Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 17, No. 4, pp. 565–575, 2021

REGULARIZED PARALLEL MATRIX-SPLITTING METHOD FOR SYMMETRIC LINEAR SECOND-ORDER CONE COMPLEMENTARITY PROBLEMS*

Guoxin Wang and Gui-Hua Lin†

Abstract: In this paper, we present a regularized parallel matrix-splitting method for solving the linear second-order cone complementarity problems. The proposed method has several advantages over some previous works: (i) The method can be implemented in parallel. (ii) The method is convergent under suitable conditions for the symmetric and positive semidefinite problems. (iii) The subproblems can be solved explicitly. Preliminary numerical experience indicates that the proposed method may be effective for large scale problems.

Key words: *regularization, matrix-splitting, parallel iterative method, second-order cone complementarity problem*

Mathematics Subject Classification: *90C33, 65F10*

1 Introduction

Consider the following linear second-order cone complementarity problem (LSOCCP):

$$\begin{array}{ll} \text{find} & z \in \mathbb{R}^n \\ \text{such that} & z \in \mathcal{K}, \quad Mz + q \in \mathcal{K}, \quad z^T(Mz+q) = 0. \end{array} \tag{1.1}$$

Here, $M \in \mathbb{R}^{n\times n}$ is a given matrix, $q \in \mathbb{R}^n$ is a given vector, $\mathcal{K} := \mathcal{K}^{n_1} \times \mathcal{K}^{n_2} \times \cdots \times \mathcal{K}^{n_m}$ is a convex cone with $n_1 + n_2 + \cdots + n_m = n$ and

$$\mathcal{K}^{n_i} := \{(x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n_i - 1} |\ x_1 \geq \|x_2\|\}, \quad i = 1, \cdots, m,$$

where $\|\cdot\|$ stands for the Euclidean norm. This problem can be viewed as a generalization of the classical linear complementarity problem (LCP)[1] and it has many practical applications in filter design, antenna array design, robust Nash equilibria, incremental quasi-static problems with unilateral frictional contact, and so on. Moreover, the LSOCCP (1.1) coincides with the Karush-Kuhn-Tucker conditions of the second-order-cone quadratic programming

*This work was supported in part by the NSFC Grants (No. 12071280, No.11901320), the Humanityand Social Science Foundation of Ministry of Education of China (No. 15YJA630034), the InnovationProgram of Shanghai Municipal Education Commission (No. 14ZS086), and the Key Scientific Research Projects of colleges and universities in Henan Province (No. 19A110027).

†Corresponding author.

problems, which has lots of applications in engineering design and portfolio optimization, etc. Throughout this paper, we always assume that M is a symmetric matrix and the LSOCCP (1.1) has a solution.

There have been proposed a list of methods for solving the LSOCCP (1.1) such as the merit function methods [3, 2], the smoothing Newton methods [4, 5, 6], the semismooth Newton methods [7, 8], and the interior-point methods [10, 9]. Recently, with the advent of the era of big data, much attention has been paid on a class of iterative methods called the matrix-splitting methods. Matrix-splitting methods were originally developed for systems of linear equations [11, 12] and, subsequently, they were extended to linear variational inequality and complementarity problems in [13, 15, 14] and to LSOCCP in [16, 17, 18] successively. In particular, Hayashi et al. [16] extended the matrix-splitting method given in [1] for LCP and presented a block successive overrelaxation method (BSOR) for symmetric LSOCCP with positive definite matrix. Xu and Zeng [17] extended the idea of multisplitting for symmetric LCP to symmetric LSOCCP with symmetric positive definite matrix. Zhang and Yang [18] presented another BSOR method for symmetric LSOCCP with positive definite matrix and, for the case where the matrix is only positive semidefinite, they suggested a strategy with a constant regularization parameter.

In this paper, we present a regularized parallel matrix-splitting method for solving the LSOCCP (1.1) in which the matrix M is assumed to be symmetric and positive semidefinite. Compared with the methods introduced above, our method has the following advantages:

The method can be implemented in parallel.

The method is convergent under suitable conditions for the symmetric and positive semidefinite problems.

The subproblems involved in the methods can be solved explicitly.

Particularly, our regularized method is different from the one given in [18] in that the regularization parameter in our method varies with the iterative step.

The paper is organized as follows: In Section 2, we briefly review the basic matrix-splitting method in [16] for symmetric LSOCCP. In Section 3, we describe our regularized parallel matrix-splitting method for the LSOCCP (1.1) with symmetric and positive semidefinite matrix and give some convergent results for the proposed method. In Section 4, we discuss how to solve the subproblems. In Section 5, we report some numerical results with the proposed method.

$\boxed{2}$ Basic Matrix-Splitting Method for Symmetric LSOCCP

In this section, we briefly review the basic matrix-splitting method presented in [16] for symmetric LSOCCP with positive definite matrix. Let (B, C) be a splitting of M, that is, $M = B + C$, where B and C do not need to be symmetric. The basic matrix-splitting method for symmetric LSOCCP is stated as follows.

Algorithm 2.1.

S0. Choose a splitting (B, C) of M, an initial point $z^0 \in \mathcal{K}$, and a tolerance $\varepsilon > 0$. Set $k := 0$.

S1. Solve the LSOCCP

$$\begin{array}{ll} \text{find} & z \in \mathbb{R}^n \\ \text{such that} & z \in \mathcal{K}, \quad Bz + Cz^k + q \in \mathcal{K}, \quad z^T(Bz + Cz^k + q) = 0 \end{array} \tag{2.1}$$

to get a solution z^{k+1}.

S2. If $\|z^{k+1} - z^k\| \le \varepsilon$, terminate. Otherwise, return to S1 with k replaced by $k+1$.

In order to guarantee Algorithm 2.1 to be well defined, we next recall some concepts. The matrix M is called a $\mathcal{K}$-Q matrix if the LSOCCP (1.1) has a solution for any $q \in \mathbb{R}^n$. A splitting (B, C) is called a $\mathcal{K}$-Q splitting if B is a $\mathcal{K}$-Q matrix. Therefore, if (B, C) is a $\mathcal{K}$-Q splitting, then LSOCCP (2.1) always has a solution. Moreover, a splitting (B, C) is said to be (weakly) regular if $B - C$ is positive (semi-)definite. The regularity of a splitting plays an important role in convergence analysis of Algorithm 2.1. The matrix M is said to be (strictly) $\mathcal{K}$-copositive if $z^T Mz \ge 0$ (> 0) for all $z \in \mathcal{K}\backslash\{0\}$. Apparently, every positive semidefinite matrix is $\mathcal{K}$-copositive and every positive definite matrix is strictly $\mathcal{K}$-copositive. The $\mathcal{K}$-copositiveness of M can be used to show the boundedness of the sequence $\{z^k\}$ generated by Algorithm 2.1.

Note that the subproblem (2.1) is not required to have a unique solution. If multiple solutions exist, any one can be picked as z^{k+1}. In addition, each subproblem (2.1) should be relatively easy to solve.

Consider the quadratic second-order cone programming problem

$$\begin{array}{ll} \min & f(z) := \frac{1}{2} z^T Mz + q^T z \\ \text{s.t.} & z \in \mathcal{K}. \end{array} \tag{2.2}$$

Since M is symmetric, the LSOCCP (1.1) coincides with the KKT conditions of problem (2.2). Since problem (2.2) apparently satisfies the Slater's constraint qualification, by the positive semi-definiteness of M, the LSOCCP (1.1) is equivalent to problem (2.2). By using this relationship, Hayashi et al. [16] obtained the following lemma, which gives a sufficient condition about the existence of accumulation points of the sequence generated by Algorithm 2.1.

Lemma 2.1. *Let M be symmetric and strictly $\mathcal{K}$-copositive. Then, for any initial point $z^0 \in \mathcal{K}$, the sequence $\{z^k\}$ generated by Algorithm* 2.1 *with regular $\mathcal{K}$-Q splitting (B, C) is bounded and its arbitrary accumulation point is a solution of the LSOCCP* (1.1).

3 Regularized Parallel Method for Symmetric LSOCCP

The matrix-splitting methods proposed in [16, 17] require M to be symmetric positive definite, while the regularized method given in [18] for the symmetric positive semidefinite case assumes that the regularization parameter is sufficiently small but fixed. In this section, we suggest a regularized parallel method for symmetric LSOCCP with positive semidefinite, in which the regularization parameter decreases to zero monotonically.

Suppose that the matrix M is symmetric and positive semidefinite. Then, for any $\delta > 0$, $M + \delta I$ is symmetric and positive definite and hence the LSOCCP (1.1) with M replaced by $M + \delta I$ has a unique solution, denoted by $z^*(\delta)$. By passing $\delta \to 0$, we may expect that $z^*(\delta)$ converges to a solution of the original LSOCCP (1.1). This is the regularization process mentioned above.

In order to solve problems in a parallel way, the splitting (B, C) used in the Jacobi method is represented as

$$B = \lambda I, \quad C = M - B.$$

Choose the parameter λ such that it is greater strictly than the maximum eigenvalue of $\frac{M}{2}$. Then both B and $B - C$ are positive definite matrices and so (B, C) is a regular $\mathcal{K}$-Q splitting of M. Since M is symmetric positive semidefinite, we have $\lambda > 0$.

We now describe the regularized parallel algorithm for the LSOCCP (1.1) as follows.

Algorithm 3.1.

S0. Choose a constant $\delta_0 > 0$, an initial point $z^0 \in \mathcal{K}$, and a tolerance $\varepsilon > 0$. Set $k := 0$.

S1. Denote by $B^k := \lambda I + \delta_k I$ and $C := M - \lambda I$. Solve the LSOCCP

$$\begin{array}{rl} \text{find} & z \in \mathbb{R}^n \\ \text{such that} & z \in \mathcal{K}, \quad B^k z + C z^k + q \in \mathcal{K}, \quad z^T(B^k z + C z^k + q) = 0 \end{array} \tag{3.1}$$

to get a solution z^{k+1}.

S2. Terminate if the stopping rule is satisfied. Otherwise, return to S1 with $\delta_{k+1} \in (0, \delta_k)$ and $k := k + 1$.

Note that, since $B^k - C = 2\lambda I - M + \delta_k I$, both B^k and $B^k - C$ are positive definite for any $\delta_k > 0$ and hence (B^k, C) is always a regular $\mathcal{K}$-Q splitting of $M + \delta_k I$. As a result, problem (3.1) has a unique solution for each k.

In [18], the stopping criteria is suggested to use the residue of the original complementarity problem, that is, the iteration is terminated whenever

$$\rho := \sum_{i=1}^{m} \max\{\|z_{i,2}^{k+1}\| - z_{i,1}^{k+1}, 0\} + \sum_{i=1}^{m} \max\{\|w_{i,2}^{k+1}\| - w_{i,1}^{k+1}, 0\} + |(z^{k+1})^T w^{k+1}| \leq \varepsilon,$$

where

$$z^{k+1} = \begin{pmatrix} z_1^{k+1} \\ z_2^{k+1} \\ \vdots \\ z_m^{k+1} \end{pmatrix}, \qquad w^{k+1} := M z^{k+1} + q = \begin{pmatrix} w_1^{k+1} \\ w_2^{k+1} \\ \vdots \\ w_m^{k+1} \end{pmatrix}$$

with $z_i^{k+1} = (z_{i,1}^{k+1}, z_{i,2}^{k+1}) \in \mathbb{R} \times \mathbb{R}^{n_i - 1}$ and $w_i^{k+1} = (w_{i,1}^{k+1}, w_{i,2}^{k+1}) \in \mathbb{R} \times \mathbb{R}^{n_i - 1}$ for each i. In our numerical experiments given in Section 5, we used the relative residue defined by

$$\rho_r := \frac{\rho}{1 + \|M\|_1 + \|q\|_1} \leq \varepsilon,$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm of matrix or vector, as the stopping criteria in S2 and obtained satisfactory numerical results.

If we partition the matrix M as

$$M = \begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1m} \\ M_{21} & M_{22} & \cdots & M_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ M_{m1} & M_{m2} & \cdots & M_{mm} \end{pmatrix},$$

where $M_{ij} \in \mathbb{R}^{n_i \times n_j} (i, j = 1, 2, \cdots, m)$, and rewrite z and q as

$$z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{pmatrix}, \quad q = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_m \end{pmatrix}$$

respectively, where $z_i \in \mathbb{R}^{n_i}$, $q_i \in \mathbb{R}^{n_i}$, $i = 1, 2, \cdots, m$, then we can solve problem (3.1) in a parallel way. In fact, by exploiting the decomposable structure of the SOC conditions, (3.1) is equivalent to the following subproblems: For $i = 1, 2, \cdots, m$,

$$\begin{array}{ll} \text{find} & z_i \in \mathbb{R}^{n_i} \\ \text{such that} & z_i \in \mathcal{K}^{n_i}, \quad (\lambda + \delta_k) z_i + h_i^k \in \mathcal{K}^{n_i}, \quad (z_i)^T [(\lambda + \delta_k) z_i + h_i^k] = 0, \end{array} \tag{3.2}$$

where

$$h_i^k := (M_{ii} - \lambda I) z_i^k + \sum_{j=1, j \neq i}^{m} M_{ij} z_j^k + q_i.$$

Next we discuss the convergence properties of Algorithm 3.1. Consider the problem with a parameter $\delta > 0$

$$\begin{array}{ll} \min & f_\delta(z) := \dfrac{1}{2} z^T (M + \delta I) z + q^T z \\ \text{s.t.} & z \in \mathcal{K}. \end{array} \tag{3.3}$$

Since $M + \delta I$ is positive definite, problem (3.3) has a unique optimal solution. Note that $f_\delta(z) \geq f(z)$ for any z, where $f(z)$ is given by (2.2). By the assumption that the solution set of (1.1) is nonempty, f_δ is bounded below over the cone $\mathcal{K}$. We further have the following lemma.

Lemma 3.1. *Let $\{z^k\}$ be a sequence generated by Algorithm 3.1. Then, for each k, we have*

$$f_{\delta_k}(z^k) - f_{\delta_{k+1}}(z^{k+1}) \geq \frac{1}{2} (z^k - z^{k+1})^T (B^k - C)(z^k - z^{k+1}) \geq 0. \tag{3.4}$$

Proof. By (3.1) and (3.3), we have

$$\begin{aligned} & f_{\delta_k}(z^k) - f_{\delta_{k+1}}(z^{k+1}) \\ &\quad = (z^k - z^{k+1})^T (q + M z^{k+1}) + \frac{1}{2} (z^k - z^{k+1})^T M (z^k - z^{k+1}) + \frac{1}{2} \delta_k \|z^k\|^2 - \frac{1}{2} \delta_{k+1} \|z^{k+1}\|^2 \\ &\quad = \frac{1}{2} (z^k - z^{k+1})^T (B^k - C)(z^k - z^{k+1}) + (z^k - z^{k+1})^T (q + C z^k + B^k z^{k+1} - \delta_k z^{k+1}) \\ &\qquad - \frac{1}{2} \delta_k \|z^k - z^{k+1}\|^2 + \frac{1}{2} \delta_k \|z^k\|^2 - \frac{1}{2} \delta_{k+1} \|z^{k+1}\|^2 \\ &\quad = \frac{1}{2} (z^k - z^{k+1})^T (B^k - C)(z^k - z^{k+1}) + (z^k)^T (q + C z^k + B^k z^{k+1}) + \frac{\delta_k - \delta_{k+1}}{2} \|z^{k+1}\|^2, \end{aligned}$$

where the third equality follows from the fact that z^{k+1} solves problem (3.1). Since $z^k \in \mathcal{K}$, $q + C z^k + B^k z^{k+1} \in \mathcal{K}$, and $\delta_k > \delta_{k+1}$, we have

$$f_{\delta_k}(z^k) - f_{\delta_{k+1}}(z^{k+1}) \geq \frac{1}{2} (z^k - z^{k+1})^T (B^k - C)(z^k - z^{k+1}).$$

The second inequality in (3.4) holds from the positive definiteness of $B^k - C$ immediately. $\square$

Using the above lemma, we establish the main theorem in this section.

Theorem 3.2. *Let M be a symmetric $\mathcal{K}$-copositive matrix. Suppose that*

$$0 \neq z \in \mathcal{K},\ Mz \in \mathcal{K},\ z^T Mz = 0 \quad \Longrightarrow \quad q^T z > 0. \tag{3.5}$$

Then the sequence $\{z^k\}$ generated by Algorithm 3.1 *is bounded and any accumulation point solves the LSOCCP* (1.1).

Proof. We first show the boundedness of $\{z^k\}$. By Lemma 3.1, the sequence $\{f_{\delta_k}(z^k)\}$ is nonincreasing. We also know that f_δ is bounded below and hence $\{f_{\delta_k}(z^k)\}$ converges. By the uniform positive definiteness of $B^k - C$ for each k sufficiently large, we have from (3.4) that $\{z^k - z^{k+1}\}$ converges to zero.

Suppose that the sequence $\{z^k\}$ is unbounded. Without loss of generality, we may assume that $\|z^k\| \to +\infty$. Consider the normalized sequence $\{z^k/\|z^k\|\}$ and suppose that it has an accumulation point $z^* \in \mathcal{K}\backslash\{0\}$. Let $\{z^{k_i+1}/\|z^{k_i+1}\|\}$ be a subsequence converging to z^*. Since z^{k_i+1} solves the LSOCCP (3.1), we have

$$(M + \delta_{k_i} I)z^{k_i+1} + C(z^{k_i} - z^{k_i+1}) + q \in \mathcal{K}, \tag{3.6}$$
$$z^{k_i+1} \in \mathcal{K}, \tag{3.7}$$
$$(z^{k_i+1})^T((M + \delta_{k_i} I)z^{k_i+1} + C(z^{k_i} - z^{k_i+1}) + q) = 0. \tag{3.8}$$

Dividing (3.6) and (3.7) by $\|z^{k_i+1}\|$ and (3.8) by $\|z^{k_i+1}\|^2$ respectively and letting $i \to +\infty$, we have

$$z^* \in \mathcal{K}, \quad Mz^* \in \mathcal{K}, \quad (z^*)^T Mz^* = 0. \tag{3.9}$$

Noting that M is $\mathcal{K}$-copositive and hence $M + \delta_{k_i} I$ is strictly $\mathcal{K}$-copositive, we have

$$\begin{array}{rcl} 0 & = & (z^{k_i+1})^T((M + \delta_{k_i} I)z^{k_i+1} + C(z^{k_i} - z^{k_i+1}) + q) \\ & > & (z^{k_i+1})^T(C(z^{k_i} - z^{k_i+1}) + q). \end{array}$$

Dividing the last inequality by $\|z^{k_i+1}\|$ and letting $i \to +\infty$, we have $q^T z^* \leq 0$. This, together with (3.9), yields a contradiction to (3.5). Consequently, the sequence $\{z^k\}$ must be bounded.

We next show that any accumulation point of $\{z^k\}$ solves the LSOCCP (1.1). Let $\tilde{z}$ be an arbitrary accumulation point of $\{z^k\}$, that is, there exists a subsequence $\{z^{k_i}\}$ of $\{z^k\}$ converging to $\tilde{z}$. From the above analysis, $\{z^k - z^{k+1}\}$ converges to zero and, therefore, the sequence $\{z^{k_i+1}\}$ also converges to $\tilde{z}$. Since $\{z^{k_i+1}\}$ satisfies (3.6)–(3.8), passing to the limit $i \to +\infty$ reveals that $\tilde{z}$ is a solution of the LSOCCP (1.1).

This completes the proof. □

A corollary is that, when M is symmetric and positive semidefinite, the conclusion of Theorem 3.2 still holds under the condition (3.5).

4 Solving Subproblems

Consider the subproblems (3.2), which can be unified as

$$\begin{array}{rl} \text{find} & z \in \mathbb{R}^l \\ \text{such that} & z \in \mathcal{K}^l, \quad bz + r \in \mathcal{K}^l, \quad z^T(bz + r) = 0, \end{array} \tag{4.1}$$

where $b > 0$. The LSOCCP (4.1) has a unique solution, say z^*. In particular, when $l = 1$, the solution of (4.1) can be easily obtained as $z^* = \max(0, -r/b)$. For general cases, inspired by Proposition 3.3 in [5], we have

$$z^* = \begin{cases} 0 & \text{if } r \in \mathcal{K}^l, \\ -b^{-1}r & \text{if } -b^{-1}r \in \mathcal{K}^l, \\ \frac{r_1 - \|r_2\|}{2b} \begin{pmatrix} -1 \\ \|r_2\|^{-1} r_2 \end{pmatrix} & \text{otherwise.} \end{cases} \tag{4.2}$$

In fact, the first two cases are easy to get. Now we suppose that $r \notin \mathcal{K}^l$ and $-b^{-1}r \notin \mathcal{K}^l$. It follows that $z^* \in \text{bd } \mathcal{K}^l \backslash \{0\}$ and $(z^*)^T(bz^* + r) = 0$. Then we have $bz^* + r \in \text{bd } \mathcal{K}^l$. Denote z^* and $bz^* + r$ as

$$z^* = \beta \begin{pmatrix} 1 \\ w \end{pmatrix}, \quad bz^* + r = \mu \begin{pmatrix} 1 \\ -w \end{pmatrix}$$

respectively, where $\beta > 0$, $\mu \geq 0$, and $w \in \mathbb{R}^{l-1}$ with $\|w\| = 1$. It follows that

$$\begin{pmatrix} \beta b + r_1 \\ \beta b w + r_2 \end{pmatrix} = \begin{pmatrix} \mu \\ -\mu w \end{pmatrix}, \tag{4.3}$$

where $r = (r_1, r_2) \in \mathbb{R} \times \mathbb{R}^{l-1}$. Eliminating μ in (4.3), we have

$$(2\beta b + r_1)w = -r_2.$$

Since $\mu = \beta b + r_1 \geq 0$, $\beta > 0$, $b > 0$, and $\|w\| = 1$, we have

$$\beta = \frac{\|r_2\| - r_1}{2b}, \quad w = -\frac{r_2}{2\beta b + r_1},$$

from which we can get (4.2) immediately.

5 Preliminary Numerical Results

We have tested Algorithm 3.1 for both the symmetric positive definite case and the symmetric positive semidefinite case on a number of examples. More precisely, we mainly considered the following two experiments:

Testing for dense, ill-conditioned symmetric and positive definite matrices.

Testing for dense, ill-conditioned symmetric and positive semidefinite matrices.

In order to show the efficiency of Algorithm 3.1, we compared it with the method 'BSOR_HYYF' proposed in [16] and the methods 'BSOR_BN_L' and 'BSOR_BN_H' proposed in [18]. The programs were run in MATLAB 7.11.0 on a computer with 2.50 GHz CPU and 4GB memory. For each tested problem, the vector q was chosen from the interval $[-1, 1]$ arbitrarily, the initial point was taken to be $z^0 = (1, 0) \in \mathbb{R} \times \mathbb{R}^{n-1}$, and the positive integer pair (n, m) satisfied $n_1 = \cdots = n_m = \frac{n}{m}$. For the matrix M, similarly as in [18], we first generated a matrix $\bar{M}$ as

$$\bar{M} := \text{diag}([1 : \delta : 1 + (n-1) * \delta].\hat{\ }0.5) * \text{orth}(\text{randn}(n, n)),$$

where $\delta := \frac{cond}{n}$ with the condition number $cond = 10^6$, and then we constructed a dense, ill-conditioned symmetric and positive definite matrix by $M := \bar{M}^T \bar{M}$, or a dense, symmetric and positive semidefinite matrix by $M := \bar{M}^T T \bar{M}$ with

$$T = \mathrm{diag}\{\underbrace{0, \cdots, 0}_{5}, \underbrace{1, \cdots, 1}_{n-10}, \underbrace{0, \cdots, 0}_{5}\}.$$

In our experiments, we set the stopping criteria $\varepsilon = 10^{-4}$ for $n = 10^4$ or $m = 1000$ and set $\varepsilon = 10^{-6}$ for other cases. In addition, we terminated the iteration whenever the iteration number $k > iter_{max} = 1000$ for all tested methods. For other parameters involved in BSOR_HYYF, BSOR_BN_L, or BSOR_BN_H, we used the same values as in [16] or [18]. For each case, we tested 10 times by generating data randomly. The numerical results are reported in Tables 1 and 2, where 'Iter' denotes the average number of iterations, 'Cpu(s)' denotes the average CPU time in second, ρ_r stands for the relative residue and '$*$' means 'out of memory' (that is, the computer could not deal with the corresponding data).

Table 1: Results for dense, ill-conditioned symmetric and positive definite matrices

		Algorithm 3.1			BSOR_HYYF			BSOR_BN_L			BSOR_BN_H		
n	m	Iter	Cpu(s)	ρ_r	Iter	Cpu(s)	ρ_r	Iter	Cpu(s)	ρ_r	Iter	Cpu(s)	ρ_r
2000	1	47.9	3.937	8.4e−14	7.9	205.367	3.9e−14	6.0	3.122	1.6e−14	10.9	9.298	1.1e−14
	10	47.2	4.055	8.8e−14	10.8	39.434	4.4e−17	703.6	17.045	7.1e−11	414	17.069	1.6e−11
	100	64.3	18.623	4.9e−14	1000	33.620	1.2e−12	1000	48.836	1.4e−8	1000	50.315	6.8e−9
	200	70.5	73.529	6.6e−14	1000	47.273	1.9e−12	1000	101.282	5.7e−8	1000	104.598	2.8e−8
4000	1	47.1	25.842	5.8e−14	7.2	1220.541	2.5e−14	6.0	12.723	1.7e−14	13.7	49.465	2.0e−15
	10	46.0	25.944	4.5e−14	11.6	42.427	1.5e−16	900.7	164.370	5.5e−11	216.7	67.801	8.5e−14
	100	46.3	37.231	5.2e−14	1000	317.671	3.8e−12	1000	87.755	4.1e−9	1000	92.412	2.4e−9
	200	64.5	84.684	5.6e−14	1000	79.918	1.2e−11	1000	139.050	1.9e−8	1000	143.249	9.9e−9
5000	1	49.0	61.569	5.3e−14	7.8	2610.784	2.9e−15	6.2	20.542	1.0e−14	13.6	82.461	9.2e−15
	10	45.8	61.065	5.4e−14	11.4	59.319	9.5e−17	801.9	247.355	4.4e−11	118.8	54.466	4.6e−14
	100	47.7	71.060	4.5e−14	1000	457.463	3.5e−13	1000	117.305	2.5e−9	1000	124.317	1.8e−9
	200	61.3	109.762	4.3e−14	1000	106.376	3.0e−12	1000	170.668	1.2e−8	1000	177.510	7.7e−9
	1000	68.1	1367.015	5.5e−12	1000	201.722	3.4e−11	1000	1285.358	5.0e−7	1000	1282.807	2.3e−7
10000	1	37.9	331.085	2.8e−12	$*$	$*$	$*$	3.3	68.975	3.0e−13	13.5	555.398	1.9e−12
	10	35.8	330.970	3.8e−12	10.0	194.744	9.8e−13	7.4	12.335	1.1e−12	15.4	37.290	3.3e−12
	100	35.4	339.004	3.0e−12	10.1	370.232	3.6e−12	1000	340.837	1.0e−9	1000	374.248	5.5e−10
	200	36.9	367.476	3.3e−12	5.1	741.939	7.4e−12	1000	407.150	3.6e−9	1000	438.763	2.4e−9
	1000	61.3	1609.788	3.0e−12	1000	397.242	2.1e−11	1000	1566.078	1.5e−7	1000	1601.517	6.4e−8

The results given in Tables 1 and 2 show clearly the efficiency of Algorithm 3.1 in terms of the CPU time and the relative accuracy. This may be because the subproblems in Algorithm 3.1 could be solved explicitly, while the subproblems in other methods were solved approximately. From our numerical experience, it is encouraging that the algorithm proposed in this paper is comparable for large scale problems with the existing methods, at least for the cases with dense, positive definite (semidefinite) matrices.

6 Conclusions

We have proposed a regularized parallel matrix-splitting method for solving the LSOCCP (1.1). Compared with the methods given in [16] and [18], the new method has some advantages, as mentioned in Section 1, and our numerical experience indicates that the new method is comparable for large scale problems, especially for the dense, ill-conditioned symmetric and positive definite or semidefinite problems. One future work is to extend the matrix-splitting approach to more general complementarity problems.

Table 2: Results for dense, ill-conditioned symmetric and positive semidefinite matrices

		Algorithm 3.1			BSOR_BN_L		
n	m	Iter	Cpu(s)	ρ_r	Iter	Cpu(s)	ρ_r
2000	1	49.8	3.956	5.7e−14	5.7	2.958	2.5e−14
	10	48.6	4.120	6.5e−14	900.7	25.733	8.6e−11
	100	65.0	19.047	7.8e−14	1000	58.875	1.1e−8
	200	73.1	71.087	7.5e−14	1000	126.203	6.0e−8
4000	1	48.4	24.859	6.1e−14	6.0	12.464	1.1e−14
	10	46.0	25.272	4.4e−14	1000	224.636	8.9e−11
	100	49.7	36.295	4.5e−14	1000	102.814	2.8e−9
	200	65.7	83.620	5.6e−14	1000	167.896	1.7e−8
5000	1	48.8	44.468	3.9e−14	6.2	20.372	7.5e−15
	10	46.8	44.954	4.8e−14	1000	304.592	1.9e−11
	100	48.0	55.942	4.1e−14	1000	116.138	3.1e−9
	200	61.8	100.882	4.9e−14	1000	170.584	1.3e−8
	1000	69.1	1386.068	4.4e−12	1000	1266.544	4.9e−7
10000	1	37.9	331.744	3.0e−12	3.9	82.475	1.8e−13
	10	36.3	332.238	3.3e−12	7.4	12.362	6.3e−13
	100	35.8	339.821	3.1e−12	1000	344.773	7.0e−10
	200	35.3	362.413	3.6e−12	1000	414.510	3.9e−9
	1000	60.5	1537.979	3.7e−12	1000	1574.975	1.5e−7

Acknowledgements. We would like to thank two anonymous referees for their valuable comments and suggestions. We also thank Professor Wei-Hong Yang of Fudan University and Professor Lei-Hong Zhang of Shanghai University of Finance and Economics for generously providing the codes of the methods BSOR_BN_L and BSOR_BN_H given in [18] and the codes of the matrix splitting method BSOR_HYYF given in [16].

References

[1] R.W. Cottle, J.S. Pang and R.E. Stone, *The Linear Complememtarity Problem*, Academic Press, Boston, MA, 1992.

[2] J.S. Chen, Two classes of merit functions for the second-order cone complementarity problem, *Math. Methods Oper. Res.* 64 (2006) 495–519.

[3] J.S. Chen and P. Tseng, An unconstrained smooth minimization reformulation of the second-order cone complementarity problem, *Math. Program.* 104 (2005) 297–327.

[4] X.D. Chen, D. Sun and J. Sun, Complementarity functions and numerical experiments for second-order cone complementarity problems, *Comput. Optim. Appl.* 25 (2003) 39–56.

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

[6] S. Hayashi, N. Yamashita and M. Fukushima, A combined smoothing and regularization method for monotone second-order cone complementarity problems, *SIAM J. Optim.* 15 (2005) 593–615.

[7] C. Kanzow, I. Ferenczi and M. Fukushima, On the local convergence of semismooth Newton methods for linear and nonlinear second-order cone programs without strict complementarity, *SIAM J. Optim.* 20 (2009) 297–320.

[8] S.H. Pan and J.S. Chen, A damped Guass-Newton method for the second-order cone complementarity problem, *Appl. Math. Optim.* 59 (2009) 293–318.

[9] R.D.C. Monteiro and T. Tsuchiya, Polynomial convergence of primal-dual algorithms for the second-order cone programs based on the MZ-family of directions, *Math. Program.* 88 (2000) 61–83.

[10] J.M. Peng, C. Roos and T. Terlaky, A new class of polynomial primal-dual interior-point methods for second-order cone optimization based on self-regular proximities, *SIAM J. Optim.* 13 (2002) 179–203.

[11] D.P. O'Leary and R.E. White, Multisplitting of matrices and parallel solution of linear system, *SIAM J. Algebraic Discrete Methods* 6 (1985) 630–640.

[12] G.H. Golub and J.M. Ortega, *Scientific Computing: An Introduction with Parallel Computing*, Academic Press, San Diego, 1993.

[13] A.N. Iusem, On the convergence of iterative methods for symmetric linear complementarity problems, *Math. Program.* 59 (1993) 33–48.

[14] Z.Q. Luo and P. Tseng, Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem, *SIAM J. Optim.* 2 (1993) 43–54.

[15] N. Machida, M. Fukushima and T. Ibaraki, A multisplitting method for symmetric linear complementarity problems, *J. Comput. Appl. Math.* 62 (1995) 217–227.

[16] S. Hayashi, T. Yamaguchi, N. Yamashita and M. Fukushima, A matrix-splitting method for symmetric affine second-order cone complementarity problem, *J. Comput. Appl. Math.* 175 (2005) 335–353.

[17] H. Xu and J. Zeng, A multisplitting method for symmetrical affine second-order cone complementarity problem, *Computer Math. Appl.* 55 (2008) 459–469.

[18] L.H. Zhang and W.H. Yang, An efficient matrix splitting method for the second-order cone complementarity problem, *SIAM J. Optim.* 24 (2014) 1178–1205.

Manuscript received 4 May 2015
revised 22 March 2016
accepted for publication 10 July 2016

Guoxin Wang
School of Mathematics and Physics
Nanyang Institute of Technology
Nanyang 473004 and School of Management
Shanghai University, Shanghai 200444, China
E-mail address: yuxinwang617@126.com

Gui-Hua Lin
School of Management, Shanghai University
Shanghai 200444, China
E-mail address: guihualin@shu.edu.cn