A two-step matrix splitting method for the mixed linear complementarity problem*

Cui-Xia Li[†], Shi-Liang Wu[‡]

[†]*School of Mathematics, Yunnan Normal University, Kunming, Yunnan, 650500, P.R. China*

[†,‡]*Yunnan Key Laboratory of Modern Analytical Mathematics and Applications, Yunnan Normal University, Kunming, 650500, P.R. China*

Abstract

In this paper, on the base of the methodology of the new modulus-based matrix splitting method in [Optim. Lett., (2022) 16:1427-1443], we establish a two-step matrix splitting (TMS) method for solving the mixed linear complementarity problem (MLCP). Three sufficient conditions to ensure the convergence of the proposed method are presented. Numerical examples are provided to illustrate the feasibility and efficiency of the proposed method.

Keywords: Mixed linear complementarity problem; new modulus-based matrix splitting method; sufficient condition; convergence

AMS classification: 65F10, 90C33

1 Introduction

The mixed linear complementarity problem is to find two vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ such that

$$\begin{cases} Ax + By + a = 0, \\ w = b + Cx + Dy \geq 0, y \geq 0, \\ w^T y = 0, \end{cases} \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$, $D \in \mathbb{R}^{m\times m}$, $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$, see [1, 2]. We denote the problem (1.1) by the MLCP(A, B, C, D, a, b). The MLCP(A, B, C, D, a, b) is

*This research was supported by National Natural Science Foundation of China (No.11961082).

†Corresponding author: lixiatkynu@126.com

‡slwuynnu@126.com

from several aspects, such as the variational inequalities, complementarity problems and a variety of mathematical programs, which is commonly considered as an important tool for the stability aspects of nonlinear complementarity problem and the Karush-Kuhn-Tucker (KKT) system of a variational problem. In addition, in a way, the MLCP(A, B, C, D, a, b) is also viewed as a generalization form of the linear complementarity problem (LCP), the vertical linear complementarity problem (VLCP), the horizontal linear complementarity problem (HLCP), respectively. For example, the MLCP(A, B, C, D, a, b) reduces to the LCP$(D - CA^{-1}B, b - cA^{-1}a)$ if A is nonsingular in (1.1), see [4, 5].

In [1,2], the authors mainly focused on the theory research of the MLCP(A, B, C, D, a, b), such as the existence of the solution, the perturbation bound of the solution, the degree theory, and so on. As is known, besides that, another important aspect is to develop some efficient numerical methods for solving the MLCP(A, B, C, D, a, b). To know our knowledge, using the iteration method for solving the MLCP(A, B, C, D, a, b) has not been discussed. In order to fill in this study gap, in this paper, inspired by the new modulus-based matrix splitting method for the LCP in [3], we establish a two-step matrix splitting (TMS) method for solving the MLCP(A, B, C, D, a, b).

Of course, as mentioned above, the MLCP(A, B, C, D, a, b) with A being nonsingular in (1.1) can reduce to the corresponding LCP. In such case, there exist many numerical methods for its LCP form, such as interior point method [7,8], projected splitting method [9–11], matrix multisplitting iteration method [6], modulus-based matrix splitting (MMS) method [12] and new modulus-based matrix splitting (NMMS) method [3]. Due to the convenience and performance of the MMS and NMMS methods, they are two power tools to gain the numerical solution of the LCP. Not only that, there exist their other forms, see [13–18] (to name a few) for more details. In addition, the MMS and NMMS methods have been used to address other complementarity problems, such as the implicit complementarity problem [19], the quasi-complementarity problem [20] and the horizontal linear complementarity problem [26]. Recent some related researches, one see [27–29].

It should be noted that if we directly use the above MMS and NMMS methods for its LCP form to obtain the numerical solution of the MLCP(A, B, C, D, a, b), then we have to use other technique for handling with the inverse of matrix A. As is known, this case should be avoided as much as possible because the computational expense for computing the inverse of matrix A is commonly large. What is worse, when we face to matrix A that is singular or is close to singular, the above MMS and NMMS methods may be failure. To avoid this disadvantage, with the help of the methodology of the NMMS method in [3], we establish a numerical method, i.e., a two-step matrix splitting (TMS) method, for solving the MLCP(A, B, C, D, a, b). To ensure the convergence of the proposed method, three coarse sufficient conditions are presented. In addition, to illustrate the feasibility and efficiency of the proposed method, some numerical examples are provided as well.

The layout of this paper unfolds below. In Section 2, a two-step matrix splitting (TMS) method for solving the MLCP(A, B, C, D, a, b) is proposed on the base of the NMMS method. In Section 3, three coarse convergence conditions of the proposed method are given. Numerical experiments are reported to verify the efficiency of the proposed method in Section 4. Finally, in Section 5, we give some conclusions to end this paper.

2 A two-step matrix splitting method

In this section, we will establish a two-step matrix splitting (TMS) method for solving the MLCP (A, B, C, D, a, b). To this end, Lemma 2.1 is required.

Lemma 2.1 [3] *Let $a, b \in \mathbb{R}$. Then $a \geq 0, b \geq 0, ab = 0$ if and only if $a + b = |a - b|$. This result carries immediately over to vectors in $\mathbb{R}^n$.*

Exploiting Lemma 2.1, we can forthrightly gain the following system with absolute value equation, see Theorem 2.1, whose proof is firsthand and omitted.

Theorem 2.1 *The MLCP(A, B, C, D, a, b) is equivalent to find two vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ such that*

$$\begin{cases} Cx + (\Omega + D)y = |(D - \Omega)y + Cx + b| - b, \\ Ax + By + a = 0, \end{cases} \tag{2.1}$$

where Ω is a positive diagonal matrix.

Naturally, in the light of two functions in Theorem 2.1, it is easy to establish a two-step matrix splitting (TMS) method for solving the MLCP(A, B, C, D, a, b), specifically see Method 2.1.

Method 2.1 Let $D = F_1 - G_1$ be a splitting of the matrix D, $A = F_2 - G_2$ be a splitting of the matrix A with F_2 being nonsingular, and let Ω be a positive diagonal matrix such that matrix $\Omega + F_1$ is nonsingular. Assume that $(x^{(0)}, y^{(0)}) \in \mathbb{R}^{n+m}$ is an arbitrary initial vector with $x^{(0)} \in \mathbb{R}^n, y^{(0)} \in \mathbb{R}^m$. For $k = 0, 1, 2, \ldots$, until the sequence of iterates $\{(x^{(k)}, y^{(k)})\}_{k=0}^{+\infty} \subset \mathbb{R}^{n+m}$ is convergent, calculate $(x^{(k+1)}, y^{(k+1)})$ by solving the following two systems

$$\begin{cases} (\Omega + F_1)y^{(k+1)} = G_1 y^{(k)} - Cx^{(k)} + |(D - \Omega)y^{(k)} + Cx^{(k)} + b| - b, \\ F_2 x^{(k+1)} = G_2 x^{(k)} - By^{(k+1)} - a. \end{cases} \tag{2.2}$$

Obviously, per iteration in Method 2.1, it is necessary to solve two linear subsystems with matrices $\Omega + F_1$ and F_2, which in practical computations compels us to quickly obtain the inverse of the matrices $\Omega + F_1$ and F_2. Fortunately, we can easily do it by the matrix splitting technique. Specially, for Method 2.1, if $U_D(U_A)$ is the minus strictly upper part of $D(A)$,

$$F_1 = \frac{1}{\alpha}(\wedge_D - \beta L_D) \text{ and } F_2 = \frac{1}{\alpha}(\wedge_A - \beta L_A),$$

where $\wedge_D(\wedge_A)$ and $L_D(L_A)$ denote the diagonal and minus strictly lower part of $D(A)$, respectively, then the two-step Jacobi (TJ), Gauss-Seidel (TGS) and successive overrelaxation (TSOR) iteration method with (α, β) being equal to $(1,0)$, $(1,1)$ and (α, α) can be obtained, respectively. Clearly, these relaxation versions of Method 2.1 easily gain the inverse of the matrices $\Omega + F_1$ and F_2. Not only that, they are quite practical and efficient for solving the large sparse MLCP(A, B, C, D, a, b) on the high-peed processor systems.

3 Convergence analysis

In this section, we will discuss the convergence property of Method 2.1. To this end, we require some necessary definitions, notations and lemmas.

Let $A = (a_{ij})$, $B = (b_{ij}) \in \mathbb{R}^{n\times n}$. If $a_{ij} \geq b_{ij}$ $(a_{ij} > b_{ij})$ for $i, j = 1, 2, \ldots, n$, we denote $A \geq B$ $(A > B)$, in particular, we call A a nonnegative (positive) matrix and denote $A \geq 0$ $(A > 0)$ if $B = 0$. Matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ is called a Z-matrix if $a_{ij} \leq 0$ $(i \neq j)$; an M-matrix if $A^{-1} \geq 0$ and A is a nonsingular Z-matrix. In addition, $|A| = (|a_{ij}|)$, $\det(A)$, $\|A\|$ and $\rho(A)$, respectively, denotes the determinant, the 2-norm and the spectral radius of the matrix A, see [21, 22].

Lemma 3.1 [23] *Let λ be any root of the quadratic equation $x^2+bx+d = 0$ with $b, d \in \mathbb{R}$. Then $|\lambda| < 1$ if and only if $|d| < 1$ and $|b| < 1 + d$.*

Lemma 3.2 [21] *Let $A \geq 0$ be an irreducible matrix. Then*

(i) A has a positive eigenvalue equal to its spectral radius;

(ii) A has an eigenvector $x > 0$ corresponding to $\rho(A)$;

(iii) $\rho(A)$ is a simple eigenvalue of A.

Lemma 3.3 [21] *Let $A \geq 0$. Then*

$$\alpha x \leq Ax, \ x \geq 0, \textit{implies } \alpha \leq \rho(A),$$

and

$$Ax \leq \beta x, \ x > 0, \textit{implies } \rho(A) \leq \beta.$$

Let

$$s_1 = \|(\Omega + F_1)^{-1}G_1\| + \|(\Omega + F_1)^{-1}\|\|D - \Omega\|, s_2 = \|(\Omega + F_1)^{-1}C\| + \|(\Omega + F_1)^{-1}\|\|C\|$$

and

$$t_1 = \|F_2^{-1}G_2\|, t_2 = \|F_2^{-1}B\|.$$

Then the following main convergence theorem for Method 2.1 is presented.

Theorem 3.1 *Let $D = F_1 - G_1$, $A = F_2 - G_2$ with $\det(F_2) \neq 0$ be a matrix splitting of the matrices D and A, respectively, and let Ω be a positive diagonal matrix such that matrix $\Omega + F_1$ is nonsingular. If*

$$t_1 s_1 < 1 \text{ and } t_2 s_2 < (1 - t_1)(1 - s_1), \tag{3.1}$$

then Method 2.1 is convergent.

Proof. Assume that (x_*, y_*) is a solution of the MLCP(A, B, C, D, a, b), then it satisfies

$$\begin{cases} (\Omega + F_1)y_* = G_1 y_* - Cx_* + |(D - \Omega)y_* + Cx_* + b| - b, \\ F_2 x_* = G_2 x_* - By_* - a. \end{cases} \tag{3.2}$$

Combining (2.2) with (3.2), we obtain

$$\begin{aligned} y^{(k+1)} - y_* =& (\Omega + F_1)^{-1}(G_1(y^{(k)} - y_*) - C(x^{(k)} - x_*) \\ & + |(D - \Omega)y^{(k)} + Cx^{(k)} + b| - |(D - \Omega)y_* + Cx_* + b|), \end{aligned}$$

and

$$x^{(k+1)} - x_* = F_2^{-1}(G_2(x^{(k)} - x_*) - B(y^{(k+1)} - y_*)).$$

Additionally,

$$\begin{aligned} \|y^{(k+1)} - y_*\| =& \|(\Omega + F_1)^{-1}(G_1(y^{(k)} - y_*) - C(x^{(k)} - x_*) \\ & + |(D - \Omega)y^{(k)} + Cx^{(k)} + b| - |(D - \Omega)y_* + Cx_* + b|)\| \\ \leq & \|(\Omega + F_1)^{-1}G_1(y^{(k)} - y_*)\| + \|(\Omega + F_1)^{-1}C(x^{(k)} - x_*)\| \\ & + \|(\Omega + F_1)^{-1}\| \| |(D - \Omega)y^{(k)} + Cx^{(k)} + b| - |(D - \Omega)y_* + Cx_* + b| \| \\ \leq & \|(\Omega + F_1)^{-1}G_1\| \|y^{(k)} - y_*\| + \|(\Omega + F_1)^{-1}C\| \|x^{(k)} - x_*\| \\ & + \|(\Omega + F_1)^{-1}\| \|(D - \Omega)(y^{(k)} - y_*) + C(x^{(k)} - x_*)\| \\ \leq & |(\Omega + F_1)^{-1}G_1||y^{(k)} - y_*| + |(\Omega + F_1)^{-1}C||x^{(k)} - x_*| \\ & + \|(\Omega + F_1)^{-1}\| \|D - \Omega\| \|y^{(k)} - y_*\| + \|(\Omega + F_1)^{-1}\| \|C\| \|x^{(k)} - x_*\| \\ \leq & (\|(\Omega + F_1)^{-1}G_1\| + \|(\Omega + F_1)^{-1}\| \|D - \Omega\|) \|y^{(k)} - y_*\| \\ & + (\|(\Omega + F_1)^{-1}C\| + \|(\Omega + F_1)^{-1}\| \|C\|) \|x^{(k)} - x_*\| \\ =& s_1 \|y^{(k)} - y_*\| + s_2 \|x^{(k)} - x_*\| \end{aligned}$$

and

$$\begin{aligned} |x^{(k+1)} - x_*| =& |F_2^{-1}(G_2(x^{(k)} - x_*) - B(y^{(k+1)} - y_*))| \\ \leq & |F_2^{-1}G_2||x^{(k)} - x_*| + |F_2^{-1}B||y^{(k+1)} - y_*| \\ =& t_1 |x^{(k)} - x_*| + t_2 |y^{(k+1)} - y_*|. \end{aligned}$$

Denote

$$e_x^{(k)} = x^{(k)} - x_*, e_y^{(k)} = y^{(k)} - y_*,$$

then

$$\|e_y^{(k+1)}\| \leq s_1\|e_y^{(k)}\| + s_2\|e_x^{(k)}\|. \tag{3.3}$$

Similarly,

$$\|e_x^{(k+1)}\| \leq t_1\|e_x^{(k)}\| + t_2\|e_y^{(k+1)}\|. \tag{3.4}$$

Thus, from (3.3) and (3.4), we have

$$\begin{pmatrix} 1 & 0 \\ -t_2 & 1 \end{pmatrix} \begin{pmatrix} \|e_y^{(k+1)}\| \\ \|e_x^{(k+1)}\| \end{pmatrix} \leq \begin{pmatrix} s_1 & s_2 \\ 0 & t_1 \end{pmatrix} \begin{pmatrix} \|e_y^{(k)}\| \\ \|e_x^{(k)}\| \end{pmatrix}. \tag{3.5}$$

Let

$$P = \begin{pmatrix} 1 & 0 \\ -t_2 & 1 \end{pmatrix}.$$

Then it is easy to find that matrix P is an M-matrix. It follows that matrix P^{-1} is a nonsingular nonnegative matrix. Left-multiplying (3.5) by P^{-1} gives

$$\begin{pmatrix} \|e_y^{(k+1)}\| \\ \|e_x^{(k+1)}\| \end{pmatrix} \leq \begin{pmatrix} 1 & 0 \\ t_2 & 1 \end{pmatrix} \begin{pmatrix} s_1 & s_2 \\ 0 & t_1 \end{pmatrix} \begin{pmatrix} \|e_x^{(k)}\| \\ \|e_y^{(k)}\| \end{pmatrix} = \begin{pmatrix} s_1 & s_2 \\ t_2 s_1 & t_2 s_2 + t_1 \end{pmatrix} \begin{pmatrix} |e_y^{(k)}| \\ |e_x^{(k)}| \end{pmatrix}.$$

Let

$$W = \begin{pmatrix} s_1 & s_2 \\ t_2 s_1 & t_2 s_2 + t_1 \end{pmatrix}.$$

Clearly, when $\rho(W) < 1$, Method 2.1 is convergent. Let λ be an eigenvalue of W. Then

$$\begin{vmatrix} \lambda - s_1 & -s_2 \\ -t_2 s_1 & \lambda - (t_2 s_2 + t_1) \end{vmatrix} = 0. \tag{3.6}$$

Further, from (3.6) we have

$$\lambda^2 - (s_1 + t_2 s_2 + t_1)\lambda + t_1 s_1 = 0. \tag{3.7}$$

Applying Lemma 3.1 to Eq. (3.7), $|\lambda| < 1$ if and only if

$$t_1 s_1 < 1$$

and

$$s_1 + t_2 s_2 + t_1 < 1 + t_1 s_1.$$

Therefore, if the condition (3.1) holds, then $\rho(W) < 1$. This completes the proof. $\square$

In addition, if we take

$$S_1 = |(\Omega + F_1)^{-1}G_1| + |(\Omega + F_1)^{-1}||D - \Omega|, S_2 = |(\Omega + F_1)^{-1}C| + |(\Omega + F_1)^{-1}||C|$$

and

$$T_1 = |F_2^{-1}G_2|, T_2 = |F_2^{-1}B|,$$

then the following convergence theorem for Method 2.1 is also presented by the proof of Theorem 3.1.

Theorem 3.2 *Let $D = F_1 - G_1$, $A = F_2 - G_2$ with $\det(F_2) \neq 0$ be a matrix splitting of the matrices D and A, respectively, and let Ω be a positive diagonal matrix such that matrix $\Omega + F_1$ is nonsingular. If*

$$\rho(W) < 1,$$

where

$$W = \begin{bmatrix} S_1 & S_2 \\ T_2 & T_1 \end{bmatrix},$$

then Method 2.1 is convergent.

Proof. Based on the proof of Theorem 3.1, we set

$$\bar{W} = \begin{bmatrix} S_1 & S_2 \\ T_2 S_1 & T_2 S_2 + T_1 \end{bmatrix}.$$

It is not difficult to find that under $\rho(\bar{W}) < 1$, Method 2.1 is convergent. According to Lemma 3.2, there is a vector

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \geq 0, \xi \neq 0,$$

such that $\bar{W}\xi = \rho(\bar{W})\xi$, i.e.,

$$\begin{bmatrix} S_1 & S_2 \\ T_2 S_1 & T_2 S_2 + T_1 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \rho(\bar{W}) \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix},$$

or

$$\begin{bmatrix} S_1 & S_2 \\ 0 & T_1 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \rho(\bar{W}) \begin{bmatrix} I & 0 \\ -T_2 & I \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}. \tag{3.8}$$

By simple passages, from (3.8), we have

$$\begin{aligned} &S_1\xi_1 + S_2\xi_2 = \rho(\bar{W})\xi_1, \\ &T_1\xi_2 = -\rho(\bar{W})T_2\xi_1 + \rho(\bar{W})\xi_2, \end{aligned}$$

which is equal to

$$\begin{bmatrix} S_1 & S_2 \\ \rho(\bar{W})T_2 & T_1 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \rho(\bar{W}) \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}. \tag{3.9}$$

If $\rho(\bar{W}) \geq 1$, then from (3.9) we have

$$\begin{aligned} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} &= \begin{bmatrix} \frac{1}{\rho(\bar{W})}S_1 & \frac{1}{\rho(\bar{W})}S_2 \\ T_2 & \frac{1}{\rho(\bar{W})}T_1 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \\ &\leq \begin{bmatrix} S_1 & S_2 \\ T_2 & T_1 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \\ &= W \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}, \end{aligned}$$

along with Lemma 3.2, we that $\rho(W) \geq 1$, which is not in line with $\rho(W) < 1$. This shows that $\rho(\bar{W}) < 1$, further, Method 2.1 is convergent. □

Based on Theorem 2.2, it is not difficult to find that we have the following result, see Theorem 2.3.

Theorem 3.3 *Let $D = F_1 - G_1$, $A = F_2 - G_2$ with $\det(F_2) \neq 0$ be a matrix splitting of the matrices D and A, respectively, and let Ω be a positive diagonal matrix such that matrix $\Omega + F_1$ is nonsingular. If*

$$\|S\|_\infty < 1, \|T\|_\infty < 1,$$

where

$$S = [S_1, S_2], T = [T_2, T_1],$$

then Method 2.1 is convergent.

4 Numerical experiments

In this section, some examples are provided to investigate the performance of Method 2.1 for solving the MLCP(A, B, C, D, a, b), which are from the KKT conditions of a quadratic program with general equality and inequality constraints, see [4] for more details.

For the sake of simplicity, we consider the following two examples.

Example 4.1 Consider the MLCP(A, B, C, D, a, b), for which A, B, C, D are given below:

$$A = \bar{A} + \mu I, \bar{A} = \begin{pmatrix} I \otimes T + T \otimes I & 0 \\ 0 & I \otimes T + T \otimes I \end{pmatrix} \in \mathbb{R}^{2m^2 \times 2m^2}$$

$$B = \begin{pmatrix} I \otimes F \\ F \otimes I \end{pmatrix} \in \mathbb{R}^{2m^2 \times m^2}, C = -B^T, D = I \otimes T + T \otimes I + \mu I \in \mathbb{R}^{m^2 \times m^2},$$

with

$$T = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m\times m} \text{ and } F = \text{tridiag}(-1, 1, 0) \in \mathbb{R}^{m\times m},$$

where $\otimes$ denotes the Kronecker product symbol, which is inspired by in [12, 24].

Example 4.2 Consider the MLCP(A, B, C, D, a, b), for which $A = (a_{kj}) \in \mathbb{R}^{q\times q}$, $D = (d_{kj}) \in \mathbb{R}^{(n-q)\times(n-q)}$ with $2q > n$ are given below:

$$a_{kj} = \begin{cases} k+1 & \text{for } j = k, \\ 1 & \text{for } |j-k| = 1,\ k, j = 1, 2, \ldots, q, \\ 0 & \text{otherwise}, \end{cases}$$

$$d_{kj} = \begin{cases} k+1 & \text{for } j = k, \\ 1 & \text{for } |j-k| = 1\ k, j = 1, 2, \ldots, n-q, \\ 0 & \text{otherwise}, \end{cases}$$

and $B = (b_{kj})$ with

$$b_{kj} = \begin{cases} j & \text{for } k = j + 2q - n, \\ 0 & \text{otherwise}, \end{cases} \quad k, j = 1, 2, \ldots, n-q,$$

and $C = -B^T$, see [25].

Numerical results consist of three aspects: the number of iteration steps (denoted by 'IT'), elapsed CPU time in seconds (denoted by 'CPU') and the relative residual error (denoted by 'RES'), which is defined by

$$\text{RES}(x^{(k)}) = \|\min(Ax^{(k)} + By^{(k)} + a, b + Cx^{(k)} + Dy^{(k)})\|_2,$$

where $x^{(k)}$ and $y^{(k)}$ is the kth approximate solution of the MLCP(A, B, C, D, a, b). In addition, we list the spectral radii $\rho(W)$ of the corresponding matrix.

In our numerical computations, the starting vectors $x^{(0)}$ and $y^{(0)}$ for Method 2.1 are set to be zero vector. Method 2.1 is stopped once $\text{RES}(z^{(k)}) \le 10^{-5}$ or the number of iterations surpass 500. The vectors a and b are to be adjusted such that $a = -Ax^* - By^*$ and $b = w^* - Cx^* - Dy^*$, where $x^* = e$ with $e = (1, 1, \ldots, 1)^T$, w^* and y^* are defined as

$$w^* = (1, 0, 1, 0 \ldots, 1, 0, \ldots)^T,\ y^* = (0, 1, 0, 1 \ldots, 0, 1, \ldots)^T.$$

With respect to the choice of Ω, we use the strategy described in [26] for Method 2.1, i.e., we take $\Omega = \wedge_D$, where $\wedge_D$ denotes the diagonal part of D. In the implementations, we consider two relaxation versions of Method 2.1, i.e., TJ and TGS are adopted to solve the MLCP(A, B, C, D, a, b). All runs are executed in R2016B.

Tables 1-3 list the numerical results (including $\rho(W)$, IT, CPU and RES) of two testing methods for Examples 4.1 and 4.2. The numerical results in Tables 1-3 verify that these two testing methods can rapidly calculate a satisfactory approximation to the solution of the MLCP(A, B, C, D, a, b) under certain conditions. That is to say, in a way,

	m	30	60	90	120
TJ	$\rho(W)$	0.8505	0.8528	0.8532	0.8534
	IT	20	19	19	19
	CPU	0.0147	0.0441	0.0865	0.1360
	RES	5.17e-6	9.30e-6	8.84e-6	8.59e-6
TGS	$\rho(W)$	0.7732	0.7758	0.7762	0.7764
	IT	13	12	12	12
	CPU	0.0113	0.0261	0.0549	0.0931
	RES	3.42e-6	9.20e-6	8.73e-6	8.47e-6

Table 1: Numerical comparison of TJ and TGS for Example 4.1 with $\mu = 4$.

	m	30	60	90	120
TJ	$\rho(W)$	0.6804	0.6822	0.6826	0.6827
	IT	15	15	15	15
	CPU	0.0116	0.0306	0.0627	0.1085
	RES	8.79e-6	7.86e-6	7.47e-6	7.25e-6
TGS	$\rho(W)$	0.5947	0.5966	0.5969	0.5970
	IT	10	10	10	10
	CPU	0.0090	0.0214	0.0456	0.0780
	RES	7.88e-6	7.00e-6	6.64e-6	6.44e-6

Table 2: Numerical comparison of TJ and TGS for Example 4.1 with $\mu = 6$.

	q	6000	7000	8000	9000
TJ	$\rho(W)$	0.8166	0.6548	0.5224	0.5223
	IT	33	19	19	19
	CPU	12.32	6.26	6.02	5.12
	RES	7.86e-6	9.99e-6	9.99e-6	9.99e-6
TGS	$\rho(W)$	0.8166	0.6548	0.5002	0.4439
	IT	31	16	11	11
	CPU	10.91	4.94	3.03	2.57
	RES	8.89e-6	4.47e-6	8.91e-6	8.91e-6

Table 3: Numerical comparison of TJ and TGS for Example 4.2 with $n = 10000$.

these numerical results in Tables 1-3 implies that Method 2.1 can be adopted to solve the MLCP(A, B, C, D, a, b).

Comparing the TJ method and the TGS method, the latter requires least iteration steps and CPU times than the former. Therefore, from the perspective of the computing efficiency, when used as a solver, the latter may be top-priority under certain conditions, compared with the former.

5 Conclusion

In this paper, we focus on the numerical solution of the mixed linear complementarity problem (MLCP) by the iteration method. Our motivation is from that at present there no exists the iteration method for solving the MLCP. In such case, to fill in the study gap, we make use of the methodology of the NMMS method in [3] to design a two-step matrix splitting (TMS) method. Under certain assumptions, three convergence conditions are obtained to ensure the convergence of the TMS method. Finally, some examples (although they are synthetic) in a way can show the efficiency of the proposed method.

Here, we notice that the convergence conditions of Theorems 3.1, 3.2 and 3.3 may be difficult to check in the implementations. How to simply the convergence conditions of Theorems 3.1, 3.2 and 3.3 is considered in the future. On the other hand, how to improve the convergence speed of Method 2.1 is further considered in the future.

References

[1] M. Seetharama Gowda, J.-S. Pang, Stability analysis of variational inequalities and nonlinear complementarity problems, via the mixed linear complementarity problem and degree Theory, Math. Oper. Res., 19 (1994) 831-879.

[2] M. Seetharama Gowda, On the extended linear complementarity problem, Math. Program., 72 (1996) 33-50.

[3] S.-L. Wu, C.-X. Li, A class of new modulus-based matrix splitting methods for linear complementarity problem, Optim. Lett., 16 (2022) 1427-1443.

[4] R.W. Cottle, J.-S. Pang, R.E. Stone, The Linear Complementarity Problem, Academic, San Diego, 1992.

[5] K.G. Murty, Linear Complementarity, Linear and Nonlinear Programming, Heldermann, Berlin, 1988.

[6] Z.-Z. Bai, D.J. Evans, Matrix multisplitting methods with applications to linear complementarity problems: Parallel asynchronous methods, Inter. J. Comput. Math., 79(2) (2002) 205-232.

[7] M. Kojima, N. Megiddo, S. Mizuno, A primal-dual infeasible-interior-point algorithm for linear programming, Math Program., 61 (1993) 263-280.

[8] E.M. Simantiraki, D.F. Shanno, An infeasible-interior-point method for linear complementarity problems, SIAM J. Optim., 7(3) (1997) 620-640.

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

[10] O.L. Mangasarian, Solution of symmetric linear complementarity problems by iterative methods, J. Optim. Theory Appl., 22 (1977) 465-485.

[11] B.H. Ahn, Solution of nonsymmetric linear complementarity problems by iterative methods, J. Optim. Theory Appl., 33(2) (1981) 175-185.

[12] Z.-Z. Bai, Modulus-based matrix splitting iteration methods for linear complementarity problems, Numer. Linear Algebra Appl., 17 (2010) 917-933.

[13] Z.-Z. Bai, L.-L. Zhang, Modulus-based synchronous multisplitting iteration methods for linear complementarity problems, Numer. Linear Algebra Appl., 20 (2013) 425-439.

[14] N. Zheng, J.-F. Yin, Accelerated modulus-based matrix splitting iteration methods for linear complementarity problems, Numer. Algor., 64 (2013) 245-262.

[15] L.-L. Zhang, Two-step modulus-based matrix splitting iteration method for linear complementarity problems, Numer. Algor., 57 (2011) 83-99.

[16] S.-L. Wu, C.-X. Li, Two-sweep modulus-based matrix splitting iteration methods for linear complementarity problems, J. Comput. Math., 302 (2016) 327-339.

[17] S.-L. Wu, L. Li, New modulus-based matrix splitting methods for implicit complementarity problem, Numer. Algor., 90 (2022) 1735-1754.

[18] C.-X. Li, S.-L. Wu, A class of modulus-based matrix splitting methods for vertical linear complementarity problem, Optim., 72 (2023) 2499-2516.

[19] J.-T. Hong, C.-L. Li, Modulus-based matrix splitting iteration methods for a class of implicit complementarity problems, Numer. Linear Algebra Appl., 23 (2016) 629-641.

[20] S.-L. Wu, P. Guo, Modulus-based matrix splitting algorithms for the quasi-complementarity problems, Appl. Numer. Math., 132 (2018) 127-137.

[21] A. Berman, R.J. Plemmons, Nonnegative Matrices in the Mathematical Sciences, Academic, New York, 1979.

[22] R.S. Varga, Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, 1962.

[23] D.M. Young, Iteratin Solution for Large Systems, Academic Press, New York, 1971.

[24] Z.-Z. Bai, G.H. Golub, J.-Y. Pan, Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems, Numer. Math., 98 (2004) 1-32.

[25] Z.-Z.Bai, G.H. Golub, L.-Z. Lu, J.-F. Yin, Block triangular and skew-Hermitian splitting methods for positive-definite linear systems, SIAM J. Sci. Comput., 26 (2005) 844-863.

[26] F. Mezzadri, E. Galligani, Modulus-based matrix splitting methods for horizontal linear complementarity problems, Numer. Algor., 83 (2020) 201-219.

[27] F. D. Fomeni, S. A. Gabriel, M. F. Anjos, An RLT approach for solving the binary-constrained mixed linear complementarity problem, Comput. Oper. Res., 110 (2019) 48-59.

[28] A. Dutta, A.K. Das, Bounded homotopy path approach to the solution of linear complementarity problems, Opsearch., (2023) https://doi.org/10.1007/s12597-023-00687-4

[29] S. Seifollahzadeh, G. Ebadi, On diagonal and off-diagonal splitting-based iteration method to solve absolute value equations, Filomat, 38:6 (2024) 2001-2013.