MATHEMATICS OF COMPUTATION
Volume 83, Number 288, July 2014, Pages 1701–1726
S 0025-5718(2013)02795-0
Article electronically published on November 7, 2013

AN EFFICIENT ALGORITHM FOR SECOND-ORDER CONE LINEAR COMPLEMENTARITY PROBLEMS

LEI-HONG ZHANG AND WEI HONG YANG

Abstract. Recently, the globally uniquely solvable (GUS) property of the linear transformation $M \in \mathbb{R}^{n\times n}$ in the second-order cone linear complementarity problem (SOCLCP) receives much attention and has been studied substantially. Yang and Yuan contributed a new characterization of the GUS property of the linear transformation, which is formulated by basic linear-algebra-related properties. In this paper, we consider efficient numerical algorithms to solve the SOCLCP where the linear transformation M has the GUS property. By closely relying on the new characterization of the GUS property, a globally convergent bisection method is developed in which each iteration can be implemented using only $2n^2$ flops. Moreover, we also propose an efficient Newton method to accelerate the bisection algorithm. An attractive feature of this Newton method is that each iteration only requires $5n^2$ flops and converges quadratically. These two approaches make good use of the special structure contained in the SOCLCP and can be effectively combined to yield a fast and efficient bisection-Newton method. Numerical testing is carried out and very encouraging computational experiments are reported.

1. Introduction

For decades, there has been considerable discussion about the linear complementarity problem (LCP) which is to find a vector $\mathbf{x} \in \mathbb{R}^n$ such that

$$LCP(M,\mathbf{q}): \quad \mathbf{x} \geq \mathbf{0},\ \mathbf{q} + M\mathbf{x} \geq \mathbf{0},\ \mathbf{x}^\top(\mathbf{q} + M\mathbf{x}) = 0, \tag{1.1}$$

where $M \in \mathbb{R}^{n\times n}$ and $\mathbf{q} \in \mathbb{R}^n$ are both given. Up to date, substantial theoretical results and various computational methods for the $LCP(M,\mathbf{q})$ have been established and we refer, e.g., to [2, 4, 7, 18, 19] and the references therein, for comprehensive discussions. Recently, considerable efforts have been made to extend the $LCP(M,\mathbf{q})$, which is essentially the LCP over the cone $\mathbb{R}^n_+$, to symmetric cones, especially to the positive semidefinite cone as well as the second-order cone (SOC); see e.g. [1, 5, 6, 8, 10, 12–14, 20, 21, 26, 31]. The second-order cone, also known as the Lorentz cone, is defined by

$$\mathcal{K}^n := \{[x_1, \mathbf{x}_2^\top]^\top \in \mathbb{R}\times\mathbb{R}^{n-1} \mid \|\mathbf{x}_2\|_2 \leq x_1\}. \tag{1.2}$$

Received by the editor June 30, 2010 and, in revised form, September 1, 2011 and October 30, 2012.

2010 *Mathematics Subject Classification.* Primary 90C33, 65K05; Secondary 65F99.

Key words and phrases. Second-order cone, Linear complementarity problem, globally uniquely solvable property, bisection iteration, Newton's iteration.

The research of the first author was supported by the Natural Science Foundation of China, NSFC-11101257.

The research of the second author was supported by the Natural Science Foundation of China, NSFC-10801040.

Therefore, for a given $M \in \mathbb{R}^{n\times n}$ and a vector $\mathbf{q} \in \mathbb{R}^n$, the LCP over SOC (SOCLCP for short) can be calculated to find that $\mathbf{x} \in \mathbb{R}^n$ such that

$$\mathbf{x} \in \mathcal{K}^n,\ \mathbf{q} + M\mathbf{x} \in \mathcal{K}^n,\ \mathbf{x}^\top(\mathbf{q} + M\mathbf{x}) = 0. \tag{1.3}$$

To simplify our presentation, we will denote (1.3) as $LCP(M, \mathcal{K}^n, \mathbf{q})$ for which the set of all solutions will also be denoted by $SOL(M, \mathcal{K}^n, \mathbf{q})$. The SOCLCP arises from many areas and we refer, e.g., to [1, 5, 6, 8, 14] for various applications.

A very important question in an LCP is how to characterize the property of M such that the solution of $LCP(M, \mathcal{K}^n, \mathbf{q})$ is unique for all vectors $\mathbf{q} \in \mathbb{R}^n$, is of great interest. If such a property exists, we will say that M has the globally uniquely solvable (GUS) property. For $LCP(M, \mathbf{q})$ (1.1), this question has been completely answered by a set of basic linear-algebra-related properties [7]. For $LCP(M, \mathcal{K}^n, \mathbf{q})$, on the other hand, Gowda *et al.* inspired this interesting work and studied the P-property, the cross commutative property and the GUS property intensively (see [11–13]). Their discussion is based on the Euclidean Jordan Algebra and their characterization of the GUS property of M turns out to be difficult to verify. Most recently, [30] has made a contribution along this line by providing basic linear-algebra-related properties of M (see also Theorem 2.2 in Section 2). A key for this new characterization is that whenever $\mathbf{q} \notin -M\mathcal{K}^n \cup \mathcal{K}^n$, finding $\mathbf{x} \in SOL(M, \mathcal{K}^n, \mathbf{q})$ is equivalent to solving the pair $(\mathbf{x}, s^*) \in bd(\mathcal{K}^n) \times \mathbb{R}_{++}$ such that

$$(M - s^* J_n)\mathbf{x} = -\mathbf{q}, \tag{1.4}$$

where $bd(\mathcal{K}^n)$ stands for the boundary of $\mathcal{K}^n$ and $J_n = \mathrm{diag}\{1, -1, \cdots, -1\} = \mathrm{diag}\{1, -I_{n-1}\}$. We point out that this equivalence is not only crucial in characterizing the GUS property of M, but also very helpful in designing an efficient algorithm for solving $LCP(M, \mathcal{K}^n, \mathbf{q})$.

Computationally, there have been a number of methods used for solving $LCP(M, \mathcal{K}^n, \mathbf{q})$. They include, for example, the smoothing Newton method [5, 8, 17], the smoothing-regularization method [14], the merit-function-based approaches [3, 6], the semismooth Newton method [25] and interior-point methods [20, 29]. It should be noted that these approaches are proposed to solve the general second-order cone complementarity problem (SOCCP) which aims at $\mathbf{x} \in \mathbb{R}^n$ such that

$$\mathbf{x} \in \mathcal{K},\ G(\mathbf{x}) \in \mathcal{K} \quad \text{and} \quad \mathbf{x}^\top G(\mathbf{x}) = 0, \tag{1.5}$$

where $G : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable mapping and $\mathcal{K} = \mathcal{K}^{n_1} \times \mathcal{K}^{n_2} \times \cdots \times \mathcal{K}^{n_m}$ with $n_i \geq 1$ and $\sum_{i=1}^m n_i = n$. As these methods target a larger class of SOCCP, fewer special properties can be exploited. On the other hand, if we are particularly interested in the solution of the SOCLCP (1.3), it is hoped that a very efficient algorithm can be designed if some specific properties can be effectively exploited. In particular, if M has the GUS property, one can expect that the basic linear-algebra-related properties of M in [30] might contain essential information for numerically solving the SOCLCP (1.3). An efficient approach for the SOCLCP (1.3) is not only of value in its own right, but also could be used to solve the general problem (1.5) with $G(\mathbf{x}) = M\mathbf{x} + \mathbf{q}$ (see Section 6). This is our motivation for this paper.

In this paper, we focus on the numerical solution of $LCP(M, \mathcal{K}^n, \mathbf{q})$ in which M has the GUS property. We will investigate the SOCLCP from a totally different perspective from various methods used for the SOCCP (1.5) in the literature. In

particular, by closely relying on the basic linear-algebra-related properties for the GUS property of M, we propose a bisection iteration for the SOCLCP which turns out to be globally convergent. An attractive feature of this method is that each iteration can be implemented via only $2n^2$ flops. Moreover, we will also show that the bisection iteration can be accelerated by an efficient Newton method, which can be implemented using only $5n^2$ flops per iteration and converges quadratically. These two approaches effectively make use of the special structure contained in the SOCLCP, and they can be perfectly combined further to yield an efficient bisection-Newton method. Numerical testing is carried out and very encouraging computational experiments are reported.

This paper is organized in the following way: In Section 2, we will first describe our notations and introduce some preliminary results of the SOCLCP when M has the GUS property. In Section 3, we will propose a bisection iteration and describe how each iteration can be implemented in a very efficient way. A special case needs to be addressed additionally for the bisection method, and we will propose an efficient and direct algorithm to handle this special case. In Section 4, a Newton method is introduced. We will provide an efficient implementation of this Newton method so that each iteration only requires $5n^2$ flops. Moreover, we will show how the bisection iteration and the Newton iteration can be effectively combined to yield a fast and efficient algorithm. Numerical testing is carried out and very encouraging computational experiments are reported in Section 5. Finally, concluding remarks are drawn and some future research topics are briefly mentioned in Section 6.

2. Notation and preliminary results

In this section, we introduce some basic notations and present preliminary results for the SOCLCP. Throughout the paper, all vectors are column vectors and are typeset in bold. For a matrix $A \in \mathbb{R}^{n\times m}$, $A^\top$ denotes its transpose, and $\mathcal{R}(A) := \{\mathbf{x} \in \mathbb{R}^n | \mathbf{x} = A\mathbf{y} \text{ for some } \mathbf{y} \in \mathbb{R}^m\}$ and $Ker(A) := \{\mathbf{y} \in \mathbb{R}^m | A\mathbf{y} = \mathbf{0}\}$ stands for the range and kernel of A, respectively. Thus $\mathcal{R}(A)^\perp = Ker(A^\top)$, where $\mathcal{R}(A)^\perp$ denotes the orthogonal complement of $\mathcal{R}(A)$. As usual, the identity matrix in $\mathbb{R}^{n\times n}$ will be denoted by I_n and the norm of A is defined by $\|A\|_2 := \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2$.

For a set $\mathcal{C} \subset \mathbb{R}^n$, we denote the boundary and the interior of $\mathcal{C}$ by $bd(\mathcal{C})$ and $int(\mathcal{C})$, respectively. If $\mathbf{x} \in \mathcal{C}$, then the normal cone (see e.g. [16], Definition 5.2.3) of $\mathcal{C}$ at $\mathbf{x}$ is defined by

$$N_{\mathcal{C}}(\mathbf{x}) := \{\mathbf{z} | \mathbf{z}^\top(\mathbf{y} - \mathbf{x}) \le 0,\ \forall \mathbf{y} \in \mathcal{C}\},$$

and therefore, if $\mathbf{y} \in int(\mathcal{C})$, the relation

$$\mathbf{z}^\top(\mathbf{y} - \mathbf{x}) < 0, \text{ for all nonzero } \mathbf{z} \in N_{\mathcal{C}}(\mathbf{x}) \tag{2.1}$$

is evident.

Related to the SOCLCP, for the matrix $M \in \mathbb{R}^{n\times n}$ and a given $s \ge 0$, we denote

$$M_s := M - sJ_n, \quad \text{where } J_n = \text{diag}\{1, -1, \cdots, -1\} = \text{diag}\{1, -I_{n-1}\},$$

and define

$$K_s := M_s\mathcal{K}^n = \{M_s\mathbf{x} | \mathbf{x} \in \mathcal{K}^n\}.$$

Obviously, K_s is a cone and $K_0 = M\mathcal{K}^n$. Moreover, Lemma 2.1 lists two straightforward properties related to $\mathcal{K}^n$ which will be frequently used in this paper.

Lemma 2.1. *For the second-order cone $\mathcal{K}^n$ and any nonzero $\mathbf{y}, \mathbf{z} \in \mathbb{R}^n$, we have*

(i) if $\mathbf{y} \in \mathcal{K}^n$, then $J_n\mathbf{y} \in \mathcal{K}^n$, and

(ii) $\mathbf{y}^\top\mathbf{z} \geq 0$, $\forall \mathbf{y}, \mathbf{z} \in \mathcal{K}^n$, and the inequality is strict if $\mathbf{y} \in int(\mathcal{K}^n)$ or $\mathbf{z} \in int(\mathcal{K}^n)$.

As we have pointed out, M has the GUS property [13] if for all $\mathbf{q} \in \mathbb{R}^n$, $LCP(M, \mathcal{K}^n, \mathbf{q})$ has a unique solution. A set of simple linear-algebra-related conditions [30] completely characterizes the GUS property. We restate these sufficient and necessary conditions in Theorem 2.2, because they not only answer the global and unique solvability of the SOCLCP but motive us to design efficient numerical algorithms.

Theorem 2.2. *For $LCP(M, \mathcal{K}^n, \mathbf{q})$, M has the GUS property if and only if it satisfies the following assumptions:*

(i) MJ_n has nonnegative eigenvalues and there exists a $\tau > 0$ such that all nonnegative eigenvalues of MJ_n are equal to τ. Moreover, $\mathrm{rank}(MJ_n - \tau I_n) = n - 1$. There exists a $\mathbf{w} \in int(\mathcal{K}^n)$ such that $\mathbf{w}$ is the eigenvector of MJ_n associated with τ;

(ii) There exists a $\mathbf{v} \in int(\mathcal{K}^n)$ such that $\mathbf{v}$ is the eigenvector of $M^\top J_n$ associated with τ;

(iii)

$$\mathbf{a}^\top M\mathbf{a} \geq 0, \quad \forall \mathbf{a} \in bd(\mathcal{K}^n);$$

(iv)

$$\mathbf{a}^\top M^{-1}\mathbf{a} \geq 0, \quad \forall \mathbf{a} \in bd(\mathcal{K}^n).$$

It has been proved previously by Gowda *et al.* ([13], Theorem 17) that any positive definite (not necessarily symmetric) matrix M has the GUS property and therefore the sufficient and necessary conditions cover all positive definite matrices. On the other hand, a concrete indefinite matrix M with the GUS property has been given in [30]. This shows that the set of the $LCP(M, \mathcal{K}^n, \mathbf{q})$ where M has the GUS property is strictly larger than the set of strongly monotone SOCLCPs. In addition, when M is positive definite or symmetric, more can be said about the eigenvalues of $M^\top J_n$ and MJ_n as we will see in Theorem 2.3.

Theorem 2.3. *Let $\theta + i\phi$ (i denotes the imaginary unit) be an arbitrary eigenvalue of $M^\top J_n$ (or MJ_n), then the following statements hold:*

(i) if M is positive definite (not necessarily symmetric) and $\theta > 0$, then $\theta = \tau$ and $\phi = 0$. That is, τ is the unique eigenvalue of $M^\top J_n$ and MJ_n with positive real part;

(ii) if M has the GUS property and is symmetric, then $\phi = 0$. That is, all eigenvalues of $M^\top J_n$ are real and τ is the unique positive eigenvalue.

Proof. Let $\mathbf{y} + i\mathbf{z}$ be the eigenvector of $M^\top J_n$ corresponding to the eigenvalue $\theta + i\phi$ where $\mathbf{y}, \mathbf{z} \in \mathbb{R}^n$, and $\mathbf{y} \neq \mathbf{0}$ or $\mathbf{z} \neq \mathbf{0}$. From

$$M^\top J_n(\mathbf{y} + i\mathbf{z}) = (\theta + i\phi)(\mathbf{y} + i\mathbf{z}),$$

it follows that

$$M^\top J_n\mathbf{y} = \theta\mathbf{y} - \phi\mathbf{z}, \tag{2.2}$$

$$M^\top J_n\mathbf{z} = \theta\mathbf{z} + \phi\mathbf{y}. \tag{2.3}$$

Let $\mathbf{w} \in int(\mathcal{K}^n)$ be the eigenvector of MJ_n associated with τ. Pre-multiplying (2.2) and (2.3) by $\mathbf{w}^\top J_n$ leads respectively to

$$\mathbf{w}^\top J_n M^\top J_n \mathbf{y} = \tau \mathbf{w}^\top J_n \mathbf{y} = \theta \mathbf{w}^\top J_n \mathbf{y} - \phi \mathbf{w}^\top J_n \mathbf{z}, \tag{2.4}$$

$$\mathbf{w}^\top J_n M^\top J_n \mathbf{z} = \tau \mathbf{w}^\top J_n \mathbf{z} = \phi \mathbf{w}^\top J_n \mathbf{y} + \theta \mathbf{w}^\top J_n \mathbf{z}. \tag{2.5}$$

Note that (2.4) and (2.5) can be rewritten as

$$(\tau - \theta)\mathbf{w}^\top J_n \mathbf{y} = -\phi \mathbf{w}^\top J_n \mathbf{z}, \tag{2.6}$$

$$(\tau - \theta)\mathbf{w}^\top J_n \mathbf{z} = \phi \mathbf{w}^\top J_n \mathbf{y}, \tag{2.7}$$

and hence one has

$$(\tau - \theta)^2 (\mathbf{w}^\top J_n \mathbf{y})(\mathbf{w}^\top J_n \mathbf{z}) = -\phi^2 (\mathbf{w}^\top J_n \mathbf{z})(\mathbf{w}^\top J_n \mathbf{y}). \tag{2.8}$$

To prove (i), we first suppose $\phi \neq 0$. Pre-multiplying $\mathbf{y}^\top J_n$ on both sides of (2.2) yields

$$\mathbf{y}^\top J_n M^\top J_n \mathbf{y} = \theta \mathbf{y}^\top J_n \mathbf{y} - \phi \mathbf{y}^\top J_n \mathbf{z}, \tag{2.9}$$

and pre-multiplying $\mathbf{z}^\top J_n$ on both sides of (2.3) gives

$$\mathbf{z}^\top J_n M^\top J_n \mathbf{z} = \theta \mathbf{z}^\top J_n \mathbf{z} + \phi \mathbf{z}^\top J_n \mathbf{y}. \tag{2.10}$$

Adding (2.9) and (2.10), and noting the positive definiteness of M, one has

$$0 < \mathbf{z}^\top J_n M^\top J_n \mathbf{z} + \mathbf{y}^\top J_n M^\top J_n \mathbf{y} = \theta(\mathbf{y}^\top J_n \mathbf{y} + \mathbf{z}^\top J_n \mathbf{z}).$$

Since $\theta > 0$, we have $\mathbf{y}^\top J_n \mathbf{y} + \mathbf{z}^\top J_n \mathbf{z} > 0$, which implies $\mathbf{y} \in int(\mathcal{K}^n)$ or $\mathbf{z} \in int(\mathcal{K}^n)$.[1] If $\mathbf{y} \in int(\mathcal{K}^n)$, then by Lemma 2.1, it follows that $\mathbf{w}^\top J_n \mathbf{y} > 0$, and by (2.8), we have

$$(\tau - \theta)^2 (\mathbf{w}^\top J_n \mathbf{z}) = -\phi^2 (\mathbf{w}^\top J_n \mathbf{z}),$$

which implies $\mathbf{w}^\top J_n \mathbf{z} = 0$. Thus from (2.7) we have $\phi \mathbf{w}^\top J_n \mathbf{y} = 0$ which contradicts $\phi \neq 0$ and $\mathbf{w}^\top J_n \mathbf{y} > 0$. Similarly, if $\mathbf{z} \in int(\mathcal{K}^n)$, then we have $\phi \mathbf{w}^\top J_n \mathbf{z} = 0$ which contradicts $\phi \neq 0$ and $\mathbf{w}^\top J_n \mathbf{z} > 0$. Therefore we conclude that if $\theta > 0$, then $\phi = 0$, and since Theorem 2.2(i) implies that τ is the unique positive eigenvalue of $M^\top J_n$ and MJ_n, we know that $\theta = \tau$ and thus completes the proof for (i).

To prove (ii), we suppose $\phi \neq 0$. Pre-multiplying $\mathbf{z}^\top J_n$ on both sides of (2.2) yields

$$\mathbf{z}^\top J_n M^\top J_n \mathbf{y} = \theta \mathbf{z}^\top J_n \mathbf{y} - \phi \mathbf{z}^\top J_n \mathbf{z}, \tag{2.11}$$

and pre-multiplying $\mathbf{y}^\top J_n$ on both sides of (2.3) gives

$$\mathbf{y}^\top J_n M^\top J_n \mathbf{z} = \theta \mathbf{y}^\top J_n \mathbf{z} + \phi \mathbf{y}^\top J_n \mathbf{y}. \tag{2.12}$$

Therefore, from (2.11), (2.12) and the symmetry of M, it follows that

$$0 = \mathbf{y}^\top J_n M^\top J_n \mathbf{z} - \mathbf{z}^\top J_n M^\top J_n \mathbf{y} = \phi(\mathbf{y}^\top J_n \mathbf{y} + \mathbf{z}^\top J_n \mathbf{z}),$$

which implies $\mathbf{y}^\top J_n \mathbf{y} + \mathbf{z}^\top J_n \mathbf{z} = 0$. This relation implies that either $\mathbf{y}^\top J_n \mathbf{y} = \mathbf{z}^\top J_n \mathbf{z} = 0$ or $(\mathbf{y}^\top J_n \mathbf{y})(\mathbf{z}^\top J_n \mathbf{z}) < 0$. The former leads to $\mathbf{y} \in bd(\mathcal{K}^n)$ or $\mathbf{z} \in bd(\mathcal{K}^n)$,[2] whereas the latter implies $\mathbf{y} \in int(\mathcal{K}^n)$ or $\mathbf{z} \in int(\mathcal{K}^n)$.[3] Therefore, the relation

[1] In the case of $\mathbf{y} \in -int(\mathcal{K}^n)$ or $\mathbf{z} \in -int(\mathcal{K}^n)$, we can simply set $\mathbf{y} = -\mathbf{y}$ and $\mathbf{z} = -\mathbf{z}$ in (2.2) and (2.3) which then guarantees $\mathbf{y} \in int(\mathcal{K}^n)$ or $\mathbf{z} \in int(\mathcal{K}^n)$.

[2] In the case of $\mathbf{y} \in -bd(\mathcal{K}^n)$ or $\mathbf{z} \in -bd(\mathcal{K}^n)$, we can simply set $\mathbf{y} = -\mathbf{y}$ and $\mathbf{z} = -\mathbf{z}$ in (2.2) and (2.3) which then guarantees $\mathbf{y} \in bd(\mathcal{K}^n)$ or $\mathbf{z} \in bd(\mathcal{K}^n)$.

[3] In the case of $\mathbf{y} \in -int(\mathcal{K}^n)$ or $\mathbf{z} \in -int(\mathcal{K}^n)$, we can simply set $\mathbf{y} = -\mathbf{y}$ and $\mathbf{z} = -\mathbf{z}$ in (2.2) and (2.3) which then guarantees $\mathbf{y} \in int(\mathcal{K}^n)$ or $\mathbf{z} \in int(\mathcal{K}^n)$.

$\mathbf{y}^\top J_n\mathbf{y} + \mathbf{z}^\top J_n\mathbf{z} = 0$, with the aid of Lemma 2.1, implies that $\mathbf{w}^\top J_n\mathbf{y} > 0$ or $\mathbf{w}^\top J_n\mathbf{z} > 0$. If $\mathbf{w}^\top J_n\mathbf{y} > 0$, from (2.8), we have

$$(\tau - \theta)^2(\mathbf{w}^\top J_n\mathbf{z}) = -\phi^2(\mathbf{w}^\top J_n\mathbf{z}),$$

which implies $\mathbf{w}^\top J_n\mathbf{z} = 0$. Thus from (2.7) we have $\phi\mathbf{w}^\top J_n\mathbf{y} = 0$ which contradicts $\phi \neq 0$ and $\mathbf{w}^\top J_n\mathbf{y} > 0$. Similarly, if $\mathbf{w}^\top J_n\mathbf{z} > 0$, then we have $\phi\mathbf{w}^\top J_n\mathbf{z} = 0$ which contradicts $\phi \neq 0$ and $\mathbf{w}^\top J_n\mathbf{z} > 0$. Therefore we conclude $\phi = 0$ and by Theorem 2.2(i), we complete the proof for (ii). □

Besides the conclusions of the eigenvalues of $M^\top J_n$ and MJ_n for the types of matrices described in Theorem 2.3, there is another interesting result concerning the eigenvalue of $M^{-\top}J_n$ and $M^{-1}J_n$ that is revealed by Theorem 2.2.

Theorem 2.4. *Suppose that M has the GUS property. Let $\mathbf{v} \in int(\mathcal{K}^n)$ and $\mathbf{w} \in int(\mathcal{K}^n)$ be the eigenvectors of $M^\top J_n$ and MJ_n associated with the unique positive eigenvalue of τ respectively. Then M^{-1} also has the GUS property and $\frac{1}{\tau}$ is the unique positive eigenvalue of $M^{-\top}J_n$ and $M^{-1}J_n$; moreover, $J_n\mathbf{v}$ and $J_n\mathbf{w}$ are the eigenvectors of $M^{-\top}J_n$ and $M^{-1}J_n$ associated with $\frac{1}{\tau}$ respectively.*

Proof. First, it is not difficult to see that $\mathbf{x} \in SOL(M, \mathcal{K}^n, \mathbf{q})$ if and only if $M\mathbf{x}+\mathbf{q} \in SOL(M^{-1}, \mathcal{K}^n, -M^{-1}\mathbf{q})$. This relation indicates that M has the GUS property if and only if M^{-1} does as well. By Theorem 2.2, we know that $M^{-\top}J_n$ and $M^{-1}J_n$ both have the unique positive eigenvalue, namely t^*. On the other hand, by Theorem 2.2 again, we note that

$$\mathbf{0} = (M^\top J_n - \tau I_n)\mathbf{v} = \tau M^\top(\frac{1}{\tau}J_n - M^{-\top})\mathbf{v} = \tau M^\top(\frac{1}{\tau}I_n - M^{-\top}J_n)J_n\mathbf{v}$$

or, equivalently,

$$(M^{-\top}J_n - \frac{1}{\tau}I_n)J_n\mathbf{v} = \mathbf{0}.$$

This implies that $t^* = \frac{1}{\tau} > 0$ is the unique positive eigenvalue of $M^{-\top}J_n$ and $J_n\mathbf{v}$ is the associated eigenvector. Following a similar argument, we know that $t^* = \frac{1}{\tau} > 0$ is the unique positive eigenvalue of $M^{-1}J_n$ with $J_n\mathbf{w}$ as the associated eigenvector. □

For a given M with the GUS property and a vector $\mathbf{q} \in \mathbb{R}^n$, our main interest in this paper is to find the unique solution $\mathbf{x} \in SOL(M, \mathcal{K}^n, \mathbf{q})$. There are two special cases that can be handled very easily: $\mathbf{q} \in \mathcal{K}^n$ and $\mathbf{q} \in -M\mathcal{K}^n$. The former gives the solution $\mathbf{x} = \mathbf{0}$ whereas the latter leads to the solution $\mathbf{x} = -M^{-1}\mathbf{q}$. Furthermore, whenever $\mathbf{q} \notin -M\mathcal{K}^n \cup \mathcal{K}^n$, it is easy to see that finding $\mathbf{x} \in SOL(M, \mathcal{K}^n, \mathbf{q})$ is equivalent to solving the pair $(\mathbf{x}, s^*) \in bd(\mathcal{K}^n) \times \mathbb{R}_{++}$ satisfying (1.4) [30]. For this reason and for the simplicity of presentation, we will call $(\mathbf{x}, s^*)$ the solution pair to the $LCP(M, \mathcal{K}^n, \mathbf{q})$. The GUS property of M implies that there is a one-to-one relation between $\mathbf{q}$ and the pair $(\mathbf{x}, s^*)$. Interestingly, the following theorem proved in [30] and its implications shed some light on this relationship which serves as the theoretical fundamental of the bisection method that we will discuss in Section 3.

Theorem 2.5. *For $LCP(M, \mathcal{K}^n, \mathbf{q})$, if M has the GUS property, then the following statements hold:*

(1) The range $\mathcal{R}(M_\tau)$ of M_τ is a hyperplane in $\mathbb{R}^n$. Moreover, $\mathcal{R}(M_\tau) = (J_n\mathbf{v})^\perp$, where $\mathbf{v} \in int(\mathcal{K}^n)$ is the eigenvector of $M^\top J_n$ associated with the unique positive eigenvalue τ.

(2) For all $0 < t < s < \tau$, $K_t \backslash \{\mathbf{0}\} \subset int(K_s)$ and K_s lie on one side of $\mathcal{R}(M_\tau)$; if $s > t > \tau$, then $-\mathcal{K}^n \backslash \{\mathbf{0}\} \subset int(K_s)$, $K_s \backslash \{\mathbf{0}\} \subset int(K_t)$ and K_t lies on the other side of $\mathcal{R}(M_\tau)$.

Theorem 2.5, together with (1.4), says a lot about the solution pair $(\mathbf{x}, s^*)$ in terms of the vector $\mathbf{q}$, the eigenvector $\mathbf{v} \in int(\mathcal{K}^n)$ and the subspace $\mathcal{R}(M_\tau)$. For example, it implies that

$$s^* = \begin{cases} < \tau, & \text{if } (-\mathbf{q})^\top J_n \mathbf{v} > 0, \\ > \tau, & \text{if } (-\mathbf{q})^\top J_n \mathbf{v} < 0, \\ = \tau, & \text{if } (-\mathbf{q})^\top J_n \mathbf{v} = 0 \text{ or equivalently, } \mathbf{q} \in \mathcal{R}(M_\tau). \end{cases} \tag{2.13}$$

Moreover, if we currently have an estimate, say $s^{(k)}$, of s^*, Theorem 2.5 reveals a geometry picture of the $LCP(M, \mathcal{K}^n, \mathbf{q})$ and serves as a guide to update the estimate $s^{(k)}$. In particular, by defining $\mathbf{x}^{(k)} := -(M - s^{(k)} J_n)^{-1}\mathbf{q}$, we know that (i) in the case of $(-\mathbf{q})^\top J_n \mathbf{v} > 0$, $s^{(k)}$ should be decreased (resp. increased) if $\mathbf{x}^{(k)} \in int(\mathcal{K}^n)$ (resp. $\mathbf{x}^{(k)} \notin \mathcal{K}^n$); while (ii) in the case of $(-\mathbf{q})^\top J_n \mathbf{v} < 0$, $s^{(k)}$ should be increased (resp. decreased) if $\mathbf{x}^{(k)} \in int(\mathcal{K}^n)$ (resp. $\mathbf{x}^{(k)} \notin \mathcal{K}^n$). These facts directly lead us to a bisection iteration which we will discuss in the next section.

3. An efficient bisection method for the SOCLCP

3.1. **Prototype of a bisection algorithm.** Recall that if $\mathbf{q} \notin -M\mathcal{K}^n \cup \mathcal{K}^n$ and $SOL(M, \mathcal{K}^n, \mathbf{q}) \neq \emptyset$, then there exists $s^* > 0$ such that $(M - s^* J_n)\mathbf{x} = -\mathbf{q}$ for $\mathbf{x} \in SOL(M, \mathcal{K}^n, \mathbf{q})$; moreover, if M has the GUS property and $\mathbf{q} \notin \mathcal{R}(M_\tau)$, it follows that $0 < s^* \neq \tau$, where τ is the unique positive eigenvalue of MJ_n. Directly based on Theorem 2.5 and its following implications, a simple bisection method can be derived for which the pesudo-code is presented in Algorithm 1.

In Algorithm 1, the special case when $s^* = \tau$ or, equivalently, $\mathbf{q} \in \mathcal{R}(M_\tau)$, needs extra care and is solved independently. In Subsection 3.3, we will show that the solution in this case can be obtained by an efficient *direct* method (Algorithm 2). For the general case, on the other hand, we first point out that the algorithm is well defined. This follows from the fact that $\tau > 0$ is the unique positive eigenvalue of MJ_n (see Theorem 2.2(i)) and from our updating scheme for $s^{(k)}$ (see Steps 4 and 5 of Algorithm 1) which ensures that $0 < s^{(k)} \neq \tau$ for each $k = 1, 2, \ldots$. Thus, if $M - s^{(k)} J_n$ is singular, then there must exist a vector $\mathbf{z}$ so that

$$\mathbf{0} = (M - s^{(k)} J_n)\mathbf{z} = (MJ_n - s^{(k)} I_n) J_n \mathbf{z}.$$

This implies that $(s^{(k)}, J_n \mathbf{z})$ is an eigenpair of MJ_n, which is impossible according to our updating procedure for $s^{(k)}$ and Theorem 2.2(i). In other words, we can guarantee that the matrix $M - s^{(k)} J_n$ involved in the iteration is nonsingular for $k = 1, 2, \ldots$. Now, we note that the main computational cost lies in solving the linear system

$$(M - s^{(k)} J_n)\mathbf{x}^{(k)} = -\mathbf{q}$$

in each iteration, which in general requires $\mathcal{O}(n^3)$ flops. Fortunately, by taking advantage of the special structure, we will show in the next subsection that this main computational cost in each iteration can be reduced to $2n^2$ flops.

Algorithm 1. Prototype of a bisection method for the SOCLCP.

INPUT: A matrix M with the GUS property, a vector $\mathbf{q}$ and a tolerance $\epsilon > 0$.
OUTPUT: The solution $\mathbf{x} \in SOL(M, \mathcal{K}^n, \mathbf{q})$ and the corresponding s^*.

Step 1: **if** $\mathbf{q} \in \mathcal{K}^n$ **then**
  $\mathbf{x} = \mathbf{0}$;
  return;
end if

Step 2: **if** $-M^{-1}\mathbf{q} \in \mathcal{K}^n$ **then**
  $\mathbf{x} = -M^{-1}\mathbf{q}$;
  return;
end if

Step 3: Find the largest real eigenvalue τ of $M^\top J_n$ and its corresponding unit eigenvector $\mathbf{v}$ with a nonnegative first component; set $Index := -\frac{\mathbf{q}^\top J_n \mathbf{v}}{\|\mathbf{q}\|_2 \|\mathbf{v}\|_2}$ and find an upper bound β_u of s^*;

Step 4: **if** $Index = 0$ **then**
  $s^* = \tau$; $\mathbf{x} = SOCLCP\tau(M, \mathbf{q}, \tau)$; {Algorithm 2}
  return;
else
  if $Index > 0$ **then**
    $\alpha = 0; \beta = \tau$;
  else
    $\alpha = \tau; \beta = \beta_u$;
  end if
end if

Step 5: **for** $k = 1, 2, \cdots, \lceil \log_2 \frac{\beta - \alpha}{\epsilon} \rceil$ **do**
  $s^{(k)} = \frac{\beta^{(k)} - \alpha^{(k)}}{2}$; $\mathbf{x}^{(k)} = -(M - s^{(k)} J_n)^{-1}\mathbf{q}$;
  if $Index > 0$ **then**
    if $\mathbf{x}^{(k)} \in int(\mathcal{K}^n)$ **then**
      $\beta^{(k)} = s^{(k)}$;
    else
      $\alpha^{(k)} = s^{(k)}$;
    end if
  else
    if $\mathbf{x}^{(k)} \in int(\mathcal{K}^n)$ **then**
      $\alpha^{(k)} = s^{(k)}$;
    else
      $\beta^{(k)} = s^{(k)}$;
    end if
  end if
end for

Step 6: $\mathbf{x} = \mathbf{x}^{(k)}$; $s^* = s^{(k)}$.

3.2. **Efficient implementation of the bisection algorithm.** To reduce the computational cost in each iteration in Algorithm 1, we try to transform the linear system $(M - s^{(k)} J_n)\mathbf{x}^{(k)} = -\mathbf{q}$ to a simpler form. Note that for different $s^{(k)}$, only the diagonal elements of $M - s^{(k)} J_n$ vary; moreover, the matrix $J_n = \mathrm{diag}\{1, -I_{n-1}\}$ motives us to transform the matrix M to an upper Hessenberg matrix first. That

is, we can find an orthogonal matrix $Q = \mathrm{diag}\{1, \bar{Q}\}$ via, for example, a sequence of Givens rotations or Householder reflections (with $\frac{10}{3}n^3$ flops, [9])[4] such that

$$QMQ^\top = H, \tag{3.1}$$

where $H \in \mathbb{R}^{n\times n}$ is upper Hessenberg. Furthermore, we note that

$$QJ_nQ^\top = J_n \tag{3.2}$$

and consequently, one has

$$(M - s^{(k)}J_n)\mathbf{x}^{(k)} = Q^\top(H - s^{(k)}J_n)Q\mathbf{x}^{(k)} = -\mathbf{q}$$

or, equivalently,

$$(H - s^{(k)}J_n)Q\mathbf{x}^{(k)} = -\bar{\mathbf{q}} := -Q\mathbf{q}. \tag{3.3}$$

The attractive feature of (3.3) is that when $s^{(k)}$ varies, the matrix $H - s^{(k)}J_n$ remains to be upper Hessenberg and therefore if we define $\mathbf{y}^{(k)} = Q\mathbf{x}^{(k)}$, the solution $\mathbf{x}^{(k)}$ can be obtained by

$$(H - s^{(k)}J_n)\mathbf{y}^{(k)} = -\bar{\mathbf{q}}, \quad \mathbf{x}^{(k)} = Q^\top\mathbf{y}^{(k)}, \tag{3.4}$$

which requires $4n^2$ flops: n^2 flops for transforming $(H - s^{(k)}J_n)\mathbf{y}^{(k)} = -\bar{\mathbf{q}}$ to an upper triangular system, n^2 flops for solving $\mathbf{y}^{(k)}$ from the resulting upper triangular system, and $2n^2$ flops for $\mathbf{x}^{(k)} = Q^\top\mathbf{y}^{(k)}$. In fact, furthermore, since in each iteration of Step 5, we only need to determine whether $\mathbf{x}^{(k)}$ is in $\mathcal{K}^n$ or not, the fact that $\mathbf{x}^{(k)} \in \mathcal{K}^n$ if and only if $\mathbf{y}^{(k)} = Q\mathbf{x}^{(k)} \in \mathcal{K}^n$ implies that we can save the $2n^2$ flops required in computing $\mathbf{x}^{(k)} = Q^\top\mathbf{y}^{(k)}$. Consequently, we only need $2n^2$ flops for each iteration in Algorithm 1. *In other words, if the matrix M is transformed to an upper Hessenberg H at the beginning of Step 5, then each iteration in Algorithm* 1 *can proceed with* $2n^2$ *flops.* Therefore, in order to achieve the accuracy $|s^{(k)} - s^*| \le \epsilon$, the overall flops used in Step 5 in Algorithm 1 now become

$$\begin{cases} 2n^2\lceil\log_2\frac{\tau}{\epsilon}\rceil, & \text{if } \mathbf{q}^\top J_n\mathbf{v} < 0, \\ 2n^2\lceil\log_2\frac{\beta_u-\tau}{\epsilon}\rceil, & \text{if } \mathbf{q}^\top J_n\mathbf{v} > 0. \end{cases} \tag{3.5}$$

As far as the storage is concerned, it is worth pointing out that we can overwrite M and $\mathbf{q}$ by the upper Hessenberg matrix H, and the vector $\bar{\mathbf{q}} = Q\mathbf{q}$, respectively; furthermore, if the orthogonal matrix Q is represented by products of Givens rotations or Householder reflections, the "essential" parts of these Givens rotations or Householder reflections can be recorded in the zeroed portion of the matrix M (see Algorithm 7.4.2 in [9] for more details). Overall, therefore, the bisection iteration does not require extra storage except for the vector $\mathbf{y}^{(k)}$ and several scalars. This is another advantage of the algorithm.

According to (3.5), we know that the computational cost of Algorithm 1 depends on the parameters τ, ϵ and β_u (if $\mathbf{q}^\top J_n\mathbf{v} > 0$). For the case $\mathbf{q}^\top J_n\mathbf{v} > 0$, a tight upper bound β_u for s^* in general is not easy to give, computationally, nevertheless, a very simple but effective strategy works well. This strategy is based on the fact that whenever $\mathbf{q}^\top J_n\mathbf{v} > 0$, there must exist a positive integer l such that

$$2^{l-1}\tau \le s^* < 2^l\tau. \tag{3.6}$$

[4]As we will see, the *only* role of the orthogonal matrix Q is to transform the original variable $\mathbf{x}$ to the new $\mathbf{y} := Q\mathbf{x}$, and it is never used during the iteration. It is therefore preferable to represent Q by products of Givens rotations or Householder reflections [9, Section 5.1], which is of advantage for large scale and sparse problems.

Therefore, if we can find the integer l, then the bisection iteration in Algorithm 1 can proceed with the setting

$$\alpha = 2^{l-1}\tau \quad \text{and} \quad \beta = \beta_u = 2^l\tau.$$

Detecting the integer l turns out to be very easy because it is the smallest positive integer satisfying

$$-(M - 2^l\tau J_n)^{-1}\mathbf{q} \notin \mathcal{K}^n,$$

which is equivalent to finding the smallest positive integer l such that

$$-(H - 2^l\tau J_n)^{-1}\bar{\mathbf{q}} \notin \mathcal{K}^n.$$

In all our numerical testings, this simple but effective strategy is adopted.

Interestingly, in the case of $\mathbf{q}^\top J_n\mathbf{v} > 0$, there is an alternative strategy to get around the trouble in choosing an upper bound β_u for $s^* > \tau$. Indeed, if $\mathbf{q} \notin -M\mathcal{K}^n \cup \mathcal{K}^n$ and $\mathbf{q}^\top J_n\mathbf{v} > 0$, then for the solution pair $(\mathbf{x}, s^*)$ of $LCP(M, \mathcal{K}^n, \mathbf{q})$, we have

$$(M - s^*J_n)\mathbf{x} = -\mathbf{q} \quad \text{or} \quad M\mathbf{x} + \mathbf{q} = s^*J_n\mathbf{x}.$$

Now, by defining

$$\widehat{\mathbf{x}} := M\mathbf{x} + \mathbf{q} \quad \text{and} \quad \widehat{\mathbf{q}} := -M^{-1}\mathbf{q},$$

one has from $M\mathbf{x} + \mathbf{q} = s^*J_n\mathbf{x}$ that

$$\widehat{\mathbf{x}} = s^*J_n(M^{-1}\widehat{\mathbf{x}} + \widehat{\mathbf{q}}) \quad \text{or equivalently} \quad (M^{-1} - \frac{1}{s^*}J_n)\widehat{\mathbf{x}} = -\widehat{\mathbf{q}},$$

which implies that $(\widehat{\mathbf{x}}, \frac{1}{s^*})$ is the solution pair to $LCP(M^{-1}, \mathcal{K}^n, \widehat{\mathbf{q}})$. But Theorem 2.4 says that M^{-1} also has the GUS property and the unique positive eigenvalue of $M^{-1}J_n$ is $\frac{1}{\tau}$ and thus $0 < \frac{1}{s^*} < \frac{1}{\tau}$. In other words, $\alpha = 0$ and $\beta = \frac{1}{\tau}$ are the lower and the upper bounds for $\frac{1}{s^*}$, respectively. This argument then implies that for the case $\mathbf{q}^\top J_n\mathbf{v} > 0$, instead of solving $LCP(M, \mathcal{K}^n, \mathbf{q})$, we can alternatively employ the bisection iteration to solve $\widehat{\mathbf{x}} \in SOL(M^{-1}, \mathcal{K}^n, \widehat{\mathbf{q}})$. According to this strategy, consequently, we can avoid choosing a *large* upper bound β_u for s^*, and in order to attain an approximation $s^{(k)}$ of s^* with $|s^{(k)} - s^*| < \epsilon$, the total flops used *during the bisection iterations* can be

$$\begin{cases} 2n^2\lceil \log_2 \frac{\tau}{\epsilon} \rceil, & \text{if } \mathbf{q}^\top J_n\mathbf{v} < 0, \\ 2n^2\lceil -\log_2(\tau\epsilon) \rceil, & \text{if } \mathbf{q}^\top J_n\mathbf{v} > 0. \end{cases} \tag{3.7}$$

Finally, we point out that a practical and efficient implementation of Algorithm 1 should also concern how to efficiently compute the eigenpair $(\tau, \mathbf{v})$ of the matrix $M^\top J_n$ in Step 3. A naive way is to compute the full eigensystem of $M^\top J_n$, which requires $\mathcal{O}(n^3)$ flops and $\mathcal{O}(n^2)$ storage. This is obviously not optimal as we only need one specific eigenpair. According to Theorem 2.3, whenever M is positive definite or symmetric, an alternative and very efficient way is the *Implicitly Restarted Arnoldi Method (IRAM)*, which is particularly appropriate for large scale problems with special structure. IRAM aims at efficiently finding a few specific eigenpairs (e.g., the largest or smallest magnitude eigenvalues on the largest or smallest real part) of a given matrix, and has been successfully incorporated into the MATLAB platform (`eigs.m`)[5]. See [9, 22, 23, 27] and the references therein for a detailed discussion.

[5]In the MATLAB platform, the command: $[\mathbf{v}, \tau] =$ `eigs`$(M^\top J_n, 1, 'lr')$, returns the desired eigenpair.

3.3. **Handling the special case: $\mathbf{q} \in \mathcal{R}(M_\tau)$.** We now assume $\mathbf{q} \in \mathcal{R}(M_\tau)$ which implies that there is a unique solution $\mathbf{x} \in SOL(M, \mathcal{K}^n, \mathbf{q})$ such that

$$(M - \tau J_n)\mathbf{x} = -\mathbf{q},\ \mathbf{x} \in bd(\mathcal{K}^n). \tag{3.8}$$

This system facilitates us to construct the solution $\mathbf{x}$ directly. In fact, we can rewrite (3.8) as

$$(MJ_n - \tau I_n)\mathbf{z} = -\mathbf{q},\ \text{where } \mathbf{z} = J_n\mathbf{x} \in bd(\mathcal{K}^n). \tag{3.9}$$

Denote $MJ_n - \tau I_n$ by L_τ. Note that τ is also an eigenvalue of MJ_n and $\text{rank}(L_\tau) = n-1$ (see Theorem 2.2). Suppose $\mathbf{w} \in \mathbb{R}^n$ is a unit eigenvector of MJ_n associated with the eigenvalue τ. Then $L_\tau\mathbf{w} = \mathbf{0}$ and therefore any row of L_τ is orthogonal to $\mathbf{w}$. Let E be a permutation matrix such that the QR factorization of $L_\tau^\top E$ is

$$L_\tau^\top E = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n] \begin{bmatrix} r_{11} & \cdots & \cdots & r_{1n} \\ & \ddots & \vdots & \vdots \\ & & r_{n-1,n-1} & r_{n-1,n} \\ & & & r_{n,n} \end{bmatrix},$$

with the diagonal elements $|r_{ii}|$ in a decreasing order. The fact that $\text{rank}(L_\tau) = n-1$ implies that $r_{n,n} = 0$ and $r_{i,i} \neq 0$ for all $i = 1, 2, \ldots, n-1$; moreover, from $L_\tau\mathbf{w} = \mathbf{0}$, one has

$$\mathbf{0} = \begin{bmatrix} r_{11} & \cdots & \cdots & r_{1n} \\ & \ddots & \vdots & \vdots \\ & & r_{n-1,n-1} & r_{n-1,n} \\ & & & 0 \end{bmatrix}^\top \begin{bmatrix} \mathbf{p}_1^\top\mathbf{w} \\ \vdots \\ \mathbf{p}_n^\top\mathbf{w} \end{bmatrix},$$

which together with $r_{i,i} \neq 0$ for $i = 1, 2, \ldots, n-1$ leads to $\mathbf{w} \perp \mathbf{p}_i$ for $i = 1, 2, \ldots, n-1$; since $\mathbf{p}_i \perp \mathbf{p}_j$ for $i \neq j$, consequently, we have $\mathbf{p}_n = \mathbf{w}$ or $\mathbf{p}_n = -\mathbf{w}$. Denote

$$W = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{n-1}], \quad \text{and} \tag{3.10}$$

$$\left[\begin{array}{c|c} R & \mathbf{r} \\ \hline \mathbf{0} & 0 \end{array}\right] = \left[\begin{array}{ccc|c} r_{11} & \cdots & r_{1,n-1} & r_{1n} \\ & \ddots & \vdots & \vdots \\ & & r_{n-1,n-1} & r_{n-1,n} \\ \hline \multicolumn{3}{c|}{\mathbf{0}} & 0 \end{array}\right],$$

and we assume $\mathbf{z} = J_n\mathbf{x}$ is expressed as

$$\mathbf{z} = \gamma\mathbf{p}_n + W\mathbf{f},\ \text{where } \gamma \in \mathbb{R},\ \mathbf{f} \in \mathbb{R}^{n-1}. \tag{3.11}$$

Our goal is to find γ and $\mathbf{f}$ such that $\mathbf{z} \in bd(\mathcal{K}^n)$ and $L_\tau\mathbf{z} = -\mathbf{q}$. From

$$-\mathbf{q} = L_\tau\mathbf{z} = L_\tau W\mathbf{f},$$

one has

$$\begin{aligned} -W^\top\mathbf{q} = (W^\top L_\tau W)\mathbf{f} &= (W^\top E \begin{bmatrix} R^\top & \mathbf{0} \\ \mathbf{r}^\top & 0 \end{bmatrix} \begin{bmatrix} W^\top \\ \mathbf{w}^\top \end{bmatrix} W)\mathbf{f} \\ &= (W^\top E \begin{bmatrix} R^\top & \mathbf{0} \\ \mathbf{r}^\top & 0 \end{bmatrix} \begin{bmatrix} I_{n-1} \\ \mathbf{0} \end{bmatrix})\mathbf{f} \\ &= (W^\top E \begin{bmatrix} R^\top \\ \mathbf{r}^\top \end{bmatrix})\mathbf{f}, \end{aligned}$$

which leads to

$$\mathbf{f} = -(W^\top E \begin{bmatrix} R^\top \\ \mathbf{r}^\top \end{bmatrix})^{-1} W^\top \mathbf{q}. \tag{3.12}$$

Substituting (3.12) for (3.11) yields

$$\mathbf{z} = \gamma \mathbf{p}_n - W(W^\top E \begin{bmatrix} R^\top \\ \mathbf{r}^\top \end{bmatrix})^{-1} W^\top \mathbf{q} := \gamma \mathbf{p}_n + \mathbf{t}, \text{ where } \mathbf{t} = W\mathbf{f}. \tag{3.13}$$

Now the scalar γ can be obtained from the condition $\mathbf{z} \in bd(\mathcal{K}^n)$. In fact, if we denote

$$\mathbf{p}_n := [\xi, \mathbf{b}^\top]^\top, \text{ and } \mathbf{t} := [\varphi, \mathbf{c}^\top]^\top, \text{ where } \xi, \varphi \in \mathbb{R}, \text{ and } \mathbf{b}, \mathbf{c} \in \mathbb{R}^{n-1},$$

then $\mathbf{z} \in bd(\mathcal{K}^n)$ leads to the simple equation,

$$\xi\gamma + \varphi = \|\gamma \mathbf{b} + \mathbf{c}\|_2, \tag{3.14}$$

from which the root γ can be expressed explicitly. Overall, these steps lead us to a direct method for this special case and we summarize the pesudo-code[6] in Algorithm 2.

Algorithm 2. $\mathbf{x} = SOCLCP\tau(M, \mathbf{q}, \tau)$

(1) Form $L_\tau = MJ_n - \tau I_n$;
(2) Compute the QR decomposition of $L_\tau^\top$ with column permutation, i.e., $[W, R, E] = qr(L_\tau^\top)$;
(3) Set $\mathbf{p}_n = W(:, n)$, $W = W(:, 1 : n-1)$, $\mathbf{r} = R(1 : n-1, n)$, and $R = R(1 : n-1, 1 : n-1)$;
(4) Solve $\mathbf{f}$ from the linear system
$$(W^\top E \begin{bmatrix} R^\top \\ \mathbf{r}^\top \end{bmatrix})\mathbf{f} = -W^\top \mathbf{q}, \text{ and set } \mathbf{t} := W\mathbf{f};$$
(5) Solve (3.14) for γ with $\xi = \mathbf{p}_n(1)$, $\varphi = \mathbf{t}(1)$, $\mathbf{b} = \mathbf{w}(2 : n)$, and $\mathbf{c} = \mathbf{t}(2 : n)$;
(6) Set $\mathbf{x} = J_n(\gamma \mathbf{p}_n + \mathbf{t})$.

3.4. **Perturbation analysis for the special case: $\mathbf{q} \in \mathcal{R}(M_\tau)$.** It is known that if M has the GUS property, then for any given $\mathbf{0} \neq \mathbf{q}_0 \in \mathcal{R}(M_\tau)$, the scalar s^* corresponding to the solution $\mathbf{x}_0 \in SOL(M, \mathcal{K}^n, \mathbf{q}_0)$ is $s^* = \tau$. This subsection is devoted to the perturbation analysis for this special case. In particular, we will provide an analysis on the sensitivity of the solution pair $(\mathbf{x}, s^*)$ in terms of the perturbation of $\mathbf{q}$ in a neighborhood of $\mathbf{q}_0 \in \mathcal{R}(M_\tau)$. Our main result reveals that: (i) the sensitivity of the scalar s^* is uniquely dependent on the angle between $\mathbf{x}_0$ and $\mathbf{v}$, and the length $\|\mathbf{x}_0\|_2$, whereas (ii) the sensitivity of the solution $\mathbf{x}$ is related to the angle between $\mathbf{x}_0$ and $\mathbf{w}$, where $\mathbf{v} \in int(\mathcal{K}^n)$ and $\mathbf{w} \in int(\mathcal{K}^n)$ are the unit eigenvectors of $M^\top J_n$, and MJ_n is associated with the eigenvalue τ, respectively.

Theorem 3.1. *Suppose M has the GUS property and $\mathbf{q}_0 \in \mathcal{R}(M_\tau)$, where $\tau > 0$ is the positive eigenvalue of $M^\top J_n$ with the unit eigenvector $\mathbf{v} \in int(\mathcal{K}^n)$. Then*

[6]This algorithm is expressed in a stylized version of MATLAB language.

(i) the solution pair $(\mathbf{x}, s^*) \in \mathbb{R}^n \times \mathbb{R}$ *to the SOCLCP is locally a continuously differentiable function of* $\mathbf{q}$ *in a neighborhood of* $\mathbf{q}_0$. *That is, there is a neighborhood* $\mathcal{N}_{\mathbf{q}_0}$ *of* $\mathbf{q}_0$ *such that* $(\mathbf{x}, s^*) = (\mathbf{x}(\mathbf{q}), s^*(\mathbf{q}))$ *is a continuously differentiable function of* $\mathbf{q} \in \mathcal{N}_{\mathbf{q}_0}$,

(ii) the gradient of $s^*(\mathbf{q})$ *at* $\mathbf{q}_0$ *is* $\nabla s^*(\mathbf{q}_0) = \frac{J_n\mathbf{v}}{\mathbf{v}^\top \mathbf{x}(\mathbf{q}_0)}$, *and*

(iii) $\mathrm{rank}(D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0)) = n-1$, *and* $\mathcal{R}(D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0)) = (J_n\mathbf{x}_0)^\perp$, *where* $D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0)$ *stands for the derivative of* $\mathbf{x}(\mathbf{q})$ *with respect to* $\mathbf{q}$ *at* $\mathbf{q}_0$.

Proof. Define a continuously differentiable mapping $F : \mathbb{R}^{n+1+n} \to \mathbb{R}^{n+1}$ by

$$F(\mathbf{x}, s, \mathbf{q}) = \begin{bmatrix} (M - sJ_n)\mathbf{x} + \mathbf{q} \\ -\frac{1}{2}\mathbf{x}^\top J_n \mathbf{x} \end{bmatrix}. \tag{3.15}$$

Suppose $\mathbf{x}_0 \in SOL(M, \mathcal{K}^n, \mathbf{q}_0)$, then in a neighborhood of $(\mathbf{x}_0, \tau, \mathbf{q}_0)$, $(\mathbf{x}, s^*)$ is a solution pair to the SOCLCP if and only if $F(\mathbf{x}, s^*, \mathbf{q}) = \mathbf{0}$. Since $F(\mathbf{x}_0, \tau, \mathbf{q}_0) = \mathbf{0}$ and the partial derivative of $F(\mathbf{x}, s, \mathbf{q})$ with respect to $(\mathbf{x}, s)$ is

$$D_{(\mathbf{x},s)}F(\mathbf{x}_0, \tau, \mathbf{q}_0) = \begin{bmatrix} M - \tau J_n & -J_n\mathbf{x}_0 \\ -(J_n\mathbf{x}_0)^\top & 0 \end{bmatrix},$$

it is sufficient to show that $D_{(\mathbf{x},s)}F(\mathbf{x}_0, \tau, \mathbf{q}_0)$ is nonsingular. To this end, we assume

$$\begin{bmatrix} M - \tau J_n & -J_n\mathbf{x}_0 \\ -(J_n\mathbf{x}_0)^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \varrho \end{bmatrix} = \mathbf{0},$$

which gives

$$(M - \tau J_n)\mathbf{a} - J_n\mathbf{x}_0\varrho = \mathbf{0}, \tag{3.16}$$

$$(J_n\mathbf{x}_0)^\top \mathbf{a} = 0. \tag{3.17}$$

We first show that $\varrho = 0$. If this is not true, then from (3.16), we have

$$\mathbf{0} \neq (M - \tau J_n)\mathbf{a} = \varrho J_n\mathbf{x}_0 \in -bd(\mathcal{K}^n) \cup bd(\mathcal{K}^n). \tag{3.18}$$

On the other hand, from Theorem 2.2, we know that $\mathrm{rank}(M - \tau J_n) = n-1$ and $\mathcal{R}(M_\tau) \perp J_n\mathbf{v}$ where $\mathbf{v} \in int(\mathcal{K}^n)$ (and hence by Lemma 2.1, $J_n\mathbf{v} \in int(\mathcal{K}^n)$) is the eigenvector of $M^\top J_n$ associated with the eigenvalue τ. This fact implies

$$\mathcal{R}(M_\tau) \cap (-bd(\mathcal{K}^n) \cup bd(\mathcal{K}^n)) = \{\mathbf{0}\},$$

which contradicts (3.18). This shows $\varrho = 0$.

We next show $\mathbf{a} = \mathbf{0}$. Since $\varrho = 0$, from (3.16), we know that

$$\mathbf{0} = (M - \tau J_n)\mathbf{a} = (MJ_n - \tau I_n)J_n\mathbf{a},$$

which leads to two scenarios: $\mathbf{a} \neq \mathbf{0}$ and $\mathbf{a} = \mathbf{0}$. If $\mathbf{a} \neq \mathbf{0}$, by $\mathrm{rank}(M - \tau J_n) = n-1$ (see Theorem 2.2(i)), we know that the nonzero vector $J_n\mathbf{a}$ is parallel to $\mathbf{w}$, the eigenvector of MJ_n associated with the eigenvalue τ, which by the fact that $\mathbf{w} \in int(\mathcal{K}^n)$ implies $J_n\mathbf{a} \in -int(\mathcal{K}^n) \bigcup int(\mathcal{K}^n)$. On the other hand, by $\mathbf{x}_0 \in bd(\mathcal{K}^n)$, we know that

$$(J_n\mathbf{x}_0)^\top \mathbf{a} = (J_n\mathbf{a})^\top \mathbf{x}_0 \neq 0,$$

which contradicts (3.17). Consequently, we conclude that $\varrho = 0$ and $\mathbf{a} = \mathbf{0}$, and therefore, $D_{(\mathbf{x},s)}F(\mathbf{x}_0, \tau, \mathbf{q}_0)$ is nonsingular. By the implicit function theorem, we know that there is a neighborhood $\mathcal{N}_{\mathbf{q}_0}$ of $\mathbf{q}_0$ such that $(\mathbf{x}, s^*) = (\mathbf{x}(\mathbf{q}), s^*(\mathbf{q}))$ is a continuously differentiable function of $\mathbf{q} \in \mathcal{N}_{\mathbf{q}_0}$.

For (ii), differentiating $F(\mathbf{x}(\mathbf{q}), s^*(\mathbf{q}), \mathbf{q}) = \mathbf{0}$ with respect to $\mathbf{q}$ at $\mathbf{q}_0$ gives

$$(M - \tau J_n) D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0) - (J_n\mathbf{x}_0) D_{\mathbf{q}} s^*(\mathbf{q}_0) + I_n = \mathbf{0}, \tag{3.19}$$

$$-(J_n\mathbf{x}_0)^\top D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0) = \mathbf{0}. \tag{3.20}$$

Since $\mathbf{v}$ is the unit eigenvector of $M^\top J_n$ associated with τ (refer to Theorem 2.2), pre-multiplying $(J_n\mathbf{v})^\top$ on the both sides of (3.19) yields

$$-(\mathbf{v}^\top\mathbf{x}_0) D_{\mathbf{q}} s^*(\mathbf{q}_0) + \mathbf{v}^\top J_n = \mathbf{0} \quad \text{or} \quad D_{\mathbf{q}} s^*(\mathbf{q}_0) = \frac{\mathbf{v}^\top J_n}{\mathbf{v}^\top\mathbf{x}_0}, \tag{3.21}$$

which proves (ii).

Finally, since $\operatorname{rank}((J_n\mathbf{x}_0) D_{\mathbf{q}} s^*(\mathbf{q}_0)) = 1$ and $\operatorname{rank}((M - \tau J_n) D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0)) \le n - 1$, (3.19) leads to $\operatorname{rank}(D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0)) \ge n - 1$. Therefore, (3.20) implies (iii) and the proof is complete. □

Remark 3.2. Theorem 3.1 leads to the following observations:

(a) The gradient $\nabla s^*(\mathbf{q}_0) = \frac{J_n\mathbf{v}}{\mathbf{v}^\top\mathbf{x}_0}$ coincides with the fact that for any $\Delta\mathbf{q} \in \mathcal{R}(M_\tau)$, $s^*(\mathbf{q}_0 + \Delta\mathbf{q}) \equiv \tau$.

(b) Since $\nabla s^*(\mathbf{q}_0) = \frac{J_n\mathbf{v}}{\|\mathbf{v}\|_2\|\mathbf{x}_0\|_2\cos\eta} = \frac{J_n\mathbf{v}}{\|\mathbf{x}_0\|_2\cos\eta}$, where η is the angle between $\mathbf{v}$ and $\mathbf{x}_0$, we know that $s^*(\mathbf{q})$ becomes sensitive either if $\mathbf{x}_0$ is nearly orthogonal to $\mathbf{v}$, or if $\|\mathbf{x}_0\|_2$ is small.

(c) If $\Delta\mathbf{q}$ is sufficiently small, then we have

$$s^*(\mathbf{q}_0 + \Delta\mathbf{q}) = \tau + \frac{\mathbf{v}^\top J_n\Delta\mathbf{q}}{\mathbf{v}^\top\mathbf{x}_0} + \mathcal{O}(\|\Delta\mathbf{q}\|_2^2),$$

and thus

$$s^*(\mathbf{q}_0 + \Delta\mathbf{q}) \approx \tau + \frac{\mathbf{v}^\top J_n\Delta\mathbf{q}}{\mathbf{v}^\top\mathbf{x}_0}$$

serves as the first-order estimate for $s^*(\mathbf{q}_0 + \Delta\mathbf{q})$.

(d) As $\mathcal{R}(D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0)) = (J_n\mathbf{x}_0)^\perp$, when $\mathbf{q}_0$ has a sufficiently small perturbation $\Delta\mathbf{q}$, then the perturbation $\Delta\mathbf{x}$ of $\mathbf{x}_0$ lies almost in $(J_n\mathbf{x}_0)^\perp$, i.e., $\Delta\mathbf{x}^\top J_n\mathbf{x}_0 \approx 0$. Moreover,

$$\mathbf{x}(\mathbf{q}_0 + \Delta\mathbf{q}) = \mathbf{x}_0 + D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0)\Delta\mathbf{q} + \mathcal{O}(\|\Delta\mathbf{q}\|_2^2),$$

and thus

$$\mathbf{x}(\mathbf{q}_0 + \Delta\mathbf{q}) \approx \mathbf{x}_0 + D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0)\Delta\mathbf{q} \tag{3.22}$$

serves as the first-order estimate for $\mathbf{x}(\mathbf{q}_0 + \Delta\mathbf{q})$.

To give a formulation of (3.22), we assume that the columns of $B \in \mathbb{R}^{n\times(n-1)}$ span $\mathcal{R}(J_n\mathbf{w})^\perp$ where $\mathbf{w}$ is the unit eigenvector of MJ_n associated with the eigenvalue τ. It is clear that computationally B could be $B = J_nW$, where $W \in \mathbb{R}^{n\times(n-1)}$ is given in (3.10). Now we write

$$D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0)\Delta\mathbf{q} = B\mathbf{d} + \vartheta J_n\mathbf{w}, \quad \mathbf{d} \in \mathbb{R}^{n-1}, \quad \vartheta \in \mathbb{R}.$$

Then from (3.20), we have

$$\vartheta = -\frac{(B\mathbf{d})^\top J_n\mathbf{x}_0}{\mathbf{w}^\top\mathbf{x}_0}.$$

By post-multiplying $\Delta\mathbf{q}$ on both sides of (3.19) and noting $(M-\tau J_n)J_n\mathbf{w}=\mathbf{0}$, one has

$$(M-\tau J_n)B\mathbf{d}-(J_n\mathbf{x}_0)D_{\mathbf{q}}s^*(\mathbf{q}_0)\Delta\mathbf{q}+\Delta\mathbf{q}=\mathbf{0},$$

and consequently from (3.21), it follows that

$$\mathbf{d}=[B^\top(M-\tau J_n)B]^{-1}B^\top(\frac{\mathbf{v}^\top J_n\Delta\mathbf{q}}{\mathbf{v}^\top\mathbf{x}_0}J_n\mathbf{x}_0-\Delta\mathbf{q}).$$

The formula of $D_{\mathbf{q}}\mathbf{x}(\mathbf{q}_0)\Delta\mathbf{q}$ then leads to another observation: the solution $\mathbf{x}\in SOL(M,\mathcal{K}^n,\mathbf{q})$ is likely to be sensitive to the perturbation of $\mathbf{q}$ in a neighborhood of $\mathbf{q}_0\in\mathcal{R}(M_\tau)$ if $\mathbf{x}_0$ is nearly orthogonal to $\mathbf{w}$ or to $\mathbf{v}$.

4. An efficient Newton method

As our Algorithm 1 follows the framework of the bisection procedure, it is only of linear convergence. In this section, an efficient Newton iteration is developed to remedy its slow convergence. We will propose an efficient procedure to implement each Newton iteration using only about $5n^2$ flops; moreover, we will show that whenever M has the GUS property, the Newton iteration locally converges quadratically for *any* $\mathbf{q}\notin -M\mathcal{K}^n\cup\mathcal{K}^n$.

Let $\mathbf{q}_0\notin -M\mathcal{K}^n\cup\mathcal{K}^n$ be given and $F(\mathbf{x},s,\mathbf{q}_0)$ be a function of $(\mathbf{x},s)$ defined by (3.15). Suppose $(\mathbf{x}_0,s^*)$ is the solution pair to the SOCLCP, then our first conclusion claims that the Jacobian of $F(\mathbf{x},s,\mathbf{q}_0)$ at $(\mathbf{x}_0,s^*)$ is nonsingular provided that M has the GUS property. To establish this result, we need the following two lemmas. Lemma 4.1 (see [30]) characterizes the normal cone N_{K_s} of the cone $K_s:=\{(M-sJ_n)\mathbf{a}|\mathbf{a}\in\mathcal{K}^n\}$ at a boundary point, and Lemma 4.2 is the key to guarantee the nonsingularity of Jacobian of $F(\mathbf{x},s,\mathbf{q}_0)$ at $(\mathbf{x}_0,s^*)$.

Lemma 4.1. *Suppose that $A\in\mathbb{R}^{n\times n}$ is nonsingular. Let $\mathcal{C}=A\mathcal{K}^n$ and let $\mathbf{a}$ be a nonzero vector in $bd(\mathcal{K}^n)$. Then the normal cone $N_{\mathcal{C}}(A\mathbf{a})$ of $\mathcal{C}$ at the point $A\mathbf{a}$ is*

$$N_{\mathcal{C}}(A\mathbf{a})=\{-tA^{-\top}J_n\mathbf{a}|t\geq 0\}.$$

Lemma 4.2. *Suppose M has the GUS property and τ is the unique positive eigenvalue of MJ_n. Then for any $0<s\neq\tau$ and for all nonzero $\mathbf{a}\in bd(\mathcal{K}^n)$, we have*

$$\mathbf{a}^\top(M-sJ_n)^{-1}\mathbf{a}\begin{cases}>0, & \text{if } 0<s<\tau,\\ <0, & \text{if } s>\tau.\end{cases}$$

Proof. The part for $0<s<\tau$ has been proved in [30]. The argument for the case $s>\tau$ follows the case $0<s<\tau$. Indeed, if $s>\tau$, by Theorem 2.2, M_s is nonsingular, and thus for any nonzero $\mathbf{a}\in bd(\mathcal{K}^n)$, Lemma 4.1 says that

$$N_{K_s}(M_sJ_n\mathbf{a})=\{-tM_s^{-\top}\mathbf{a}|t\geq 0\}.$$

From Theorem 2.5, we know that for any $t>s$, $M_tJ_n\mathbf{a}\in int(K_s)$, which together with $-M_s^{-\top}\mathbf{a}\in N_{K_s}(M_sJ_n\mathbf{a})$ and (2.1), implies

$$(-M_s^{-\top}\mathbf{a})^\top(M_tJ_n\mathbf{a}-M_sJ_n\mathbf{a})=-(s-t)\mathbf{a}^\top M_s^{-1}\mathbf{a}<0.$$

This shows $\mathbf{a}^\top(M-sJ_n)^{-1}\mathbf{a}<0$ for any $s>\tau$, and we complete the proof. □

Based on Lemma 4.2, we claim

Theorem 4.3. *Suppose M has the GUS property, then for any $\mathbf{q}_0 \notin -M\mathcal{K}^n \cup \mathcal{K}^n$, the Jacobian of $F(\mathbf{x}, s, \mathbf{q}_0)$ at the solution pair $(\mathbf{x}_0, s^*)$,*

$$D_{(\mathbf{x},s)}F(\mathbf{x}_0, s^*, \mathbf{q}_0) = \begin{bmatrix} M - s^*J_n & -J_n\mathbf{x}_0 \\ -(J_n\mathbf{x}_0)^\top & 0 \end{bmatrix}$$

is nonsingular.

Proof. Suppose there exists a vector $[\mathbf{a}^\top, \varrho]^\top \in \mathbb{R}^{n+1}$ satisfying

$$\begin{bmatrix} M - s^*J_n & -J_n\mathbf{x}_0 \\ -(J_n\mathbf{x}_0)^\top & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \varrho \end{bmatrix} = \mathbf{0}.$$

In the case $s^* = \tau$, Theorem 3.1(i) indicates $[\mathbf{a}^\top, \varrho]^\top = \mathbf{0}$. If $s^* \neq \tau$, then it follows that $M - s^*J_n$ is nonsingular and thus one has

$$\mathbf{a} = -\varrho(M - s^*J_n)^{-1}(J_n\mathbf{x}_0)$$

and

$$0 = -\varrho(J_n\mathbf{x}_0)^\top(M - s^*J_n)^{-1}(J_n\mathbf{x}_0).$$

This relation together with Lemma 4.2 implies $\varrho = 0$ which leads to $\mathbf{a} = \mathbf{0}$. Therefore, our conclusion follows. □

Theorem 4.3 implies that whenever M has the GUS property and $\mathbf{q}_0 \notin -M\mathcal{K}^n \cup \mathcal{K}^n$, there is a neighborhood $\mathcal{N}_{(\mathbf{x}_0,s^*)}$ of the solution pair $(\mathbf{x}_0, s^*)$ such that for any initial point $(\mathbf{x}^{(0)}, s^{(0)}) \in \mathcal{N}_{(\mathbf{x}_0,s^*)}$, the Newton iteration

$$\begin{cases} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \Delta\mathbf{x}, \\ s^{(k+1)} &= s^{(k)} + \Delta s, \end{cases} \tag{4.1}$$

where

$$\begin{bmatrix} M - s^{(k)}J_n & -J_n\mathbf{x}^{(k)} \\ -(J_n\mathbf{x}^{(k)})^\top & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta s \end{bmatrix} = -F^{(k)} := -F(\mathbf{x}^{(k)}, s^{(k)}, \mathbf{q}_0), \tag{4.2}$$

converges quadratically to $(\mathbf{x}_0, s^*)$. On the other hand, as the bisection method is of global convergence, it is able to provide a good initial point $(\mathbf{x}^{(0)}, s^{(0)})$ for the Newton iteration. Moreover, by taking advantage of the special structure in (4.2), we will see that a single Newton iteration only requires $5n^2$ flops.

To describe the detailed computational procedure, recall that there is an orthogonal matrix $Q = \operatorname{diag}\{1, \bar{Q}\}$ such that $QJ_nQ^\top = J_n$ and $QMQ^\top = H$ is an upper Hessenberg matrix. Based on this fact, one has from (4.2) that

$$\begin{bmatrix} Q & \\ & 1 \end{bmatrix} \begin{bmatrix} M - s^{(k)}J_n & -J_n\mathbf{x}^{(k)} \\ -(J_n\mathbf{x}^{(k)})^\top & 0 \end{bmatrix} \begin{bmatrix} Q^\top & \\ & 1 \end{bmatrix} \begin{bmatrix} Q & \\ & 1 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta s \end{bmatrix} = - \begin{bmatrix} Q & \\ & 1 \end{bmatrix} F^{(k)} := -\bar{F}^{(k)}$$

or, equivalently,

$$\begin{bmatrix} H - s^{(k)}J_n & -QJ_n\mathbf{x}^{(k)} \\ -(QJ_n\mathbf{x}^{(k)})^\top & 0 \end{bmatrix} \begin{bmatrix} Q\Delta\mathbf{x} \\ \Delta s \end{bmatrix} = -\bar{F}^{(k)}.$$

Furthermore, we note that

$$QJ_n\mathbf{x}^{(k)} = QJ_nQ^\top Q\mathbf{x}^{(k)} = J_nQ\mathbf{x}^{(k)},$$

and the update (4.1) could be rewritten as

$$\begin{cases} Q\mathbf{x}^{(k+1)} &= Q\mathbf{x}^{(k)} + Q\Delta\mathbf{x}, \\ s^{(k+1)} &= s^{(k)} + \Delta s. \end{cases}$$

Therefore, if we define the new variable $\mathbf{y} = Q\mathbf{x}$ (see also (3.4)), the Newton iteration (4.1) for $(\mathbf{x}^{(k)}, s^{(k)})$ could be realized via $(\mathbf{y}^{(k)}, s^{(k)})$; that is, we can define the Newton iteration for $(\mathbf{y}^{(k)}, s^{(k)})$ as

$$\begin{cases} \mathbf{y}^{(k+1)} &= \mathbf{y}^{(k)} + \Delta\mathbf{y}, \\ s^{(k+1)} &= s^{(k)} + \Delta s, \end{cases} \tag{4.3}$$

where $[\Delta\mathbf{y}^\top, \Delta s]^\top$ is the solution of the linear system

$$\begin{bmatrix} H - s^{(k)}J_n & -J_n\mathbf{y}^{(k)} \\ -(J_n\mathbf{y}^{(k)})^\top & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{y} \\ \Delta s \end{bmatrix} = -\bar{F}^{(k)}. \tag{4.4}$$

The attractive feature of the system (4.4) is that the solution $[\Delta\mathbf{y}^\top, \Delta s]^\top$ can be obtained using only about $3n^2$ flops. To see this more clearly, we note that the $(1,1)$ block of the coefficient matrix of (4.4) remains to be upper Hessenberg as $s^{(k)}$ varies. Therefore, it takes about $2n^2$ flops to transform this system to an upper triangular system for which n^2 flops are needed for solving $[\Delta\mathbf{y}^\top, \Delta s]^\top$. To complete a single Newton step, we should also update the system (4.4). For this step, the relation

$$\begin{aligned} \bar{F}^{(k)} &= \begin{bmatrix} Q & \\ & 1 \end{bmatrix} F^{(k)} \\ &= \begin{bmatrix} Q(M - s^{(k)}J_n)\mathbf{x}^{(k)} + Q\mathbf{q}_0 \\ -\frac{1}{2}(\mathbf{x}^{(k)})^\top J_n\mathbf{x}^{(k)} \end{bmatrix} = \begin{bmatrix} (H - s^{(k)}J_n)Q\mathbf{x}^{(k)} + Q\mathbf{q}_0 \\ -\frac{1}{2}(\mathbf{x}^{(k)})^\top Q^\top J_n Q\mathbf{x}^{(k)} \end{bmatrix} \\ &= \begin{bmatrix} (H - s^{(k)}J_n)\mathbf{y}^{(k)} + Q\mathbf{q}_0 \\ -\frac{1}{2}(\mathbf{y}^{(k)})^\top J_n\mathbf{y}^{(k)} \end{bmatrix} \end{aligned}$$

implies that updating $\bar{F}^{(k)}$ to $\bar{F}^{(k+1)}$ could be done using only $2n^2$ flops (the vector $\bar{\mathbf{q}} = Q\mathbf{q}_0$ is constant in each iteration and needs not to be updated). Consequently, we conclude that a Newton step for updating $(\mathbf{y}^{(k)}, s^{(k)})$ to $(\mathbf{y}^{(k+1)}, s^{(k+1)})$ requires totally about $5n^2$ flops.

Based on our discussion in 3.2, we know that the initial point $(\mathbf{y}^{(k)}, s^{(k)})$ for the Newton iteration (4.3) could be efficiently generated by the bisection method (see (3.4)), and the orthogonal matrix Q, the upper Hessenberg matrix H and the vector $\bar{\mathbf{q}} = Q\mathbf{q}_0$ which are also needed in (3.4) can still be used in the Newton iteration, and therefore these methods can be perfectly combined. To conclude this section, we combine our described techniques and present the complete bisection-Newton algorithm for $LCP(M, \mathcal{K}^n, \mathbf{q})$ in Algorithm 3.

5. Numerical experiments

In this section, we will present our preliminary but very encouraging numerical experiments of the bisection-Newton algorithm (Algorithm 3). It is known that there exist various methods for solving the second-order cone complement problem (1.5). In order to evaluate the numerical performance and demonstrate clearly the efficiency of the bisection-Newton algorithm (BN for short) for solving the SOCLCP,

Algorithm 3. A bisection-Newton method for the SOCLCP.

INPUT: A matrix $M \in \mathbb{R}^{n\times n}$ with the GUS property, a vector $\mathbf{q} \in \mathbb{R}^n$ and tolerances $\epsilon_i, \epsilon_b, \epsilon_n > 0$.

OUTPUT: The solution $\mathbf{x} \in SOL(M, \mathcal{K}^n, \mathbf{q})$ and the corresponding s^*.

Step 1: **if** $\mathbf{q} \in \mathcal{K}^n$ **then**
  $\mathbf{x} = \mathbf{0}$;
  return;
end if

Step 2: **if** $-M^{-1}\mathbf{q} \in \mathcal{K}^n$ **then**
  $\mathbf{x} = -M^{-1}\mathbf{q}$;
  return;
end if

Step 3: Find the eigenpair $(\tau, \mathbf{v})$ of $M^\top J_n$ where $\tau > 0$ and $\mathbf{v} \in int(\mathcal{K}^n)$; set $Index := -\frac{\mathbf{q}^\top J_n \mathbf{v}}{\|\mathbf{q}\|_2\|\mathbf{v}\|_2}$;

Step 4: Find the orthogonal matrix Q such that $QMQ^\top = H$ is upper Hessenberg; set $\bar{\mathbf{q}} := Q\mathbf{q}$;

Step 5: **if** $|Index| < \epsilon_i$ **then**
  $s^* = \tau$; $\mathbf{x} = SOCLCP\tau(M, \mathbf{q}, \tau)$; {Algorithm 2}
  return;
else
  if $Index > \epsilon_i$ **then**
    $\alpha = 0; \beta = \tau$;
  else
    Find the smallest integer $l > 0$ satisfying $-(H - 2^l\tau J_n)^{-1}\bar{\mathbf{q}} \notin \mathcal{K}^n$;
    set $\alpha = 2^{l-1}\tau$ and $\beta = 2^l\tau$;
  end if
end if

Step 6: **for** $k = 1, 2, \cdots, \lceil \log_2 \frac{\beta-\alpha}{\epsilon_b} \rceil$ **do**
  $s^{(k)} = \frac{\beta^{(k)} - \alpha^{(k)}}{2}$; solve $\mathbf{y}^{(k)}$ from $(H - s^{(k)}J_n)\mathbf{y}^{(k)} = -\bar{\mathbf{q}}$;
  if $Index > 0$ **then**
    if $\mathbf{y}^{(k)} \in int(\mathcal{K}^n)$ **then**
      $\beta^{(k)} = s^{(k)}$;
    else
      $\alpha^{(k)} = s^{(k)}$;
    end if
  else
    if $\mathbf{y}^{(k)} \in int(\mathcal{K}^n)$ **then**
      $\alpha^{(k)} = s^{(k)}$;
    else
      $\beta^{(k)} = s^{(k)}$;
    end if
  end if
end for

Step 7: **while** $(|(\mathbf{y}^{(k)})^\top J_n \mathbf{y}^{(k)}|/2 + \|(H - s^{(k)}J_n)\mathbf{y}^{(k)} + \bar{\mathbf{q}}\|_2) > \epsilon_n$ **do**
  solve $[\Delta\mathbf{y}^\top, \Delta s]^\top$ from

$$\begin{bmatrix} H - s^{(k)}J_n & -J_n\mathbf{y}^{(k)} \\ -(J_n\mathbf{y}^{(k)})^\top & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{y} \\ \Delta s \end{bmatrix} = - \begin{bmatrix} (H - s^{(k)}J_n)\mathbf{y}^{(k)} + \bar{\mathbf{q}} \\ -\frac{1}{2}(\mathbf{y}^{(k)})^\top J_n \mathbf{y}^{(k)} \end{bmatrix};$$

  update $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \Delta\mathbf{y}$; $s^{(k+1)} = s^{(k)} + \Delta s$; $k = k + 1$;
end while

Step 8: $\mathbf{x} = Q^\top \mathbf{y}^{(k)}$; $s^* = s^{(k)}$.

we will also present the numerical results from the smoothing Newton method [17] (SNM for short), the smoothing-regularization method [14] (SRM[7] for short) and a descent method based on the Fischer-Burmeister merit function (FBMF) [3] (DM[8] for short). On the other hand, since Algorithm 3 basically involves two different procedures, we will separately present our numerical experiences in two parts: one for the general case $\mathbf{q} \notin \mathcal{R}(M_\tau)$ and the other for the special case $\mathbf{q} \in \mathcal{R}(M_\tau)$. All of our tests are carried out in MATLAB 7.1 on a PC with Intel(R) Core(R)i3 CPU 550@3.20GHz, 3.20GHz.

5.1. **Numerical testing for the bisection-Newton iteration.** To evaluate the efficiency of the bisection-Newton iteration of Algorithm 3, we vary the problem size n from 100 to 1000 and generate randomly $LCP(M, \mathcal{K}^n, \mathbf{q})$. In particular, for every given n, 100 symmetric and positive definite matrices $M \in \mathbb{R}^{n\times n}$ together with 100 corresponding random $\mathbf{q} \notin -M\mathcal{K}^n \cup \mathcal{K}^n$ are generated with each element uniformly distributed in the interval $[-1, 1]$. Therefore, for each n, we totally have 100 test SOCLCPs. The average numerical performance of each tested method over 100 random testings is evaluated and compared.

For the parameters involved in the SNM [17], SRM [14], and DM [3], we use the values and adopt the stopping criteria that have been suggested and tested previously. For Algorithm 3, we set $\epsilon_i = 10^{-9}$ (Step 5 in Algorithm 3) and $\epsilon_n = 10^{-10}$ (Step 7 in Algorithm 3). Besides these parameters, we should also point out that the termination criterion for the bisection procedure (Step 6 in Algorithm 3) is another important factor that determines the performance of the bisection-Newton iteration. Finding a proper parameter ϵ_b is related to the problem of detecting the neighborhood (called the *basin of attraction*) in which the Newton iteration converges quadratically. Such a problem is generally believed to be a hard problem in the literature, and even for finding the zeros of the general real polynomial, there are no known fail-safe rules for selecting initial values [28]. The bisection iteration is usually employed to provide an initial guess for the Newton iteration, which can refine the approximation from the bisection procedure [28]. There are some sufficient conditions for the local quadratic convergence of the Newton iteration in the literature, for example, the Newton-Kantorovich Theorem (see e.g., [24, 28]), which states that the basin of attraction of the Newton iteration is dependent on $F(\mathbf{x}, s, \mathbf{q})$ defined in (3.15) and the Jacobian $D_{(\mathbf{x},s)}F(\mathbf{x}_0, s^*, \mathbf{q}_0)$ given in Theorem 4.3 as well. However, it is expensive to check these sufficient conditions in Newton-Kantorovich Theorem, and in many problems, these sufficient conditions are not fulfilled. According to these observations, we suggest and test a simple *conservative* rule: $\epsilon_b = 10^{-\nu}$. In our testing, to evaluate the bisection-Newton method and compare it with others, we choose $\nu = 1, 2, 3$ and set additionally the maximal number of bisection steps as 50. It can be expected that as ν gets large, less Newton steps are required to refine the approximation from the bisection iteration. This is observed in our numerical experiments.

In Table 1, we summarize the average numbers of iterations (labeled as "Iter#") for every method. Since Algorithm 3 consists of two iterative schemes, for each

[7]The code of the SRM is available at: http://www-optima.amp.i.kyoto-u.ac.jp/∼hayashi/index_e.html

[8]The code of the DM is available at: http://math.ntnu.edu.tw/∼jschen/Publications.html

criterion ϵ_b, we list the average number of bisection iterations used in Step 6 (labeled as "Biter#") and the average number of Newton iterations required in Step 7 (labeled as "Niter#") separately. The corresponding CPU times of each method are summarized in Table 2. To check the accuracy of the computed solution $\mathbf{x} = [x_1, \mathbf{x}_2^\top]^\top$, we define

$$fc := |\mathbf{x}^\top \mathbf{g}| + |x_1 - \|\mathbf{x}_2\|_2| + |g_1 - \|\mathbf{g}_2\|_2|, \tag{5.1}$$

where $\mathbf{g} = [g_1, \mathbf{g}_2^\top]^\top := M\mathbf{x} + \mathbf{q}$ with $g_1 \in \mathbb{R}$ and $\mathbf{g}_2 \in \mathbb{R}^{n-1}$. As we have pointed out in (1.4), for our tested case $\mathbf{q} \notin -M\mathcal{K}^n \cup \mathcal{K}^n$, $fc = 0$ implies $\mathbf{x} \in SOL(M, \mathcal{K}^n, \mathbf{q})$, and hence fc measures the feasibility of $\mathbf{x}, \mathbf{g} \in bd(\mathcal{K}^n)$ and the complementarity of the computed solution $\mathbf{x}$. In Table 3, the average values of fc over 100 random tests are listed.

From these tables, we can see that all the tested methods succeed in solving these problems but their numerical performances are different. In particular, we observe that

(i) the bisection procedure terminates within 50 iterations ($20 \sim 30$ iterations in most cases),
(ii) due to the quadratic convergence of the Newton iteration, less than 3 Newton steps (1 or 2 steps in most cases) are required to refine a moderately accurate approximation obtained from the bisection iteration to our given termination rule,
(iii) as ν gets larger, more bisection steps but less Newton steps are needed, and
(iv) the bisection-Newton algorithm converges fastest to a highly accurate solution.

TABLE 1. Average number of iterations with various problem sizes.

	SNM	SRM	DM	BN($\nu = 1$)		BN($\nu = 2$)		BN($\nu = 3$)	
n	Iter#	Iter#	Iter#	Biter#	Niter#	Biter#	Niter#	Biter#	Niter#
100	9.42	6.23	2044.69	25.17	1.10	26.66	0.88	28.08	0.81
200	10.80	6.47	7614.48	24.96	1.20	26.66	1.00	28.21	0.91
300	11.74	7.00	15262.70	24.51	1.20	26.24	1.06	28.19	0.88
400	12.66	7.00	25182.70	25.41	1.19	27.12	1.03	28.70	0.85
500	13.36	7.00	28767.80	24.64	1.23	26.37	1.12	28.20	0.60
600	14.09	7.05	32302.20	28.40	0.93	29.65	0.82	31.20	0.62
700	14.88	7.43	48914.30	25.92	1.19	27.84	1.05	29.53	0.77
800	15.34	7.45	58566.20	28.55	0.91	30.04	0.83	31.28	0.61
900	16.40	7.08	86321.20	23.75	1.32	25.74	1.27	27.67	1.01
1000	16.44	7.07	97901.50	25.98	1.13	27.57	1.08	29.36	0.81

TABLE 2. Average CPU time(s) with various problem sizes.

n	SNM	SRM	DM	BN($\nu = 1$)	BN($\nu = 2$)	BN($\nu = 3$)
100	0.0404	0.0489	1.0521	0.0162	0.0162	0.0162
200	0.2962	0.3638	5.5027	0.0469	0.0478	0.0479
300	1.0412	1.2385	14.8425	0.1365	0.1361	0.1421
400	2.8023	3.4515	37.6897	0.3117	0.3130	0.3164
500	5.5273	6.6480	57.6000	0.5813	0.5813	0.5879
600	8.9139	10.4772	132.9670	0.8472	0.8536	0.8621
700	14.8391	19.2171	311.8070	1.2603	1.2738	1.2853
800	29.5577	46.8406	708.1530	2.6413	2.6209	2.6334
900	46.0424	79.7097	1298.5300	3.6265	3.6561	3.6838
1000	59.2010	95.7390	1637.2500	4.6585	4.6287	4.6587

TABLE 3. Accuracy of the computed solution with various problem sizes.

n	SNM fc	SRM fc	DM fc	BN($\nu = 1$) fc	BN($\nu = 2$) fc	BN($\nu = 3$) fc
100	1.0272e-07	2.4143e-09	9.0935e-06	3.1427e-11	2.6201e-11	9.8735e-11
200	1.4924e-07	2.8807e-10	1.3903e-05	3.9967e-11	1.5882e-10	2.7511e-10
300	2.3428e-07	9.9837e-11	1.8749e-05	4.3617e-11	2.1658e-11	1.1533e-09
400	2.8157e-07	1.5812e-10	2.0837e-05	3.0345e-11	2.0588e-11	3.3914e-10
500	1.9250e-07	6.6678e-11	3.0552e-05	6.2700e-11	5.8400e-11	3.4225e-10
600	2.5014e-07	1.2745e-10	2.8555e-05	4.3533e-11	3.2509e-11	1.8751e-10
700	2.1447e-07	4.3511e-10	1.6376e-04	4.4745e-11	4.7116e-11	3.9005e-10
800	2.0841e-07	1.2327e-09	2.6978e-04	4.0576e-11	4.6273e-10	2.8714e-10
900	2.0235e-07	2.2823e-09	4.3328e-04	3.5671e-11	1.7896e-11	3.6038e-10
1000	3.3524e-07	4.2692e-09	7.8158e-04	4.0370e-11	2.6522e-11	1.3811e-10

TABLE 4. Average number of iterations and CPU time with various problem sizes.

Problem size	SRM		BN
n	Iter#	CPU(s)	CPU(s)
100	6.00	0.0431	0.0130
200	6.01	0.3122	0.0445
300	6.00	1.0258	0.1299
400	6.05	2.5958	0.2976
500	6.06	4.9600	0.5718
600	6.23	8.7318	0.9793
700	6.36	14.6201	1.6444
800	6.40	22.1725	2.5364
900	6.59	34.7423	3.9893
1000	6.42	48.3565	5.5946

5.2. **Numerical testing for the special case $\mathbf{q} \in \mathcal{R}(M_\tau)$.** This subsection is devoted to testing the direct algorithm (Algorithm 2) for the special case $\mathbf{q} \in \mathcal{R}(M_\tau)$. For this purpose, we first describe our four-step-procedure in generating the test problems as follows:

(1) generate randomly a symmetric and positive definite matrix $M \in \mathbb{R}^{n\times n}$;
(2) find the largest eigenvalue τ of MJ_n;
(3) generate a random $\mathbf{x} \in bd(\mathcal{K}^n)$;
(4) set $\mathbf{q} = -(M - \tau J_n)\mathbf{x}$.

According to this procedure, we generate 100 testing problems for each n varying from 100 to 1000. As we observed that there are some cases for which the SNM and the DM fail to converge within the given stopping criteria, we only present numerical results from the SRM and the BN. In Table 4, we list average numbers of iterations and average CPU times for the SRM and BN methods. Furthermore, in Figure 1, the relation between $\log_{10} fc$ and the problem size n is plotted for both methods.

6. Conclusions and future work

In this paper, we have investigated the LCP over the second-order cone (1.3) from a new perspective. Our new development on the SOCLCP benefits from the basic linear-algebra-related properties which characterize the GUS property of M. The mechanics behind the bisection iteration is from the geometry knowledge of the $LCP(M, \mathcal{K}^n, \mathbf{q})$. As Theorems 2.2 and 2.5 serve as the theoretical fundamental for the bisection procedure, the success of the bisection method in this paper can also be viewed as a numerical verification of these theoretical results.

Finally, we point out that although our algorithm is currently designed specially for the SOCLCP (1.3), the idea and the techniques might be used and be extended

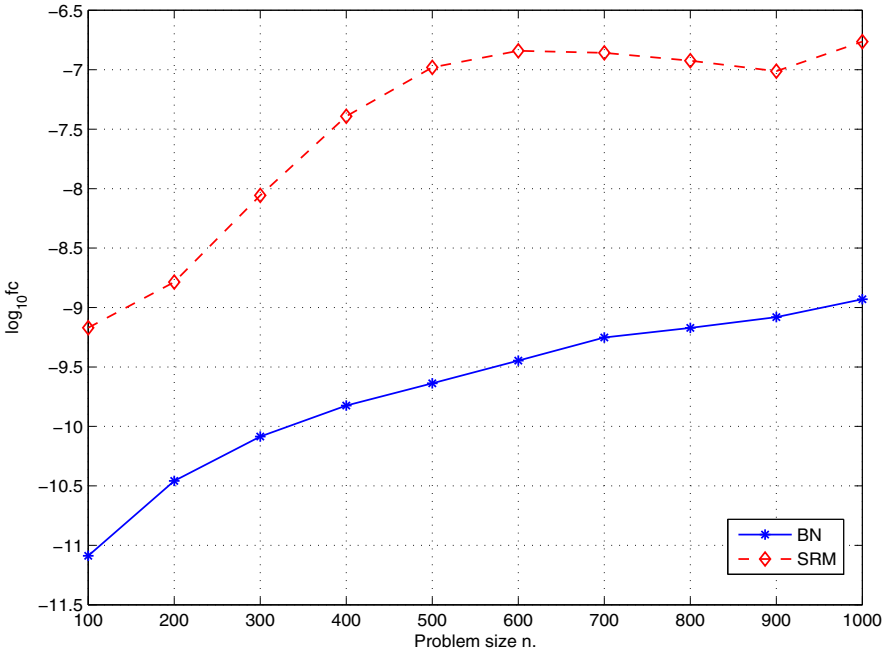

FIGURE 1. $\log_{10} fc$ of the SRM and the BN methods with respect to the problem size n.

to solve the following general SOCLCP on the product of multiple second-order cones:

$$\text{Find } \mathbf{x} \in \mathcal{K} \text{ such that } M\mathbf{x} + \mathbf{q} \in \mathcal{K} \quad \text{and} \quad \mathbf{x}^\top (M\mathbf{x} + \mathbf{q}) = 0, \tag{6.1}$$

where $\mathcal{K} = \mathcal{K}^{n_1} \times \mathcal{K}^{n_2} \times \cdots \times \mathcal{K}^{n_m}$ with $n_i \geq 1$ and $\sum_{i=1}^m n_i = n$. One natural idea is to decouple (6.1) into a sequence of SOCLCPs (1.3), in which the following equivalent relation [14] is helpful

$$\begin{aligned} &\mathbf{x} \in \mathcal{K},\ \mathbf{g} \in \mathcal{K} \text{ and } \mathbf{x}^\top \mathbf{g} = 0 \\ \Longleftrightarrow\ &\mathbf{x}^i \in \mathcal{K}^{n_i},\ \mathbf{g}^i \in \mathcal{K}^{n_i} \text{ and } (\mathbf{x}^i)^\top \mathbf{g}^i = 0,\ i = 1, 2, \ldots, m, \end{aligned} \tag{6.2}$$

where $\mathbf{x} = [(\mathbf{x}^1)^\top, \ldots, (\mathbf{x}^m)^\top]^\top \in \mathbb{R}^n$ with $\mathbf{x}^i \in \mathbb{R}^{n_i}$ being subvectors of $\mathbf{x}$. To apply (6.2) for solving (6.1), we should note that each subvector $\mathbf{g}^i \in \mathbb{R}^{n_i}$ of $\mathbf{g} := M\mathbf{x} + \mathbf{q}$ also involves the subvectors $\mathbf{x}^j \in \mathbb{R}^{n_j}$ for $j \neq i$. The matrix splitting method, which is widely used in the classical LCP [7], is an effective approach to get around that trouble, and has been employed in [15] to solve (6.1) with a symmetric and positive definite M. In particular, the block successive over-relaxation (SOR) method [15] first splits the matrix M as

$$M = B + C = \begin{bmatrix} B_{11} & & & \\ B_{21} & B_{22} & & \\ \vdots & \vdots & \ddots & \\ B_{m1} & \cdots & \cdots & B_{mm} \end{bmatrix} + C, \quad B_{ij} \in \mathbb{R}^{n_i \times n_j}, \tag{6.3}$$

and then solves problem (6.1) by the following iterative procedure ([15, Algorithm 2.1]):

Algorithm 4. The basic matrix-splitting method [15] for SOCLCP (6.1).

Step 1: Choose $\mathbf{x}^{(0)} \in \mathcal{K}$; set $k = 0$.
Step 2: Find $\mathbf{x}^{(k+1)} \in \mathcal{K}$ such that

$$B\mathbf{x}^{(k+1)} + \mathbf{q}^{(k)} \in \mathcal{K} \text{ and } (\mathbf{x}^{(k+1)})^\top (B\mathbf{x}^{(k+1)} + \mathbf{q}^{(k)}) = 0,$$

where $\mathbf{q}^{(k)} := \mathbf{q} + C\mathbf{x}^{(k)}$.
Step 3: Stop and return the approximate solution $\mathbf{x}^{(k+1)}$ if the given stopping criterion is met; otherwise, set $k = k + 1$ and goto *Step 2*.

The convergence is established in [15]; moreover, with the block structure of B in (6.3), it is known [15] by (6.2) that the solution $\mathbf{x} \in \mathcal{K}$ in Step 2 of Algorithm 4 can be equivalently obtained via solving sequentially the following subproblems for $i = 1, 2, \ldots, m$:

$$\mathbf{x}^i \in \mathcal{K}^{n_i},\ B_{ii}\mathbf{x}^i + \mathbf{r}_i^{(k)} \in \mathcal{K}^{n_i},\ (\mathbf{x}^i)^\top (B_{ii}\mathbf{x}^i + \mathbf{r}_i^{(k)}) = 0, \tag{6.4}$$

where

$$\mathbf{r}_i^{(k)} := \begin{cases} \mathbf{q}_1^{(k)}, & \text{if } i = 1, \\ \sum_{j=1}^{i-1} B_{ij}(\mathbf{x}^{(k+1)})^j + \mathbf{q}_i^{(k)}, & \text{if } i > 1, \end{cases}$$

and $\mathbf{q}^{(k)} = \mathbf{q} + C\mathbf{x}^{(k)} = [(\mathbf{q}_1^{(k)})^\top, \ldots, (\mathbf{q}_m^{(k)})^\top]^\top$ with $\mathbf{q}_i^{(k)} \in \mathbb{R}^{n_i}$. It is clear that for each i in (6.4), $\mathbf{x}^i$ is the solution of $LCP(B_{ii}, \mathcal{K}^{n_i}, \mathbf{r}_i^{(k)})$, for which our current bisection-Newton algorithm is applicable. In [15], in order to solve each subproblem (6.4) efficiently, the diagonal block matrices B_{ii} are assumed to be of the special form

$$B_{ii} = \begin{bmatrix} b_1 & \mathbf{0}^\top \\ \mathbf{b}_2 & B_3 \end{bmatrix}, \quad b_1 \in \mathbb{R},\ \mathbf{b}_2 \in \mathbb{R}^{n_i - 1},\ B_3 \in \mathbb{R}^{(n_i-1)\times(n_i-1)},$$

which potentially restricts the application and the efficiency of the matrix-splitting method. As our bisection-Newton method can efficiently solve each subproblem (6.4), it seems that the matrix-splitting method, equipped with our bisection-Newton algorithm, is an appealing approach for the SOCLCP (6.1), and the detailed implementation, theoretical properties and numerical investigation will be part of our future work.

Acknowledgments

The authors would like to thank two anonymous referees for the careful reading and helpful comments and suggestions that have improved the presentation of the manuscript. One referee also motivated us to consider applying our bisection-Newton method to solve the general SOCLCP (6.1). We also thank Dr. Zheng-Hai Huang for providing us the code for the smoothing-Newton algorithm [17] for our testing in Section 5.

References

[1] Alfred Auslender, *Variational inequalities over the cone of semidefinite positive symmetric matrices and over the Lorentz cone*, Optim. Methods Softw. **18** (2003), no. 4, 359–376, DOI 10.1080/1055678031000122586. The Second Japanese-Sino Optimization Meeting, Part II (Kyoto, 2002). MR2019034 (2004i:90101)

[2] James V. Burke and Song Xu, *The global linear convergence of a noninterior path-following algorithm for linear complementarity problems*, Math. Oper. Res. **23** (1998), no. 3, 719–734, DOI 10.1287/moor.23.3.719. MR1653734 (99h:90103)

[3] Jein-Shan Chen and Shaohua Pan, *A descent method for a reformulation of the second-order cone complementarity problem*, J. Comput. Appl. Math. **213** (2008), no. 2, 547–558, DOI 10.1016/j.cam.2007.01.029. MR2392576 (2008k:90131)

[4] X. Chen, L. Qi, and D. Sun, *Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities*, Math. Comp. **67** (1998), no. 222, 519–540, DOI 10.1090/S0025-5718-98-00932-6. MR1458218 (98g:90034)

[5] X. D. Chen, D. Sun, and J. Sun, *Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementarity problems*, Comput. Optim. Appl. **25** (2003), no. 1-3, 39–56, DOI 10.1023/A:1022996819381. A tribute to Elijah (Lucien) Polak. MR1996662 (2004f:90142)

[6] Jein-Shan Chen and Paul Tseng, *An unconstrained smooth minimization reformulation of the second-order cone complementarity problem*, Math. Program. **104** (2005), no. 2-3, Ser. B, 293–327, DOI 10.1007/s10107-005-0617-0. MR2179239 (2006g:90123)

[7] Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone, *The linear complementarity problem*, Computer Science and Scientific Computing, Academic Press Inc., Boston, MA, 1992. MR1150683 (93f:90001)

[8] Masao Fukushima, Zhi-Quan Luo, and Paul Tseng, *Smoothing functions for second-order-cone complementarity problems*, SIAM J. Optim. **12** (2001/02), no. 2, 436–460 (electronic), DOI 10.1137/S1052623400380365. MR1885570 (2002m:90084)

[9] Gene H. Golub and Charles F. Van Loan, *Matrix computations*, 3rd ed., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 1996. MR1417720 (97g:65006)

[10] M. Seetharama Gowda and Yoon Song, *On semidefinite linear complementarity problems*, Math. Program. **88** (2000), no. 3, Ser. A, 575–587, DOI 10.1007/PL00011387. MR1782157 (2001e:90111)

[11] M. Seetharama Gowda and Roman Sznajder, *Automorphism invariance of* **P**- *and* **GUS**-*properties of linear transformations on Euclidean Jordan algebras*, Math. Oper. Res. **31** (2006), no. 1, 109–123, DOI 10.1287/moor.1050.0182. MR2205521 (2007i:90086)

[12] M. Seetharama Gowda and R. Sznajder, *Some global uniqueness and solvability results for linear complementarity problems over symmetric cones*, SIAM J. Optim. **18** (2007), no. 2, 461–481, DOI 10.1137/060653640. MR2338447 (2008k:49020)

[13] M. Seetharama Gowda, Roman Sznajder, and J. Tao, *Some* **P**-*properties for linear transformations on Euclidean Jordan algebras*, Linear Algebra Appl. **393** (2004), 203–232, DOI 10.1016/j.laa.2004.03.028. MR2098615 (2005h:17059)

[14] Shunsuke Hayashi, Nobuo Yamashita, and Masao Fukushima, *A combined smoothing and regularization method for monotone second-order cone complementarity problems*, SIAM J. Optim. **15** (2004/05), no. 2, 593–615 (electronic), DOI 10.1137/S1052623403421516. MR2144183 (2006f:90081)

[15] Shunsuke Hayashi, Takahiro Yamaguchi, Nobuo Yamashita, and Masao Fukushima, *A matrix-splitting method for symmetric affine second-order cone complementarity problems*, J. Comput. Appl. Math. **175** (2005), no. 2, 335–353, DOI 10.1016/j.cam.2004.05.018. MR2108579 (2005h:90140)

[16] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal, *Convex analysis and minimization algorithms. I*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 305, Springer-Verlag, Berlin, 1993. Fundamentals. MR1261420 (95m:90001)

[17] Zheng-Hai Huang and Tie Ni, *Smoothing algorithms for complementarity problems over symmetric cones*, Comput. Optim. Appl. **45** (2010), no. 3, 557–579, DOI 10.1007/s10589-008-9180-y. MR2600896 (2011a:90119)

[18] Christian Kanzow, *Some noninterior continuation methods for linear complementarity problems*, SIAM J. Matrix Anal. Appl. **17** (1996), no. 4, 851–868, DOI 10.1137/S0895479894273134. MR1410705 (97g:90148)

[19] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, *A unified approach to interior point algorithms for linear complementarity problems*, Lecture Notes in Computer Science, vol. 538, Springer-Verlag, Berlin, 1991. MR1226025 (94e:90004)

[20] Masakazu Kojima, Susumu Shindoh, and Shinji Hara, *Interior-point methods for the monotone semidefinite linear complementarity problem in symmetric matrices*, SIAM J. Optim. **7** (1997), no. 1, 86–125, DOI 10.1137/S1052623494269035. MR1430559 (97m:90088)

[21] Lingchen Kong, Jie Sun, and Naihua Xiu, *A regularized smoothing Newton method for symmetric cone complementarity problems*, SIAM J. Optim. **19** (2008), no. 3, 1028–1047, DOI 10.1137/060676775. MR2460730 (2009i:90150)

[22] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Anal. Appl. **17** (1996), no. 4, 789–821, DOI 10.1137/S0895479895281484. MR1410702 (97k:65091)

[23] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK users' guide*, Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods. Software, Environments, and Tools, vol. 6, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998. MR1621681

[24] James M. Ortega, *The Newton-Kantorovich theorem*, Amer. Math. Monthly **75** (1968), 658–660. MR0231218 (37 #6773)

[25] Shaohua Pan and Jein-Shan Chen, *A damped Gauss-Newton method for the second-order cone complementarity problem*, Appl. Math. Optim. **59** (2009), no. 3, 293–318, DOI 10.1007/s00245-008-9054-9. MR2491700 (2010d:90117)

[26] Jong-Shi Pang, Defeng Sun, and Jie Sun, *Semismooth homeomorphisms and strong stability of semidefinite and Lorentz complementarity problems*, Math. Oper. Res. **28** (2003), no. 1, 39–63, DOI 10.1287/moor.28.1.39.14258. MR1961266 (2004a:90094)

[27] G. W. Stewart, *Matrix algorithms. Vol. II*, Eigensystems, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001. MR1853468

[28] J. Stoer and R. Bulirsch, *Introduction to numerical analysis, 2nd ed.*, Springer-Verlag, NY, 1991.

[29] K. C. Toh, M. J. Todd, and R. H. Tütüncü, *SDPT3—a MATLAB software package for semidefinite programming, version 1.3*, Optim. Methods Softw. **11/12** (1999), no. 1-4, 545–581, DOI 10.1080/10556789908805762. Interior point methods. MR1778429

[30] W. H. Yang and Y. Yuan, *The GUS-property of second-order cone linear complementarity problems,* Math. Programming Ser. A, DOI: 10.1007/s10107-012-0523-1.

[31] Akiko Yoshise, *Interior point trajectories and a homogeneous model for nonlinear complementarity problems over symmetric cones*, SIAM J. Optim. **17** (2006), no. 4, 1129–1153 (electronic), DOI 10.1137/04061427X. MR2274506 (2007k:90137)

Department of Applied Mathematics, Shanghai University of Finance and Economics, 777 Guoding Road, Shanghai 200433, People's Republic of China

E-mail address: longzlh@gmail.com

Corresponding author. School of Mathematical Sciences, Fudan University, Shanghai, 200433, People's Republic of China

E-mail address: whyang@fudan.edu.cn