mu,\varepsilon}\) is a global minimum of \(\Psi_{\mu,\varepsilon}\).

**Proposition 4.1.** If \(f : \mathbb{R}^n \to \mathbb{R}^n\) is monotone, then, for any \(\mu > 0\) and \(\varepsilon \geq 0\), every stationary point \((\overline{x}, \overline{y})\) of the function \(\Psi_{\mu,\varepsilon}\) satisfies \(\Psi_{\mu,\varepsilon}(\overline{x}, \overline{y}) = 0\).

**Proof.** Note that \(\nabla \Psi_{\mu,\varepsilon}(\overline{x}, \overline{y}) = \nabla H_{\mu,\varepsilon}(\overline{x}, \overline{y}) H_{\mu,\varepsilon}(\overline{x}, \overline{y}) = 0\). By Proposition 6.1 of [11], \(\nabla H_{\mu,\varepsilon}(\overline{x}, \overline{y})\) is nonsingular. Hence, we have \(H_{\mu,\varepsilon}(\overline{x}, \overline{y}) = 0\), that is, \(\Psi_{\mu,\varepsilon}(\overline{x}, \overline{y}) = (1/2)||H_{\mu,\varepsilon}(\overline{x}, \overline{y})||^2 = 0\).

We proceed to showing the global convergence property of Algorithm 1. To this end, we introduce the weak univalence property. We say function \(H : D \subseteq \mathbb{R}^n \to \mathbb{R}^n\) to be weakly univalent if it is continuous and there exists a sequence of continuous and injective functions \(\{H_k\}\) converging to \(H\) uniformly on a bounded subset of \(D\). In particular, if \(f\) is monotone, then \(H_{\text{min}}\) is weakly univalent from the following reasons: For any \(\mu > 0\) and \(\varepsilon \geq 0\), \(\nabla H_{\mu,\varepsilon}(x, y)\) is nonsingular for any \((x, y) \in \mathbb{R}^n \times \mathbb{R}^n\) [11, Proposition 6.1], and hence, \(H_{\mu,\varepsilon}\) is injective. Moreover, \(H_{\mu,\varepsilon}\) converges to \(H_{\text{min}}\) uniformly on a bounded set as \((\mu, \varepsilon) \downarrow (0, 0)\) since we have from (3.5)

\[
\|H_{\mu,\varepsilon}(x, y) - H_{\text{min}}(x, y)\| \leq \|\varphi_{\mu}(x, y) - \varphi_{\text{min}}(x, y)\| + \|f_{\varepsilon}(x) - f(x)\| 
\leq \nu \mu + \varepsilon \|x\|. \tag{4.2}
\]

The next lemma indicates a property which weakly univalent functions possess.

**Lemma 4.2.** [9, Corollary 3.6.5] Let \(H : \mathbb{R}^n \to \mathbb{R}^n\) be a weakly univalent function such that the inverse image \(H^{-1}(0)\) is nonempty and compact. Then, for any \(\varepsilon > 0\), there exists \(\delta > 0\) such that the following statement holds: For every weakly univalent function \(\overline{H} : \mathbb{R}^n \to \mathbb{R}^n\) such that \(\|H(z) - \overline{H}(z)\| \leq \delta\) for any \(z \in \text{cl}(H^{-1}(0) + B(0, \varepsilon))\), we have \(\emptyset \neq \overline{H}^{-1}(0) \subseteq H^{-1}(0) + B(0, \varepsilon)\), and \(H^{-1}(0)\) is connected.

By using this lemma, we establish the global convergence of Algorithm 1.

**Theorem 4.3.** Let \(f : \mathbb{R}^n \to \mathbb{R}^n\) be a monotone function. Assume that the solution set \(\mathcal{S}\) of SOCCP (3.1) is nonempty and bounded. Let \(\{(x^{(k)}, y^{(k)})\}\) be a sequence generated by Algorithm 1. Then, \(\{(x^{(k)}, y^{(k)})\}\) is bounded, and every accumulation point is a solution of SOCCP (3.1).

**Proof.** From a simple continuity argument, we can easily show that every accumulation point of \(\{(x^{(k)}, y^{(k)})\}\) is a solution of SOCCP (3.1). So we only show the
boundedness of \( \{(x^k, y^k)\} \). For any \( \varepsilon > 0 \), let \( \Omega := \text{cl}(H^{-1}_\text{sn}(0) + B(0, \varepsilon)) \), which is nonempty and compact by the given assumption. Then, there exists \( \delta > 0 \) such that Lemma 4.2 holds for \( H = H_{\text{sn}} \). Let continuous and injective functions \( G_k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{2n} \) be defined by \( G_k(x, y) := H_{\mu_k-1, \varepsilon_k-1}(x, y) - H_{\mu_k-1, \varepsilon_k-1}(x^k, y^k) \). From (4.1) and (4.2), there exists \( \delta \) such that \( \|G_k(x, y) - H_{\text{sn}}(x, y)\| \leq \|H_{\mu_k-1, \varepsilon_k-1}(x, y) - H_{\text{sn}}(x, y)\| + \|H_{\mu_k-1, \varepsilon_k-1}(x^k, y^k)\| \leq \|H_{\mu_k-1, \varepsilon_k-1}(x, y)\| + \|H_{\mu_k-1, \varepsilon_k-1}(x^k, y^k)\| \leq \delta \) for any \( k \geq \delta \) and \( (x, y) \in \Omega \). Moreover, \( G_k \) is a weakly univalent function. Hence, by Lemma 4.2 with \( \tilde{H} = G_k \) and \( H = H_{\text{sn}} \), we have \( G_k^{-1}(0) \subseteq \Omega \) for all \( k \geq \delta \). This together with \( (x^k, y^k) \in G_k^{-1}(0) \) implies the boundedness of \( \{(x^k, y^k)\} \). \( \Box \)

### 4.2. Semismoothness and strong semismoothness

Semismoothness is a generalized concept of the smoothness, which was originally introduced by Mifflin [14] for functionals and extended to vector-valued functions by Qi and Sun [20]. Strong semismoothness is also a generalized concept of the smoothness, which is stronger than the semismoothness. These concepts play an important role in establishing fast local convergence of the algorithm presented later. In this subsection, we first show that the function \( P_{\kappa} \) is strongly semismooth at an arbitrary point. Then, by using this result, we show the strong semismoothness of the function \( H_{\text{sn}} \) defined by (3.2).

Let \( H : \mathbb{R}^n \to \mathbb{R}^n \) be a locally Lipschitzian function. Then \( H \) is differentiable almost everywhere by Rademacher’s Theorem [6]. Let \( D_H \subseteq \mathbb{R}^n \) be the set of points where \( H \) is differentiable. Then, Clarke subdifferential \( \partial H(x) \) of \( H \) at \( x \) is defined by \( \partial H(x) := \text{co} \{ \lim_{z \to x, z \in D_H} \nabla H(z) \} \) [6, 18]. We note that \( \partial H(x) = \{ \nabla H(x) \} \) if \( H \) is continuously differentiable at \( x \). By using the concept of subdifferential, we give the definitions of semismoothness and strong semismoothness.

**Definition 4.4.** A directionally differentiable and locally Lipschitzian function \( H : \mathbb{R}^n \to \mathbb{R}^n \) is said to be semismooth at \( z \) if \( V^T d - H'(z; d) = o(\|d\|) \) for any \( d \in \mathbb{R}^n \setminus \{0\} \) sufficiently small and \( V \in \partial H(z + d) \), where \( H'(z; d) := \lim_{h \to 0} (H(z + hd) - H(z))/h \) is the directional derivative of \( H \) at \( z \) along the direction \( d \). In particular, if \( o(\|d\|) \) can be replaced by \( O(\|d\|^2) \), then function \( H \) is said to be strongly semismooth.

It is known that any piecewise smooth function\(^1\) is semismooth. Moreover, any piecewise smooth function is strongly semismooth if all the derivatives of smooth pieces are locally Lipschitzian. Next we show that the projection function \( P_{\kappa} \) is semismooth everywhere.

**Proposition 4.5.** The projection function \( P_{\kappa} \) is strongly semismooth at any \( z \in \mathbb{R}^n \).

**Proof.** Let \( \lambda_1 \) and \( \lambda_2 \) be the spectral values of \( z \). Since we have from (2.4)

\[
P_{\kappa}(z) = \begin{cases} 
  z & (\lambda_1 \geq 0, \lambda_2 \geq 0) \\
  \frac{1}{2}(z_1 + \|z_2\|) \left(1, \frac{z_2}{\|z_2\|}\right) & (\lambda_1 < 0, \lambda_2 > 0) \\
  0 & (\lambda_1 \leq 0, \lambda_2 \leq 0)
\end{cases}
\]

\( P_{\kappa} \) is continuously differentiable at any \( z \) such that \( \lambda_1 \neq 0 \) and \( \lambda_2 \neq 0 \), and its derivative is locally Lipschitzian. Moreover, \( P_{\kappa} \) is piecewise smooth at any \( z \) such that \( \lambda_2 > \lambda_1 = 0 \) or \( \lambda_1 < \lambda_2 = 0 \). Hence, we have only to show the strong semismoothness at the origin. Note that \( P_{\kappa}(tz) = tP_{\kappa}(z) \) for any \( t \geq 0 \). Since any

\(^1\)Function \( H : \mathbb{R}^n \to \mathbb{R}^n \) is said to be piecewise smooth at \( z \) if there exist \( \varepsilon > 0 \) and a finite number of continuously differentiable functions \( H_i : B(z, \varepsilon) \to \mathbb{R}^m \) \( (i = 1, \ldots, N) \) such that for any \( z' \in B(z, \varepsilon) \) there exists an index \( i \) such that \( H(z') = H_i(z') \).
locally Lipschitzian positively homogeneous function is strongly semismooth at the origin [23], $P_{\kappa_n}$ is strongly semismooth at the origin.

Recently, Chen, Sun and Sun [5] also showed the strong semismoothness of $P_{\kappa_n}$, and Chen, Chen and Tseng [2] gave more general results. We note, however, that their approaches are quite different from ours. Now, we are ready to prove the strong semismoothness of $H_{sn}$.

**Theorem 4.6.** The function $H_{sn}$ defined by (3.2) is semismooth at every point $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$. Moreover, if $\nabla f : \mathbb{R}^n \to \mathbb{R}^{n \times n}$ is locally Lipschitzian, then $H_{sn}$ is strongly semismooth at every point $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$.

**Proof.** Let $(\xi, \eta) \in \mathbb{R}^n \times \mathbb{R}^n$ be a pair of vectors sufficiently small, and $U$ be an arbitrary element of $\partial H_{sn}(x + \xi, y + \eta)$. Note that $P_{\kappa_n}$ is strongly semismooth at $x - y$. Then, since

$$\partial H_{sn}(x + \xi, y + \eta) = \left\{ \begin{pmatrix} I - V & \nabla f(x + \xi) \\ V & -I \end{pmatrix} \mid V \in \partial P_{\kappa_n}((x - y) + (\xi - \eta)) \right\}$$

and

$$H'_{sn}((x, y); (\xi, \eta)) = \begin{pmatrix} -P'_{\kappa_n}(x - y; \xi - \eta) \\ \nabla f(x)^T \xi - \eta \end{pmatrix},$$

we have

$$U^T \begin{pmatrix} \xi \\ \eta \end{pmatrix} - H'_{sn}((x, y); (\xi, \eta)) = \begin{pmatrix} I - V & \nabla f(x) \\ V & -I \end{pmatrix}^T \begin{pmatrix} \xi \\ \eta \end{pmatrix} - \begin{pmatrix} -P'_{\kappa_n}(x - y; \xi - \eta) \\ \nabla f(x)^T \xi - \eta \end{pmatrix}$$

$$= \begin{pmatrix} P'_{\kappa_n}(x - y; \xi - \eta) - V^T(\xi - \eta) \\ (\nabla f(x + \xi) - \nabla f(x))^T \xi \end{pmatrix}$$

$$= \begin{pmatrix} O(\|\xi - \eta\|^2) \\ o(\|\xi\|) \end{pmatrix}$$

$$= o(\|\xi, \eta\|),$$

where the last equality follows from $\|\xi - \eta\|^2 \leq 2\|\xi\|^2 + 2\|\eta\|^2 = 2(\|\xi\|^2 + \|\eta\|^2)$ and $\|\xi\| \leq \|\xi, \eta\|$. Hence, $H_{sn}$ is semismooth at $(x, y)$. Furthermore, if $\nabla f$ is locally Lipschitzian, then the strong semismoothness of $H_{sn}$ at $(x, y)$ readily follows since $o(\|\xi\|)$ in (4.3) can be replaced by $O(\|\xi\|^2)$.

**4.3. Jacobian consistency of the smoothing function.** Jacobian consistency, which was first introduced by Chen, Qi and Sun [4], is a concept relating the generalized Jacobian of a nonsmooth function with Jacobian of a smoothing function. Like the strong semismoothness, the Jacobian consistency plays an important role in establishing rapid convergence of smoothing methods. In this subsection, we show that the function $H_{\mu, \varepsilon}$ defined by (3.7) enjoys the Jacobian consistency, and give some results that will be useful in constructing a rapidly convergent algorithm.

**Definition 4.7.** Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a continuous function. Let $F_{\mu, \varepsilon} : \mathbb{R}^n \to \mathbb{R}^n$ be a function such that $F_{\mu, \varepsilon}$ is continuously differentiable for any $\mu > 0$ and $\varepsilon \geq 0$, and that $\lim_{(\mu, \varepsilon) \to (0, 0)} F_{\mu, \varepsilon}(z) = F(z)$ for any $z \in \mathbb{R}^n$. Then we say $F_{\mu, \varepsilon}$ satisfies the Jacobian consistency if

$$\lim_{(\mu, \varepsilon) \to (0, 0)} \text{dist}(\nabla F_{\mu, \varepsilon}(z), \partial F(z)) = 0$$

holds for any $z \in \mathbb{R}^n$. Here $\text{dist}(X, S)$ denotes $\min\{\|X - Y\| \mid Y \in S\}$.
We thus have \( \partial F_1(z) \times \cdots \times \partial F_n(z) \supseteq \partial F(z) \) instead of \( \partial F(z) \). Moreover, their definition contains only a smoothing parameter \( \mu \).

In order to show the Jacobian consistency of \( H_{\mu, \varepsilon} \), we need the following proposition. Since it can be proved from (2.4) and (3.10) in a straightforward manner, we omit the proof.

**Proposition 4.8.** Clarke subdifferential of the projection function \( P_{\kappa, n} \) and the matrix \( J^P_{\mu}(z) := \lim_{t \to 0} \nabla P_{\mu}(z) \) are given as follows:

\[
\partial P_{\kappa, n}(z) = \begin{cases}
\{ I \} & (\lambda_1 > 0, \lambda_2 > 0) \\
\{(\lambda_2/(\lambda_1 + \lambda_2))I + Z\} & (\lambda_1 < 0, \lambda_2 > 0) \\
\{ O \} & (\lambda_1 < 0, \lambda_2 < 0) \\
\co\{ I, I + Z \} & (\lambda_1 = 0, \lambda_2 > 0) \\
\co\{ O, Z \} & (\lambda_1 < 0, \lambda_2 = 0) \\
\co\{(O) \cup \{ I \} \cup S\} & (\lambda_1 = 0, \lambda_2 = 0),
\end{cases}
\]

\[
J^P_{\mu}(z) = \begin{cases}
I & (\lambda_1 > 0, \lambda_2 > 0) \\
(\lambda_2/(\lambda_1 + \lambda_2))I + Z & (\lambda_1 < 0, \lambda_2 > 0) \\
O & (\lambda_1 < 0, \lambda_2 < 0) \\
I + (1 - \tilde{g}'(0))Z & (\lambda_1 = 0, \lambda_2 > 0) \\
\tilde{g}'(0)Z & (\lambda_1 < 0, \lambda_2 = 0) \\
\tilde{g}'(0)I & (\lambda_1 = 0, \lambda_2 = 0),
\end{cases}
\]

where \( (r_1, r_2) := (z_1, z_2)/\|z_2\| \),

\[
Z = \frac{1}{2} \begin{pmatrix} -r_1 & r_Z^T \\ r_Z & -r_1 r_Z r_Z^T \end{pmatrix},
\]

\[
S := \left\{ \frac{1}{2} (1 + \beta) I + \frac{1}{2} \begin{pmatrix} -\beta & w^T \\ w & -\beta w w^T \end{pmatrix} \mid -1 \leq \beta \leq 1, \|w\| = 1 \right\}.
\]

By using this proposition, we show the Jacobian consistency of \( H_{\mu, \varepsilon} \).

**Theorem 4.9.** \( H_{\mu, \varepsilon} \) satisfies the Jacobian consistency.

**Proof.** The formulas (4.4), (4.5) and \( 0 < \tilde{g}'(0) < 1 \) from (3.3) yield \( J^P_{\mu}(z) \in \partial P_{\kappa, n}(z) \) for any \( z \in \mathbb{R}^n \). Moreover, we have

\[
J^P_{\mu}(x, y) := \lim_{(\mu, \varepsilon) \to (0, 0)} \nabla H_{\mu, \varepsilon}(x, y) = \begin{pmatrix} I - J^P_{\mu}(x - y) & \nabla f(x) \\ J^P_{\mu}(x - y) & -I \end{pmatrix}
\]

from (3.9), and

\[
\partial H_{\kappa, n}(x, y) = \left\{ \begin{pmatrix} I - D & \nabla f(x) \\ D & -I \end{pmatrix} \mid D \in \partial P_{\kappa, n}(x - y) \right\}.
\]

We thus have \( J^P_{\mu}(x, y) \in \partial H_{\kappa, n}(x, y) \) for any \( (x, y) \in \mathbb{R}^n \times \mathbb{R}^n \), which implies that \( H_{\mu, \varepsilon} \) satisfies the Jacobian consistency. \( \blacksquare \)
The Jacobian consistency of $H_{\mu,\varepsilon}$ guarantees that, for any $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ and $\delta' > 0$, there exist $\mu > 0$ and $\varepsilon > 0$ such that $\text{dist}(\nabla H_{\mu,\varepsilon}(x, y), \partial H_{\text{sm}}(x, y)) < \delta'$. To find such $\mu$ and $\varepsilon$ for given $\delta'$ and $(x, y)$ is an essential issue for constructing a rapidly convergent algorithm. In the remainder of this subsection, we study how we can find such $\mu$ and $\varepsilon$.

In what follows, we use the following notation for convenience. Let $\hat{g} : \mathbb{R} \rightarrow \mathbb{R}$ be a continuously differentiable convex function satisfying (3.3). Then we denote for any $\mu > 0$ that
\[(4.7) \quad \gamma_{\mu}(\alpha) := \mu \hat{g}(\alpha/\mu).\]

Note that, as $\mu \downarrow 0$, there exist limits of $\gamma_{\mu}(\alpha)$ and $\gamma'_{\mu}(\alpha)$ for any fixed $\alpha$, which we denote
\[(4.8) \quad \gamma_{0}(\alpha) := \lim_{\mu \downarrow 0} \gamma_{\mu}(\alpha) = \max\{0, \alpha\}, \quad \gamma^{+}_{0}(\alpha) := \lim_{\mu \downarrow 0} \gamma'_{\mu}(\alpha) = \begin{cases} 0 & (\alpha < 0) \\ \hat{g}'(0) & (\alpha = 0) \\ 1 & (\alpha > 0). \end{cases}\]

Moreover, we suppose that the function $\hat{g}$ satisfies not only (3.3) but also
\[(4.10) \quad \hat{g}(\alpha) - \alpha = \hat{g}(-\alpha)\]
for any $\alpha \in \mathbb{R}$. For example, $\hat{g}_{1}(\alpha) = (\sqrt{\alpha^2 + 4} + \alpha)/2$ and $\hat{g}_{2}(\alpha) = \ln(e^\alpha + 1)$ satisfy (3.3) and (4.10). For such functions, we have the following lemma.

**Lemma 4.10.** Let $\hat{g}$ be a continuously differentiable convex function satisfying (3.3) and (4.10). Let $\gamma_{\mu}$, $\gamma_{0}$ and $\gamma^{+}_{0}$ be defined by (4.7)–(4.9). Then it holds that (a) $\gamma_{\mu}(\alpha) - \gamma_{0}(\alpha) = \gamma_{\mu}(-\alpha) - \gamma_{0}(-\alpha)$ for any $\alpha \in \mathbb{R}$, and (b) $|\gamma'_{\mu}(0) - \gamma^{+}_{0}(0)| = 0 < |\gamma'_{\mu}(\alpha_2) - \gamma^{+}_{0}(\alpha_2)| \leq |\gamma'_{\mu}(\alpha_1) - \gamma^{+}_{0}(\alpha_1)|$ for any $(\alpha_1, \alpha_2) \in \mathbb{R} \times \mathbb{R}$ such that $0 < |\alpha_1| \leq |\alpha_2|$.

**Proof.** Since (a) can be easily seen from (4.7), (4.8) and (4.10), we only show (b). Let $(\alpha_1, \alpha_2) \in \mathbb{R} \times \mathbb{R}$ be arbitrary scalars such that $0 < |\alpha_1| \leq |\alpha_2|$. From (4.10), we have $\hat{g}'(\alpha) - 1 = -\hat{g}'(-\alpha)$ for any $\alpha \in \mathbb{R}$. This, together with $\gamma'_{\mu}(\alpha) = \hat{g}'(\alpha/\mu)$ and (4.9), yields $\gamma'_{\mu}(\alpha) - \gamma^{+}_{0}(\alpha) = -\{\gamma'_{\mu}(-\alpha) - \gamma^{+}_{0}(-\alpha)\}$. We therefore have $|\gamma'_{\mu}(0) - \gamma^{+}_{0}(0)| = 0$ and
\[(4.11) \quad |\gamma'_{\mu}(\alpha_i) - \gamma^{+}_{0}(\alpha_i)| = |\gamma'_{\mu}(|\alpha_i|) - \gamma^{+}_{0}(|\alpha_i|)|\]
for each $i = 1, 2$. Now, we note that, for any $\beta > 0$, $|\gamma'_{\mu}(\beta) - \gamma^{+}_{0}(\beta)| = 1 - \gamma'_{\mu}(\beta) > 0$, where the inequality follows from $\gamma'_{\mu}(\beta) = \hat{g}'(\beta/\mu)$ and (3.3). Moreover, $1 - \gamma'_{\mu}(\beta)$ is monotonically nonincreasing since $\gamma_{\mu}$ is convex. Hence, we have $0 < |\gamma'_{\mu}(\alpha_2)| - |\gamma^{+}_{0}(\alpha_2)| \leq |\gamma'_{\mu}(\alpha_1)| - |\gamma^{+}_{0}(\alpha_1)|$. Combining this inequality with (4.11), we have (b). \[\square\]

We further define function $\tilde{\lambda} : \mathbb{R}^n \rightarrow [0, +\infty)$ by
\[(4.12) \quad \tilde{\lambda}(z) := \begin{cases} \min_{i \in I(z)} |\lambda_i(z)| & (I(z) \neq \emptyset) \\ 0 & (I(z) = \emptyset), \end{cases}\]
where $\lambda_i(z)$ $(i = 1, 2)$ are the spectral values of $z$, and $I(z) \subseteq \{1, 2\}$ is the index set defined by $I(z) := \{i \mid \lambda_i(z) \neq 0\}$. Then we can easily see from Lemma 4.10 (b) that
\[(4.13) \quad |\gamma'_{\mu}(\lambda_i(z)) - \gamma^{+}_{0}(\lambda_i(z))| \leq |\gamma'_{\mu}(\tilde{\lambda}(z)) - \gamma^{+}_{0}(\tilde{\lambda}(z))| \quad (i = 1, 2).\]
From this fact, we can estimate the upper bound of dist(∇H_{µ,ε}(x, y), ∂H_{SN}(x, y)).

Proposition 4.11. Let \( \tilde{g} \) be a function satisfying (3.3) and (4.10), and \( \tilde{\lambda} \) be the function defined by (4.12). Then, there exists \( M > 0 \) such that

\[
\text{dist}(\nabla H_{µ,ε}(x, y), ∂H_{SN}(x, y)) \leq M(|γ_µ'(\tilde{\lambda}(x - y)) - γ_0^+(\tilde{\lambda}(x - y))| + ε)
\]

for any \( µ > 0 \), \( ε ≥ 0 \) and \( (x, y) \) ∈ \( \mathbb{R}^n × \mathbb{R}^n \).

Proof. By (3.9), (4.6) and the Jacobian consistency of \( H_{µ,ε} \), there exists \( M' > 0 \) such that dist(∇H_{µ,ε}(x, y), ∂H_{SN}(x, y)) ≤ M'(||∇P_µ(x - y) - J_0^P(x - y)|| + ε) for any \( µ > 0 \), \( ε ≥ 0 \) and \( (x, y) \) ∈ \( \mathbb{R}^n × \mathbb{R}^n \). So it suffices to show the existence of \( N > 0 \) such that

\[
||∇P_µ(z) - J_0^P(z)|| ≤ N|γ_µ'(\tilde{\lambda}(z)) - γ_0^+(\tilde{\lambda}(z))|
\]

for any \( µ > 0 \) and \( z \) ∈ \( \mathbb{R}^n \). Let \( λ_1 \) and \( λ_2 \) be the spectral values of \( z \), and \( a_µ, b_µ \) and \( c_µ \) be defined by (3.11). Moreover, let

\[
a_0 := \lim_{µ \to 0} a_µ = \frac{γ_0(λ_2) - γ_0(λ_1)}{λ_2 - λ_1},
\]

(4.14)

\[
b_0 := \lim_{µ \to 0} b_µ = \frac{1}{2}(γ_0^+(λ_2) + γ_0^+(λ_1)),
\]

(4.15)

\[
c_0 := \lim_{µ \to 0} c_µ = \frac{1}{2}(γ_0^+(λ_2) - γ_0^+(λ_1)).
\]

(4.16)

When \( z_2 = 0 \), that is, \( z_1 = λ_1 = λ_2 \), we have ||∇P_µ(z) - J_0^P(z)|| = ||g'(z_1/µ)I - γ_0^+(z_1)I|| = ||γ_µ'(z_1) - γ_0^+(z_1)|| = ||γ_µ'(\tilde{λ}(z)) - γ_0^+(\tilde{λ}(z))||, where the first equality follows from (3.10), (4.5) and (4.9), and the last equality follows from \( \tilde{λ}(z) = |z_1| \) and (4.11).

When \( z_2 ≠ 0 \), that is, \( λ_1 < λ_2 \), by (3.10) and (4.5), we have for some \( N' > 0 \)

\[
||∇P_µ(z) - J_0^P(z)|| ≤ N'(a_µ - a_0)I + \left((b_µ - b_0) - (a_µ - a_0)\right)_{r_2} + \left((c_µ - c_0)r_2^T\right)_{r_2} + \left|\left|a_µ - a_0\right| + |b_µ - b_0| + |c_µ - c_0|\right|
\]

(4.17)

where \( r_2 := z_2/||z_2|| \). First we consider the case where \( 0 ≤ λ_1 < λ_2 \). Note that there exists \( \lambda \) ∈ \( [λ_1, λ_2] \) such that \( γ_µ'(\lambda) = (γ_µ(λ_2) - γ_µ(λ_1))/(λ_2 - λ_1) = a_µ \). Moreover, we have from (4.9), (4.14) and \( \lambda ∈ [λ_1, λ_2] \) that \( a_0 = 1 = γ_0^+(\lambda) \). Hence, we obtain \( |a_µ - a_0| = |γ_µ'(\lambda) - γ_0^+(\lambda)| \). Furthermore, by (3.11) and (4.15), we have \( |b_µ - b_0| ≤ (1/2)|γ_µ'(λ_2) - γ_0^+(λ_2)| + (1/2)|γ_µ'(λ_1) - γ_0^+(λ_1)| + (1/2)|γ_µ'(λ_1) - γ_0^+(λ_2)| + (1/2)|γ_µ'(λ_1) - γ_0^+(λ_2)| \). Combining these inequalities, we have

\[
|a_µ - a_0| + |b_µ - b_0| + |c_µ - c_0| ≤ 3|γ_µ'(\tilde{λ}(z)) - γ_0^+(\tilde{λ}(z))|,
\]

where the last inequality follows from \( 0 ≤ λ_1 < λ_2 \), \( \lambda ∈ [λ_1, λ_2] \) and Lemma 4.10 (b). Hence, by (4.17), we have for some \( N > 0 \) that \( ||∇P_µ(z) - J_0^P(z)|| ≤ N|γ_µ'(\tilde{λ}(z)) - γ_0^+(\tilde{λ}(z))| \). When \( λ_1 < λ_2 ≤ 0 \), we can obtain the desired result in a similar way.
Finally, we consider the case where \( \lambda_1 < 0 < \lambda_2 \). First we assume \( 0 < \lambda_1 \leq \lambda_2 \). When \( |\lambda_1| < \lambda_2 \), there exists \( \overline{\lambda} \in [\lambda_1, \lambda_2] = [-\lambda_1, \lambda_2] \) such that

\[
|a_\mu - a_0| = \left| \frac{\gamma_\mu(\lambda_2) - \gamma_\mu(\lambda_1)}{\lambda_2 - \lambda_1} \cdot \frac{\gamma_0(\lambda_2) - \gamma_0(\lambda_1)}{\lambda_2 - \lambda_1} \right|
\]

\[(4.18)\]

where the second equality follows from Lemma 4.10 (a), the third equality follows from (4.9), and the inequality follows from \((4.8)\), the fourth equality follows from (4.9), and the inequality follows from \((4.8)\). When \( \lambda_1 = \lambda_2 \), we have \( |a_\mu - a_0| = 0 \) from (4.18) and Lemma 4.10 (a). In a way similar to the previous case, we have \( |b_\mu - b_0| \leq (1/2)|\gamma_\mu(\lambda_2) - \gamma_\mu^+(\lambda_2)| + (1/2)|\gamma_\mu(\lambda_1) - \gamma_\mu^+(\lambda_1)| \) and \( |c_\mu - c_0| \leq (1/2)|\gamma_\mu(\lambda_2) - \gamma_\mu^+(\lambda_2)| + (1/2)|\gamma_\mu(\lambda_1) - \gamma_\mu^+(\lambda_1)| \). Hence, by \( 0 < \lambda_1 \leq \lambda_2 \), \( \overline{\lambda} \in [\lambda_1, \lambda_2] \), Lemma 4.10 (b) and (4.13), we have \( \|\nabla P_\mu(z) - J^\mu_\mu(z)\| \leq N' \gamma_\mu^+(\overline{\lambda}(z)) - \gamma_\mu^+(\overline{\lambda}(z)) \) for some \( N > 0 \). In the case where \( \lambda_1 < 0 < \lambda_2 \) and \( 0 < \lambda_2 \leq |\lambda_1| \), we can also show the desired result in a similar manner.

Proposition 4.11 indicates that, for given \( \alpha \in \mathcal{R} \), if we can reduce the value of \( |\gamma_\mu(\alpha) - \gamma_\mu^+(\alpha)| \) arbitrarily close to zero by choosing a sufficiently small \( \mu > 0 \), then we can make \( \nabla H_{\mu,\alpha}(x, y) \) arbitrarily close to \( \partial H_{\mu,\alpha}(x, y) \). Since \( \gamma_\mu^+(\alpha) = \lim_{\mu \to 0} \gamma_\mu(\alpha) \), for any \( \delta > 0 \) and \( \alpha \in \mathcal{R} \), there exists \( \overline{\alpha}(\alpha, \delta) > 0 \) such that \( |\gamma_\mu(\alpha) - \gamma_\mu^+(\alpha)| < \delta \) for any \( \mu \in (0, \overline{\alpha}(\alpha, \delta)) \). The following proposition gives an explicit expression of such \( \overline{\alpha}(\alpha, \delta) \) when \( \hat{g}(\alpha) \) is given by \( \hat{g}(\alpha) = (\alpha + \sqrt{\alpha^2 + 4})/2 \).

**Proposition 4.12.** Let the function \( \hat{g} \) be defined by \( \hat{g}(\alpha) = (\alpha + \sqrt{\alpha^2 + 4})/2 \), and \( \gamma_\mu \) and \( \gamma_\mu^+ \) be defined by (4.7) and (4.9), respectively. Moreover, let \( \overline{\alpha}(\alpha, \delta) \) be defined by

\[
\overline{\alpha}(\alpha, \delta) := \begin{cases} +\infty & (\delta \geq 1/2 \text{ or } \alpha = 0) \\ \frac{1}{2}|\alpha|\sqrt{\delta} & (\delta < 1/2 \text{ and } \alpha \neq 0) \end{cases}
\]

Then, for any \( \alpha \in \mathcal{R} \), \( \delta > 0 \) and \( \mu \in (0, \overline{\alpha}(\alpha, \delta)) \), we have

\[(4.19)\]

\[
|\gamma_\mu(\alpha) - \gamma_\mu^+(\alpha)| < \delta.
\]

**Proof.** Let \( \mu \) be an arbitrary scalar in the interval \((0, \overline{\alpha}(\alpha, \delta))\). By easy calculation, we have \( \gamma_\mu(\alpha) = (\alpha + \sqrt{\alpha^2 + 4\mu^2})/2 \), \( \gamma_\mu^+(\alpha) = (1 + \alpha/\sqrt{\alpha^2 + 4\mu^2})/2 \) and

\[
\gamma_\mu^+(\alpha) = \begin{cases} 1 & (\alpha > 0) \\ 1/2 & (\alpha = 0) \\ 0 & (\alpha < 0) \end{cases}
\]
Moreover, these equalities yield
\[
|\gamma'_\mu(\alpha) - \gamma'_0(\alpha)| = \begin{cases} 
0 & (\alpha = 0) \\
\frac{1}{2} \left( 1 - \frac{|\alpha|}{\sqrt{\alpha^2 + 4\mu^2}} \right) & (\alpha \neq 0).
\end{cases}
\]
When $\alpha = 0$, (4.19) trivially holds since $|\gamma'_\mu(\alpha) - \gamma'_0(\alpha)| = 0 < \delta$. When $\delta \geq 1/2$, (4.19) always holds since $|\gamma'_\mu(\alpha) - \gamma'_0(\alpha)| < 1/2$ for any $\delta > 0$ and $\mu > 0$. When $\alpha \neq 0$ and $0 < \delta < 1/2$, we have
\[
|\gamma'_\mu(\alpha) - \gamma'_0(\alpha)| - \delta = \frac{1}{2} \left( 1 - \frac{|\alpha|}{\sqrt{\alpha^2 + 4\mu^2}} \right) - \delta
\]
\[
< \frac{1}{2} \left( 1 - \frac{|\alpha|}{\sqrt{\alpha^2 + \alpha^2 \delta}} \right) - \delta
\]
\[
= \frac{1}{2} \left( 1 - \frac{1}{\sqrt{1 + \delta}} \right) - \delta
\]
\[
< 0,
\]
where the first inequality follows from $\mu < \pi(\alpha, \delta) = (1/2)|\alpha|/\sqrt{\delta}$, and the last inequality follows since the function $f(\delta) := (1/2)(1 - (1 + \delta)^{-1/2}) - \delta$ satisfies $f(0) = 0$ and $f'(\delta) = (1/4)(1 + \delta)^{-3/2} - 1 < 0$ for any $\delta > 0$. This completes the proof. \( \square \)

4.4. Quadratically convergent algorithm with Newton’s method. In this subsection, based on Algorithm 1, we propose a more specific algorithm in which Newton’s method is applied for solving subproblem (4.1). Moreover, we show that the algorithm is quadratically convergent under appropriate assumptions including the strong semismoothness and the Jacobian consistency. We first state the algorithm. For convenience, we denote
\[
w := \begin{pmatrix} x \\ y \end{pmatrix}, \quad w^{(k)} := \begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix}.
\]

**Algorithm 2.** Choose $\eta$, $\rho \in (0, 1)$, $\eta \in (0, \eta]$, $\sigma \in (0, 1/2)$, $\kappa > 0$ and $\hat{k} > 0$.

**Step 0** Choose $w^{(0)} \in \mathbb{R}^{2n}$ and $\beta_0 \in (0, \infty)$. Let $\mu_0 := \|H_{ss}(w^{(0)})\|$ and $\varepsilon_0 := \|H_{ss}(w^{(0)})\|$. Set $k := 0$.

**Step 1** Terminate if $\|H_{ss}(w^{(k)})\| = 0$.

**Step 2**

**Step 2.0** Set $\psi^{(0)} := w^{(k)}$ and $j := 0$.

**Step 2.1** Find a vector $\tilde{d}^{(j)}$ such that
\[
H_{\mu_k, \varepsilon_k}(\psi^{(j)}) + \nabla H_{\mu_k, \varepsilon_k}(\psi^{(j)})^T \tilde{d}^{(j)} = 0.
\]

**Step 2.2** If $\|H_{\mu_k, \varepsilon_k}(\psi^{(j)} + \tilde{d}^{(j)})\| \leq \beta_k$, then let $w^{(k+1)} := \psi^{(j)} + \tilde{d}^{(j)}$ and go to Step 3. Otherwise, go to Step 2.3.

**Step 2.3** Find the smallest nonnegative integer $m$ such that
\[
\Psi_{\mu_k, \varepsilon_k}(\psi^{(j)} + \rho^m \tilde{d}^{(j)}) \leq (1 - 2\sigma\rho^m)\Psi_{\mu_k, \varepsilon_k}(\psi^{(j)}).
\]
Let $m_j := m$, $\tau_j := \rho^m$, and $w^{(j+1)} := \psi^{(j)} + \tau_j \tilde{d}^{(j)}$. 

Let $m_j := m$, $\tau_j := \rho^m$, and $w^{(j+1)} := \psi^{(j)} + \tau_j \tilde{d}^{(j)}$. 

\[
\Psi_{\mu_k, \varepsilon_k}(\psi^{(j)} + \rho^m \tilde{d}^{(j)}) \leq (1 - 2\sigma\rho^m)\Psi_{\mu_k, \varepsilon_k}(\psi^{(j)}).
\]
Step 2.4 If
\[ \| H_{\mu_k, \varepsilon_k}(w(j+1)) \| \leq \beta_k, \]
then let \( w^{(k+1)} := v^{(j+1)} \) and go to Step 3. Otherwise, set \( j := j + 1 \) and go back to Step 2.1.

Step 3 Update the parameters as follows:
\[
\begin{align*}
\mu_{k+1} &:= \min \left\{ \kappa \| H_{\text{sn}}(w^{(k+1)}) \|^2, \mu_0 T_k^{2 \sigma_k} \right\}, \\
\varepsilon_{k+1} &:= \min \left\{ \kappa \| H_{\text{sn}}(w^{(k+1)}) \|^2, \varepsilon_0 T_k^{2 \sigma_k} \right\}, \\
\beta_{k+1} &:= \beta_k T_k^{2 \sigma_k}.
\end{align*}
\]

Set \( k := k + 1 \). Go back to Step 1.

In Step 3, \( \lambda \) is the function given by (4.12), and \( T(\alpha, \delta) \) is determined so that \( |\gamma_\mu^\gamma(\alpha) - \gamma_0^\gamma(\alpha)| < \delta \) for any \( \mu \in (0, T(\alpha, \delta)) \). An explicit formula of \( T(\alpha, \delta) \) is given as in Proposition 4.12 when \( g(\alpha) = (a + \sqrt{a^2 + 4})/2 \).

In the inner iterations Steps 2.0–2.4, a damped Newton method seeks a point \( w^{(k+1)} \) such that \( \| H_{\mu_k, \varepsilon_k}(w^{(k+1)}) \| \leq \beta_k \). Note that, by letting \( \alpha_k := \beta_k^2/2 \), the termination criterion (4.20) for the inner iterations becomes equivalent to (4.1) in Algorithm 1. Step 3 specifies the updating rule of the parameters, where \( \{\beta_k\}, \{\mu_k\} \) and \( \{\varepsilon_k\} \) converge to 0 since \( 0 < \bar{\eta} < \eta < 1 \). Algorithm 2 is well-defined in the sense that Steps 2.0–2.4 find \( v^{(j+1)} \) satisfying (4.20) in a finite number of iterations for each \( k \).

Now, we show that Algorithm 2 is quadratically convergent under appropriate assumptions. Note that, from (4.2) and \( \| x \| \leq \| (x, y) \| \), there exists \( \nu > 0 \) such that
\[ \| H_{\mu, \varepsilon}(w) - H_{\text{sn}}(w) \| \leq \nu \mu + \varepsilon \| w \| \]
for any \( w \in \mathbb{R}^{2n} \). By using this inequality and the two properties shown in Subsections 4.2 and 4.3, we establish the main theorem of this subsection.

Theorem 4.13. Let \( f : \mathbb{R}^n \to \mathbb{R}^n \) be a monotone function such that \( \nabla f \) is locally Lipschitzian. Let \( \{w(k)\} \) be a sequence generated by Algorithm 2. Moreover, suppose that the following assumptions hold true:

(i) The solution set of SOCCP (3.1) is nonempty and bounded.
(ii) Every accumulation point of \( \nabla H_{\mu_k, \varepsilon_k}(w(k)) \) is nonsingular.

Then we have the following two statements:

(a) For all \( k \) sufficiently large, the inequality in Step 2.2 of Algorithm 2 holds for \( j = 0 \);
(b) The sequence \( \{w(k)\} \) converges to a solution \( w^* \) of SOCCP (3.1) quadratically.

Proof. By assumption (i) and Theorem 4.3, \( \{w(k)\} \) is bounded and an arbitrary accumulation point \( w^* \) is a solution of SOCCP (3.1), that is, \( H_{\text{sn}}(w^*) = 0 \). Let \( B \) be a positive number such that \( \| w(k) \| \leq B \) for all \( k \). Let \( L > 0 \) be a Lipschitzian constant of \( H_{\text{sn}} \) on a bounded set \( \Omega \supset \{w(k)\} \). Let \( d(k) := d(0) - \nabla H_{\mu_k, \varepsilon_k}(w(k)) - T H_{\mu_k, \varepsilon_k}(w(k)) \), where \( J^T \) denotes \( J^{-T} := (J^T)^{-1} = (J^{-1})^T \). From the nonsingularity of \( \nabla H_{\mu_k, \varepsilon_k}(w(k)) \) [11, Proposition 6.1] and assumption (ii), there exists \( C > 0 \) such that
\[ \| \nabla H_{\mu_k, \varepsilon_k}(w(0))^{-T} \| = \| \nabla H_{\mu_k, \varepsilon_k}(w(k))^{-1} \| \leq C. \]
for all \( k \).

First we show \( \|w^{(k)} + d^{(k)} - w^*\| = O(\|w^{(k)} - w^*\|^2) \). Let \( V_k \) be an element of \( \partial H_{\text{sn}}(w^{(k)}) \) such that \( \|\nabla H_{\mu_k, \varepsilon_k}(w^{(k)}) - V_k\| = \text{dist}(\nabla H_{\mu_k, \varepsilon_k}(w^{(k)}), \partial H_{\text{sn}}(w^{(k)})) \). It then follows from (4.22) and \( H_{\text{sn}}(w^*) = 0 \) that

\[
\|w^{(k)} + d^{(k)} - w^*\|
\leq \left\| \nabla H_{\mu_k, \varepsilon_k}(w^{(k)}) - V_k \right\| - \|H_{\mu_k, \varepsilon_k}(w^{(k)}) - H_{\mu_k, \varepsilon_k}(w^{(k)})\|
\leq C \left\{ \|\nabla H_{\mu_k, \varepsilon_k}(w^{(k)}) - V_k\|^2 + \|V_k^2(w^{(k)} - w^*) - H_{\text{sn}}(w^*; w^{(k)} - w^*)\| + \left\| H_{\text{sn}}(w^*) - H_{\text{sn}}(w^{(k)}) \right\| \right\}.
\]

Moreover, each term of (4.23) is \( O(\|w^{(k)} - w^*\|^2) \) by the following reasons. Since \( H_{\mu, \varepsilon} \) satisfies the Jacobian consistency, we have for some \( M > 0 \) that

\[
\left\| \nabla H_{\mu_k, \varepsilon_k}(w^{(k)}) - V_k \right\| \leq M \left( \gamma_4 \left( \hat{\lambda}(x^{(k)} - y^{(k)}) \right) + \gamma_0 \right) + \varepsilon_k \leq M (\hat{\kappa} \|H_{\text{sn}}(w^{(k)})\| + \|H_{\text{sn}}(w^{(k)})\|^2)
\leq M (\hat{\kappa} + \kappa \|H_{\text{sn}}(w^{(k)})\|) \|H_{\text{sn}}(w^{(k)}) - H_{\text{sn}}(w^*)\|
\leq M (\hat{\kappa} + \kappa \|H_{\text{sn}}(w^{(k)})\|) L \|w^{(k)} - w^*\|
\]

where the first inequality follows from Proposition 4.11, the second inequality follows from Step 3 of the algorithm, the third inequality follows from \( H_{\text{sn}}(w^*) = 0 \), and the last inequality is due to the local Lipschitz continuity of \( H_{\text{sn}} \). By (4.24) and the boundedness of \( \{|H_{\text{sn}}(w^{(k)})|\} \), we have \( \|\nabla H_{\mu_k, \varepsilon_k}(w^{(k)}) - V_k\|^2 = O(\|w^{(k)} - w^*\|^2) \). The second and third terms of (4.23) are also \( O(\|w^{(k)} - w^*\|^2) \) since \( H_{\text{sn}} \) is strongly semismooth and directionally differentiable. Finally, we consider the fourth term of (4.23). From (4.21) and Step 3 of Algorithm 2, we have \( \|H_{\text{sn}}(w^{(k)}) - H_{\mu_k, \varepsilon_k}(w^{(k)})\| \leq \nu \mu_k + B \varepsilon_k = O(\|H_{\text{sn}}(w^{(k)})\|^2) \). Moreover, \( O(\|H_{\text{sn}}(w^{(k)})\|^2) \) and local Lipschitz continuity of \( H_{\text{sn}} \). Therefore, the fourth term of (4.23) is \( O(\|w^{(k)} - w^*\|^2) \). Consequently, we have

\[
\|w^{(k)} + d^{(k)} - w^*\| = O(\|w^{(k)} - w^*\|^2).
\]

Next we show (a), that is, \( \|H_{\mu_k, \varepsilon_k}(w^{(k)} + d^{(k)})\| \leq \beta_k \) for all \( k \) sufficiently large. Note that \( \{d^{(k)}\} \) is bounded from the inequality \( \|d^{(k)}\| \leq \|w^{(k)} + d^{(k)} - w^*\| + \|w^{(k)} - w^*\| \) and (4.25). We therefore have for some \( \Gamma > 0 \) that

\[
\|H_{\mu_k, \varepsilon_k}(w^{(k)} + d^{(k)})\|
\leq \|H_{\mu_k, \varepsilon_k}(w^{(k)} + d^{(k)}) - H_{\text{sn}}(w^{(k)} + d^{(k)})\| + \|H_{\text{sn}}(w^{(k)} + d^{(k)})\|
\leq \Gamma^2 \|H_{\text{sn}}(w^{(k)})\|^2 + \|H_{\text{sn}}(w^{(k)} + d^{(k)})\|,
\]

where the second inequality follows from (4.21), the boundedness of \( \{w^{(k)} + d^{(k)}\} \) and Step 3 of Algorithm 2. Hence, it suffices to show \( \|H_{\text{sn}}(w^{(k)} + d^{(k)})\| = O(\|H_{\text{sn}}(w^{(k)}\|^2) \) and \( \|H_{\text{sn}}(w^{(k)})\| = O(\gamma^k) \). Notice that assumption (ii) implies that \( H_{\text{sn}} \) satisfies the local error bound property at \( w^* \), that is, \( \|w^{(k)} - w^*\| = O(\|H_{\text{sn}}(w^{(k)})\|) \). Then, we have \( \|H_{\text{sn}}(w^{(k)} + d^{(k)})\| \leq \Gamma \|w^{(k)} + d^{(k)} - w^*\| = O(\|w^{(k)} - w^*\|^2) = O(\|H_{\text{sn}}(w^{(k)})\|^2) \),
where \( \overline{L} \) is a Lipschitzian constant for \( H_{sn} \) on a bounded set \( \overline{\Omega} \supset \{ w^{(k)} + d^{(k)} \} \). On the other hand, \( \| H_{sn}(w^{(k)}) \| = O(\eta^k) \) since:

\[
\| H_{sn}(w^{(k)}) \| \leq \| H_{\mu_k, \varepsilon_k}(w^{(k)}) \| + \| H_{\mu_k, \varepsilon_k}(w^{(k)}) - H_{sn}(w^{(k)}) \|
\]

\[
\leq \beta_{k-1} + \nu \mu_{k-1} + B \varepsilon_{k-1}
\]

\[
\leq \beta_0 \eta^{k-1} + \nu \mu_0 \eta^{k-1} + B \varepsilon_0 \eta^{k-1}
\]

\[
= \eta^{-1}(\beta_0 + \nu \mu_0 + B \varepsilon_0) \eta^{k},
\]

where the second inequality follows from Step 2.4 of Algorithm 2 and (4.21), and the third inequality follows from Step 3 in Algorithm 2 and \( 0 < \eta \leq \eta < 1 \). Combining these results with (4.26), we have \( \| H_{\mu_k, \varepsilon_k}(w^{(k)} + d^{(k)}) \| \leq \| H_{sn}(w^{(k)}) \| O(\eta^k) \), which implies (a). Finally, (b) directly follows from (4.25) and (a).

We note that assumption (ii) holds if \( f \) is strongly monotone around a solution. This can be observed as follows. From [11, Proposition 6.1], we have \( O < \nabla P_\mu(z) < I \) for any \( \mu > 0 \) and \( z \in \mathbb{R}^n \), which implies \( O \leq J_\mu^o(z) \leq I \). Moreover, the matrix \( J_\mu^o(x, y) \) defined by (4.6) is nonsingular since the matrix \( I - J_\mu^o(x-y) + J_\mu^o(x-y) \nabla f(x) \) is nonsingular by the positive definiteness of \( \nabla f(x) \) and [26, Proposition 2.1(b)].

There have been studied many Newton-type methods with smoothing and regularization techniques for solving NCPs and box constrained variational inequalities (BVI). Our algorithm is regarded as an extension of those methods to SOCCP. However, it is different from the existing Newton-type methods for NCP and BVI in some respects. The Jacobian smoothing method [4, 12] is one of the popular Newton-type methods. In this method, an approximate Newton direction \( d = -\nabla H_\mu(z)^{-T} H(z) \) of \( H \) is used instead of Newton direction \( -\nabla H_\mu(z)^{-T} H_\mu(z) \) of \( H_\mu \) itself, which our algorithm adopts. Another approach [17, 22] is to consider the nonlinear equation \( G(x, \mu) = 0 \) where \( G(x, \mu) := (H_\mu(z)) \), and solve it by Newton’s method that treats parameter \( \mu \) as a variable. Moreover, in [17], a regularization parameter is also included. On the other hand, our method distinguishes between parameters and variables strictly.

5. Numerical experiments. In order to evaluate the efficiency of Algorithm 2, we have conducted some numerical experiments. In our experiments, we chose \( g(\alpha) = (\sqrt{\alpha^2 + 1} + \alpha)/2 \) as the function satisfying (3.3) and (4.10), and employed \( \| H_{sn}(w^{(k)}) \| < 10^{-8} \) as the termination criterion. Moreover, we adopted:

\[
\overline{\eta}(\alpha, \delta) := \begin{cases} 
10^{10} & (\delta \geq 1/2 \text{ or } \alpha = 0) \\
1/2 |\alpha| \sqrt{\delta} & (\delta < 1/2 \text{ and } \alpha \neq 0).
\end{cases}
\]

in Step 3 of the algorithm. Note that, when \( \overline{\lambda}(x^{(k)} - y^{(k)}) \) defined by (4.12) is very small but positive while \( \| H_{sn}(x^{(k)}, y^{(k)}) \| \) is not small enough, \( \mu_k \) becomes almost as small as \( \overline{\lambda}(x^{(k)} - y^{(k)}) \), although \( (x^{(k)}, y^{(k)}) \) is not sufficiently close to a solution. Since this may diminish the effects of smoothing, we regard \( \lambda_i(x^{(k)} - y^{(k)}) \) as 0 when \( \lambda_i(x^{(k)} - y^{(k)}) / H_{sn}(x^{(k)}, y^{(k)}) \| \leq \overline{\delta} \) holds for a sufficiently small \( \overline{\delta} > 0 \). In our implementation, we therefore modified the definition (4.12) of \( \overline{\lambda}(z) \) such that the index set \( \mathcal{I}(z) := \{ i | \lambda_i(z) \geq 10^{-4} \| H_{sn}(x, y) \| \} \) is used instead of \( \mathcal{I}(z) = \{ i | \lambda_i(z) \neq 0 \} \). The program was coded in MATLAB 6.5 and run on a machine with AMD Athlon(tm) XP 2000+ CPU and 1GB RAM.
5.1. Linear case. We solved the following linear SOCCP: Find \((x, y)\in \mathbb{R}^n \times \mathbb{R}^n\) such that
\[
(5.1) \quad x \in \mathcal{K}^n, \ y \in \mathcal{K}^n, \ x^T y = 0, \ y = Mx + q,
\]
where \(M \in \mathbb{R}^{n \times n}\) is a rank-deficient positive semidefinite matrix. We chose \(\eta = 0.01\), \(\bar{\eta} = 0.001\), \(\rho = 0.5\), \(\sigma = 0.4\), \(\kappa = 0.01\) and \(\bar{\kappa} = 1\) in Algorithm 2.

In order to obtain a positive semidefinite matrix \(M\) with rank \(r < n\), we let \(M := \frac{nBB^T}{\|BB^T\|}\), where \(B \in \mathbb{R}^{n \times r}\) is a matrix of which components are randomly chosen from the interval \([-1, 1]\). Furthermore, we let \(q := 10^n n^{1/2} p - M e\), where \(e = (1, 0, \ldots, 0)^T \in \mathbb{R}^n\), \(p\) is a vector such that \(p \in \text{int} \mathcal{K}^n\) and \(\|p\| = 1\), and \(\alpha\) is randomly chosen from the interval \([-1, 1]\). Then, SOCP (5.1) has a solution, since \(M\) is positive semidefinite and there exist \(x \in \text{int} \mathcal{K}^n\) and \(y \in \mathcal{K}^n\) such that \(y = Mx + q\). In the experiments, we determined \(p\) as follows: Let \(\theta\) be randomly chosen from \((0, \pi/2]\), let \(w\) be an \((n-1)\)-dimensional vector whose components are randomly chosen from \([-1, 1]\), and put \(p := 2^{-1/2} \cos \theta (w/\|w\|) + 2^{-1/2} \sin \theta (1 - w/\|w\|)\). It is not difficult to see that a vector \(p\) thus determined satisfies \(\|p\| = 1\) and \(p \in \text{int} \mathcal{K}^n\).

In our first experiments, we generated 100 problem instances for each of \(n = 100, 200, \ldots, 1000\), and solved each problem instance by using 100 randomly chosen initial points. We let the rank \(r\) of \(M\) be an integer randomly chosen from \([0.9n, n-1]\), and selected an initial point \((x^{(0)}, y^{(0)})\) as \((x^{(0)}, y^{(0)}) := 10^3 (a, b)/\|(a, b)\|\), where \(\beta\) is randomly chosen from \([-3, 3]\) and each component of \((a, b) \in \mathbb{R}^n \times \mathbb{R}^n\) is randomly chosen from \([-1, 1]\). Table 5.1 shows the results of our experiments, in which \(n\) denotes the size of problems, \#Ite denotes the number of outer iterations, \#Newton denotes the total number of inner Newton iterations, and cpu(s) denotes the CPU time in second. The values of \#Ite, \#Newton and cpu(s) are the average of 100 runs for each \(n\). Table 5.1 reveals that the problem size only slightly affects the number of iterations.

\[
\begin{array}{cccc}
 n & \#Ite & \#Newton & \text{cpu(s)} \\
 100 & 5.28 & 7.12 & 0.151 \\
 200 & 5.58 & 7.93 & 1.026 \\
 300 & 5.86 & 8.48 & 2.848 \\
 400 & 5.65 & 8.80 & 5.960 \\
 500 & 5.77 & 8.89 & 10.603 \\
 600 & 5.80 & 9.12 & 17.346 \\
 700 & 5.96 & 9.37 & 26.252 \\
 800 & 5.99 & 9.36 & 37.185 \\
 900 & 5.93 & 9.71 & 52.223 \\
 1000 & 5.98 & 9.52 & 67.316 \\
\end{array}
\]

In our second experiments, we fix the size of problems at 100 and varied the rank of \(M \in \mathbb{R}^{100 \times 100}\) as \(r = 10, 20, \ldots, 90\) and 99. We solved 1000 different problem instances for each \(r\), where initial points are selected in a way similar to the previous experiments. Table 5.2 shows the results of the experiments. The numbers of iterations, \#Ite and \#Newton, are the average of 1000 runs for each \(r\). Table 5.2 indicates that neither \#Ite nor \#Newton is affected by the rank of matrix \(M\).
Table 5.2

Results for linear SOCCPs with various degrees of rank deficiency

<table>
<thead>
<tr>
<th>r</th>
<th>$\frac{|x|}{|x|}$</th>
<th>Newton</th>
</tr>
</thead>
<tbody>
<tr>
<td>99</td>
<td>5.396</td>
<td>7.346</td>
</tr>
<tr>
<td>90</td>
<td>5.361</td>
<td>7.260</td>
</tr>
<tr>
<td>80</td>
<td>5.284</td>
<td>7.207</td>
</tr>
<tr>
<td>70</td>
<td>5.245</td>
<td>7.115</td>
</tr>
<tr>
<td>60</td>
<td>5.242</td>
<td>7.189</td>
</tr>
<tr>
<td>50</td>
<td>5.172</td>
<td>7.004</td>
</tr>
<tr>
<td>40</td>
<td>5.233</td>
<td>7.128</td>
</tr>
<tr>
<td>30</td>
<td>5.116</td>
<td>6.988</td>
</tr>
<tr>
<td>20</td>
<td>5.072</td>
<td>6.809</td>
</tr>
<tr>
<td>10</td>
<td>4.894</td>
<td>6.371</td>
</tr>
</tbody>
</table>

5.2. Comparison with interior point method. We solved the following SOCP by our method and interior point method.

\[
\begin{align*}
\text{Minimize} & \quad c^T z \\
\text{subject to} & \quad z \in \mathcal{K}_1, \quad Az + b \in \mathcal{K}_2,
\end{align*}
\]

where $A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^n$, $\mathcal{K}_1 := \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_N} \subset \mathbb{R}^n$ and $\mathcal{K}_2 := \mathcal{K}^{m_1} \times \cdots \times \mathcal{K}^{m_M} \subset \mathbb{R}^n$. Since the KKT conditions for SOCP (5.2) are written as

\[
\left( \begin{array}{c}
\lambda \\
\end{array} \right) \in \mathcal{K}_1 \times \mathcal{K}_2, \quad \left( \begin{array}{cc}
O & -A^T \\
A & O \\
\end{array} \right) \left( \begin{array}{c}
\lambda \\
\end{array} \right) + \left( \begin{array}{c}
c \\
b \\
\end{array} \right) \in \mathcal{K}_1 \times \mathcal{K}_2,
\]

we can obtain a solution of SOCP (5.2) by solving SOCCP (5.3). In this experiment, we chose eight pairs of $(\mathcal{K}_1, \mathcal{K}_2)$ such that $n_1 = \cdots = n_N$ and $m_1 = \cdots = m_M$ and generated 100 problem instances for each $(\mathcal{K}_1, \mathcal{K}_2)$, where matrix $A$ is randomly chosen so that its components are contained in the interval $[-1,1]$, and vectors $b$ and $c$ are randomly chosen so that SOCP (5.2) is solvable.

In applying Algorithm 2 to SOCCP (5.3), we chose parameters and initial points in a way similar to the previous experiment. As the interior point method for SOCP, we used SDPT3 solver [24]. We give the results in Table 5.3, in which $(n, N)$ denotes the dimension of $\mathcal{K}_1$ and the number of second-order cones comprising $\mathcal{K}_1$, $(m, M)$ applies similarly to $\mathcal{K}_2$, and the numbers of iterations are the average of 100 runs for each $(\mathcal{K}_1, \mathcal{K}_2)$. As this table shows, our method tends to require fewer iterations than the interior point method when the dimension of second-order cone is large.

5.3. Nonlinear case. We solved the following nonlinear SOCP: Find $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ such that

\[
\begin{align*}
x \in \mathcal{K}, \quad & y \in \mathcal{K}, \quad x^T y = 0, \quad y = f(x),
\end{align*}
\]
Table 5.3  
Comparison of Algorithm 2 and the interior point method on linear SOCPs

\begin{tabular}{|c|c|c|c|}
\hline
(n, N) & (m, M) & Our method \( \bar{\text{Ite}} \) & SDPT3 \( \bar{\text{Ite}} \) \\
\hline
(100, 2) & (100, 4) & 5.07 & 11.07 & 12.58 \\
(100, 20) & (100, 25) & 5.21 & 18.55 & 15.81 \\
(400, 8) & (300, 5) & 5.30 & 9.23 & 14.05 \\
(400, 40) & (300, 50) & 5.53 & 11.52 & 15.24 \\
(800, 4) & (900, 3) & 5.74 & 9.67 & 15.19 \\
(800, 50) & (900, 60) & 5.71 & 12.24 & 16.12 \\
\hline
\end{tabular}

where \( \mathcal{K} = \mathcal{K}^3 \times \mathcal{K}^2 \) and \( f : \mathbb{R}^5 \to \mathbb{R}^5 \) is given by

\[
\begin{align*}
\tilde{f}(x) := & \begin{pmatrix}
24(2x_1 - x_2)^3 + \exp(x_1 - x_3) - 4x_4 + x_5 \\
-12(2x_1 - x_2)^3 + 3(3x_2 + 5x_3)/\sqrt{1 + (3x_2 + 5x_3)^2} - 6x_4 - 7x_5 \\
- \exp(x_1 - x_3) + 5(3x_2 + 5x_3)/\sqrt{1 + (3x_2 + 5x_3)^2} - 3x_4 + 5x_5 \\
4x_1 + 6x_2 + 3x_3 - 1 \\
-x_1 + 7x_2 - 5x_3 + 2
\end{pmatrix}.
\end{align*}
\]

We note that, in view of the KKT conditions (6.4) for the SOCP (6.1) (see Section 6), SOCCP (5.4) is equivalent to the following SOCP:

\[
\begin{align*}
\text{Minimize} & \quad \exp(z_1 - z_3) + 3(2z_1 - z_2)^4 + \sqrt{1 + (3z_2 + 5z_3)^2} \\
\text{subject to} & \quad \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} \in \mathcal{K}^3, \quad \begin{pmatrix} 4 & 6 & 3 \\ -1 & 7 & -5 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} + \begin{pmatrix} -1 \\ 2 \end{pmatrix} \in \mathcal{K}^2.
\end{align*}
\]

Since the objective function of this SOCP is convex, we can easily see that the function \( f \) is monotone. In the experiments, we set the parameters as \( \rho = 0.5, \sigma = 0.4, \kappa = 0.01 \) and \( \tilde{\kappa} = 1. \)

We used nine different pairs \((\eta, \bar{\eta})\) and ran Algorithm 2 with 100 initial points \((x^{(0)}, y^{(0)})\) for each \((\eta, \bar{\eta})\) determined as \((x^{(0)}, y^{(0)}) := \Gamma(a, b)/\| (a, b) \|\), where \( \Gamma \) is randomly chosen from \([0, 10]\) and each component of \((a, b) \in \mathbb{R}^2 \times \mathbb{R}^3 \) is randomly chosen from \([-1, 1]\). Table 5.4 shows the results of the experiments, in which \( \bar{\text{Ite}} \) and \( \bar{\text{Newton}} \) are the average of 100 runs for each \((\eta, \bar{\eta})\). As Table 5.4 shows, the number of outer iterations decreases as \( \eta \) becomes smaller. On the other hand, the number of inner iterations decreases as \( \bar{\eta} \) becomes smaller and \( \eta \) becomes larger. Since the computation time is largely dependent on the number of inner iterations, it would be recommended to choose \( \eta \) as large as possible and \( \bar{\eta} \) as small as possible. However, it should be kept in mind that, when \( \bar{\eta} \) is too small, \( \mu \) and \( \varepsilon \) tend to 0 very rapidly, which may deteriorate the advantage of smoothing and regularization methods.

6. Concluding remarks. In this section, we make some comments that complement the results obtained in the paper.

In this paper, the function \( F \) in SOCCP (1.1) is assumed to be of the form \( F(x, y, \zeta) = f(x) - y \). This assumption may seem rather restrictive. However, the KKT conditions for the SOCP

\[
\begin{align*}
\text{Minimize} & \quad \theta(z) \\
\text{subject to} & \quad \gamma(z) \in \mathcal{K}
\end{align*}
\]
SMOOTHING AND REGULARIZATION METHOD FOR MONOTONE SOCP

Table 5.4
Results for nonlinear SOCCPs with various choices of $\eta, \eta$

<table>
<thead>
<tr>
<th>$\eta$</th>
<th>$\eta$</th>
<th>$\xi$</th>
<th>$\xi$ Newton</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>16.79</td>
<td>26.34</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td>12.20</td>
<td>13.70</td>
</tr>
<tr>
<td>0.5</td>
<td>0.01</td>
<td>10.50</td>
<td>10.75</td>
</tr>
<tr>
<td>0.5</td>
<td>0.001</td>
<td>9.58</td>
<td>9.85</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>8.35</td>
<td>19.11</td>
</tr>
<tr>
<td>0.1</td>
<td>0.01</td>
<td>8.13</td>
<td>13.51</td>
</tr>
<tr>
<td>0.01</td>
<td>0.01</td>
<td>7.66</td>
<td>11.25</td>
</tr>
<tr>
<td>0.01</td>
<td>0.001</td>
<td>5.99</td>
<td>14.32</td>
</tr>
<tr>
<td>0.01</td>
<td>0.001</td>
<td>5.73</td>
<td>12.35</td>
</tr>
</tbody>
</table>

can be written as the SOCCP with $F(x, y, z) = f(x) - y$ as follows: In SOCP (6.1), let $z = z' - z''$ with $z' \in \mathbb{R}^n_+$ and $z'' \in \mathbb{R}^n_+$, and denote $\tilde{z} := (\tilde{z}') \in \mathbb{R}^{2n}_+$, where $\mathbb{R}^n_+$ is the $n$-dimensional nonnegative orthant. Moreover, define $\tilde{\theta} : \mathbb{R}^{2n} \to \mathbb{R}$ by $\tilde{\theta}(\tilde{z}) = \theta(z' - z'')$, and $\tilde{\gamma} : \mathbb{R}^{2n} \to \mathbb{R}^n$ by $\tilde{\gamma}(\tilde{z}) = \gamma(z' - z'')$. Then SOCP (6.1) can be reformulated as

\[
\begin{array}{ll}
\text{Minimize} & \tilde{\theta}(\tilde{z}) \\
\text{subject to} & \left(\tilde{\gamma}(\tilde{z})\right) \in \mathcal{K} \times \mathbb{R}^{2n}_+, \\
& \left(\tilde{\lambda}_1 \right)^T \left(\tilde{\gamma}(\tilde{z})\right) = 0.
\end{array}
\]

(6.2)

and the KKT conditions for (6.2) are written as

\[
\begin{array}{l}
\nabla \tilde{\theta}(\tilde{z}) - \left(\nabla \tilde{\gamma}(\tilde{z}) \mathbf{I}\right) \left(\tilde{\lambda}_1 \right) = 0, \\
\left(\tilde{\lambda}_1 \right) \in \mathcal{K} \times \mathbb{R}^{2n}_+, \\
\left(\tilde{\gamma}(\tilde{z})\right) \in \mathcal{K} \times \mathbb{R}^{2n}_+, \\
\left(\tilde{\lambda}_1 \right)^T \left(\tilde{\gamma}(\tilde{z})\right) = 0.
\end{array}
\]

(6.3)

Now, let $\mu_1 = \tilde{\gamma}(\tilde{z})$, and notice that (2.7) holds. Then, (6.3) can be rewritten as

\[
\begin{array}{l}
\left(\nabla \tilde{\theta}(\tilde{z}) - \nabla \tilde{\gamma}(\tilde{z}) \tilde{\lambda}_1 \right) = \left(\tilde{\mu}_1 \right), \\
\left(\tilde{\lambda}_1 \right) \in \mathcal{K} \times \mathbb{R}^{2n}_+, \\
\left(\tilde{\mu}_1 \right) \in \mathcal{K} \times \mathbb{R}^{2n}_+, \\
\left(\tilde{\lambda}_1 \right)^T \left(\tilde{\mu}_1 \right) = 0.
\end{array}
\]

(6.4)

Setting

\[
x = \left(\tilde{\lambda}_1 \right), \quad y = \left(\tilde{\mu}_1 \right), \quad f(x) = \left(\nabla \tilde{\theta}(\tilde{z}) - \nabla \tilde{\gamma}(\tilde{z}) \tilde{\lambda}_1 \right),
\]

the KKT conditions (6.4) for SOCP (6.1) can be reduced to the SOCCP with $F(x, y, z) = f(x) - y$. Note that the KKT conditions (6.4) for SOCP (6.1) contain more variables than the original KKT conditions. Furthermore, some desirable properties of the functions involved in SOCP (6.1) may be lost. For example, even if $\theta$ and $\gamma$ in SOCP (6.1) are strictly convex, $\theta$ and $\gamma$ in (6.2) are merely convex. Hence, it would be useful to
develop a method that can directly deal with the original KKT conditions, or more generally, SOCCP involving the function $F(x, y, \zeta)$ which is not restricted to be of the form $F(x, y, \zeta) = f(x) - y$.

In Sections 3 and 4, it is also assumed that $K = K^n$. For the general case where $K = K^{n_1} \times \cdots \times K^{n_m}$, $H_{\mu,c}(x, y)$ and $\nabla H_{\mu,c}(x, y)$ are given by

\[
H_{\mu,c}(x, y) = \begin{pmatrix}
\varphi_\mu(x^1, y^1) \\
\vdots \\
\varphi_\mu(x^m, y^m) \\
f_c(x) - y
\end{pmatrix},
\]

\[
\nabla H_{\mu,c}(x, y) = \begin{pmatrix}
I - \text{diag}(\nabla P_\mu(x^j - y^j))_{j=1}^m \\
\text{diag}(\nabla P_\mu(x^j - y^j))_{j=1}^m \\
-I
\end{pmatrix},
\]

where $x = (x^1, \ldots, x^m) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m}$, $y = (y^1, \ldots, y^m) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m}$, and $\text{diag}(\nabla P_\mu(x^j - y^j))_{j=1}^m$ denotes the block-diagonal matrix with block elements $\nabla P_\mu(x^j - y^j)$. Moreover, the definition (4.12) of $\tilde{\lambda}(z)$ is replaced by

\[
\tilde{\lambda}(z) := \begin{cases}
\min_{(i,j) \in \mathcal{I}(z)} |\lambda_i(z^j)| & (\mathcal{I}(z) \neq \emptyset) \\
0 & (\mathcal{I}(z) = \emptyset)
\end{cases},
\]

where $z = (z^1, \ldots, z^m) \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m}$, $\lambda_i(z^j)$ ($i = 1, 2$) are the spectral values of $z^j$, and $\mathcal{I}(z) \subseteq \{1, 2\} \times \{1, \ldots, m\}$ is the index set defined by $\mathcal{I}(z) := \{(i, j) \mid \lambda_i(z^j) \neq 0\}$. Then, we can also show that Theorems 4.3 and 4.13 hold for monotone $f$ in a similar way.

Acknowledgments. The authors are grateful to the anonymous referees for their very helpful comments.

REFERENCES