Complementarity Functions and Numerical Experiments on Some Smoothing Newton Methods for Second-Order-Cone Complementarity Problems

X.D. Chen[*] D. Sun[†] and J. Sun[‡]

October 26, 2002

Abstract

Two results on the second-order-cone complementarity problem are presented. We show that the squared smoothing function is strongly semismooth. Under monotonicity and strict feasibility we provide a new proof, based on a penalized natural complementarity function, for the solution set of the second-order-cone complementarity problem being bounded. Numerical results of squared smoothing Newton algorithms are reported.

Keywords: complementarity function, soc, smoothing Newton method, quadratic convergence

Dedication: I still remember the first optimization book in English that I read is "Computational Methods in Optimization. A Unified Approach", written by Lucien in 1971. So I was very excited when I first met Lucien in 1995 in Coogee Beach, an eastern suburb in Sydney, where Liqun Qi introduced him to me. Lucien has a broad interest in optimization both in theory and applications. He always likes to tackle challenging problems. It is my great honor to dedicate this paper to Lucien on the occasion of his 72nd birthday. – Defeng Sun

1 Introduction

Optimization problems with second order cone constraints have wide range of applications in engineering, control, and management science [11]. In this paper we focus on the second order cone complementarity problem (SOCCP for short), which includes the Karush-Kuhn-Tucker (KKT)

[*]Department of Applied Mathematics, Tongji University, Shanghai, China. Email: chenxiongda@yahoo.com. His research is supported by SMA.

[†]Department of Mathematics, National University of Singapore, Republic of Singapore. Email: matsundf@nus.edu.sg. His research is partially supported by Grant R146-000-035-101 from NUS.

[‡]SMA and Department of Decision Sciences, National University of Singapore, Republic of Singapore. Email: jsun@nus.edu.sg. His research is partially supported by SMA and Grants RP314000-028/042-112 from NUS.

system of the second order cone programming problem as a special case. Through the so-called complementarity function (C-function for short), the SOCCP can be transformed into a system of nonsmooth equations [7]. For applications of C-functions in vector spaces, see [6, 13]. C-functions are usually smoothed and regularized, say, by convolution and Tikhonov regularization [16] in solving vector complementarity problems in order to facilitate the so-called smoothing Newton methods. Using the same framework, we show that the SOCCP can also be effectively solved by a smoothing Newton method. In Section 2, some preliminary results for the second order cone are presented and two C-functions of the second order cone are analyzed. The boundedness property of the level set of one of the C-functions is proved in Section 3, which is used to characterize the boundedness of the solution set of the SOCCP. Section 4 shows the strong semismoothness of the squared smoothing function. The algorithms and the numerical experiments are presented in Section 5 and 6, respectively.

2 Preliminary results

The second order cone (SOC) in $\mathcal{R}^n$, also called Lorentz cone or ice–cream cone, is defined by

$$\mathcal{K}^n = \{(x_1, x_2^T)^T \mid x_1 \in \mathcal{R}, x_2 \in \mathcal{R}^{n-1} \text{ and } x_1 \geq \|x_2\|\}. \tag{2.1}$$

Here and below, $\|\cdot\|$ denotes the 2-norm. If there is no ambiguity, for convenience, we write $x = (x_1, x_2)$ instead of $x = (x_1, x_2^T)^T$. We are interested in complementarity problems involving the second order cone in its constraints. In general, the SOCCP has the following form:

$$\text{Find an } x \in \mathcal{K}, \text{ such that } F(x) \in \mathcal{K} \text{ and } x^T F(x) = 0, \tag{2.2}$$

where $F : \mathcal{R}^N \to \mathcal{R}^N$ is a continuously differentiable function, and

$$\mathcal{K} = \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_p} \tag{2.3}$$

with $p, n_1, \ldots, n_p \geq 1$ and $n_1 + \cdots + n_p = N$. Unless otherwise specified, in the following analysis we assume that $\mathcal{K} = \mathcal{K}^n$. This, however, does not lose any generality because our analysis can be easily extended to the general case.

The SOC is associated with its Jordan algebra. A Jordan algebra is a vector space with the following property [5]:

Definition 2.1 *A real vector space V is called a Jordan algebra if a bilinear mapping (Jordan product) $(x, y) \to x \circ y$ from $V \times V$ into V is defined with the following property*

$$x^2 \circ (x \circ y) - x \circ (x^2 \circ y) = 0, \ \forall\ x, y \in V, \tag{2.4}$$

where $x^2 = x \circ x$.

Some basic properties of the Jordan algebra can be found in [4, 5]. In the context of the SOC, the Jordan product is defined as

$$x \circ y = \begin{pmatrix} x^T y \\ y_1 x_2 + x_1 y_2 \end{pmatrix} \tag{2.5}$$

if $x = (x_1, x_2)$ and $y = (y_1, y_2)$. An important character of Jordan algebra is its eigen-decomposition. The eigen-decomposition of the SOC is defined in the following way. For any $x \in \mathcal{R}^n$, $x = \lambda_1 u_1 + \lambda_2 u_2$ is called the eigen-decomposition of x in which

$$\lambda_i = x_1 + (-1)^i \|x_2\| \quad \text{and} \quad u_i = \begin{cases} \frac{1}{2}(1, (-1)^i \frac{x_2}{\|x_2\|}), & \text{if } x_2 \neq 0, \\ \frac{1}{2}(1, (-1)^i \frac{\omega}{\|\omega\|}), & \text{otherwise, for any } \omega \neq 0. \end{cases} \tag{2.6}$$

By using the eigen-decomposition, a scalar function can be extended to an SOC function. Given a function $f(\cdot) : \mathcal{R} \to \mathcal{R}$, the corresponding SOC function is defined as

$$f(x) = f(\lambda_1)u_1 + f(\lambda_2)u_2. \tag{2.7}$$

For convenience of discussion, we denote

$$x_+ = (\lambda_1)_+ u_1 + (\lambda_2)_+ u_2,$$

$$|x| = |\lambda_1| u_1 + |\lambda_2| u_2,$$

and

$$x_- = (\lambda_1)_- u_1 + (\lambda_2)_- u_2,$$

where for any scalar $\alpha \in \mathcal{R}$, $\alpha_+ = \max\{0, \alpha\}$ and $\alpha_- = \max\{0, -\alpha\}$. It can be seen that $x = x_+ - x_-$, $x_+, x_- \in \mathcal{K}^n$ and $x_+ \circ x_- = 0$. For any $x \in \mathcal{K}^n$, it is obvious that both eigenvalues of x are nonnegative, therefore we define

$$x^{1/2} = \lambda_1^{1/2} u_1 + \lambda_2^{1/2} u_2.$$

It has been shown that the following results hold, e.g., see Fukushima, Luo and Tseng [7].

Proposition 2.2

(a) $|x| = (x^2)^{1/2}$.

(b) $x^2 = (\lambda_1)^2 u_1 + (\lambda_2)^2 u_2$.

(c) x_+ *is the projection of* x *onto* $\mathcal{K}^n$ *and* $x_+ = (x + |x|)/2$.

(d) $x, y \in \mathcal{K}^n$ *and* $x^T y = 0 \iff x, y \in \mathcal{K}^n$ *and* $x \circ y = 0 \iff x - [x - y]_+ = 0$.

A function $\psi : \mathcal{R}^n \to \mathcal{R}^n$ is called an SOC C-function if $x^T y = 0,\ x, y \in \mathcal{K} \iff \psi(x, y) = 0$. The result in Proposition 2.2 (d) shows that $\phi(x, y) := x - [x - y]_+$ is a C-function. Let the penalized natural function be defined as

$$\phi_p(x, y) := x - [x - y]_+ + x_+ \circ y_+.$$

In the following, we will discuss some properties of ϕ and ϕ_p in Propositions 2.3 and 2.4 and show that ϕ_p is a C-function in Proposition 2.5. Note that ϕ_p is the counterpart of penalized C-function given in [1] in the context of nonlinear complementarity problems.

Proposition 2.3 *Both* $\phi(x,y) := x-[x-y]_+$ *and* $\phi_p(x,y) := x-[x-y]_+ + x_+ \circ y_+$ *are symmetric in* x *and* y*, i.e.,*

$$\phi(x,y) = \phi(y,x) \quad \textit{and} \quad \phi_p(x,y) = \phi_p(y,x). \tag{2.8}$$

Proof. We have

$$\begin{array}{rcl} \phi(x,y) & = & x-[x-y]_+ \\ & = & x-(x-y+|x-y|)/2 \\ & = & \frac{1}{2}(x+y-|x-y|) \end{array} \tag{2.9}$$

and $\phi_p(x,y) = \phi(x,y) + x_+ \circ y_+ = \phi(y,x) + y_+ \circ x_+ = \phi_p(y,x)$. Hence, the conclusion follows. $\square$

Proposition 2.4

$$\|\phi_p(x,y)\| \geq \max\{\|x_-\|, \|y_-\|\}.$$

Proof. By Proposition 2.3, it suffices to prove that

$$\|\phi_p(x,y)\| \geq \|x_-\|. \tag{2.10}$$

We have

$$\begin{array}{rcl} \|\phi_p(x,y)\|^2 & = & \|x_+ - x_- - [x-y]_+ + x_+ \circ y_+\|^2 \\ & = & \|x_-\|^2 + \|x_+ - [x-y]_+ + x_+ \circ y_+\|^2 + 2[-x_-]^T(x_+ - [x-y]_+ + x_+ \circ y_+) \\ & \geq & \|x_-\|^2 - 2[x_-]^T x_+ + 2[x_-]^T[x-y]_+ - 2[x_-]^T(x_+ \circ y_+) \\ & \geq & \|x_-\|^2 - 2([x_-] \circ [x_+])^T y_+ \\ & = & \|x_-\|^2, \end{array}$$

where the first inequality holds due to the nonnegativity of norm, the second inequality due to Proposition 2.2 and the nonnegativity of an inner product of two elements in $\mathcal{K}^n$, and the last equality follows from the property of Jordan product [7]:

$$(x \circ y)^T z = (y \circ z)^T x = (z \circ x)^T y \tag{2.11}$$

for all $x, y, z \in \mathcal{R}^n$. $\square$

It can easily be seen that we have $\|\phi(x,y)\| \geq \max\{\|x_-\|, \|y_-\|\}$ similarly.

Proposition 2.5 *The following two statements are equivalent:*

(a) $x \in \mathcal{K}^n$, $y \in \mathcal{K}^n$ *and* $x^T y = 0$.

(b) $\phi_p(x,y) = x - [x-y]_+ + x_+ \circ y_+ = 0$.

Proof. (a)⇒ (b): In this case, $\phi_p(x,y) = \phi(x,y)$. By Proposition 2.2, (b) holds.

(b)⇒ (a): From Proposition 2.4, if either $x \notin \mathcal{K}^n$ or $y \notin \mathcal{K}^n$, then $\|\phi_p(x,y)\| > 0$. Thus $x, y \in \mathcal{K}^n$. In such case

$$[\phi_p(x,y)]_1 = \begin{cases} y_1 + x^T y, & \text{if } x-y \in \mathcal{K}^n; \\ (x_1 + y_1 - \|x_2 - y_2\|)/2 + x^T y, & \text{if } x-y \notin \mathcal{K}^n \cup (-\mathcal{K}^n); \\ x_1 + x^T y, & \text{if } x-y \in -\mathcal{K}^n. \end{cases} \tag{2.12}$$

Since for $x, y \in \mathcal{K}^n$, x_1, y_1 and $x_1 + y_1 - \|x_2 - y_2\|$ are all nonnegative, one has $[\phi_p(x,y)]_1 = 0$ only if $x^T y = 0$. $\square$

Note that we obtain problem (2.2) if we substitute $y = F(x)$ in above proposition. Therefore, to solve problem (2.2) is to find the solution to $\phi_p(x, F(x)) = 0$.

3 Boundedness properties of an SOC function

In this section, we show that under certain conditions the level sets of the merit function based on ϕ_p are bounded. As a consequence of Proposition 2.5, the solution set of problem (2.2) is also bounded.

For a given function $F : \mathcal{R}^n \to \mathcal{R}^n$, define

$$\Phi_p(x) = x - [x - F(x)]_+ + x_+ \circ [F(x)]_+. \tag{3.1}$$

In the following, we will use the fact that if $x^k \in \mathcal{K}^n$, then the boundedness of $\{x^k\}$ is equivalent to the boundedness of $\{x_1^k\}$.

Recall that a function F is said to be monotone on $\mathcal{R}^n$ if

$$(x - y)^T (F(x) - F(y)) \geq 0, \ \forall\, x, y \in \mathcal{R}^n \tag{3.2}$$

and strictly monotone if the above inequality holds strictly for any $x \neq y$.

Lemma 3.1 *Suppose that the SOCCP has a strictly feasible point $\hat{x}$, i.e., $\hat{x} \in \operatorname{int} \mathcal{K}^n$ and $F(\hat{x}) \in \operatorname{int} \mathcal{K}^n$, and that F is a monotone function. For any sequence $\{x^k\}$ satisfying $\|x^k\| \to \infty$, $\limsup_{k\to\infty} \|x_-^k\| < \infty$ and $\limsup_{k\to\infty} \|F_-(x^k)\| < \infty$, we have*

$$(x_+^k)^T F_+(x^k) \to \infty. \tag{3.3}$$

Proof. Since F is monotone, the following statement holds for all x^k,

$$(x^k - \hat{x})^T (F(x^k) - F(\hat{x})) \geq 0, \tag{3.4}$$

which can be expanded as

$$(x^k)^T F(\hat{x}) + (\hat{x})^T F(x^k) \leq (x^k)^T F(x^k) + (\hat{x})^T F(\hat{x}). \tag{3.5}$$

Since $x^k = x_+^k - x_-^k$ and $F(x^k) = F_+(x^k) - F_-(x^k)$, we have

$$(x_+^k)^T F(\hat{x}) - (x_-^k)^T F(\hat{x}) + (\hat{x})^T F_+(x^k) - (\hat{x})^T F_-(x^k) \leq (x^k)^T F(x^k) + (\hat{x})^T F(\hat{x}). \tag{3.6}$$

Note that

$$\begin{array}{rcl} (x_+^k)^T F(\hat{x}) & = & [x_+^k]_1 [F(\hat{x})]_1 + [x_+^k]_2^T [F(\hat{x})]_2 \\ & \geq & [x_+^k]_1 [F(\hat{x})]_1 - \|[x_+^k]_2\| \|[F(\hat{x})]_2\| \\ & \geq & [x_+^k]_1 [F(\hat{x})]_1 - [x_+^k]_1 \|[F(\hat{x})]_2\| \\ & = & [x_+^k]_1 \left([F(\hat{x})]_1 - \|[F(\hat{x})]_2\|\right). \end{array} \tag{3.7}$$

Since $\|x_+^k\| \geq \|x^k\| - \|x_-^k\|$, the assumptions on $\{x^k\}$ imply that $\|x_+^k\| \to \infty$ and $[x_+^k]_1 \to \infty$. Because $\hat{x}$ is a strictly feasible point of problem (2.2), we have

$$(x_+^k)^T F(\hat{x}) \to \infty. \tag{3.8}$$

Moreover, $\limsup_{k\to\infty}(x_-^k)^T F(\hat{x}) < \infty$, $\limsup_{k\to\infty} \hat{x}^T F_-(x^k) < \infty$ and $\hat{x}^T F_+(x^k) \geq 0$, which, together with (3.8), gives that

$$(x^k)^T F(x^k) \to \infty. \tag{3.9}$$

Further expanding $(x^k)^T F(x^k)$, we have

$$(x^k)^T F(x^k) = (x_+^k)^T F_+(x^k) + (x_-^k)^T F_-(x^k) - (x_+^k)^T F_-(x^k) - (x_-^k)^T F_+(x^k). \tag{3.10}$$

Since the third and fourth terms of the right hand side are negative and the second is bounded from above, we have $(x_+^k)^T F_+(x^k) \to \infty$. $\square$

Now we are ready to prove the boundedness of level sets of the natural merit function $\|\Phi_p(x)\|$.

Theorem 3.2 *The level set $L = \{x \mid \|\Phi_p(x)\| \leq C\}$ is bound provided that F is monotone and that problem (2.2) has a strictly feasible point, where $C \geq 0$ is a constant.*

Proof. It is sufficient to prove that $\|\Phi_p(x)\| \to \infty$ whenever $\|x\| \to \infty$. If $\|x_-\| \to \infty$ or $\|F_-(x)\| \to \infty$, the result holds by Proposition 2.4.

On the other hand, if $\limsup \|x_-\| < \infty$ and $\limsup \|F_-(x)\| < \infty$, then there exists a constant C_0 such that $\max\{\|x_-\|, \|F_-(x)\|\} < C_0$. Hence,

$$\begin{aligned}
&[x - (x - F(x))_+]_1 \\
=& \begin{cases} x_1, & \text{if } x - F(x) \in -\mathcal{K}^n, \\ F_1(x), & \text{if } x - F(x) \in \mathcal{K}^n, \\ \frac{1}{2}(x_1 + F_1(x) - \|x_2 - F_2(x)\|), & \text{if } x - F(x) \notin \mathcal{K}^n \cup -\mathcal{K}^n \end{cases} \\
\geq& \begin{cases} [x_+]_1 - [x_-]_1, & \text{if } x - F(x) \in -\mathcal{K}^n, \\ [F_+(x)]_1 - [F_-(x)]_1, & \text{if } x - F(x) \in \mathcal{K}^n, \\ \frac{1}{2}([x_+]_1 + [F_+(x)]_1 - \|[x_+]_2 - [F_+(x)]_2\|) & \\ \quad -\frac{1}{2}([x_-]_1 + [F_-(x)]_1 + \|[x_-]_2 - [F_-(x)]_2\|), & \text{if } x - F(x) \notin \mathcal{K}^n \cup -\mathcal{K}^n \end{cases} \\
\geq& \begin{cases} [x_+]_1 - C_0, & \text{if } x - F(x) \in -\mathcal{K}^n, \\ [F_+(x)]_1 - C_0, & \text{if } x - F(x) \in \mathcal{K}^n, \\ \frac{1}{2}([x_+]_1 + [F_+(x)]_1 - \|[x_+]_2\| - \|[F_+(x)]_2\|) - 2C_0, & \text{if } x - F(x) \notin \mathcal{K}^n \cup -\mathcal{K}^n, \end{cases}
\end{aligned} \tag{3.11}$$

which, means

$$\liminf [x - (x - F(x))_+]_1 > -\infty.$$

Since Lemma 3.1 implies that $x_+^T F_+(x) \to \infty$, we have $[\Phi_p(x)]_1 \to \infty$, which in turn implies that $\|\Phi_p(x)\| \to \infty$. $\square$

As a natural consequence, we have the following result, which has been proved in [12, 3, 8].

Corollary 3.3 *If F is monotone and if problem (2.2) has a strictly feasible point, then the solution set of problem (2.2) is bounded.*

Since Φ_p is a C-function and its level sets are bounded, one may expect to solve the SOCCP by solving $\Phi_p(x) = 0$ like in the case of nonlinear complementarity problems [1]. However, unlike the vector case, there is a complication: The Jacobian of $\Phi_p(x)$ is not necessarily a P-matrix or quasi-P matrix. Consider the following example.

Example 3.4 *Let $F : \mathcal{R}^3 \to \mathcal{R}^3$ be defined as*

$$F(x) = x + (2, 4, 8)^T. \tag{3.12}$$

Obviously, $F'(x) = I$ is a positive definite matrix and

$$\nabla_x \Phi_p(x)|_{x=(2,-4,0)^T} = \frac{1}{4}\begin{bmatrix} 22 & -21 & 24 \\ -18 & 25 & -6 \\ 18 & -12 & 28 \end{bmatrix}, \tag{3.13}$$

which is neither a P-matrix nor a quasi-P matrix since its determinant is $-\frac{29}{16}$, a negative number.

As a result, if we apply Newton-type methods to Φ_p, the Jacobian could be singular, causing failure of the algorithms. We will list our experimental results in Section 6, which do include several failure examples.

4 Strong semismoothness of smoothing functions

Different from function Φ_p, the function $\Phi(x) = x - [x - F(x)]_+$ can guarantee the nonsingularity of the Jacobian if $F(x)$ is strongly monotone, which is shown in [18]. In addition, as we will show now, the smoothing function of $\Phi(x)$, denoted by

$$\Phi(x, \mu) = \frac{1}{2}\Big(x + F(x) - \sqrt{(x - F(x))^2 + 4\mu^2 e}\Big), \quad e = (1, 0) \in \mathcal{R} \times \mathcal{R}^{n-1} \tag{4.1}$$

(forgive the abused notation) is strongly semismooth provided that F' is locally Lipschitz continuous, therefore the corresponding smoothing Newton methods will have quadratic convergence (see Section 5). For the original definition of strong semismoothness and its relation to quadratic convergence, see [14].

It is clear that in order to show the strong semismoothness of $\Phi(x, \mu)$ it is sufficient to show that $f(x, \mu) = \sqrt{x^2 + \mu^2 e}$ is strongly semismooth. Let D_f be the set of points where f is differentiable. It is shown in [17] that f is strongly semismooth at (x, μ) if it is locally Lipschitz, directionally differentiable at (x, μ) and for $(h, \varepsilon) \to 0$ with $(h, \varepsilon) \in \mathcal{R}^n \times \mathcal{R}$ and $(x+h, \mu+\varepsilon) \in D_f$ the following relation holds

$$f(x + h, \mu + \varepsilon) - f(x, \mu) - f'(x + h, \mu + \varepsilon)\begin{pmatrix} h \\ \varepsilon \end{pmatrix} = O(\|(h, \varepsilon)\|^2). \tag{4.2}$$

First we introduce the associated matrix (operator) $\mathcal{L}_x$ of any $x \in \mathcal{R}^n$ under the SOC structure [7]:

$$\mathcal{L}_x = \begin{bmatrix} x_1 & x_2^T \\ x_2 & x_1 I \end{bmatrix}. \tag{4.3}$$

Notice that $\mathcal{L}_x y = x \circ y$ holds for all y.

Lemma 4.1 *If $x = \lambda_1 u_1 + \lambda_2 u_2$ is the eigen-decomposition of $x \in \mathcal{R}^n$, then $\mathcal{L}_x$ has the eigen-decomposition of the following form:*

$$\mathcal{L}_x = Q \operatorname{diag}\left(\lambda_1, \lambda_2, \frac{\lambda_1+\lambda_2}{2}, \cdots, \frac{\lambda_1+\lambda_2}{2}\right) Q^T, \tag{4.4}$$

where, $Q_{n\times(n-2)} \in \mathcal{R}^{n\times(n-2)}$ is any matrix so that $Q = \left[\sqrt{2}u_1, \sqrt{2}u_2, Q_{n\times(n-2)}\right]$ is orthonormal.

Proof. It can be verified by simple calculations. $\square$

Theorem 4.2 *The function $f(x,\mu) = \sqrt{x^2 + \mu^2 e}$ is strongly semismooth on $(x,\mu) \in \mathcal{R}^n \times \mathcal{R}$.*

Proof. We need to show that f is locally Lipschitz and directionally differentiable at (x,μ) and (4.2) holds.

By the eigen-decomposition of the SOC, one can easily show the Lipschitz continuity of $f(x,\mu)$.

To show the directional differentiability of $f(x,\mu)$, first we note the following fact: $f(x,\mu)$ is directional differentiable at $(0,0)$ due to its positive homogeneity and

$$f'((0,0);(h,\varepsilon)) = f(h,\varepsilon).$$

Moreover, if either $\mu \neq 0$ or x is neither on the boundary of $\mathcal{K}^n$ nor $-\mathcal{K}^n$, f is continuously differentiable around (x,μ), and is thus directionally differentiable at (x,μ).

Next, we consider the directional differentiability of f at (x,μ) with $\mu = 0$ and a nonzero x on the boundary of $\mathcal{K}^n$ or $-\mathcal{K}^n$. In this case we have $\|x_2\| > 0$. Let $x = \lambda_1 u_1 + \lambda_2 u_2$ and $x(t) = x + th = \lambda_1(t)u_1(t) + \lambda_2(t)u_2(t)$ be the SOC eigen-decompositions of x and $x(t)$, respectively, where for $i = 1, 2$,

$$\lambda_i = x_1 + (-1)^i \|x_2\|, \quad u_i = \frac{1}{2}(1, (-1)^i \frac{x_2}{\|x_2\|}), \tag{4.5}$$

$$\lambda_i(t) = x_1 + th_1 + (-1)^i \|x_2 + th_2\|, \quad \text{and} \quad u_i(t) = \frac{1}{2}(1, (-1)^i \frac{x_2 + th_2}{\|x_2 + th_2\|}). \tag{4.6}$$

We only prove the case that x is on the boundary of $\mathcal{K}^n$. Therefore, $\lambda_1 = 0$ and $\lambda_2 \neq 0$. The case that x is on the boundary of $-\mathcal{K}^n$ can be proved similarly.

We have the following facts:

$$\lambda_i(t) - \lambda_i = 2tu_i^T h + O(t^2) \tag{4.7}$$

and

$$u_i(t) - u_i = \frac{t}{2}\left[\begin{array}{c} 0 \\ (-1)^i(I - \frac{x_2x_2^T}{\|x_2\|^2})\frac{h_2}{\|x_2\|} \end{array}\right] + O(t^2), \tag{4.8}$$

which can be obtained by direct calculations. Therefore,

$$\begin{array}{rl} & f(x+th, t\varepsilon) - f(x,0) \\ = & \sqrt{\lambda_2^2(t) + (t\varepsilon)^2}u_2(t) + \sqrt{\lambda_1^2(t) + (t\varepsilon)^2}u_1(t) - |\lambda_2|u_2 \\ = & \sqrt{\lambda_2^2(t) + (t\varepsilon)^2}[u_2(t) - u_2] + [\sqrt{\lambda_2^2(t) + (t\varepsilon)^2} - |\lambda_2|]u_2 + \sqrt{\lambda_1^2(t) + (t\varepsilon)^2}u_1 \\ = & \frac{t|\lambda_2|}{2}\left[\begin{array}{c} 0 \\ (I - \frac{x_2x_2^T}{\|x_2\|^2})\frac{h_2}{\|x_2\|} \end{array}\right] + 2\text{sign}(\lambda_2)t(u_2^Th)u_2 + \sqrt{\lambda_1^2(t) + (t\varepsilon)^2}u_1 + O(t^2) \\ = & \frac{t\ \text{sign}(\lambda_2)}{2}\left[\begin{array}{c} 2u_2^Th \\ \frac{\lambda_2 h_2}{\|x_2\|} + (h_1 - x_1\frac{x_2^Th_2}{\|x_2\|^2})\frac{x_2}{\|x_2\|} \end{array}\right] + |t|\sqrt{4(u_1^Th)^2 + \varepsilon^2}u_1 + O(t^2), \end{array} \tag{4.9}$$

where sign$(\cdot)$ is a scalar function taking values $-1, 0$, or $+1$ if the variable is negative, zero or positive, respectively. The above equation implies the directional differentiability of f at $(x, 0)$.

We next prove (4.2). Notice that by Lemma 4.1 that $\mathcal{L}_{f(x,\mu)}$ is nonsingular if and only if f is differentiable at (x,μ), i.e., $(x,\mu) \in D_f$. Moreover, for any $(x,\mu) \in D_f$, we have

$$f'_x(x,\mu) = \mathcal{L}^{-1}_{f(x,\mu)}\mathcal{L}_x \quad \text{and} \quad f'_\mu(x,\mu) = \mu\mathcal{L}^{-1}_{f(x,\mu)}e, \tag{4.10}$$

which can be obtained by finding the derivatives on both sides of

$$f(x,\mu) \circ f(x,\mu) = x^2 + \mu^2 e. \tag{4.11}$$

Now, letting $(x+h, \mu+\varepsilon) \in D_f$ and denoting $f = f(x,\mu)$ and $\hat{f} = f(x+h, \mu+\varepsilon)$, we have the following equality:

$$\begin{array}{rl} & f(x+h,\mu+\varepsilon) - f(x,\mu) - f'_x(x+h,\mu+\varepsilon)h - f'_\mu(x+h,\mu+\varepsilon)\varepsilon \\ = & \mathcal{L}^{-1}_{\hat{f}}\left[\mathcal{L}_{\hat{f}}(\sqrt{(x+h)^2 + (\mu+\varepsilon)^2e} - \sqrt{x^2+\mu^2e}) - (x+h)\circ h - (\mu+\varepsilon)\varepsilon e\right] \\ = & \mathcal{L}^{-1}_{\hat{f}}\left[(x+h)^2 + (\mu+\varepsilon)^2e - (x+h)\circ h - (\mu+\varepsilon)\varepsilon e - \sqrt{(x+h)^2+(\mu+\varepsilon)^2e}\circ\sqrt{x^2+\mu^2e}\right] \\ = & \mathcal{L}^{-1}_{\hat{f}}\left[\frac{1}{2}(\sqrt{(x+h)^2+(\mu+\varepsilon)^2e} - \sqrt{x^2+\mu^2e})^2 - \frac{1}{2}(h^2+\varepsilon^2e)\right]. \end{array} \tag{4.12}$$

We consider the following three cases:

Case a. If $(x,\mu) = 0$, the right hand side of (4.12) vanishes.

Case b. Consider the case when $\mathcal{L}_f$ is invertible. In this case, $\mathcal{L}^{-1}_{\hat{f}}$ is uniformly bounded when $(h,\varepsilon) \to 0$. Then, by the Lipschitz property of f, we have

$$f(x+h,\mu+\varepsilon) - f(x,\mu) - f'_x(x+h,\mu+\varepsilon)h - f'_\mu(x+h,\mu+\varepsilon)\varepsilon = O(\|(h,\varepsilon)\|^2).$$

Case c. $\mathcal{L}_f$ is singular. Since $f \in \mathcal{K}^n$, it can only happen if $\mu = 0$ and either

$$\lambda_2(f) > \lambda_1(f) = 0 \quad \text{or} \quad \lambda_2(f) = \lambda_1(f) = 0 \tag{4.13}$$

by Lemma 4.1. The second subcase in (4.13) reduces to Case a. In the first subcase, we have $f = |x| \in \partial\mathcal{K}^n$, the boundary of $\mathcal{K}^n$. Let $\hat{\lambda}_1\hat{u}_1 + \hat{\lambda}_2\hat{u}_2$ be the eigen-decomposition of $\hat{x} := x + h$. Then $\mathcal{L}_{\hat{f}} = Q\hat{\mathcal{F}}Q^T$, where

$$\hat{\mathcal{F}} = \operatorname{diag}\left(\sqrt{\hat{\lambda}_2^2+\varepsilon^2}, \sqrt{\hat{\lambda}_1^2+\varepsilon^2}, \frac{\sqrt{\hat{\lambda}_2^2+\varepsilon^2}+\sqrt{\hat{\lambda}_1^2+\varepsilon^2}}{2}, \cdots, \frac{\sqrt{\hat{\lambda}_2^2+\varepsilon^2}+\sqrt{\hat{\lambda}_1^2+\varepsilon^2}}{2}\right) \tag{4.14}$$

and Q is an orthonormal matrix with $Q_{n\times(n-2)} \in \mathcal{R}^{n\times(n-2)}$:

$$Q = \left[\sqrt{2}\hat{u}_2, \sqrt{2}\hat{u}_1, Q_{n\times(n-2)}\right]. \tag{4.15}$$

Therefore, when $\mu = 0$ and $x \in \partial\mathcal{K}^n$, the right hand side of (4.12) is

$$\begin{array}{rl} & \mathcal{L}_{\hat{f}}^{-1}\left(x\circ(\hat{x}-\hat{f})\right) \\ = & \left[\frac{2\hat{u}_2\hat{u}_2^T}{\sqrt{\hat{\lambda}_2^2+\varepsilon^2}} + \frac{2\hat{u}_1\hat{u}_1^T}{\sqrt{\hat{\lambda}_1^2+\varepsilon^2}} + \frac{2Q_{n\times(n-2)}Q_{n\times(n-2)}^T}{\sqrt{\hat{\lambda}_2^2+\varepsilon^2}+\sqrt{\hat{\lambda}_1^2+\varepsilon^2}}\right]\left[x\circ\left((\hat{\lambda}_2-\sqrt{\hat{\lambda}_2^2+\varepsilon^2})\hat{u}_2 + (\hat{\lambda}_1-\sqrt{\hat{\lambda}_1^2+\varepsilon^2})\hat{u}_1\right)\right] \\ = & \frac{2x^T\hat{u}_2(\hat{\lambda}_2-\sqrt{\hat{\lambda}_2^2+\varepsilon^2})}{\sqrt{\hat{\lambda}_2^2+\varepsilon^2}}\hat{u}_2 + \frac{2x^T\hat{u}_1(\hat{\lambda}_1-\sqrt{\hat{\lambda}_1^2+\varepsilon^2})}{\sqrt{\hat{\lambda}_1^2+\varepsilon^2}}\hat{u}_1 \\ = & O(\|(h,\varepsilon)\|^2), \end{array} \tag{4.16}$$

where, the first equality is by (4.14), the second equality is by (2.11) and the third equality holds because $x^T\hat{u}_2 \to \lambda_2$ and $\hat{\lambda}_2 \to \lambda_2 > 0$, $\hat{\lambda}_1 = O(\|h\|)$ when $h \to 0$ and

$$\begin{array}{rl} x^T\hat{u}_1 & = (\lambda_1 u_1 + \lambda_2 u_2)^T\hat{u}_2 \\ & = \frac{\lambda_2}{4}(1, -\frac{x_2+h_2}{\|x_2+h_2\|})^T(1, \frac{x_2}{\|x_2\|}) \\ & = \frac{\lambda_2}{4\|x_2\|\|x_2+h_2\|}(\|x_2\|\|x_2+h_2\| - x_2^T(x_2+h_2)) \\ & = \frac{\lambda_2}{8\|x_2\|\|x_2+h_2\|}(\|x_2-(x_2+h_2)\|^2 - (\|x_2\|-\|x_2+h_2\|)^2) \\ & = O(\|h_2\|^2). \end{array} \tag{4.17}$$

Similarly, when $\mu = 0$ and $x \in -\partial\mathcal{K}^n$, the right hand side of (4.12) is of order $O(\|(h,\varepsilon)\|^2)$.

Overall, we have proved that for all $(x+h, \mu+\varepsilon) \in D_f$ and $(h,\varepsilon) \to 0$,

$$f(x+h,\mu+\varepsilon) - f(x,\mu) - f'(x+h,\mu+\varepsilon)\begin{pmatrix} h \\ \varepsilon \end{pmatrix} = O(\|(h,\varepsilon)\|^2). \tag{4.18}$$

This completes the proof. □

As a simple consequence by fixing $\mu = 0$, we have

Proposition 4.3 *The SOC functions $[x]_+$ and $|x|$ are strongly semismooth at $x \in \mathcal{R}^n$.*

5 A squared smoothing Newton method

Given a continuously differentiable function $F : \mathcal{R}^n \to \mathcal{R}^n$, in order to solve the SOCCP, let $z = (x, \varepsilon) \in \mathcal{R}^n \times \mathcal{R}$ and define

$$H(z) = \begin{pmatrix} \Phi(x,\varepsilon) \\ \varepsilon \end{pmatrix}, \tag{5.1}$$

where $\Phi(x,\varepsilon) = \frac{1}{2}\Big(x + F(x) - \sqrt{(x - F(x))^2 + 4\varepsilon^2 e}\Big)$. We also define the merit function $\psi(z) : \mathcal{R}^n \times \mathcal{R} \to \mathcal{R}_+$ by $\psi(z) = \|H(z)\|^2$ and $\beta : \mathcal{R}^n \times \mathcal{R} \to \mathcal{R}_+$ by

$$\beta(z) = \gamma \min\{1, \psi(z)\} \tag{5.2}$$

in which $\gamma \in (0,1)$. Here, to find the root of (5.1) implies that $\varepsilon = 0$ and $\Phi(x, 0)$ is nothing but our C-function given in Proposition 2.2(d). Therefore, the SOCCP is equivalent to the system $H(z) = 0$.

Algorithm 5.1

Step 0. *Choose constants $\delta \in (0,1)$ and $\sigma \in (0, \frac{1}{2})$. Choose $\varepsilon^0 = \bar{\varepsilon} \in \mathcal{R}_{++} = \{\varepsilon \mid \varepsilon > 0\}$ and $\gamma \in (0,1)$ such that $\gamma\bar{\varepsilon} < 1$. Given an arbitrary $x^0 \in \mathcal{R}^n$, $\bar{z} = z^0 = (x^0, \varepsilon^0)$. Set $k = 0$.*

Step 1. *If $H(z^k) = 0$, stop. Otherwise, let $\beta_k = \beta(z^k)$.*

Step 2. *Compute $\Delta z^k = (\Delta x^k, \Delta\varepsilon^k) \in \mathcal{R}^n \times \mathcal{R}$ by*

$$H(z^k) + H'(z^k)\Delta z^k = \beta_k \bar{z}. \tag{5.3}$$

Step 3. *Let l_k be the smallest nonnegative integer l satisfying*

$$\psi(z^k + \delta^l \Delta z^k) \le [1 - 2\sigma(1 - \gamma\bar{\varepsilon})\delta^l]\psi(z^k). \tag{5.4}$$

Set $z^{k+1} = z^k + \delta^{l_k}\Delta z^k$.

Step 4. *Set $k = k + 1$ and go to step 1.*

Theorem 5.2 *Suppose that $H'(z)$ is nonsingular for all $z = (x, \varepsilon)$ with $\varepsilon \neq 0$. Then Algorithm 5.1 is well defined and any accumulation point z^* of the sequence $\{z^k\}$ generated by this algorithm is a solution of $H(z) = 0$. Moreover, if all $V \in \partial H(z^*)$ are nonsingular at some $z^* = (x^*, 0)$, where ∂H stands for the generalized Jacobian of H in the sense of Clarke [2], and F' is Lipschitz continuous around x^*, then the whole sequence $\{z^k\}$ converge to z^* Q-quadratically.*

The proof of the above theorem is similar to that in [15] and [16], and we omit it. In addition, we have shown in [18] for a more general setting that the quadratic convergence will occur if F' is Lipschitz continuous around x^* and $F'(x^*)$ is positive definite on the affine hull of the critical cone of F at x^*. As a special case, the whole sequence $\{z^k\}$ converge to z^* Q-quadratically if F' is Lipschitz continuous around x^* and F is strongly monotone.

6 Numerical experiments

In our numerical experiments, we test the second order cone programs of the following form:

$$\begin{array}{ll} \min & \sum_{i=1}^{p} c_i^T x_i \\ \text{s.t.} & \sum_{i=1}^{p} A_i x_i = b \\ & x_i \in \mathcal{K}^{n_i}, i = 1, \cdots, p, \end{array} \tag{6.1}$$

where $c_i, x_i \in \mathcal{R}^{n_i}$, $b \in \mathcal{R}^m$ and $A_i \in \mathcal{R}^{m \times n_i}$. Let $y \in \mathcal{R}^m$ be the dual variable of the equality constraint. The KKT system of the above problem is

$$\begin{array}{rl} \sum_{i=1}^{p} A_i x_i = b, & x_i^T (c_i - A_i^T y) = 0, \quad i = 1, \cdots, p, \\ x_i \in \mathcal{K}^{n_i}, & c_i - A_i^T y \in \mathcal{K}^{n_i}, \quad i = 1, \cdots, p. \end{array} \tag{6.2}$$

Let $N = \sum_{i=1}^{p} n_i$, $c = (c_1^T, \cdots, c_p^T)^T$, $x = (x_1^T, \cdots, x_p^T)^T$ and $A = (A_1^T, \cdots, A_p^T)^T$. Then to solve the above KKT system is equivalent to find a root of

$$H \begin{pmatrix} y \\ x \\ \varepsilon \end{pmatrix} = \begin{pmatrix} b - Ax \\ x + c - A^T y - \sqrt{(x - c + A^T y)^2 + 4\varepsilon^2 e} \\ \varepsilon \end{pmatrix} = 0. \tag{6.3}$$

Note that all the vector operations are those on the SOC, see definitions in section 2. We note that Huang et al. [9] recently proposed a revised version of Algorithm 5.1 for solving linear complementarity problems that assures the superlinear convergence under either the nonsingularity of all generalized Jacobian or the strict complementarity condition at the solution. We applied their techniques to the SOCCP in our numerical implementation.

We use randomly generated problems in which the involved matrices are not sparse. The test problems are generated with sizes $N(=2m)$ from 100 to 800 with each $n_i = 5$. (By our numerical observation, other n_i leads to similar numerical results. We only report the result of $n_i = 5$ here.) The random problems of each size are generated 10 times, and thus we have totally 80 random problems. The problems are generated by choosing vectors b and c so that they are feasible and their optimal values are obtainable. In detail, we generate a random matrix A and a random vector x in the SOC which gives a right hand side $b = Ax$ and so the SOCCP is feasible. Moreover, we generate a random vector c in the SOC so the optimal value of the SOCCP is obtainable. There are different types of initial points. One is to set variables x to be $0.2\mathbf{e}$, $0.5\mathbf{e}$ and $1.0\mathbf{e}$ and y to be the zero vector, where $\mathbf{e}$ is the unit vector in the feasible cone. The other is to set x and y randomly with x in the interior of the feasible cone. By using these different types of initializations, one can show that the initial points play more important roles in the interior point methods than they do in the smoothing Newton methods. The algorithm is coded in the similar style as SDPT3 used by Toh, Todd and Tütüncü [19]. Although there are some differences between the interior-point method and the smoothing Newton method, the kernel of both algorithms are similar. Both require to solve a Newton equation obtained from a KKT system at each iteration. We also adopt the "primal feasibility projection" used in [10], i.e., if

$$\|A(x + \Delta x) - b\| > \|Ax - b\|, \tag{6.4}$$

we replace Δx by its orthogonal projection onto the null space $\{u \mid Au = 0\}$. We use

$$\|H(z)\| \leq \texttt{restol} \tag{6.5}$$

as our stopping criterion, where the residual tolerance $\texttt{restol} = 10^{-6}$ and $z = (y, x, \varepsilon)$. Also we set the minimal step length and maximal iteration number as $\texttt{steptol} = 10^{-6}$ and $\texttt{maxit} = 100$. Other parameters used in Algorithm 5.1 are as follows: $\sigma = 0.35$, $\delta = 0.95$, $\gamma = 0.2$, and $\bar{\varepsilon} = 1.0$. Since we adopt the techniques in [9], the additional parameters used are: $t = 0.2$, $\kappa = 2.0$ and $\tau = 0.1$. We reduce γ and τ by half until they satisfy the initial condition:

$$\gamma \|H(z^0)\| + \tau\sqrt{n} < 1, \tag{6.6}$$

in which z^0 is the initial z. See [9] for more details.

The numerical tests are done on a HP workstation installed Windows 2000 and the source code is written in MATLAB 6.1.

Figure 6.1 The logarithm of residual norm $\|H(z)\|$ by iterations

Figure 6.1 shows the convergence behavior of one of the largest test problems, i.e., $N = 800$ and $m = 400$, with initial points $x = 1.0\mathbf{e}$ and $y = 0$. Tables 6.2—6.5 show the average iteration numbers (iter) and average CPU (cpu) time in second for 10 test problems of each size, for the different initializations respectively. The computational results are certainly preliminary. However, they show that the smoothing Newton method could in general perform very efficiently in practice.

The results in Tables 6.2—6.5 show that our algorithm takes more cpu times than the SDPT3 does, while less iteration numbers are needed. However, we stress that the convergence criteria of these two algorithms are different. We also point out that there is room for improvement of the code of Algorithm 5.1 by refining the costly matrix computations as in SDPT3 [19].

		Algorithm5.1			SDPT3	
m	N	iter	cpu(s)		iter	cpu(s)
50	100	8.7	0.420		19.2	0.310
100	200	7.9	1.147		19.0	0.922
150	300	7.9	2.700		18.0	2.356
200	400	7.8	5.536		18.0	4.843
250	500	8.1	11.214		19.0	9.548
300	600	7.8	17.077		17.0	13.391
350	700	8.1	29.381	*	21.0	25.523
400	800	8.0	39.435	*	20.0	35.159

Table 6.2 Average Performances of Algorithm 5.1 and SDPT3 for 10 problems with initial point $x = 0.2\mathbf{e}$ and $y = 0$. (Through our numerical report, * stands for that one random problem fails to converge for too short steps. The average is based on the successful instances.)

		Algorithm5.1			SDPT3	
m	N	iter	cpu(s)		iter	cpu(s)
50	100	7.8	0.333		13.2	0.218
100	200	7.5	0.987		13.0	0.640
150	300	7.7	2.383		12.0	1.611
200	400	7.9	5.358		14.0	3.901
250	500	8.5	11.580		13.0	6.709
300	600	8.9	19.304		14.0	11.106
350	700	8.1	28.864	*	14.0	17.272
400	800	8.5	41.779		13.0	23.140

Table 6.3 The same as Table 6.2 except the initial points $x = 0.5\mathbf{e}$.

		Algorithm5.1			SDPT3	
m	N	iter	cpu(s)		iter	cpu(s)
50	100	8.2	0.311		10.1	0.176
100	200	8.1	0.986		10.0	0.506
150	300	8.7	2.642		10.0	1.353
200	400	9.2	6.233		10.0	2.880
250	500	9.2	12.508		10.0	5.203
300	600	10.5	23.639		10.0	8.147
350	700	10.1	38.437	*	11.0	13.859
400	800	10.0	50.803	*	10.0	18.186

Table 6.4 The same as Table 6.2 except the initial points $x = 1.0\mathbf{e}$.

		Algorithm5.1		SDPT3	
m	N	iter	cpu(s)	iter	cpu(s)
50	100	8.9	0.312	10.1	0.206
100	200	9.0	1.094	10.0	0.502
150	300	9.2	2.958	11.0	1.485
200	400	9.0	6.409	10.0	2.853
250	500	8.9	13.070	11.0	5.665
300	600	9.1	19.260	10.0	8.131
350	700	8.9	33.107	11.0	13.823
400	800	8.8	44.847	11.0	39.879

Table 6.5 The same as Table 6.2 except the initial points are randomly set.

We also did some tests for the penalized C-function ϕ_p. By using

$$\Phi(x,\varepsilon) = \frac{1}{2}(x + F(x) - \sqrt{(x - F(x))^2 + 4\varepsilon^2 e} + \theta[x]_+ \circ [F(x)]_+), \tag{6.7}$$

we have the penalized smoothing Newton algorithm for the SOCCP. The parameter $\theta \in [0,1]$. If $\theta = 0$, then we go back to the squared smoothing method in (5.1), whereas $\theta > 0$, we have the penalized natural squared smoothing method. As stated in Example 3.4, the algorithm may generate poor search directions for the merit function. We observed that the smoothing Newton algorithm behaves better when θ is close to 0. To show the effects of different penalized parameters, we tested the problems of each size with initial $x = 0.2\mathbf{e}$ and $y = 0$, and report the results in Table 6.6. (One can compare these results with those in Table 6.2). Note that the Jacobians are singular in some cases even with small parameter θ. With these parameters, the algorithm generates ill conditioned Jacobians at some iterates, and fails due to too short steps. It is a future topic of research whether one can have a "scaled" penalized natural C-function with nonsingular Jacobians and bounded level sets.

θ		0.02		0.04		0.06		0.08		0.10	
m	N	iter	cpu(s)	iter	cpu(s)	iter	cpu(s)	iter	cpu(s)	iter	cpu(s)
50	100	10	1.016	12	1.156	15	1.344	18	1.531	22	1.813
100	200	10	3.297	12	3.531	15	4.625	19	5.500	26	7.735
150	300	10	7.438	13	9.750	16	10.843	21	15.672	29	23.265
200	400	10	14.156	13	20.000	17	27.578	21	30.781	29	56.718
250	500	11	30.125	13	—	21	75.578	29	—	49	—
300	600	10	39.578	13	59.640	17	87.452	21	92.593	27	129.248
350	700	11	—	14	93.655	18	132.826	24	191.997	36	324.606
400	800	12	—	15	170.185	17	—	21	218.997	5	—

Table 6.6 Numerical results for penalized C-function
(— stands for "Jacobian Singular at a certain iterate")

Acknowledgment: Many thanks to Professor K.-C. Toh for his helpful discussion on the numerical implementation.

References

[1] B. Chen, X. Chen and C. Kanzow, *A penalized Fischer-Burmeister NCP-function*, Math. Prog., **88** (2000), 211–216.

[2] F.H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[3] J.-P. Crouzeix, *Pseudomonotone variational inequality problems: existence of solutions*, Math. Prog., **78** (1997), 305–314.

[4] J. Faraut and A. Koranyi, *Analysis on symmetric cones*, Clarendon Press, Oxford, 1994.

[5] L. Faybusovich, *Linear systems in Jordan algebras and primal-dual interior-point algorithms*, Journal of Comp. and Appl. Math., **86** (1997), 149–175.

[6] M.C. Ferris and C. Kanzow, *Complementarity and related problems: A survey*, In P.M. Pardalos and M.G.C. Resende (eds.): Handbook of Applied Optimization, 514–530, Oxford University Press, New York, 2002.

[7] M. Fukushima, Z.Q. Luo and P. Tseng, *Smoothing functions for second-order-cone complementarity problems*, SIAM Journal on Optimization, **12** (2002), 436–460.

[8] Y.R. He and K.F. Ng, *Characterize the boundedness of solution set of generalized complementarity problems*, Math. Prog., to appear.

[9] Z. Huang, L. Qi and D. Sun, *Sub-quadratic convergence of a smoothing Newton algorithm for the P_0- and monotone LCP*, `http://www.math.nus.edu.sg/~matsundf/`.

[10] C. Kanzow and C. Nagel, *Semidefinite programs: new search directions, smoothing-type methods, and numerical results*, `http://ifamus.mathematik.uni-wuerzburg.de/~kanzow/`.

[11] M.S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, *Applications of second-order cone programming*, Linear Alg. Appl., **284** (1998), 193–228.

[12] L. McLinden, *Stable monotone variational inequalities*, Math. Prog., **48** (1990), 303–338.

[13] J.S. Pang, *Complementarity problems*, In R. Horst and P. Pardalos, (eds.): Handbook in Global Optimization, Kluwer Academic Publishers, Boston, 1994.

[14] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Math. Prog., **58** (1993), 353–367.

[15] L. Qi, D. Sun and G. Zhou, *A new look at smoothing Newton methods for nonlinear complementarity problems and box constrained variational inequalities*, Math. Prog., **87** (2001), 1–35.

[16] D. Sun and L. Qi, *Solving variational inequality problems via smoothing-nonsmooth reformulations*, J. Comput. Appl. Math., **129** (2001), 37–62.

[17] D. Sun and J. Sun, *Semismooth matrix valued functions*, Math. Oper. Res., **27** (2002), 150–169.

[18] J. Sun, D. Sun and L. Qi, *Quadratic convergence of a squared smoothing Newton method for nonsmooth matrix equations and its applications in semidefinite optimization problems*, Technical Report, Department of Decision Sciences, National University of Singapore (2002), `http://www.fba.nus.edu.sg/depart/ds/sunjiehomepage/`.

[19] K.C. Toh, M.J. Todd and R.H. Tütüncü, *SDPT3— a MatLab software package for semidefinite programming, version 2.1*, Optimization Methods and Software, **11** (1999), 545–581.