A Multisplitting Method for Symmetric Linear Complementarity Problems

Naoki Machida,[1] Masao Fukushima[2] and Toshihide Ibaraki[1]

[1]Department of Applied Mathematics and Physics,
Faculty of Engineering, Kyoto University,
Kyoto 606, Japan

[2]Graduate School of Information Science,
Nara Institute of Science and Technology,
Ikoma, Nara 630-01, Japan

February 26, 1994

Abstract

Over the years, many methods for solving the linear complementarity problem (LCP) have been developed. Most of these methods have their origin in solving a system of linear equations. In particular, much attention has recently been paid on the class of iterative methods called the splitting method, which is an extension of the matrix splitting method for solving a system of linear equations such as Jacobi, Gauss-Seidel and SOR methods. Furthermore, as a method for solving a system of linear equations, O'Leary and White have proposed a parallel iterative method called the multisplitting method. This method makes use of a set of different splittings of the coefficient matrix, which may be dealt with independently of each other. The results obtained from those splitting iterations are combined to define the multisplitting iterates. Thus the method may be effectively implemented on multiprocessors.

In this paper, we extend the idea of the multisplitting to the symmetric LCP. In particular, we establish some convergence results for the multisplitting method, which generalize the corresponding convergence results for the splitting method for LCP. We also report some computational results with the proposed method.

Keywords: Linear complementarity problem, splitting, multisplitting, parallel computation.
Abbreviated Title: Multisplitting method for LCP

1 Introduction

We consider the symmetric linear complementarity problem LCP(q, M): Find $z \in R^n$ such that

$$Mz + q \geq 0, \quad z \geq 0, \quad z^T(Mz + q) = 0, \tag{1.1}$$

where $M \in R^{n \times n}$ is a given symmetric matrix and $q \in R^n$ is a given vector. Associated with LCP(q, M) is the following quadratic programming problem:

$$\begin{aligned} \text{minimize} \quad & f(z) = \frac{1}{2} z^T M z + q^T z, \\ \text{subject to} \quad & z \geq 0. \end{aligned} \tag{1.2}$$

It is easily seen that (1.1) is a necessary optimality condition for (1.2), and it is also sufficient if M is positive semidefinite.

Over the years, many methods for solving LCP have been developed. Most of these methods have their origin in the solution of a system of linear equations and may be classified into two categories, pivoting methods and iterative methods. For a comprehensive study of LCP, the reader is referred to the recent book [3] by Cottle, Pang and Stone. Iterative methods, which generate an infinite sequence converging to a solution of the problem, are particularly effective for large and sparse problems. Recently much attention has been paid on the class of iterative methods called the splitting method [1, 2, 4, 8, 9, 10, 15], which is an extension of the matrix splitting method for solving a system of linear equations such as Jacobi, Gauss-Seidel and SOR methods. As to theoretical convergence of the splitting method, Mangassarian [10] established a subsequential convergence result under fairly weak assumptions. The convergence of the entire sequence had been an open question for a long time, but recently it was resolved by Luo and Tseng [9] and Iusem [8] independently.

The multisplitting method was proposed by O'Leary and White [13] as a parallel iterative method for solving a system of linear equations. This method makes use of a set of different splittings of the coefficient matrix, which may be dealt with independently of each other. The results obtained from those splitting iterations are combined to define the multisplitting iterates. Thus the method may be effectively implemented on multiprocessors. More results about the multisplitting method for a system of linear equations can be found in [5, 6, 7, 12, 14, 16, 17, 18]. In this paper, we extend the idea of the multisplitting to the symmetric LCP. In particular, we establish some convergence results for the multisplitting method, which generalize the corresponding subsequential convergence results [3, Theorem 5.3.3 and Lemma 5.3.4] for the splitting method for LCP.

Moreover, we apply the multisplitting method to the parallel successive overrelaxation (SOR) method [11], which is one of the splitting methods for solving the symmetric LCP. By the computational experiments with this method, we also examine the effectiveness of multisplitting.

This paper is organized as follows. In Section 2, we briefly review the splitting method for solving the symmetric LCP. In Section 3, we propose a multisplitting method for solving the symmetric LCP and establish some convergence results for the proposed method. In Section 4, we present an application of the proposed method. In Section 5, we report some computational results with the method presented in Section 4. Finally we conclude the paper in Section 6.

2 Splitting Method for the Symmetric LCP

In this section, we briefly review the splitting method for solving the symmetric LCP. For detail, the readers may refer to Cottle, Pang and Stone [3].

Let M be symmetric and B and C be any matrices satisfying $M = B + C$. Then (B, C) is called a *splitting* of M. Matrix M is called a *Q-matrix* if LCP(q, M) has a solution for all $q \in R^n$. A sufficient condition for M to be a Q-matrix is that M is a strictly copositive matrix, i.e.,

$$z \geq 0, \quad z \neq 0 \quad \Rightarrow \quad z^T M z > 0.$$

A splitting (B, C) is said to be a *Q-splitting* if B is a Q-matrix, while (B, C) is said to be *regular* if $B - C$ is positive definite.

Throughout this section, we assume that (B, C) is a regular Q-splitting of M. We can state the splitting method for LCP(q, M) as follows.

Algorithm 2.1 [3] **(Splitting method for** LCP(q, M)**)**

Step 1. Choose an arbitrary nonnegative vector $z(0) \in R^n$. Let $t := 0$.

Step 2. Given $z(t) \geq 0$, let $z(t+1)$ be an arbitrary solution of LCP$(q(t),B)$:

$$\begin{aligned} Bz + q(t) &\geq 0, \\ z &\geq 0, \\ z^T(Bz + q(t)) &= 0, \end{aligned}$$

where $q(t) = Cz(t) + q$.

Step 3. If $z(t+1) = z(t)$, stop. Otherwise, set $t := t + 1$ and return to Step 2.

The following lemma says that the sequence $\{f(z(t))\}$ generated by Algorithm 2.1 is monotone nonincreasing. This feature is important in analyzing the convergence of Algorithm 2.1.

Lemma 2.1 [3, Lemma 5.3.2] *Let $\{z(t)\}$ be a sequence generated by Algorithm 2.1. Then we have*

$$f(z(t)) - f(z(t+1)) \geq \frac{1}{2}(z(t) - z(t+1))^T (B - C)(z(t) - z(t+1)) \geq 0 \tag{2.1}$$

for all t. Moreover, $f(z(t)) = f(z(t+1))$ if and only if $z(t) = z(t+1)$.

The following convergence theorem is well known for the splitting method for solving the symmetric LCP.

Theorem 2.1 [3, Theorem 5.3.3 and Lemma 5.3.4] *Suppose that*

(a) *$f(z)$ is bounded below on $z \geq 0$;*

(b) *$0 \neq z \geq 0$, $Mz \geq 0$ and $z^T Mz = 0$ imply $q^T z > 0$.*

Then any sequence $\{z(t)\}$ generated by Algorithm 2.1 is bounded and has at least one accumulation point. Moreover, any such accumulation point solves LCP(q, M).

If M is strictly copositive, then assumptions (a) and (b) in Theorem 2.1 are satisfied and hence the convergence of Algorithm 2.1 is guaranteed [3, Theorem 5.3.5]. If M is positive semidefinite, then assumptions (a) and (b) in Theorem 2.1 can be replaced by the condition that there exists z satisfying $Mz + q > 0$ [3, Theorem 5.3.9]. Note that more general results than Theorem 2.1 have recently been obtained by Iusem [8] and Luo and Tseng [9]. However we shall not elaborate on those results here, because the convergence results to be established in the next section is a direct extension of Theorem 2.1 to the multisplitting method for LCP(q, M).

3 A Multisplitting Method for the Symmetric LCP

In this section, we propose a multisplitting method for solving LCP(q, M). First, we give the definition of multisplitting.

Definition 3.1 *Let t be an iteration number and B_k, C_k and $E_k(t), k = 1, \ldots, K$, be $n \times n$ matrices. Then $\{(B_k, C_k, E_k(t)); k = 1, \ldots, K\}$ is called a* multisplitting *of M if*

(1) *for all k, $M = B_k + C_k$ and (B_k, C_k) are regular Q-splittings of M;*

(2) *for each* t, $E_k(t), k = 1, \ldots, K$ *are diagonal and* $\sum_{k=1}^{K} E_k(t) = I$.

Note that this definition is somewhat different from that of the multisplitting for the system of linear equations considered in [13]. Here, the "weight" matrices $E_k(t)$ are allowed to vary with iteration t and may not necessarily satisfy $E_k(t) \geq 0$, while the definition in [13] required that $E_k(t)$ are fixed and nonnegative through the iterations. Also, instead of the nonsingularity condition on B_k in [13], we require (B_k, C_k) to be regular Q-splittings. The latter condition is peculiar to LCP.

Using a multisplitting $\{(B_k, C_k, E_k(t)); k = 1, \ldots, K\}$ of M, we have the following algorithm.

Algorithm 3.1 (Multisplitting method for LCP(q, M)**)**

Step 1. Choose an arbitrary nonnegative vector $z(0) \in R^n$. Let $t := 0$.

Step 2. For each k, given $z(t) \geq 0$, let $y_k(t)$ be an arbitrary solution of LCP$(q_k(t),B_k)$:

$$\begin{aligned} B_k z + q_k(t) &\geq 0, \\ z &\geq 0, \\ z^T(B_k z + q_k(t)) &= 0, \end{aligned}$$

where $q_k(t) = C_k z(t) + q$.

Step 3. If $y_k(t) = z(t)$ for some k, stop. Otherwise, let

$$z(t+1) = \sum_{k=1}^{K} E_k(t) y_k(t)$$

and $t := t + 1$. Return to Step 2.

Since B_k is a Q-matrix, LCP$(q_k(t),B_k)$ always has a solution for each k. Thus Algorithm 3.1 is well-defined. In particular, if M is positive semidefinite, then $B_k = (M + B_k - C_k)/2$ is positive definite and hence LCP$(q_k(t),B_k)$ must have a unique solution for each k.

The following assumption is introduced to ensure convergence of the algorithm.

Assumption 3.1 *At each iteration* t, $E_k(t), k = 1, \ldots, K$ *satisfy the following conditions:*

(a) $\sum_{k=1}^{K} E_k(t) y_k(t) \geq 0$;

(b) $f(\sum_{k=1}^{K} E_k(t) y_k(t)) \leq \max_{1 \leq k \leq K} f(y_k(t))$.

Various choices of $E_k(t)$ satisfy this assumption. For now, let $E_k(t)$ be expressed as

$$E_k(t) = \alpha_k(t)I, \quad k = 1, \ldots, K,$$

where $\alpha_k(t)$ are real numbers such that

$$\sum_{k=1}^{K} \alpha_k(t) = 1. \tag{3.1}$$

Then, one of the possibilities is to choose at each iteration t an index $k(t) \in \{1, \ldots, K\}$ and let

$$\alpha_k(t) = \begin{cases} 1 & \text{for } k = k(t), \\ 0 & \text{for } k \neq k(t). \end{cases}$$

In this case, we have

$$z(t+1) = y_{k(t)}(t).$$

Namely, at each iteration, the next iterate is determined precisely by one of the splittings (B_k, C_k). As to the rule of choosing $k(t)$, one could choose $k(t)$ randomly at each iteration or in a certain predetermined order such as the cyclic rule. Alternatively one could choose $k(t)$ based on the function values $f(y_k(t)), k = 1, \ldots, K$. For example, we may choose $k(t)$ that minimizes $f(y_k(t))$ over $k = 1, \ldots, K$. More generally, one might choose $\alpha_k(t), k = 1, \ldots, K$ so as to minimize the function $f(\sum_{k=1}^{K} \alpha_k(t) y_k(t))$ subject to the constraints $\sum_{k=1}^{K} \alpha_k(t) = 1$ and $\sum_{k=1}^{K} \alpha_k(t) y_k(t) \geq 0$. The latter strategies require the evaluation of the function f, but the algorithm is expected to converge in fewer iterations compared with the random or cyclic rule.

When M is positive semidefinite, any nonnegative $\alpha_k(t)$ satisfying (3.1) may be used to determine $E_k(t)$. In fact, if M is positive semidefinite, then $f(z)$ is a convex function and hence

$$\begin{aligned} f(\sum_{k=1}^{K} E_k(t) y_k(t)) &= f(\sum_{k=1}^{K} \alpha_k(t) y_k(t)) \\ &\leq \sum_{k=1}^{K} \alpha_k(t) f(y_k(t)) \\ &\leq \max_{1 \leq k \leq K} f(y_k(t)). \end{aligned}$$

Thus Assumption 3.1 (b) is always satisfied. As to the rule of choosing $\alpha_k(t), k = 1, \ldots, K$, one could simply put $\alpha_k(t) = \frac{1}{K}$ for all k.

Throughout this section, we assume that $E_k(t), k = 1, \ldots, K$, satisfy the conditions in Assumption 3.1. Now we present two lemmas that will be useful in proving convergence of Algorithm 3.1. The second lemma is an extension of Theorem 5.3.3 in [3].

Lemma 3.1 *At each iteration t, we obtain*

$$\begin{aligned} f(z(t)) - f(z(t+1)) &\geq f(z(t)) - f(y_{\hat{k}(t)}(t)) && (3.2)\\ &\geq \frac{1}{2}(z(t) - y_{\hat{k}(t)}(t))^T (B_{\hat{k}(t)} - C_{\hat{k}(t)})(z(t) - y_{\hat{k}(t)}(t)) && (3.3)\\ &\geq 0, && (3.4) \end{aligned}$$

where $\hat{k}(t)$ is an index such that $f(y_{\hat{k}(t)}(t)) = \max_{1\leq k\leq K} f(y_k(t))$.

Proof. By Assumption 3.1(b), we have

$$\begin{aligned} f(z(t)) - f(z(t+1)) &= f(z(t)) - f(\sum_{k=1}^{K} E_k(t) y_k(t)) \\ &\geq f(z(t)) - \max_k f(y_k(t)) \\ &= f(z(t)) - f(y_{\hat{k}(t)}(t)). \end{aligned}$$

This shows the inequality (3.2). Moreover, since (B_k, C_k) are regular Q-splittings for all k, it follows from Lemma 2.1 that

$$f(z(t)) - f(y_{\hat{k}(t)}(t)) \geq \frac{1}{2}(z(t) - y_{\hat{k}(t)}(t))^T (B_{\hat{k}(t)} - C_{\hat{k}(t)})(z(t) - y_{\hat{k}(t)}(t)) \geq 0.$$

This establishes the inequalities (3.3) and (3.4). □

Lemma 3.2 *Every accumulation point of the sequence $\{z(t)\}$ generated by Algorithm 3.1 is a solution of* LCP(q, M).

Proof. Let $\tilde{z}$ be an arbitrary accumulation point of $\{z(t)\}$ and $\{z(t_i)\}$ be a subsequence converging to $\tilde{z}$. Since $\{f(z(t_i))\}$ converges to $f(\tilde{z})$ and since $\{f(z(t))\}$ is nonincreasing by Lemma 3.1, the entire sequence $\{f(z(t))\}$ converges. Let $\hat{k}(t)$ be defined as in Lemma 3.1. Then taking a further subsequence if necessary, we may assume that there exists some index $\hat{k}$ such that $\hat{k}(t_i) = \hat{k}$ for all i. Since $(B_{\hat{k}}, C_{\hat{k}})$ is a regular Q-splitting, the sequence $\{z(t_i) - y_{\hat{k}}(t_i)\}$ converges to zero by (3.2) – (3.4), and hence $\{y_{\hat{k}}(t_i)\}$ converges to $\tilde{z}$. Since $y_{\hat{k}}(t_i)$ is a solution of LCP$(C_{\hat{k}} z(t_i) + q, B_{\hat{k}})$, we have

$$\begin{aligned} B_{\hat{k}} y_{\hat{k}}(t_i) + C_{\hat{k}} z(t_i) + q &\geq 0, \\ y_{\hat{k}}(t_i) &\geq 0, \\ (y_{\hat{k}}(t_i))^T (B_{\hat{k}} y_{\hat{k}}(t_i) + C_{\hat{k}} z(t_i) + q) &= 0. \end{aligned}$$

Since $B_{\hat{k}} + C_{\hat{k}} = M$, and since $y_{\hat{k}}(t_i) \to \tilde{z}$ and $z(t_i) \to \tilde{z}$ as $t_i \to \infty$, it follows that $\tilde{z}$ solves LCP(q, M). □

We now state the main convergence theorem for the multisplitting method (Algorithm 3.1), which is a natural extension of Theorem 2.1 for the splitting method (Algorithm 2.1).

Theorem 3.1 *Suppose that*

(a) *$f(z)$ is bounded below on $z \geq 0$;*

(b) *$0 \neq z \geq 0$, $Mz \geq 0$ and $z^T Mz = 0$ imply $q^T z > 0$.*

Then the sequence $\{z(t)\}$ generated by Algorithm 3.1 is bounded and any accumulation point of it solves LCP(q, M).

Proof. The proof is a modification of that of Lemma 5.3.4 in [3]. First, suppose to the contrary that the sequence $\{z(t)\}$ is unbounded. Then there exists a subsequence $\{z(t_i)\}$ such that $\|z(t_i)\| \to \infty$. By assumption (a) and Lemma 3.1, $\{f(z(t))\}$ converges. Let $\hat{k}$ be chosen as in the proof of Lemma 3.2. Since $(B_{\hat{k}}, C_{\hat{k}})$ is a regular Q-splitting, the sequence $\{z(t_i) - y_{\hat{k}}(t_i)\}$ converges to zero by (3.2)–(3.4). This in particular implies that $\|y_{\hat{k}}(t_i)\| \to \infty$ because $\|z(t_i)\| \to \infty$.

Now consider the corresponding normalized sequence $\{y_{\hat{k}}(t_i)/\|y_{\hat{k}}(t_i)\|\}$, which is bounded and hence has an accumulation point $\tilde{z}$ such that $\|\tilde{z}\| = 1$ and $\tilde{z} \geq 0$. Assume without loss of generality that $\{y_{\hat{k}}(t_i)/\|y_{\hat{k}}(t_i)\|\}$ converges to $\tilde{z}$. Then since $y_{\hat{k}}(t_i) - z(t_i) \to 0$ and $y_{\hat{k}}(t_i)/\|y_{\hat{k}}(t_i)\| \to \tilde{z}$, we have $z(t_i)/\|y_{\hat{k}}(t_i)\| \to \tilde{z}$. Moreover, since $y_{\hat{k}}(t_i)$ is a solution of the LCP$(C_{\hat{k}} z(t_i) + q, B_{\hat{k}})$, we have

$$B_{\hat{k}} y_{\hat{k}}(t_i) + C_{\hat{k}} z(t_i) + q \geq 0, \tag{3.5}$$

$$y_{\hat{k}}(t_i) \geq 0, \tag{3.6}$$

$$(y_{\hat{k}}(t_i))^T (B_{\hat{k}} y_{\hat{k}}(t_i) + C_{\hat{k}} z(t_i) + q) = 0. \tag{3.7}$$

Then passing to the limit $t_i \to \infty$, we have

$$M\tilde{z} \geq 0, \quad \tilde{z} \geq 0, \quad \tilde{z}^T M\tilde{z} = 0. \tag{3.8}$$

On the other hand, if assumption (a) is satisfied, M satisfies the inequality $z^T Mz \geq 0$ for any $z \geq 0$, by Proposition 3.7.14 in [3]. Therefore, since $M = B_{\hat{k}} + C_{\hat{k}}$ and $y_{\hat{k}}(t_i) \geq 0$ by (3.6), it follows from (3.7) that

$$(y_{\hat{k}}(t_i))^T M y_{\hat{k}}(t_i) = -(y_{\hat{k}}(t_i))^T (C_{\hat{k}}(z(t_i) - y_{\hat{k}}(t_i)) + q) \geq 0. \tag{3.9}$$

Since $y_{\hat{k}}(t_i) - z(t_i) \to 0$, dividing (3.9) by $\|y_{\hat{k}}(t_i)\|$ and passing to the limit $t_i \to \infty$ yield $q^T \tilde{z} \leq 0$. But, this together with (3.8) contradicts assumption (b), indicating that the sequence $\{z(t)\}$ must be bounded. The rest of the proof is immediate from Lemma 3.2. □

4 An Application

In this section, we apply the multisplitting method to the parallel successive overrelaxation method [11], which is one of the splitting methods for solving the symmetric LCP. Let $P^{(k)} \in R^{n\times n}$ be arbitrary permutation matrices and let

$$\begin{aligned} M^{(k)} &= P^{(k)} M (P^{(k)})^T, \\ q^{(k)} &= P^{(k)} q. \end{aligned}$$

Note that solving LCP(q, M) is equal to solving LCP$(q^{(k)}, M^{(k)})$. Let $\{I_1, \ldots, I_N\}$ be a partition of $\{1, \ldots, n\}$ such that $i < j$ for $i \in I_l, j \in I_{l+1}$ and $l = 1, \ldots, N-1$. Here, we consider a single partitioning of $\{1, \ldots, n\}$. This is for notational simplicity only. We note that a different partition of $\{1, \ldots, n\}$ may be associated with each k. Break $M^{(k)}$ into N blocks of rows, and further break those blocks into N blocks of columns, as follows:

$$M^{(k)} = \begin{bmatrix} M^{(k)}_{I_1} \\ M^{(k)}_{I_2} \\ \vdots \\ M^{(k)}_{I_N} \end{bmatrix} = \begin{bmatrix} M^{(k)}_{I_1 I_1} & M^{(k)}_{I_1 I_2} & \cdots & M^{(k)}_{I_1 I_N} \\ M^{(k)}_{I_2 I_1} & M^{(k)}_{I_2 I_2} & & \\ \vdots & & \ddots & \\ M^{(k)}_{I_N I_1} & & & M^{(k)}_{I_N I_N} \end{bmatrix},$$

where $M^{(k)}_{I_i} \in R^{|I_i|\times n}$ and $M^{(k)}_{I_i I_j} \in R^{|I_i|\times|I_j|}$. We further partition the diagonal blocks $M^{(k)}_{I_i I_i}$ as follows:

$$M^{(k)}_{I_i I_i} = L^{(k)}_{I_i I_i} + D^{(k)}_{I_i I_i} + U^{(k)}_{I_i I_i}, \tag{4.1}$$

where $L^{(k)}_{I_i I_i}$ is the strictly lower triangular part of $M^{(k)}_{I_i I_i}$, $D^{(k)}_{I_i I_i}$ its diagonal part and $U^{(k)}_{I_i I_i}$ its strictly upper triangular part. With the partitioning (4.1), define a block diagonal matrix $H^{(k)}$ by

$$H^{(k)} = \begin{bmatrix} L^{(k)}_{I_1 I_1} & & & \\ & L^{(k)}_{I_2 I_2} & & \\ & & \ddots & \\ & & & L^{(k)}_{I_N I_N} \end{bmatrix} \tag{4.2}$$

and let

$$H_k = (P^{(k)})^T H^{(k)} P^{(k)}.$$

Consider the splitting (B_k, C_k) for each k such that

$$B_k = \omega_k^{-1} D + H_k, \tag{4.3}$$

$$C_k = M - \omega_k^{-1} D - H_k, \tag{4.4}$$

where ω_k is the relaxation parameter and D is the diagonal part of M which is assumed positive. We assume that, for each k, the parameter ω_k satisfies

$$0 < \omega_k < \min_{1\le i\le N} \min_{l\in I_i} 2 \Big/ \left[1 + \sum_{s\in \bar{I}_i} |M^{(k)}_{ls}|/D^{(k)}_{ll}, \right]$$

where $\bar{I}_i = \bigcup_{j\neq i} I_j$. Then it can be seen that for each k the splitting (B_k, C_k) defined by (4.3) and (4.4) is a regular Q-splitting. Under the above setting, Algorithm 3.1 reduces to the following algorithm:

Algorithm 4.1

Step 1. Choose an arbitrary nonnegative vector $z(0) \in R^n$. Let $t := 0$.

Step 2. For each k, given $z^{(k)}(t) = P^{(k)}z(t) \geq 0$, solve LCP$(q^{(k)}_{I_i}(t), B^{(k)}_{I_iI_i})$ for all blocks i, to obtain their solutions $y^{(k)}_{I_i}(t), i = 1, \ldots, N$:

$$\begin{aligned} B^{(k)}_{I_iI_i} z + q^{(k)}_{I_i}(t) &\geq 0, \\ z &\geq 0, \\ z^T (B^{(k)}_{I_iI_i} z + q^{(k)}_{I_i}(t)) &= 0, \end{aligned}$$

where

$$\begin{aligned} B^{(k)}_{I_iI_i} &= \omega_k^{-1} D^{(k)}_{I_iI_i} + L^{(k)}_{I_iI_i}, \\ q^{(k)}_{I_i}(t) &= M^{(k)}_{I_i} z^{(k)}(t) - B^{(k)}_{I_iI_i} z^{(k)}_{I_i}(t) + q^{(k)}_{I_i}, \end{aligned}$$

and let

$$y_k(t) = (P^{(k)})^T \begin{bmatrix} y^{(k)}_{I_1}(t) \\ y^{(k)}_{I_2}(t) \\ \vdots \\ y^{(k)}_{I_N}(t) \end{bmatrix}.$$

Step 3. If $y_k(t) = z(t)$ for some k, stop. Otherwise, let

$$z(t+1) = \sum_{k=1}^{K} E_k(t) y_k(t)$$

and $t := t + 1$. Return to Step 2.

We note that the $K \times N$ subproblems LCP$(q^{(k)}_{I_i}(t), B^{(k)}_{I_iI_i})$, $i = 1, \ldots, N, k = 1, \ldots, K$, may be dealt with independently in Step 2 of the algorithm.

5 Numerical Results

In this section, we present some computational results for Algorithm 4.1. In our test problems, matrix M is determined by $M = AA^T + \gamma D$, where A is a square matrix, D is a diagonal matrix and γ is a positive parameter used to control the condition of matrix M. By suitably adjusting the sparsity of A, sparse matrices M are formed so that their nonzero density is approximately 1%. The nonzero elements of A and the diagonal elements of D are chosen randomly from intervals $[-5, 5]$ and $[1, 10]$, respectively. The elements of q are random integers in the interval $[-100, 100]$. The initial solution $z(0)$ is always set to be 0. The termination criterion for the algorithm is

$$\min_{0 \le k \le K} \|z(t) - y_k(t)\|_\infty < 10^{-5},$$

where $\|\cdot\|_\infty$ denotes the l_∞-norm of a vector.

We only consider the case where the number of splittings K is 2. The 'weight' matrices $E_k(t)$ are chosen to be

$$E_1(t) = (1 - \alpha)I \quad \text{and} \quad E_2(t) = \alpha I,$$

where α is a parameter which is not necessarily nonnegative. We examine how the choice of parameter α affects the performance of the algorithm. In this experiment, we fix the number of blocks N as 2, and relaxation parameter ω_k as 1.0 for each k. We generate five test problems for each pair (n,γ) with $n = 5000$ and 7000 and $\gamma = 1$ and 5. Those problems are solved with several values of parameter α such as -0.5, 0.0, 0.5, 1.0 and 1.5. When $\alpha = 0.0$ or $\alpha = 1.0$, the multisplitting method can be regarded as the corresponding splitting method with $k = 1$ or $k = 2$, respectively. When $\alpha = -0.5$ or $\alpha = 1.5$, Assumption 3.1 (a) may not be satisfied. But it has often been observed that the algorithm still converges to a solution of the problem for such choice of α. The results are summarized in Tables 1-4. It is observed that the choice of parameter α affects the performance of the algorithm. If we choose a suitable α, the algorithm may converge in fewer iterations compared with the corresponding splitting method, though it is not easy to know the optimal choice of parameter α.

6 Conclusion

In this paper, we have extended the idea of multisplitting for a system of linear equations to the symmetric LCP and established a convergence theorem for the multisplitting method for LCP under some appropriate assumptions. From a practical viewpoint, however, the computational results reported in the previous section suggest that it is not necessarily easy to obtain satisfactory

Table 1: Number of iterations ($n = 5000, \gamma = 1, \omega_k = 1.0$)

Problem	parameter α				
	-0.5	0.0	0.5	1.0	1.5
#1	195	88	122	103	div
#2	194	61	69	72	139
#3	289	78	70	78	div
#4	div	75	65	73	div
#5	210	62	58	72	283

Table 2: Number of iterations ($n = 5000, \gamma = 5, \omega_k = 1.0$)

Problem	parameter α				
	-0.5	0.0	0.5	1.0	1.5
#1	54	48	55	49	66
#2	50	36	37	43	48
#3	68	52	46	48	111
#4	100	42	37	41	74
#5	46	36	36	39	71

Table 3: Number of iterations ($n = 7000, \gamma = 1, \omega_k = 1.0$)

Problem	parameter α				
	-0.5	0.0	0.5	1.0	1.5
#1	214	58	59	62	div
#2	262	67	62	69	div
#3	599	71	70	74	345
#4	div	85	76	93	199
#5	149	72	61	62	189

Table 4: Number of iterations ($n = 7000, \gamma = 5, \omega_k = 1.0$)

Problem	parameter α				
	-0.5	0.0	0.5	1.0	1.5
#1	50	36	36	33	50
#2	54	38	34	48	100
#3	57	37	38	44	86
#4	65	49	46	50	66
#5	48	44	39	36	46

results by the naive use of multisplitting, which parallels the observation made in a recent paper by Frommer and Mayer [7] for the multisplitting method for linear equations.

References

[1] R. W. Cottle, G. H. Golub and R. S. Sacher, On the solution of large, structured linear complementarity problems: The block partitioned case, *Applied Mathematics and Optimization* **4** (1978) 347-363.

[2] R. W. Cottle and J. S. Pang, On the convergence of a block successive overrelaxation method for a class of linear complementarity problems, *Mathematical Programming Study* **17** (1980) 126-138.

[3] R. W. Cottle, J. S. Pang and R. E. Stone, *The Linear Complementarity Problem* (Academic Press, San Diego, 1992).

[4] C. W. Cryer, The solution of a quadratic programming problem using systematic overrelaxation, *SIAM Journal on Control* **9** (1971) 385-392.

[5] L. Elsner, Comparisons of weak regular splittings and multisplitting methods, *Numerische Mathematik* **56** (1989) 283-289.

[6] W. A. Frommer and K. G. Mayer, Convergence of relaxed parallel multisplitting methods, *Linear Algebra and its Applications* **119** (1989) 141-152.

[7] W. A. Frommer and K. G. Mayer, On the theory and practice of multisplitting methods in parallel computation, *Computing* **49** (1992) 63-74.

[8] A. N. Iusem, On the convergence of iterative methods for symmetric linear complementarity problems, *Mathematical Programming* **59** (1993) 33-48.

[9] Z. Luo and P. Tseng, On the convergence of a matrix splitting algorithm for the symmetric monotone linear complementarity problem, *SIAM J. Control and Optimization* **29** (1991) 1037-1060.

[10] O. L. Mangasarian, Solution of symmetric linear complementarity problems by iterative methods, *Journal of Optimization Theory and Applications* **22** (1977) 465-485.

[11] O. L. Mangasarian and R. De Leone, Parallel successive overrelaxation methods for symmetric linear complementarity problems and linear programs, *Journal of Optimization Theory and Applications* **54** (1987) 437-446.

[12] M. Neumann and R. J. Plemmonos, Convergence of parallel multisplitting iterative methods for M-matrices, *Linear Algebra and Applications* **88-89** (1987) 559-573.

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

[14] S. Papatheodorou and Y. G. Saridakis, Parallel algorithms and architectures for multisplitting iterative methods, *Parallel Computing* **12** (1989) 171-182.

[15] J. S. Pang, Necessary and sufficient conditions for the convergence of iterative methods for the linear complementarity problem, *Journal of Optimization Theory and Applications* **42** (1984) 1-17.

[16] D. Wang, On the convergence of the parallel multisplitting AOR algorithm, *Linear Algebra and Applications* **154-156** (1991) 473-486.

[17] R. E. White, Multisplitting with different weighting schemes, *SIAM Journal on Matrix Analysis and Applications* **10** (1989) 481-493.

[18] R. E. White, Multisplitting of a symmetric positive definite matrix, *SIAM Journal on Matrix Analysis and Applications* **11** (1990) 69-84.