Numerical Algorithms (2021) 86:1791–1810
https://doi.org/10.1007/s11075-020-00954-1

ORIGINAL PAPER

A two-step modulus-based matrix splitting iteration method for horizontal linear complementarity problems

Hua Zheng[1] · **Seakweng Vong**[2]

Received: 30 December 2019 / Accepted: 20 May 2020 / Published online: 29 May 2020

Abstract
In this paper, for solving horizontal linear complementarity problems, a two-step modulus-based matrix splitting iteration method is established. The convergence analysis of the proposed method is presented, including the case of accelerated over-relaxation splitting. Numerical examples are reported to show the efficiency of the proposed method.

Keywords Horizontal linear complementarity problem · Modulus-based method · H_+-matrix

1 Introduction

Given $A, B \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$, the horizontal linear complementarity problem (abbreviated as HLCP) is to find vectors $z, r \in \mathbb{R}^n$ such that

$$Az - Br + q = 0,\ z, r \geq 0 \text{ and } z^T r = 0, \tag{1}$$

where for two matrices $F = (f_{ij}), G = (g_{ij}) \in \mathbb{R}^{m\times n}$ the order $F \geq (>)G$ means $f_{ij} \geq (>)g_{ij}$ for any i and j.

Clearly, when either A or B is an $n \times n$ identity matrix, the HLCP reduces to the well-known linear complementarity problem (abbreviated as LCP) [9]. On the other hand, the HLCP can be viewed as the simplest case of the extended HLCP [8], and

✉ Seakweng Vong
swvong@umac.mo

Hua Zheng
hzhengjlfdu@gmail.com

[1] School of Mathematics and Statistics, Shaoguan University, Shaoguan, People's Republic of China

[2] Department of Mathematics, University of Macau, Macau, People's Republic of China

it has wide applications in the fields of control theory, hydrodynamic lubrication, nonlinear networks (see [12–15, 26, 27] for details).

Recently, for solving HLCPs, some matrix splitting-based methods were introduced. In particular, the projected splitting methods were introduced in [22], while modulus-based matrix splitting (MMS) methods were introduced in [23]. Specially, the MMS method was shown to be more efficient than the interior-point method [21] and the projected splitting method with global convergence. Weaker assumptions and larger convergence domain of the MMS method were given in [41] when the system matrices are assumed to be H_+-matrices. The convergence of modulus-based Jacobi (MJ) and modulus-based Gauss-Seidel (MGS) methods was discussed in [25] for the HLCP from hydrodynamic lubrication. The MMS technique had also been used as a successful tool for LCPs [3–5, 10, 18–20, 30, 34, 39, 42], and some classes of nonlinear complementarity problems (NCP) [17, 29, 32, 35, 36, 38, 40] in recent years. On the other hand, without matrix splitting, the HLCP was solved by nonsmooth Newton's method based on the equivalent modulus equation (see [24]). The technique of nonsmooth Newton's iteration had also been successfully used in LCPs [33] and NCPs [37] with locally fast convergence rate.

In this paper, we aim at further accelerating of the MMS method for the HLCP. To achieve higher computing efficiency, and to make full use of the information contained in the system matrices, the technique of two-step splitting had been used for LCPs and NCPs (see [30, 36, 39] for details). It is therefore interesting to study the two-step MMS method for the HLCP as well. Thus, we first propose the two-step MMS method for the HLCP by employing two-step matrix splittings in Section 2. In Section 3, we present the convergence analysis for the proposed method. Specifically, the accelerated overrelaxation (AOR) splitting is analyzed. The proposed method and theorems are shown to generalized the existing theories of the MMS method. Next, by numerical examples, we show that the proposed method is more efficient than the MMS method in Section 4. Finally, Section 5 concludes this work.

Next, some needed notations, definitions and known results are given.

Let e_n be an $n \times 1$ vector whose elements are all equal to 1. Let $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ and $A = D_A - L_A - U_A = D_A - C_A$, where $D_A, -L_A, -U_A$ and $-C_A$ denote the diagonal, the strictly lower-triangular, the strictly upper-triangular and nondiagonal matrices of A, respectively. tridiag$(a; b; c)$ denotes a tridiagonal matrix with vectors a, b, c forming its subdiagonal, main diagonal and superdiagonal entries, respectively, while blktridiag$(A; B; C)$ denotes a block tridiagonal matrix with block matrices $I \otimes A, I \otimes B, I \otimes C$ as its corresponding block entries, where"$\otimes$" is the Kronecker product and I is the identity matrix.

By $|A|$ we denote $|A| = (|a_{ij}|)$ and the comparison matrix of A is $\langle A\rangle = (\langle a_{ij}\rangle)$, where $\langle a_{ij}\rangle = |a_{ij}|$ if $i = j$ and $\langle a_{ij}\rangle = -|a_{ij}|$ if $i \neq j$. A is called a Z-matrix if $a_{ij} \leq 0$ for any $i \neq j$, a nonsingular M-matrix if it is a nonsingular Z-matrix with $A^{-1} \geq 0$, an H-matrix if $\langle A\rangle$ is a nonsingular M-matrix (e.g., see [6]). A is called an H_+-matrix if A is an H-matrix with $a_{ii} > 0$ for every i (e.g., see [2]). If $|a_{ii}| > \sum_{j\neq i} |a_{ij}|$ for all $1 \leq i \leq n$, A is called a strictly diagonal dominant (s.d.d.)

matrix. By $\rho(A)$ we denote the spectral radius of A. $A = M - N$ is called an H-splitting if $\langle M\rangle - |N|$ is a nonsingular M-matrix; and an H-compatible splitting if $\langle A\rangle = \langle M\rangle - |N|$ (e.g., see [31]).

2 Two-step method

First, we introduce the MMS method for solving the HLCP.

Let $A = M_A - N_A$, $B = M_B - N_B$ be two splittings of A and B, respectively. Then, with $z = \frac{1}{\gamma}(|x| + x)$ and $r = \frac{1}{\gamma}\Omega(|x| - x)$, the HLCP can be equivalently transformed into a system of fixed-point equations

$$(M_B\Omega + M_A)x = (N_B\Omega + N_A)x + (B\Omega - A)|x| - \gamma q, \tag{2}$$

where Ω is a positive diagonal parameter matrix and γ is a positive constant (see [23] for more details). Based on (2), the MMS method is presented as follows:

Method 1 [23] Let $\Omega \in \mathbb{R}^{n\times n}$ be a positive diagonal matrix, γ be a positive constant and $A = M_A - N_A$, $B = M_B - N_B$ be two splittings of the matrix $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$, respectively. Given an initial vector $x^{(0)} \in \mathbb{R}^n$, compute $x^{(k+1)} \in \mathbb{R}^n$ by solving the linear system

$$(M_B\Omega + M_A)x^{(k+1)} = (N_B\Omega + N_A)x^{(k)} + (B\Omega - A)|x^{(k)}| - \gamma q.$$

Then set

$$z^{(k+1)} = \frac{1}{\gamma}(|x^{(k+1)}| + x^{(k+1)}) \quad \text{and} \quad r^{(k+1)} = \frac{1}{\gamma}\Omega(|x^{(k+1)}| - x^{(k+1)})$$

for $k = 0, 1, 2, \ldots$, until the iteration sequence $\{(z^{(k)}, r^{(k)})\}_{k=1}^{+\infty}$ is convergent.

With the same idea used in LCPs and NCPs, in order to achieve high computing efficiency, we consider making use of the information in the matrices A and B. The two-step modulus-based matrix splitting (TMMS) iteration method for solving the HLCP is presented as follows:

Method 2 Two-step modulus-based matrix splitting iteration method for the HLCP For any given positive diagonal matrix $\Omega \in \mathbb{R}^{n\times n}$ and $\gamma > 0$, let $A = M_{A_1} - N_{A_1} = M_{A_2} - N_{A_2}$ be two splittings of the matrix $A \in \mathbb{R}^{n\times n}$, while $B = M_{B_1} - N_{B_1} = M_{B_2} - N_{B_2}$ be two splittings of the matrix $B \in \mathbb{R}^{n\times n}$. Given an initial vector $x^{(0)} \in \mathbb{R}^n$, compute $x^{(k+1)} \in \mathbb{R}^n$ by solving the two linear systems

$$\begin{cases} (M_{B_1}\Omega + M_{A_1})x^{(k+\frac{1}{2})} = (N_{B_1}\Omega + N_{A_1})x^{(k)} + (B\Omega - A)|x^{(k)}| - \gamma q, \\ (M_{B_2}\Omega + M_{A_2})x^{(k+1)} = (N_{B_2}\Omega + N_{A_2})x^{(k+\frac{1}{2})} + (B\Omega - A)|x^{(k+\frac{1}{2})}| - \gamma q. \end{cases} \tag{3}$$

Then set

$$z^{(k+1)} = \frac{1}{\gamma}(|x^{(k+1)}| + x^{(k+1)}) \quad \text{and} \quad r^{(k+1)} = \frac{1}{\gamma}\Omega(|x^{(k+1)}| - x^{(k+1)})$$

for $k = 0, 1, 2, \ldots$, until the iteration sequence $\{(z^{(k)}, r^{(k)})\}_{k=1}^{+\infty}$ is convergent.

Note that if we take $M_{A_1} = M_{A_2}, N_{A_1} = N_{A_2}, M_{B_1} = M_{B_2}$ and $N_{B_1} = N_{B_2}$, Method 2 reduces to Method 1 by combining the two iterations in (3).

Furthermore, by the same definitions as those in [30] of two-step matrix splittings, taking

$$\begin{cases} M_{A_1} = \frac{1}{\alpha}(D_A - \beta L_A), M_{A_2} = \frac{1}{\alpha}(D_A - \beta U_A), \\ M_{B_1} = \frac{1}{\alpha}(D_B - \beta L_B), M_{B_2} = \frac{1}{\alpha}(D_B - \beta U_B). \end{cases} \tag{4}$$

we can obtain the two-step modulus-based accelerated overrelaxation (TMAOR) iteration method. Taking $\alpha = \beta$ and $\alpha = \beta = 1$, the TMAOR iteration method reduces to the two-step modulus-based successive overrelaxation (TMSOR) iteration method and the two-step modulus-based Gauss-Seidel (TMGS) iteration method, respectively. Furthermore, to achieve fast convergence rate, one can generalize the MAOR iteration method by setting more choices of the relaxation parameters than (4) in the two-step matrix splittings, e.g.,

$$\begin{cases} M_{A_1} = \frac{1}{\alpha_1}(D_A - \beta_1 L_A), M_{A_2} = \frac{1}{\alpha_1}(D_A - \beta_1 U_A), \\ M_{B_1} = \frac{1}{\alpha_2}(D_B - \beta_2 L_B), M_{B_2} = \frac{1}{\alpha_2}(D_B - \beta_2 U_B). \end{cases} \tag{5}$$

3 Convergence analysis

It is well known that the HLCP has a unique solution for any known vector q if and only if $\{A, B\}$ satisfies the $\mathcal{W}$-property [28], i.e., if all column-representative matrices of $\{A, B\}$ have determinants that are either all positive or all negative. As remarked in [23], that both A and B are H_+-matrices is in such cases. In the following discussion, the convergence analysis of Method 2 is presented when the HLCP is always assumed to have a unique solution with both A and B being H_+-matrices.

We present some useful lemmas first.

Lemma 1 [11] *Let A be an H-matrix. Then $|A^{-1}| \leq \langle A \rangle^{-1}$.*

Lemma 2 [16] *Let $B \in \mathbb{R}^{n\times n}$ be an s.d.d. matrix. Then, $\forall C \in \mathbb{R}^{n\times n}$, $||B^{-1}C||_\infty \leq \max\limits_{1\leq i\leq n} \frac{(|C|e)_i}{(\langle B\rangle e)_i}$ holds.*

Lemma 3 [6] *Assume that A is a Z-matrix. Then the following statements are equivalent:*

- *A is an M-matrix;*
- *There exists a positive diagonal matrix D, such that AD is an s.d.d. matrix with positive diagonal entries;*
- *If $A = M - N$ satisfies $M^{-1} \geq 0$ and $N \geq 0$, then $\rho(M^{-1}N) < 1$.*

Let (z^*, r^*) be a solution of the HLCP. Then, by (2) and straightforward computation, we can get that $x^* = \frac{\gamma}{2}(z^* - \Omega^{-1}r^*)$ satisfies the implicit fixed-point equations

$$\begin{cases} (M_{B_1}\Omega + M_{A_1})x^* = (N_{B_1}\Omega + N_{A_1})x^* + (B\Omega - A)|x^*| - \gamma q, \\ (M_{B_2}\Omega + M_{A_2})x^* = (N_{B_2}\Omega + N_{A_2})x^* + (B\Omega - A)|x^*| - \gamma q. \end{cases} \tag{6}$$

By subtracting (6) from (3), we have

$$\begin{cases} (M_{B_1}\Omega + M_{A_1})(x^{(k+\frac{1}{2})} - x^*) = (N_{B_1}\Omega + N_{A_1})(x^{(k)} - x^*) \\ \qquad\qquad\qquad\qquad\qquad\quad +(B\Omega - A)(|x^{(k)}| - |x^*|) \\ (M_{B_2}\Omega + M_{A_2})(x^{(k+1)} - x^*) = (N_{B_2}\Omega + N_{A_2})(x^{(k+\frac{1}{2})} - x^*) \\ \qquad\qquad\qquad\qquad\qquad\quad +(B\Omega - A)(|x^{(k+\frac{1}{2})}| - |x^*|). \end{cases} \tag{7}$$

To prove $\lim_{k\to+\infty} z^{(k)} = z^*$ and $\lim_{k\to+\infty} r^{(k)} = r^*$, we need only to prove $\lim_{k\to+\infty} x^{(k)} = x^*$.

Theorem 1 *Let $\Omega = (\omega_{jj})$ be an $n \times n$ positive diagonal matrix and $A, B \in \mathbb{R}^{n\times n}$ be two H_+-matrices. Let D be a positive diagonal matrix such that $\langle A\rangle D$ is an s.d.d. matrix. Assume that:*

(I) *$A = M_{A_1} - N_{A_1} = M_{A_2} - N_{A_2}$ and $B = M_{B_1} - N_{B_1} = M_{B_2} - N_{B_2}$ are two H-compatible splittings of A and B, respectively;*

(II) *$|b_{ij}|\omega_{jj} \le |a_{ij}|$ $(i \ne j)$ and $sign(b_{ij}) = sign(a_{ij})$ $(b_{ij} \ne 0)$, for all i, j.*

Then the iteration sequence $\{(z^{(k)}, r^{(k)})\}_{k=1}^{+\infty}$ generated by Method 2 converges to the unique solution (z^*, r^*) of the HLCP for any initial vector $x^{(0)} \in \mathbb{R}^n$ provided

$$\Omega e > D_B^{-1}D_A e - D_B^{-1}D^{-1}\langle A\rangle De. \tag{8}$$

Proof By the assumption (I), we have

$$\begin{aligned} &\langle M_{B_1}\Omega + M_{A_1}\rangle De \\ \ge\ &(\langle M_{B_1}\rangle\Omega + \langle M_{A_1}\rangle)De \\ =\ &(\langle A\rangle + \langle B\rangle\Omega + |N_{A_1}| + |N_{B_1}|\Omega)De \\ \ge\ &(D_A + D_B\Omega - |C_A| - |C_B|\Omega)De. \end{aligned} \tag{9}$$

On the other hand, assumption (II) implies that

$$|C_A| \ge |C_B|\Omega. \tag{10}$$

If $\Omega \ge D_B^{-1}D_A$, by (9) and (10), we get

$$\langle M_{B_1}\Omega + M_{A_1}\rangle De \ge 2(D_A - |C_A|)De = 2\langle A\rangle De > 0. \tag{11}$$

If $D_B^{-1}D_Ae > \Omega e > D_B^{-1}D_Ae - D_B^{-1}D^{-1}\langle A\rangle De$ holds, we have

$$\begin{aligned} & DD_B\left[\Omega - (D_B^{-1}D_A - D_B^{-1}D^{-1}\langle A\rangle D)\right]e > 0 \\ \Rightarrow & \left[DD_B\Omega D^{-1} - (DD_AD^{-1} - \langle A\rangle)\right]De > 0 \\ \Rightarrow & \big(D_B\Omega - |C_A|\big)De > 0. \end{aligned} \tag{12}$$

By (9), (10) and 12, we get

$$\langle M_{B_1}\Omega + M_{A_1}\rangle De \geq (D_A - |C_B|\Omega)De \geq (D_A - |C_A|)De = \langle A\rangle De > 0. \tag{13}$$

Thus, if (8) holds, by (11) and (13), we have that $\langle M_{B_1}\Omega + M_{A_1}\rangle D$ is an s.d.d. matrix, which implies that $M_{B_1}\Omega + M_{A_1}$ is an H-matrix. Then, by Lemma 1 and the first equality of (7), we have

$$\begin{aligned} & |x^{(k+\frac{1}{2})} - x^*| \\ = & \left|(M_{B_1}\Omega + M_{A_1})^{-1}\left[(N_{B_1}\Omega + N_{A_1})(x^{(k)} - x^*) + (B\Omega - A)(|x^{(k)}| - |x^*|)\right]\right| \\ \leq & \langle M_{B_1}\Omega + M_{A_1}\rangle^{-1}(|N_{B_1}\Omega + N_{A_1}| + |B\Omega - A|)|x^{(k)} - x^*| \\ \doteq & \mathcal{P}_1|x^{(k)} - x^*|, \end{aligned}$$

where

$$\mathcal{P}_1 = \mathcal{M}_1^{-1}\mathcal{N}_1, \mathcal{M}_1 = \langle M_{B_1}\Omega + M_{A_1}\rangle, \mathcal{N}_1 = |N_{B_1}\Omega + N_{A_1}| + |B\Omega - A|.$$

By Lemma 2, we have

$$||D^{-1}\mathcal{P}_1D||_\infty = ||(\mathcal{M}_1D)^{-1}(\mathcal{N}_1D)||_\infty \leq \max_{1\leq i\leq n}\frac{(\mathcal{N}_1De)_i}{(\mathcal{M}_1De)_i}. \tag{14}$$

If (8) holds, we obtain

$$\begin{aligned} & \mathcal{M}_1De - \mathcal{N}_1De \\ = & (\langle M_{B_1}\Omega + M_{A_1}\rangle - |N_{B_1}\Omega + N_{A_1}| - |B\Omega - A|)De \\ \geq & (\langle M_{B_1}\Omega\rangle + \langle M_{A_1}\rangle - |N_{B_1}\Omega| - |N_{A_1}| - |D_B\Omega - D_A| - |C_A| + |C_B\Omega|)De \\ = & (D_B\Omega + \langle A\rangle - |C_A| - |D_B\Omega - D_A|)De \\ = & \begin{cases} 2\langle A\rangle De, & \text{if } \Omega \geq D_AD_B^{-1}; \\ 2\big(D_B\Omega - |C_A|\big)De, & \text{if } D_B^{-1}D_Ae > \Omega e > D_B^{-1}D_Ae - D_B^{-1}D^{-1}\langle A\rangle De. \end{cases} \end{aligned}$$

Then by (11), (12) and (14), we have $||D^{-1}\mathcal{P}_1D||_\infty < 1$.

Similarly, by the second equality of (7), we have

$$|x^{(k+1)} - x^*| \leq \mathcal{P}_2|x^{(k+\frac{1}{2})} - x^*|,$$

where

$$\mathcal{P}_2 = \mathcal{M}_2^{-1}\mathcal{N}_2, \mathcal{M}_2 = \langle M_{B_2}\Omega + M_{A_2}\rangle, \mathcal{N}_2 = |N_{B_2}\Omega + N_{A_2}| + |B\Omega - A|.$$

Thus, we get

$$|x^{(k+1)} - x^*| \leq \mathcal{P}_2\mathcal{P}_1|x^{(k)} - x^*|, \tag{15}$$

By proceeding the same deduction, we can get that $\langle \mathcal{M}_2 \rangle D$ is an s.d.d. matrix. With the same discussion, if (8) holds, we have $||D^{-1}\mathcal{P}_2 D||_\infty < 1$, too. Hence, the next inequality holds:

$$\begin{aligned}\rho(\mathcal{P}_2\mathcal{P}_1) &= \rho(D^{-1}\mathcal{P}_2\mathcal{P}_1 D)\\ &\le ||D^{-1}\mathcal{P}_2 D D^{-1}\mathcal{P}_1 D||_\infty\\ &\le ||D^{-1}\mathcal{P}_2 D||_\infty ||D^{-1}\mathcal{P}_1 D||_\infty < 1,\end{aligned}$$

which implies that $\lim_{k\to+\infty} x^{(k)} = x^*$ by (15), proving the claim. □

Next, we present the convergence results for the TMAOR.

Lemma 4 *Let A, B be two H_+-matrices. If $0 < \alpha < \frac{1}{\rho\left[(D_A+D_B\Omega)^{-1}(D_B\Omega+|C_A|)\right]}$, there exists a positive diagonal matrix $\bar{D}$, such that*

$$\left[\frac{1+\alpha-|1-\alpha|}{\alpha}D_A + \frac{1-\alpha-|1-\alpha|}{\alpha}D_B\Omega - 2|C_A|\right]\bar{D}$$

is an s.d.d. matrix.

Proof Since A is an H_+-matrix, we have $\langle A\rangle = D_A - |C_A| = (D_A + D_B\Omega) - |D_B\Omega + C_A|$ is an M-matrix. By Lemma 3, we can get $\rho\left[(D_A + D_B\Omega)^{-1}(D_B\Omega + |C_A|)\right] < 1$.

If $0 < \alpha \le 1$, then we have $\frac{1+\alpha-|1-\alpha|}{\alpha}D_A + \frac{1-\alpha-|1-\alpha|}{\alpha}D_B\Omega - 2|C_A| = 2\langle A\rangle$ is an M-matrix.

If $1 < \alpha < \frac{1}{\rho\left[(D_A+D_B\Omega)^{-1}(D_B\Omega+|C_A|)\right]}$, then we have $\frac{1}{\alpha}I - (D_A+D_B\Omega)^{-1}(D_B\Omega + |C_A|)$ is an M-matrix, which implies that

$$\begin{aligned}&\frac{1+\alpha-|1-\alpha|}{\alpha}D_A + \frac{1-\alpha-|1-\alpha|}{\alpha}D_B\Omega - 2|C_A|\\ &= 2(D_A + D_B\Omega)\left[\frac{1}{\alpha}I - (D_A + D_B\Omega)^{-1}(D_B\Omega + |C_A|)\right]\end{aligned}$$

is an M-matrix. Therefore, by Lemma 3, such $\bar{D}$ in the conclusion exists. □

Remark 1 For computing the positive diagonal matrix $\bar{D}$ given in the assumption of Lemma 4, readers can refer to [1, 7, 18].

Theorem 2 *Let $A, B \in \mathbb{R}^{n\times n}$ be two H_+-matrices and $\Omega \in \mathbb{R}^{n\times n}$ be a positive diagonal matrix satisfying $\Omega \ge D_A D_B^{-1}$. Furthermore, for $i, j = 1, 2, \ldots, n$, let $|b_{ij}|\omega_{jj} \le |a_{ij}|(i \ne j)$ and $sign(b_{ij}) = sign(a_{ij})(b_{ij} \ne 0)$. Then, the iteration sequence $(z^{(k)}, r^{(k)})_{k=1}^{+\infty}$ generated by the TMAOR method converges to the unique solution (z^*, r^*) of (1) for any initial vector $x^{(0)} \in \mathbb{R}^n$ provided*

$$0 < \beta \le \alpha < \frac{1}{\rho\left[(D_A + D_B\Omega)^{-1}(D_B\Omega + |C_A|)\right]}. \tag{16}$$

Proof With the same notations and discussion as the proof of Theorem 1, let $\bar{D}$ be the positive diagonal matrix given by Lemma 4. If $\Omega \geq D_A D_B^{-1}$, by Lemma 4 and (16), we have

$$\begin{aligned}
&\mathcal{M}_1\bar{D} - \mathcal{N}_1\bar{D}e \\
&= \left[\langle M_{A_1} + M_{B_1}\Omega\rangle - |N_{A_1} + N_{B_1}\Omega| - |B\Omega - A|\right]\bar{D}e \\
&= \left[\frac{1}{\alpha}(D_A + D_B\Omega) - \frac{\beta}{\alpha}|L_A + L_B\Omega| - \frac{|1-\alpha|}{\alpha}(D_A + D_B\Omega) - \frac{|\beta-\alpha|}{\alpha}|L_A + L_B\Omega|\right. \\
&\quad \left. -|U_A + U_B\Omega| - |D_B\Omega - D_A| - |C_B\Omega - C_A|\right]\bar{D}e \\
&= \left[\frac{1+\alpha-|1-\alpha|}{\alpha}D_A + \frac{1-\alpha-|1-\alpha|}{\alpha}D_B\Omega - |C_B\Omega + C_A| - |C_B\Omega - C_A|\right]\bar{D}e \\
&= \left[\frac{1+\alpha-|1-\alpha|}{\alpha}D_A + \frac{1-\alpha-|1-\alpha|}{\alpha}D_B\Omega - 2|C_A|\right]\bar{D}e \\
&> 0.
\end{aligned}$$

Hence $\mathcal{M}_1\bar{D}$ is an s.d.d. matrix. Then by Lemma 1, we have $||\bar{D}^{-1}\mathcal{P}_1\bar{D}||_\infty < 1$. Furthermore, by (4) and the same discussion as above, it is easy to get $||\bar{D}^{-1}\mathcal{P}_2\bar{D}||_\infty < 1$ too. Therefore we have

$$\begin{aligned}
\rho(\mathcal{P}_2\mathcal{P}_1) &= \rho(V^{-1}\mathcal{P}_2\mathcal{P}_1 V) \\
&\leq ||\bar{D}^{-1}\mathcal{P}_2\bar{D}\bar{D}^{-1}\mathcal{P}_1\bar{D}||_\infty \\
&\leq ||\bar{D}^{-1}\mathcal{P}_2\bar{D}||_\infty||\bar{D}^{-1}\mathcal{P}_1\bar{D}||_\infty < 1,
\end{aligned}$$

which implies the TMAOR method is convergent. □

Note that, there is no assumption on the two-step splittings in Theorem 2 and its proof. Thus, we have the next corollary.

Corollary 1 *With the same notations and assumptions as Theorem 2, the MAOR method is convergent.*

Remark 2 It is well known that, when the system matrix is an H_+-matrix, H-splitting or H-compatible splitting, including the classic AOR splitting, is usually the sufficient condition to guarantee the convergence of matrix splitting iteration methods (see [6] for the linear equations and the references mentioned in Section 1 for LCPs or NCPs). Note that Corollary 1 discusses the convergence conditions of the MAOR method for the HLCP for the first time. Thus, Corollary 1 provides a more explicit theoretical analysis of Method 1 than those given in [23] and [41].

Remark 3 In Theorems 3.1 and 3.2, the elements of A and B are linked by some relationship where the assumption (II) with $D_B\Omega \geq D_A$ had been called "B is more or equally diagonally dominant than A" in [23]. The condition even excludes the simple case when $A = I$. However, if we denote the HLCP given in (1) by HLCP(A, B, q), it is easy to see that (z, r) is a solution of HLCP(A, B, q) if and only if (r, z) is a solution of HLCP(B, A, $-q$). Therefore, for HLCP(B, A, $-q$), with the same discussion

as that given in [23], the proposed method and convergence analysis can be reformulated by replacing all A, B, q by $B, A, -q$, respectively. For example, the equivalent modulus equation of HLCP($B, A, -q$) is

$$(M_B + M_A\Omega)x = (N_B + N_A\Omega)x + (A\Omega - B)|x| + \gamma q,$$

and the assumption (II) should be changed to "$|a_{ij}|\omega_{jj} \geq |b_{ij}| \quad (i \neq j)$ and $\text{sign}(b_{ij}) = \text{sign}(a_{ij}) \quad (a_{ij} \neq 0)$".

4 Numerical examples

In this section, numerical examples are given to show the efficiency of the proposed method. The computations were run on an Intel® Core™, where the CPU is 2.50 GHz and the memory is 4.00 GB.

Let I_n be the identity matrix of order n. Consider the following three examples.

Example 1 [23, 41] Let $n = m^2$, $A = \hat{A} + \mu I_n$, $B = \hat{B} + \nu I_n$ and $q = Az^* - Bw^*$, where μ, ν are real parameters, $\hat{A} = \text{blktridiag}(-I_m, S, -I_m)$, $\hat{B} = I_m \otimes S$ and $S = \text{tridiag}(-e_m; 4e_m; -e_m) \in \mathbb{R}^{m\times m}$.

Example 1 can come from the discretization of a 2-D boundary problem like

$$\triangle z + \frac{\partial^2 r}{\partial^2 u} + \mu z + \nu r - q = 0, z \geq 0, r \geq 0 \text{ and } z^T r = 0.$$

by five-point difference scheme with suitable boundary conditions, where $z(u, v), r(u, v)$ and $q(u, v)$ are three 2-D mapings. The similar examples had been analyzed in [4] and [17] for the LCP and the NCP, respectively. It is easy to see that both A and B are symmetric positive definite matrices.

Example 2 [23, 41] Let $n = m^2$, $A = \hat{A} + \mu I_n$, $B = \hat{B} + \nu I_n$ and $q = Az^* - Bw^*$, where μ, ν are real parameters, $\hat{A} = \text{blktridiag}(-1.5I_m; S; -0.5I_m)$, $\hat{B} = I_m \otimes S$ and $S = \text{tridiag}(-1.5e_m; 4e_m; -0.5e_m) \in \mathbb{R}^{m\times m}$.

Note that the system matrices A and B in Example 2 belong to asymmetric positive definite cases.

Example 3 Consider Problem 1 in the numerical experiments of [21], where the HLCP is from the discretization of the Reynolds equation [15], which models a 1-D test problem in hydrodynamic lubrication with Convergent-Divergent profile.

For this example, it is shown by [25] that the assumption (II) of Theorem 1 cannot be satisfied. Although the convergence of the MJ and MGS methods had been proved by another way in [25], it is difficult to prove the convergence of the MSOR method or the MAOR method. Nevertheless, we still show the numerical results of Method 2 based on the MJ and MGS methods for Example 3.

Table 1 Abbreviations of testing methods

	Examples 1 and 2			Example 3	
(α, β)	Method 1	AMMS	Method 2	$(\alpha_1, \beta_1, \alpha_2, \beta_2)$	Method 2
(1.2,1.2)	$MSOR_1$	$AMSOR_1$	$TMSOR_1$	(1.0,0.0,0.9,0.9)	$TMJSOR_1$
(1.0,1.0)	$MSOR_2$	$AMSOR_2$	$TMSOR_2$	(1.0,0.0,1.0,1.0)	$TMJSOR_2$
(0.8,0.8)	$MSOR_3$	$AMSOR_3$	$TMSOR_3$	(1.0,0.0,1.1,1.1)	$TMJSOR_3$
(1.3,1.1)	$MAOR_1$	$AMAOR_1$	$TMAOR_1$	(1.0,0.0,1.2,1.2)	$TMJSOR_4$
(1.1,0.9)	$MAOR_2$	$AMAOR_2$	$TMAOR_2$	(1.0,1.0,1.0,0.0)	TMGSJ
(0.9,0.7)	$MAOR_3$	$AMAOR_3$	$TMAOR_3$	(1.0,1.0,0.9,0.9)	$TMGSSOR_1$
				(1.0,1.0,1.1,1.1)	$TMGSSOR_2$
				(1.0,1.0,1.2,1.2)	$TMGSSOR_3$

Besides Method 1, the accelerated modulus-based matrix splitting method (AMMS) is also included in the numerical experiments to compare with Method 2. By utilizing a pair of suitable matrix splitting to the matrices A and B in (2), the AMMS method was proposed in [23] and shown to converge faster than the MMS method. Specially, let $A = M_{A_1} - N_{A_1} = M_{A_2} - N_{A_2}$ and $B = M_{B_1} - N_{B_1} = M_{B_2} - N_{B_2}$ be two pairs of matrix splittings of A and B, respectively, where

$$\begin{cases} M_{A_1} = \frac{1}{\alpha}(D_A - \beta L_A), M_{B_1} = \frac{1}{\alpha}(D_B - \beta L_B), \\ M_{A_2} = D_A - U_A, M_{B_2} = D_B - U_B. \end{cases}$$

The accelerated modulus-based accelerated overrelaxation (AMAOR) iteration method is based on solving the next equation in the kth iteration:

$$(M_{B_1}\Omega + M_{A_1})x^{(k+1)} + (N_{B_2}\Omega - N_{A_2})|x^{(k+1)}| = (N_{B_1}\Omega + N_{A_1})x^{(k)} + (M_{B_2}\Omega - M_{A_2})|x^{(k)}| - \gamma q.$$

In the following, we first compare the TMAOR method with the MAOR method and the AMAOR method with different pairs of relaxation parameters for Example 1

Table 2 Upper bounds of α in (16)

m	$\Omega = \tau D_A D_B^{-1}$	Example 1 $(\mu = \nu = 4)$	Example 1 $(\mu = \nu = 0)$	Example 2 $(\mu = \nu = 4)$
128	$\tau = 1.20$	1.29	1.00	1.34
	$\tau = 1.00$	1.33	1.00	1.39
256	$\tau = 1.20$	1.29	1.00	1.34
	$\tau = 1.00$	1.33	1.00	1.39
512	$\tau = 1.20$	1.29	1.00	1.34
	$\tau = 1.00$	1.33	1.00	1.39

Table 3 Numerical results of Example 1 when $\mu = \nu = 4$ and $\Omega = \tau D_A D_B^{-1}$

		$m = 128$		$m = 256$		$m = 512$			$m = 128$		$m = 256$		$m = 512$	
τ	Method	IT	CPU	IT	CPU	IT	CPU	Method	IT	CPU	IT	CPU	IT	CPU
1.2	$MSOR_1$	32	0.1000	32	0.3242	32	1.8762	$MAOR_1$	47	0.1227	48	0.4820	49	2.7035
	$AMSOR_1$	27	0.0844	27	0.2735	27	1.5830	$AMAOR_1$	38	0.0992	39	0.3897	40	2.1858
	$TMSOR_1$	17	0.0780	17	0.2512	17	1.3928	$TMAOR_1$	24	0.0950	24	0.4065	25	2.0274
	$MSOR_2$	28	0.0482	29	0.2820	30	1.5922	$MAOR_2$	25	0.0750	26	0.2429	26	1.4346
	$AMSOR_2$	26	0.0448	27	0.2619	28	1.4785	$AMAOR_2$	22	0.0660	23	0.2138	23	1.2624
	$TMSOR_2$	14	0.0377	15	0.2189	15	1.1795	$TMAOR_2$	13	0.0604	13	0.2087	13	1.0638
	$MSOR_3$	42	0.0923	43	0.4064	44	2.3671	$MAOR_3$	36	0.1132	37	0.3604	38	2.1589
	$AMSOR_3$	40	0.0879	41	0.3870	42	2.2544	$AMAOR_3$	34	0.1069	35	0.3404	36	2.0390
	$TMSOR_3$	21	0.0728	22	0.3336	22	1.7735	$TMAOR_3$	18	0.0780	19	0.3134	19	1.6003
1.0	$MSOR_1$	29	0.0927	30	0.3147	30	1.5486	$MAOR_1$	43	0.1315	43	0.4901	44	2.4621
	$AMSOR_1$	26	0.0831	27	0.2821	27	1.3884	$AMAOR_1$	37	0.1132	37	0.4217	38	2.1186
	$TMSOR_1$	14	0.0641	14	0.2232	15	1.1294	$TMAOR_1$	20	0.0856	21	0.4081	21	1.7182
	$MSOR_2$	24	0.0520	24	0.2243	26	1.4005	$MAOR_2$	24	0.0497	24	0.2362	25	1.3861
	$AMSOR_2$	22	0.0477	22	0.2056	24	1.2838	$AMAOR_2$	22	0.0456	22	0.2165	23	1.2706
	$TMSOR_2$	12	0.0357	12	0.1923	13	1.0011	$TMAOR_2$	12	0.0419	12	0.1824	12	0.9846
	$MSOR_3$	38	0.1042	39	0.3524	40	2.1826	$MAOR_3$	33	0.0967	33	0.3216	34	1.9707
	$AMSOR_3$	36	0.0987	37	0.3339	38	2.0677	$AMAOR_3$	31	0.0908	31	0.3021	32	1.8513
	$TMSOR_3$	19	0.0811	20	0.2997	20	1.5913	$TMAOR_3$	17	0.0758	17	0.2846	17	1.4109
0.8	$MSOR_1$	39	0.1133	40	0.3802	40	2.1254	$MAOR_1$	63	0.1535	64	0.6542	65	3.6814
	$AMSOR_1$	33	0.0959	34	0.3217	34	1.7984	$AMAOR_1$	47	0.1145	48	0.4881	48	2.7464
	$TMSOR_1$	19	0.0822	19	0.2858	20	1.5937	$TMAOR_1$	30	0.1097	30	0.4795	31	2.5010
	$MSOR_2$	22	0.0507	22	0.2321	23	1.2287	$MAOR_2$	31	0.0969	32	0.3308	32	1.8493

Table 3 (continued)

		$m = 128$		$m = 256$		$m = 512$			$m = 128$		$m = 256$		$m = 512$	
τ	Method	IT	CPU	IT	CPU	IT	CPU	Method	IT	CPU	IT	CPU	IT	CPU
	$AMSOR_2$	21	0.0484	21	0.2216	22	1.1729	$AMAOR_2$	27	0.0844	28	0.2881	28	1.6107
	$TMSOR_2$	11	0.0343	11	0.1668	12	0.9483	$TMAOR_2$	15	0.0744	15	0.2549	16	1.3474
	$MSOR_3$	35	0.1025	36	0.3538	37	2.0534	$MAOR_3$	30	0.0828	31	0.2780	32	1.8217
	$AMSOR_3$	34	0.0996	35	0.3437	36	1.9947	$AMAOR_3$	29	0.0800	30	0.2687	31	1.7610
	$TMSOR_3$	18	0.0780	18	0.3132	19	1.5102	$TMAOR_3$	15	0.0686	16	0.2481	16	1.3718

and Example 2. On the other hand, besides the TMAOR method, more choices of the two-step matrix splittings in the TMMS method given by (5) are tested for Example 3. The abbreviations of the corresponding terminologies are shown in Table 1. Note that the AOR splitting reduces to the SOR splitting when $\alpha = \beta$, and to Gauss-Seidel splitting (the MSOR_2, the AMSOR_2 and the TMSOR_2) when $\alpha = \beta = 1$, respectively. Let $\gamma = 1$ and $\Omega = \tau D_A D_B^{-1}$, where $\tau = 0.8, 1.0, 1.2$. All initial iteration vectors are chosen to be $x^{(0)} = e_n$, the tolerance is set at 10^{-10} and the maximum iteration step is 30,000.

In order to compare with the theoretical result, the upper bounds of α in (16) are presented in Table 2 for Example 1 and Example 2, while the ones of Example 3 are not presented because the assumption of Theorem 1 is not satisfied. Note that $\Omega \geq D_A D_B^{-1}$ is one assumption of Theorem 2. Thus, we only show the upper bounds of α for the cases of $\tau = 1$ and $\tau = 1.2$. It is remarked that, although the upper bounds of Example 1 with $\mu = \nu = 0$ listed in Table 2 are all equal to 1 with 2 decimal digit accuracy, the exact bounds are all larger than 1, which is also guaranteed by the proof of Lemma 4.

Numerical results are reported in Tables 3, 4, 5, and 6, where "IT" and "CPU" denote the number of iteration steps and the elapsed CPU time in seconds, respectively.

For Example 1 and Example 2, it is shown by Tables 3, 4, and 5 that all methods are convergent. In each comparison, the number of iteration steps of the MAOR is nearly twice as long as that of the TMAOR. This fact is due to the two linear systems solved in each iteration of the TMAOR. For the efficiency, we notice that the TMAOR always converges faster than the MAOR. On the other hand, the TMAOR converges faster than the AMAOR in most cases, except the TMAOR_1 with $\tau = 0.8$, $\mu = \nu = 0$ for Example 1 and the TMAOR_2 with $\tau = 0.8, 1.2$ for Example 2. Pass to the relaxation parameters (α, β). By the results in Table 2, (16) can be satisfied with the choices of (α, β) in Table 1 both for the two examples when $\mu = \nu = 4$. On the other hand, for Example 1 with $\mu = \nu = 0$, (16) cannot be satisfied when $\alpha > 1$. Meanwhile, when $\tau = 0.8$, the assumption $\Omega \geq D_A D_B^{-1}$ in Theorem 2 does not hold. However, the TMAOR still converges and also performs better than the MAOR in these cases.

For Example 3, focus on Table 6, where the iteration steps of some cases of the MJ and the AMJ exceed 30,000, denoted by "–" with their CPU time. One can see that the MJ and the MGS perform better than the AMJ and the AMGS, respectively. Based on the first step iteration of Jacobi splittings, by introducing another pair of matrix splittings as the second step iteration, all the five methods (the TMJ, the TMJSOR_1, the TMJSOR_2, the TMJSOR_3, and the TMJSOR_4) converge faster than the MJ. On the other hand, for the cases of the first step iteration being Gauss-Seidel splittings, the TMGSJ, the TMGSSOR_1 and the TMGS take more CPU time than the MGS, while as the relaxation parameters in the matrix splittings of the second step iteration being larger than 1, the TMGSSOR_2 and the TMGSSOR_3 converge faster than the MGS. Thus, the numerical results in Table 6 show that Method 2 can also accelerate Method 1 by suitable choices of the two-step matrix splittings.

Table 4 Numerical results of Example 1 when $\mu = \nu = 0$ and $\Omega = \tau D_A D_B^{-1}$

		$m = 128$		$m = 256$		$m = 512$			$m = 128$		$m = 256$		$m = 512$	
τ	Method	IT	CPU	IT	CPU	IT	CPU	Method	IT	CPU	IT	CPU	IT	CPU
1.2	$MSOR_1$	66	0.1755	67	0.6126	69	3.8274	$MAOR_1$	66	0.1620	68	0.6538	69	3.9289
	$AMSOR_1$	55	0.1463	56	0.5105	58	3.1895	$AMAOR_1$	53	0.1387	55	0.5410	55	3.4998
	$TMSOR_1$	36	0.1309	37	0.4723	37	3.0455	$TMAOR_1$	40	0.1302	41	0.5226	42	3.3372
	$MSOR_2$	92	0.2145	94	0.8393	96	5.1550	$MAOR_2$	86	0.2215	89	0.8447	91	4.9408
	$AMSOR_2$	81	0.1889	83	0.7389	85	4.5386	$AMAOR_2$	76	0.1957	79	0.7465	80	4.3663
	$TMSOR_2$	46	0.1548	47	0.7068	48	3.7819	$TMAOR_2$	44	0.1703	45	0.6658	46	3.6404
	$MSOR_3$	131	0.3106	134	1.2030	137	7.5609	$MAOR_3$	120	0.2783	123	1.1145	126	6.6717
	$AMSOR_3$	121	0.2869	124	1.1112	127	6.9837	$AMAOR_3$	110	0.2551	113	1.0216	116	6.1157
	$TMSOR_3$	66	0.2249	67	0.9804	69	6.2123	$TMAOR_3$	60	0.2089	62	0.9005	63	5.0036
1.0	$MSOR_1$	58	0.1474	60	0.5557	61	3.2205	$MAOR_1$	85	0.2148	86	0.8296	88	4.7458
	$AMSOR_1$	49	0.1245	51	0.4695	52	2.7208	$AMAOR_1$	54	0.1661	55	0.7019	56	3.3912
	$TMSOR_1$	30	0.1064	30	0.4308	31	2.3998	$TMAOR_1$	40	0.1554	40	0.6927	41	3.3217
	$MSOR_2$	83	0.1998	85	0.7582	87	4.6139	$MAOR_2$	79	0.2044	81	0.7805	83	4.5723
	$AMSOR_2$	74	0.1781	76	0.6760	78	4.1136	$AMAOR_2$	70	0.1811	72	0.6916	74	4.0514
	$TMSOR_2$	42	0.1461	43	0.6215	44	3.4234	$TMAOR_2$	40	0.1493	41	0.6346	42	3.4508
	$MSOR_3$	120	0.2656	123	1.0204	126	6.4994	$MAOR_3$	111	0.2655	114	1.0567	116	6.2495
	$AMSOR_3$	111	0.2457	114	0.9439	117	6.0119	$AMAOR_3$	102	0.2440	105	0.9710	107	5.7428
	$TMSOR_3$	60	0.1949	62	0.8369	63	4.8235	$TMAOR_3$	56	0.2037	57	0.8632	58	4.6161
0.8	$MSOR_1$	72	0.1803	73	0.6929	74	4.0907	$MAOR_1$	202	0.4563	205	1.8670	208	11.2323
	$AMSOR_1$	44	0.1419	45	0.6009	45	3.1281	$AMAOR_1$	68	0.1536	69	0.6285	70	3.7812
	$TMSOR_1$	35	0.1315	35	0.5848	36	2.9025	$TMAOR_1$	91	0.3086	93	1.5490	94	9.1355
	$MSOR_2$	75	0.1885	77	0.7112	79	4.2835	$MAOR_2$	71	0.1883	73	0.7096	75	4.2177

Table 4 (continued)

		$m = 128$		$m = 256$		$m = 512$			$m = 128$		$m = 256$		$m = 512$	
τ	Method	IT	CPU	IT	CPU	IT	CPU	Method	IT	CPU	IT	CPU	IT	CPU
	$AMSOR_2$	68	0.1709	70	0.6448	72	3.8837	$AMAOR_2$	65	0.1724	67	0.6496	69	3.8613
	$TMSOR_2$	38	0.1365	39	0.5587	40	3.1106	$TMAOR_2$	36	0.1342	37	0.5678	38	3.0905
	$MSOR_3$	111	0.2444	114	0.9702	116	6.0574	$MAOR_3$	103	0.2448	105	0.9816	108	5.8318
	$AMSOR_3$	104	0.2290	107	0.9090	109	5.6754	$AMAOR_3$	97	0.2305	99	0.9244	102	5.4921
	$TMSOR_3$	56	0.1826	57	0.8169	58	4.4648	$TMAOR_3$	52	0.1873	53	0.8321	54	4.2993

Table 5 Numerical results of Example 2 when $\mu = \nu = 4$ and $\Omega = \tau D_A D_B^{-1}$

		$m = 128$		$m = 256$		$m = 512$			$m = 128$		$m = 256$		$m = 512$	
τ	Method	IT	CPU	IT	CPU	IT	CPU	Method	IT	CPU	IT	CPU	IT	CPU
1.2	$MSOR_1$	37	0.1073	38	0.3593	39	2.1643	$MAOR_1$	51	0.1391	53	0.5175	54	3.0308
	$AMSOR_1$	26	0.0754	27	0.2891	27	1.5209	$AMAOR_1$	33	0.1129	34	0.3519	35	2.0216
	$TMSOR_1$	17	0.0684	17	0.2741	17	1.3999	$TMAOR_1$	24	0.1009	24	0.3436	25	2.0172
	$MSOR_2$	24	0.0748	25	0.2237	26	1.4246	$MAOR_2$	25	0.0793	26	0.2499	26	1.4596
	$AMSOR_2$	20	0.0623	21	0.1899	22	1.1872	$AMAOR_2$	18	0.0571	19	0.1799	19	1.0509
	$TMSOR_2$	13	0.0609	14	0.1861	14	1.1007	$TMAOR_2$	13	0.0633	13	0.2024	13	1.0721
	$MSOR_3$	38	0.1147	39	0.3989	40	2.2551	$MAOR_3$	33	0.0944	34	0.3538	35	2.0283
	$AMSOR_3$	35	0.1056	36	0.3674	37	2.0771	$AMAOR_3$	30	0.0858	31	0.3216	32	1.8439
	$TMSOR_3$	21	0.0838	22	0.3412	22	1.7825	$TMAOR_3$	18	0.0805	19	0.2990	19	1.6037
1.0	$MSOR_1$	29	0.0871	29	0.2703	30	1.6173	$MAOR_1$	38	0.1081	39	0.3923	40	2.2938
	$AMSOR_1$	20	0.0701	20	0.2372	21	1.1311	$AMAOR_1$	27	0.0844	28	0.3398	28	1.7567
	$TMSOR_1$	14	0.0661	14	0.2216	14	1.1123	$TMAOR_1$	20	0.0830	21	0.3346	21	1.7357
	$MSOR_2$	20	0.0432	20	0.1841	21	1.1557	$MAOR_2$	19	0.0459	19	0.2129	19	1.1082
	$AMSOR_2$	16	0.0381	16	0.1611	17	0.9810	$AMAOR_2$	17	0.0411	17	0.1905	17	1.0215
	$TMSOR_2$	12	0.0368	12	0.1571	12	0.9694	$TMAOR_2$	12	0.0360	12	0.1847	12	1.0076
	$MSOR_3$	34	0.1023	34	0.3552	35	2.0481	$MAOR_3$	29	0.0606	30	0.2729	30	1.6029
	$AMSOR_3$	31	0.0933	31	0.3239	32	1.8674	$AMAOR_3$	26	0.0543	27	0.2447	27	1.4371
	$TMSOR_3$	19	0.0816	19	0.3074	20	1.6434	$TMAOR_3$	16	0.0569	17	0.2325	17	1.4003
0.8	$MSOR_1$	33	0.0757	34	0.3499	34	1.9109	$MAOR_1$	49	0.1131	50	0.4481	50	2.6827
	$AMSOR_1$	25	0.0689	26	0.2909	26	1.6990	$AMAOR_1$	36	0.0965	37	0.4112	37	2.5580
	$TMSOR_1$	20	0.0677	20	0.2858	20	1.6254	$TMAOR_1$	31	0.0931	31	0.4098	32	2.5387

Table 5 (continued)

		$m = 128$		$m = 256$		$m = 512$			$m = 128$		$m = 256$		$m = 512$	
τ	Method	IT	CPU	IT	CPU	IT	CPU	Method	IT	CPU	IT	CPU	IT	CPU
	$MSOR_2$	17	0.0461	18	0.2666	18	1.0352	$MAOR_2$	25	0.0562	26	0.2538	26	1.4165
	$AMSOR_2$	16	0.0434	17	0.2509	17	0.9743	$AMAOR_2$	21	0.0472	22	0.2132	22	1.1899
	$TMSOR_2$	10	0.0388	11	0.2224	11	0.8869	$TMAOR_2$	15	0.0473	15	0.2305	16	1.3328
	$MSOR_3$	38	0.1119	39	0.4032	40	2.2686	$MAOR_3$	26	0.0780	27	0.2714	27	1.5185
	$AMSOR_3$	30	0.0883	31	0.3636	32	1.8291	$AMAOR_3$	25	0.0750	26	0.2610	26	1.4601
	$TMSOR_3$	21	0.0848	22	0.3521	22	1.7987	$TMAOR_3$	15	0.0676	15	0.2337	16	1.3721

Table 6 Numerical results of Example 3

	$n = 100$		$n = 200$		$n = 300$	
Method	IT	CPU	IT	CPU	IT	CPU
MJ	6905	0.2522	25904	1.3130	–	–
AMJ	–	–	–	–	–	–
TMJ	3453	0.1694	12952	0.9408	28482	2.1462
TMJSOR_1	2575	0.1295	9228	0.5945	19628	1.5372
TMJSOR_2	2384	0.1239	8545	0.5433	18178	1.3622
TMJSOR_3	2206	0.1133	7917	0.5116	16846	1.3299
TMJSOR_4	2035	0.1062	7327	0.4914	15605	1.3059
MGS	1344	0.0765	4907	0.2558	10495	0.6254
AMGS	2782	0.1203	9967	0.5221	21202	1.3199
TMGSJ	1095	0.0815	3950	0.2707	8425	0.6677
TMGSSOR_1	1065	0.0802	3831	0.2665	8166	0.7578
TMGS	1033	0.0780	3710	0.2612	7906	0.6394
TMGSSOR_2	1001	0.0753	3589	0.2490	7645	0.6186
TMGSSOR_3	968	0.0719	3467	0.2457	7385	0.6092

5 Conclusions

By employing two-step matrix splittings, we have constructed the two-step MMS method for the HLCP. The proposed method extends the MMS method in [23]. We also give the convergence theorems, which include and generalize some existing results. Specially, the convergence analysis of the TMAOR method is given, which enriches the theory of the MMS method. Finally, the effectiveness of the proposed method is shown by numerical examples. Note that by the numerical results in the previous section, in Theorem 1, the assumption (II) may be weakened and the convergence range of α may be enlarged. How to improve the convergence theorems is worth studying in the future.

Acknowledgments The authors would like to thank the reviewers for their helpful suggestions.

Funding information This work was supported by the Major Projects of Guangdong Education Department for Foundation Research and Applied Research (No. 2018KZDXM065); University of Macau with (No. MYRG2018-00047-FST); the Science and Technology Development Fund, Macau SAR (No. 0005/2019/A); Guangdong provincial Natural Science Foundation (Grant No. 2018A0303100015); and the Young Innovative Talents Project from Guangdong Provincial Department of Education (Nos. 2018KQNCX230, 2018KQNCX233).

References

1. Alanelli, M., Hadjidimos, A.: A new iterative criterion for H-matrices. SIAM J. Matrix Anal. Appl. **29**, 160–176 (2006)

2. Bai, Z.-Z.: On the convergence of the multisplitting methods for the linear complementarity problem. SIAM J. Matrix Anal. Appl. **21**, 67–78 (1999)
3. Bai, Z.-Z.: Modulus-based matrix splitting iteration methods for linear complementarity problems. Numer. Linear Algebra Appl. **17**, 917–933 (2010)
4. Bai, Z.-Z., Zhang, L.-L.: Modulus-based synchronous multisplitting iteration methods for linear complementarity problems. Numer. Linear Algebra Appl. **20**, 425–439 (2013)
5. Bai, Z.-Z., Zhang, L.-L.: Modulus-based synchronous two-stage multisplitting iteration methods for linear complementarity problems. Numer. Algorithm. **62**, 59–77 (2013)
6. Berman, A., Plemmons, R.J.: Nonnegative Matrix in the Mathematical Sciences. SIAM Publisher, Philadelphia (1994)
7. Bru, R., Giménez, I., Hadjidimos, A.: Is $a \in c^{n\times n}$ a general H-matrix? Linear Algebra Appl. **436**, 364–380 (2012)
8. Cottle, R.W., Dantzig, G.B.: A generalization of the linear complementarity problem. J. Comb. Theory **8**, 79–90 (1970)
9. Cottle, R.W., Pang, J.S., Stone, R.E.: The Linear Complementarity Problem. Academic, San Diego (1992)
10. Fang, X.-M., Zhu, Z.-w.: The modulus-based matrix double splitting iteration method for linear complementarity problems. Comput. Math. Appl. **78**(11), 3633–3643 (2019)
11. Frommer, A., Mayer, G.: Convergence of relaxed parallel multisplitting methods. Linear Algebra Appl. **119**, 141–152 (1989)
12. Fujisawa, T., Kuh, E.S.: Piecewise-linear theory of nonlinear networks. SIAM J. Appl. Math. **22**, 307–328 (1972)
13. Fujisawa, T., Kuh, E.S., Ohtsuki, T.: A sparse matrix method for analysis of piecewise-linear resistive networks. IEEE Trans. Circ. Theory **19**, 571–584 (1972)
14. Gao, X., Wang, J.: Analysis and application of a one-layer neural network for solving horizontal linear complementarity problems. Int. J. Comput. Int. Sys. **7**(4), 724–732 (2014)
15. Giacopini, M., Fowell, M., Dini, D., Strozzi, A.: A mass-conserving complementarity formulation to study lubricant films in the presence of cavitation. J. Tribol. **132**, 041702 (2010)
16. Hu, J.-G.: Estimates of $||b^{-1}c||_\infty$ and their applications. Math. Numer. Sin. **4**, 272–282 (1982)
17. Huang, N., Ma, C.-F.: The modulus-based matrix splitting algorithms for a class of weakly nonlinear complementarity problems. Numer. Linear Algebra Appl. **23**, 558–569 (2016)
18. Li, W.: A general modulus-based matrix splitting method for linear complementarity problems of H-matrices. Appl. Math. Lett. **26**, 1159–1164 (2013)
19. Li, W., Zheng, H.: A preconditioned modulus-based iteration method for solving linear complementarity problems of H-matrices. Linear Multilinear Algebra **64**, 1390–1403 (2016)
20. Liu, S.-M., Zheng, H., Li, W.: A general accelerated modulus-based matrix splitting iteration method for solving linear complementarity problems. Calcolo **53**, 189–199 (2016)
21. Mezzadri, F., Galligani, E.: An inexact Newton method for solving complementarity problems in hydrodynamic lubrication. Calcolo **55**, 1 (2018)
22. Mezzadri, F., Galligani, E.: Splitting methods for a class of horizontal linear complementarity problems. J. Optim. Theory Appl. **180**(2), 500–517 (2019)
23. Mezzadri, F., Galligani, E.: Modulus-based matrix splitting methods for horizontal linear complementarity problems. Numer. Algorithm. **83**, 201–219 (2020)
24. Mezzadri, F., Galligani, E.: A modulus-based nonsmooth Newton's method for solving horizontal linear complementarity problems. Optimiz. Lett. Published online. https://doi.org/10.1007/s11590-019-01515-9
25. Mezzadri, F., Galligani, E.: On the convergence of modulus-based matrix splitting methods for horizontal linear complementarity problems in hydrodynamic lubrication. Math. Comput. Simulat. Published online. https://doi.org/10.1016/j.matcom.2020.01.014
26. Sun, M.: Monotonicity of Mangasarian's iterative algorithm for generalized linear complementarity problems. J. Math. Anal. Appl. **144**, 474–485 (1989)
27. Sun, M.: Singular control problems in bounded intervals. Stochastics **21**, 303–344 (1987)
28. Sznajder, R., Gowda, M.S.: Generalizations of p_0- and P-properties; extended vertical and horizontal linear complementarity problems. Linear Algebra Appl. **223**(/224), 695–715 (1995)
29. Xia, Z.-C., Li, C.-L.: Modulus-based matrix splitting iteration methods for a class of nonlinear complementarity problem. Appl. Math. Comput. **271**, 34–42 (2015)

30. Zhang, L.-L.: Two-step modulus based matrix splitting iteration for linear complementarity problems. Numer. Algorithm. **57**, 83–99 (2011)
31. Zhang, L.-L., Ren, Z.-R.: Improved convergence theorems of modulus-based matrix splitting iteration methods for linear complementarity problems. Appl. Math. Lett. **26**, 638–642 (2013)
32. Zheng, H.: Improved convergence theorems of modulus-based matrix splitting iteration method for nonlinear complementarity problems of H-matrices. Calcolo **54**, 1481–1490 (2017)
33. Zheng, H., Li, W.: The modulus-based nonsmooth Newton's method for solving linear complementarity problems. J. Comput. Appl. Math. **288**, 116–126 (2015)
34. Zheng, H., Li, W., Vong, S.: A relaxation modulus-based matrix splitting iteration method for solving linear complementarity problems. Numer. Algorithm. **74**, 137–152 (2017)
35. Zheng, H., Li, W., Vong, S.: An iteration method for nonlinear complementarity problems. J. Comput. Appl. Math. **372**, 112681 (2020)
36. Zheng, H., Liu, L.: A two-step modulus-based matrix splitting iteration method for solving nonlinear complementarity problems of h_+-matrices. Comput. Appl. Math. **37**(4), 5410–5423 (2018)
37. Zheng, H., Vong, S.: The modulus-based nonsmooth Newton's method for solving a class of nonlinear complementarity problems of P-matrices. Calcolo. **55**, 37 (2018)
38. Zheng, H., Vong, S., Liu, L.: A direct precondition modulus-based matrix splitting iteration method for solving linear complementarity problems. Appl. Math. Comput. **353**, 396–405 (2019)
39. Zheng, H., Vong, S.: Improved convergence theorems of the two-step modulus-based matrix splitting and synchronous multisplitting iteration methods for solving linear complementarity problems. Linear Multilinear Algebra. **67**(9), 1773–1784 (2019)
40. Zheng, H., Vong, S.: A modified modulus-based matrix splitting iteration method for solving implicit complementarity problems. Numer. Algorithm. **82**(2), 573–592 (2019)
41. Zheng, H., Vong, S.: On convergence of the modulus-based matrix splitting iteartion method for horizontal linear complementarity problem of h_+-matrices. Appl. Math. Comput. **369**, 124890 (2020)
42. Zheng, N., Yin, J.-F.: Accelerated modulus-based matrix splitting iteration methods for linear complementarity problems. Numer. Algorithm. **64**, 245–262 (2013)