Filomat 38:13 (2024), 4495–4509
https://doi.org/10.2298/FIL2413495H

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Two classes of modulus-based methods for solving linear complementarity problems

Maryam Bashirizadeh[a], Masoud Hajarian[a,*]

[a]*Department Applied Mathematics, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran.*

Abstract. This study focuses on developing efficient numerical methods to solve linear complementarity problems (LCP). There are many problems in various fields like engineering, economics, and science that lead to an LCP. Modulus-based methods are powerful computational tools for solving such problems. In this paper, the schemes for solving LCPs are based on modulus. The new methods utilize two initial guesses and update each of the initial guesses in separate steps. Convergence of new methods is expressed under special conditions when the system matrix is an H_+-matrix. Also, the presented numerical results confirm the efficiency of the new techniques compared to the modulus-based and two-step modulus-based methods.

1. Introduction

In the process of solving many scientific, engineering, economic, and computing problems, solving a linear complementarity problem is inevitable [3, 9, 16, 24, 36]. Therefore, it is very important to present and evaluate efficient numerical methods to solve these problems. The linear complementarity problem for the given matrix $A \in R^{n\times n}$ and vector $q \in R^n$ denoted by LCP(q, A) is to find a vector $z \in R^n$, so that the following conditions are met:

$$z \geq 0, \quad w := Az + q \geq 0, \quad z^T w = 0. \tag{1}$$

For solving LCP, methods are divided into two categories: direct methods and iterative methods. Iterative methods are well suited for solving LCP(q, A), especially when the system matrix is large and sparse. Iterative methods for solving LCPs were first proposed by Cryer [10] and many researchers followed his work [11, 18, 20, 27, 34, 41]. Then in 1980 Bokhon [35] presented the modulus splitting method. Based on this approach, modified modulus [13] and extrapolated modulus methods [19] were proposed. In 2010 Bai [2] introduced a general structure called modulus-based methods, these approaches cover the previous modulus methods. Modulus-based methods have become a powerful tool for solving LCPs. Today, we widely see the presentation of new methods based on this technique. For example, two-step, two-sweep, two-step two-sweep, general two-sweep, and preconditioner methods are of this category [7, 21, 26, 31, 32, 38, 42]. Also, multisplitting methods that were initially proposed to solve systems of linear equations have been

2020 *Mathematics Subject Classification*. Primary 90C33; Secondary 65F10.
Keywords. Linear complementarity problem, Modulus-based, H_+-matrix, LCP
Received: 28 October 2022; Revised: 07 November 2023; Accepted: 12 February 2024
Communicated by Predrag Stanimirović
* Corresponding author: Masoud Hajarian
Email addresses: m_bashirizadeh@sbu.ac.ir (Maryam Bashirizadeh), m_hajarian@sbu.ac.ir (Masoud Hajarian)

developed to solve LCPs [4, 6, 12, 44]. Convergence analysis of modulus-based methods is often discussed when the system matrix is an H_+-matrix or a positive definite matrix[23, 43, 45].
In addition to solve LCP, modulus-based methods have been used to solve different branches of complementarity problems, including nonlinear complementarity problems [21, 39, 45, 47], horizontal linear complementary problems [15, 30, 46], implicit complementarity [25], quasi-complementary problems [33], and second-order cone linear complementary problems [28]. Considering the importance of these techniques in solving linear complementarity problems, this paper focuses on modulus-based iterative methods for solving linear complementarity problems. So far, many valuable works have been done in this field [22, 29, 40]. The aim of this paper is to introduce new classes of efficient modulus-based methods that have a significant advantage in terms of CPU time and iteration steps (IT) compared to the modulus-based method presented by Bai [2] and the two-step modulus-based method introduced by Zhang [42]. New techniques have two initial guesses, each of which is updated separately. One initial vector is updated in the first step and another initial vector is updated in the second step. By defining the vector $X = \begin{bmatrix} v \\ v \end{bmatrix} \in \mathbb{R}^{2n}$ the convergence of the presented methods under certain conditions is proved in a different way from the previous methods. The presented numerical results confirm the efficiency of the new methods.
The subsequent sections of this paper include the following:
Section 2: Basic concepts, lemmas, and theorems.
Section 3: Description of new modulus-based methods.
Section 4: Discussion on convergence and relevant theorems.
Section 5: Presentation of numerical results.
Section 6: Concluding.

2. Preliminaries

This section is organized in a way to express the fundamental concepts, lemmas, and theorems that are utilized throughout the article. A matrix $A = (a_{ij}) \in R^{n\times n}$ is called [1, 8, 37]

- A nonnegative matrix if for all i and j, $a_{ij} \geq 0$.
- A Z-matrix if $a_{ij} \leq 0$ for all $i \neq j$.
- An $M-$matrix if A is a Z-matrix and nonsingular such that $A^{-1} \geq 0$.
- An H-matrix if its comparison matrix $\langle A\rangle$ (if $i = j$ then $a_{ij} = |a_{ij}|$, if $i \neq j$ then $a_{ij} = -|a_{ij}|$) is an M-matrix.
- An H_+-matrix if A is an H-matrix with positive diagonal entries.

For any two matrices $A, H \in R^{n\times n}$ if $A \geq H (A > H)$ then $A - H \geq 0$ $(A - H > 0)$. Also, we have $|AH| \leq |A||H|$. We denote the spectral radius and the absolute value of A by $\rho(A)$ and $|A| = (|a_{ij}|)$, respectively.
Let $A = F - G$ be the splitting of matrix $A \in R^{n\times n}$ if [8, 14, 43]

- $\langle F\rangle - |G|$ is an M-matrix, then the splitting is an H-splitting.
- F is a nonsingular M-matrix and $G \geq 0$, then the splitting is an M-splitting.
- $\langle A\rangle = \langle F\rangle - |G|$, then the splitting is an H-compatible splitting.

Lemma 2.1. *[8, 28] For any nonnegative matrix K and any vectors $u, v \in R^n$ such that $u \leqslant v$, the inequality $Ku \leqslant Kv$ holds.*

Lemma 2.2. *[17] Assume that $A = D - B$ is an H-matrix. Then the following statements hold*

- *A is nonsingular and $|A^{-1}| \leq \langle A\rangle^{-1}$.*
- *$|D|$ is nonsingular and $\rho(|D|^{-1}|B|) < 1$.*

Lemma 2.3. *[8] Let A be a Z-matrix. Then the following statements are equivalent*

- *A is an M-matrix.*
- *There exists a positive vector x, such that $Ax > 0$.*

Lemma 2.4. *[5] Let A be an H_+-matrix. Then the LCP(q, A) has a unique solution z^*.*

Lemma 2.5. *[23] For a nonnegative matrix $A \in R^{n\times n}$, if there exists a positive vector $x \in R^n$ such that $Ax < x$, then $\rho(A) < 1$.*

Lemma 2.6. *[2] Let $A = F - G$ be a splitting of the matrix $A \in R^{n\times n}, \Lambda$ be a positive diagonal matrix and γ be a positive constant. For the LCP(q, A), the following statements hold true:*

- *If (z, ω) is a solution of the LCP(q, A), then $x = \frac{1}{2}\gamma(z - \Lambda^{-1}\omega)$, with $|x| = \frac{1}{2}\gamma(z + \Lambda^{-1}\omega)$, satisfies the implicit fixed-point equation*

$$(\Lambda + F)x = Gx + (\Lambda - A)|x| - \gamma q. \tag{2}$$

- *If x satisfies the implicit fixed-point equation* (2)*, then*

$$z = \gamma^{-1}(|x| + x), \quad \omega = \gamma^{-1}\Lambda(|x| - x),$$

is a solution of the LCP(q, A).

3. Modulus-based methods for solving LCP(q, A)

This section introduces two new classes of modulus-based methods for solving LCP(q, A). Modulus-based technique was first presented by Bai [2] as follows. For any initial guess $x^{(0)}$, and $k = 0, 1, ...$ until the iteration sequence $\left\{z^{(k)}\right\}$ converges, the computation involves solving:

$$(\Lambda + F)x^{(k+1)} = Gx^{(k)} + (\Lambda - A)|x^{(k)}| - \gamma q,$$

and subsequently setting

$$z^{(k+1)} = (\frac{1}{\gamma})(|x^{(k+1)}| + x^{(k+1)}),$$

where $A = F - G$ is a splitting of matrix $A \in R^{n\times n}$, Λ be a positive diagonal matrix and γ be a positive constant, as stated in Lemma 2.6.
We propose two classes of modulus-based iteration methods for solving LCP(q, A), by using two splittings $A = F_1 - G_1 = F_2 - G_2$ of matrix $A \in R^{n\times n}$.

Method 3.1. *Modulus-based matrix splitting iteration method I for LCP(q, A).*
For any given two initial guesses $x^{(0)}$, $y^{(0)} \in R^n$, and k=0,1,2,... until the iteration sequence $\left\{z^{(k)}\right\}$ is convergent, compute

$$\begin{cases} (\Lambda + F_1)y^{(k+1)} = G_1 y^{(k)} + (\Lambda - A)|x^{(k)}| - \gamma q, \\ (\Lambda + F_2)x^{(k+1)} = G_2 y^{(k+1)} + (\Lambda - A)|x^{(k)}| - \gamma q, \end{cases} \tag{3}$$

and set

$$z^{(k+1)} = (\frac{1}{\gamma})(|x^{(k+1)}| + x^{(k+1)}).$$

Method 3.2. *Modulus-based matrix splitting iteration method II for LCP(q, A).*
For any given two initial guesses $x^{(0)},\ y^{(0)} \in R^n$ and k=0,1,2,... until the iteration sequence $\left\{z^{(k)}\right\}$ is convergent, compute

$$\begin{cases} (\Lambda + F_1)y^{(k+1)} = G_1 y^{(k)} + (\Lambda - A)|x^{(k)}| - \gamma q, \\ (\Lambda + F_2)x^{(k+1)} = G_2 y^{(k+1)} + (\Lambda - A)|y^{(k+1)}| - \gamma q, \end{cases} \tag{4}$$

and set

$$z^{(k+1)} = (\frac{1}{\gamma})(|x^{(k+1)}| + x^{(k+1)}).$$

Remark 3.3. *With different choices for the splittings, various methods can be obtained from the presented methods. In fact, the proposed methods provide general structures. For example, if we put:*

$$\begin{cases} F_1 = \frac{1}{\alpha}(D - \beta L), \quad G_1 = \frac{1}{\alpha}((1-\alpha)D + (\alpha - \beta)L + \alpha U), \\ F_2 = \frac{1}{\alpha}(D - \beta U), \quad G_2 = \frac{1}{\alpha}((1-\alpha)D + (\alpha - \beta)U + \alpha L), \end{cases} \tag{5}$$

then we obtain accelerated over-relaxation modulus-based method I and II (MAOR I and MAOR II). In addition, whenever $(\alpha, \beta) = (\alpha, \alpha)$, $(\alpha, \beta) = (1, 1)$, and $(\alpha, \beta) = (1, 0)$, it will be obtained successive over-relaxation modulus-based method I and II (MSOR I and MSOR II), Gauss-Seidel modulus-based method I and II (MGS I and MGS II), and Jacobi modulus-based method I and II (MJ I and MJ II), respectively.

4. Convergence results

In this section, we discuss the convergence of the proposed methods when the system matrix is an H_+-matrix.
Also, we consider Λ be a positive diagonal matrix, γ be a positive constant,

$$D = diag(A), \quad e_y^{(k)} = y^{(k)} - y^*, \quad e_x^{(k)} = x^{(k)} - x^*.$$

Theorem 4.1. *Let $A \in R^{n\times n}$ be an H_+-matrix, $A = F_1 - G_1 = F_2 - G_2$ be two H-compatible splittings of A (i.e., $\langle A\rangle = \langle F_1\rangle - |G_1| = \langle F_2\rangle - |G_2|$). For any two initial guesses $x^{(0)},\ y^{(0)} \in R^n$, Method 3.1 is convergent if any of the following conditions are met:*

- *$\Lambda \geq D$.*
- *A positive vector $v \in R^n$ exists such that $|B|v < \Lambda v < Dv$.*

Proof. Let (x^*, y^*) be the solution of (3) and satisfies the implicit fixed point equations

$$\begin{cases} y^* = (\Lambda + F_1)^{-1} G_1 y^* + (\Lambda + F_1)^{-1}(\Lambda - A)|x^*| - \gamma q, \\ x^* = (\Lambda + F_2)^{-1} G_2 y^* + (\Lambda + F_2)^{-1}(\Lambda - A)|x^*| - \gamma q. \end{cases} \tag{6}$$

As $(\Lambda + F_i)$, $(i = 1, 2)$ are H_+-matrices, by Lemma 2.2 and Equations (3), (6), we obtain

$$\begin{cases} |e_y^{(k+1)}| \leqslant (\Lambda + \langle F_1\rangle)^{-1}|G_1||e_y^{(k)}| + (\Lambda + \langle F_1\rangle)^{-1}|\Lambda - A||e_x^{(k)}|, \\ |e_x^{(k+1)}| \leqslant (\Lambda + \langle F_2\rangle)^{-1}|G_2|(\Lambda + \langle F_1\rangle)^{-1}|G_1||e_y^{(k)}| + \\ \qquad (\Lambda + \langle F_2\rangle)^{-1}(|\Lambda - A| + |G_2|(\Lambda + \langle F_1\rangle)^{-1}|\Lambda - A|)|e_x^{(k)}|, \end{cases} \tag{7}$$

or

$$\left|\begin{bmatrix} e_y^{(k+1)} \\ e_x^{(k+1)} \end{bmatrix}\right| \leq E \left|\begin{bmatrix} e_y^{(k)} \\ e_x^{(k)} \end{bmatrix}\right|, \tag{8}$$

where

$$E = \begin{bmatrix} (\Lambda + \langle F_1 \rangle)^{-1}|G_1| & (\Lambda + \langle F_1 \rangle)^{-1}|\Lambda - A| \\ (\Lambda + \langle F_2 \rangle)^{-1}|G_2|(\Lambda + \langle F_1 \rangle)^{-1}|G_1| & (\Lambda + \langle F_2 \rangle)^{-1}(|\Lambda - A| + |G_2|(\Lambda + \langle F_1 \rangle)^{-1}|\Lambda - A|) \end{bmatrix} \in R^{2n \times 2n}. \tag{9}$$

To demonstrate the convergence of Method 3.1, it suffices to show that $\rho(E) < 1$. Now, let's consider the following two cases.

1. When $\Lambda \geq D$.
   As $\langle A \rangle$ is an M-matrix, by Lemma 2.3 there exists a positive vector $v \in R^n$ such that $\langle A \rangle v > 0$.
   Let $X = \begin{bmatrix} v \\ v \end{bmatrix} \in R^{2n}$, as E is a nonnegative matrix and X is a positive vector, based on Lemma 2.5, if $EX < X$, then $\rho(E) < 1$. By direct operation, we obtain

$$\begin{aligned} EX &= \begin{bmatrix} (\Lambda + \langle F_1 \rangle)^{-1}(|G_1| + \Lambda - \langle A \rangle)v \\ (\Lambda + \langle F_2 \rangle)^{-1}[|G_2|(\Lambda + \langle F_1 \rangle)^{-1}(|G_1| + \Lambda - \langle A \rangle) + \Lambda - \langle A \rangle]v \end{bmatrix} \\ &= \begin{bmatrix} (I - 2(\Lambda + \langle F_1 \rangle)^{-1}\langle A \rangle)v \\ (\Lambda + \langle F_2 \rangle)^{-1}[|G_2|(I - 2(\Lambda + \langle F_1 \rangle)^{-1}\langle A \rangle) + \Lambda - \langle A \rangle]v \end{bmatrix}. \end{aligned} \tag{10}$$

   Also, we have

$$(I - 2(\Lambda + \langle F_1 \rangle)^{-1}\langle A \rangle)v < v, \tag{11}$$

   by using Lemma 2.1

$$\begin{aligned} &(\Lambda + \langle F_2 \rangle)^{-1}[|G_2|(I - 2(\Lambda + \langle F_1 \rangle)^{-1}\langle A \rangle) + \Lambda - \langle A \rangle]v \\ &\leqslant (\Lambda + \langle F_2 \rangle)^{-1}[|G_2|v + \Lambda v - \langle A \rangle v] \\ &= (I - 2(\Lambda + \langle F_2 \rangle)^{-1}\langle A \rangle)v < v. \end{aligned} \tag{12}$$

   Thus $EX < \begin{bmatrix} v \\ v \end{bmatrix} = X$. Lemma 2.5 implies that $\rho(E) < 1$.
2. When there is a positive vector $v \in R^n$ such that $|B|v < \Lambda v < Dv$.
   Let $X = \begin{bmatrix} v \\ v \end{bmatrix}$ then we have

$$\begin{aligned} EX &= \begin{bmatrix} (\Lambda + \langle F_1 \rangle)^{-1}(|G_1| + |\Lambda - A|)v \\ (\Lambda + \langle F_2 \rangle)^{-1}[|G_2|(\Lambda + \langle F_1 \rangle)^{-1}(|G_1| + |\Lambda - A|) + |\Lambda - A|]v \end{bmatrix} \\ &= \begin{bmatrix} (I - 2(\Lambda + \langle F_1 \rangle)^{-1}(\Lambda - |B|))v \\ (\Lambda + \langle F_2 \rangle)^{-1}[|G_2|(I - 2(\Lambda + \langle F_1 \rangle)^{-1}(\Lambda - |B|)) + |\Lambda - A|]v \end{bmatrix}. \end{aligned} \tag{13}$$

Similarly, as

$$(I - 2(\Lambda + \langle F_1 \rangle)^{-1}(\Lambda - |B|))v < v, \tag{14}$$

and

$$\begin{aligned} &(\Lambda + \langle F_2 \rangle)^{-1}[|G_2|(I - 2(\Lambda + \langle F_1 \rangle)^{-1}(\Lambda - |B|)) + |\Lambda - A|]v \\ &\leqslant (\Lambda + \langle F_2 \rangle)^{-1}[|G_2|v + |\Lambda - A|v] \\ &= (I - 2(\Lambda + \langle F_2 \rangle)^{-1}(\Lambda - |B|))v < v. \end{aligned} \tag{15}$$

Hence $EX < X$, i.e., $\rho(E) < 1$. The proof is completed. □

Theorem 4.2. *Suppose that A is an H_+-matrix, $A = F_1 - G_1 = F_2 - G_2$ are two H-compatible splittings of A, and Λ is a positive diagonal matrix. For any initial two guesses $x^{(0)}$, $y^{(0)}$, Method 3.2 is convergent if any of the following conditions are met:*

- $\Lambda \geq D$.
- *A positive vector $v \in R^n$ exists such that $|B|v < \Lambda v < Dv$.*

Proof. Similar to the proof of Theorem 4.1, let (x^*, y^*) be the solution of (4) and satisfy the implicit fixed point equations

$$\begin{cases} y^* = (\Lambda + F_1)^{-1}G_1y^* + (\Lambda + F_1)^{-1}(\Lambda - A)|x^*| - \gamma q, \\ x^* = (\Lambda + F_2)^{-1}G_2y^* + (\Lambda + F_2)^{-1}(\Lambda - A)|y^*| - \gamma q. \end{cases} \tag{16}$$

From (4) and (16)

$$\begin{cases} |e_y^{(k+1)}| \leqslant (\Lambda + \langle F_1\rangle)^{-1}|G_1||e_y^{(k)}| + (\Lambda + \langle F_1\rangle)^{-1}|\Lambda - A||e_x^{(k)}|, \\ |e_x^{(k+1)}| \leqslant (\Lambda + \langle F_2\rangle)^{-1}(|G_2| + |\Lambda - A|)(\Lambda + \langle F_1\rangle)^{-1}|G_1||e_y^{(k)}| + \\ \qquad (\Lambda + \langle F_2\rangle)^{-1}(|G_2| + |\Lambda - A|)(\Lambda + \langle F_1\rangle)^{-1}|\Lambda - A||e_x^{(k)}|, \end{cases} \tag{17}$$

or

$$\left|\begin{bmatrix} e_y^{(k+1)} \\ e_x^{(k+1)} \end{bmatrix}\right| \leq \hat{E} \left|\begin{bmatrix} e_y^{(k)} \\ e_x^{(k)} \end{bmatrix}\right|, \tag{18}$$

where

$$\hat{E} = \begin{bmatrix} (\Lambda + \langle F_1\rangle)^{-1}|G_1| & (\Lambda + \langle F_1\rangle)^{-1}|\Lambda - A| \\ (\Lambda + \langle F_2\rangle)^{-1}(|G_2| + |\Lambda - A|)(\Lambda + \langle F_1\rangle)^{-1}|G_1| & (\Lambda + \langle F_2\rangle)^{-1}(|G_2| + |\Lambda - A|)(\Lambda + \langle F_1\rangle)^{-1}|\Lambda - A| \end{bmatrix}. \tag{19}$$

We only need to verify the validity of $\rho(\hat{E}) < 1$. Obviously, $\hat{E} \geq 0$. Now, let's consider the following two cases.

1. If $\Lambda \geq D$.
   Similar to the proof of Theorem 4.1 there exists a positive vector v, such that $\langle A\rangle v > 0$. Assume that $X = \begin{bmatrix} v \\ v \end{bmatrix}$, by straightforward calculations

$$\begin{aligned} \hat{E}X &= \begin{bmatrix} (\Lambda + \langle F_1\rangle)^{-1}(|G_1| + |\Lambda - A|)v \\ (\Lambda + \langle F_2\rangle)^{-1}(|G_2| + |\Lambda - A|)(\Lambda + \langle F_1\rangle)^{-1}(|G_1| + |\Lambda - A|)v \end{bmatrix} \\ &= \begin{bmatrix} (I - 2(\Lambda + \langle F_1\rangle)^{-1}\langle A\rangle)v \\ (I - 2(\Lambda + \langle F_2\rangle)^{-1}\langle A\rangle)(I - 2(\Lambda + \langle F_1\rangle)^{-1}\langle A\rangle)v \end{bmatrix} < \begin{bmatrix} v \\ v \end{bmatrix} = X. \end{aligned} \tag{20}$$

   Then $\rho(\hat{E} < 1)$.
2. If there exists a positive vector $v \in R^n$ such that $|B|v < \Lambda v < Dv$, then the following holds.

$$\begin{aligned} \hat{E}X &= \begin{bmatrix} (\Lambda + \langle F_1\rangle)^{-1}(|G_1| + |\Lambda - A|)v \\ (\Lambda + \langle F_2\rangle)^{-1}(|G_2| + |\Lambda - A|)(\Lambda + \langle F_1\rangle)^{-1}(|G_1| + |\Lambda - A|)v \end{bmatrix} \\ &= \begin{bmatrix} (I - 2(\Lambda + \langle F_1\rangle)^{-1}(\Lambda - |B|))v \\ (I - 2(\Lambda + \langle F_2\rangle)^{-1}(\Lambda - |B|))(I - 2(\Lambda + \langle F_1\rangle)^{-1}(\Lambda - |B|))v \end{bmatrix} < \begin{bmatrix} v \\ v \end{bmatrix} = X. \end{aligned} \tag{21}$$

Hence, $\rho(\hat{E}) < 1$. The proof is completed. □

Theorem 4.3. *Let $A = D - B$ be an H_+-matrix ($D = diag(A)$), $A = F_1 - G_1 = F_2 - G_2$ be two splittings of A, $\lambda = \rho(D^{-1}|B|)$, and Λ be a positive diagonal matrix that met $\Lambda \geq \frac{1}{2\alpha}D$. The MAOR I iteration method is convergent when one of the following conditions are met.*

1. $0 \leqslant \beta \leqslant \alpha,\ \frac{1}{2\alpha}D \leqslant \Lambda < \frac{1}{\alpha}D,\ \lambda < \frac{1}{2},\ \frac{1}{2(1-\lambda)} < \alpha < \frac{3}{2(1+\lambda)}$.
2. $0 \leqslant \beta \leqslant \alpha,\ \ \Lambda \geq \frac{1}{\alpha}D,\ \lambda < 1,\ 0 < \alpha < \frac{2}{1+\lambda}$.
3. $0 < \alpha \leqslant \beta,\ \Lambda \geq \frac{1}{\alpha}D,\ \ \lambda < \frac{1}{2\beta},\ 2\beta\lambda < \alpha < 2 - 2\beta\lambda$.
4. $0 < \alpha \leqslant \beta,\ \frac{1}{2\alpha}D \leqslant \Lambda < \frac{1}{\alpha}D,\ \lambda < \frac{1}{4\beta},\ \frac{4\beta\lambda+1}{2} < \alpha < \frac{3-4\beta\lambda}{2}$.

Proof. Let

$$\begin{cases} \tilde{F}_i = \Lambda + \langle F_i \rangle = \Lambda + \frac{1}{\alpha}(D - \beta|L_i|) \quad i = 1,2, \\ \tilde{G}_i = |G_i| + |\Lambda - A| \\ \quad \leqslant |\Lambda - \frac{1}{\alpha}D| + \dfrac{\beta}{\alpha}|L_i| + \dfrac{2}{\alpha}\left[|1-\alpha|D + |\alpha - \beta||L_i| + \alpha|U_i|\right], \end{cases} \tag{22}$$

where F_i and G_i are defined in (5). Denote

$$Q_i = (\Lambda + \langle F_i \rangle)^{-1}(|G_i| + |\Lambda - A|) = I - (\tilde{F}_i)^{-1}S_i, \qquad i = 1,2 \tag{23}$$

where

$$S_i = \Lambda + \frac{(1 - 2|1-\alpha|)D}{\alpha} - \frac{2|\alpha-\beta|}{\alpha}|L_i| - 2|U_i| - |\Lambda - \frac{1}{\alpha}D| - \frac{2\beta}{\alpha}|L_i|. \tag{24}$$

Obviously, $\tilde{F}_i$ $(i = 1,2)$ are M-matrices and based on the proof of Theorem 4.3 in [38], $S_i \geq K$ $(i = 1,2)$, where

- If $0 \leqslant \beta \leqslant \alpha,\ \frac{1}{2\alpha}D \leqslant \Lambda < \frac{1}{\alpha}D,\ \lambda < \frac{1}{2},\ \frac{1}{2(1-\lambda)} < \alpha < \frac{3}{2(1+\lambda)}$, then $K = \frac{1-2|1-\alpha|}{\alpha}D - 2|B|$, and K is an M-matrix.
- If $0 \leqslant \beta \leqslant \alpha,\ \ \Lambda \geq \frac{1}{\alpha}D,\ \lambda < 1,\ 0 < \alpha < \frac{2}{1+\lambda}$, then $K = \frac{2-2|1-\alpha|}{\alpha}D - 2|B|$, and K is an M-matrix.
- If $0 < \alpha \leqslant \beta,\ \Lambda \geq \frac{1}{\alpha}D,\ \ \lambda < \frac{1}{2\beta},\ 2\beta\lambda < \alpha < 2 - 2\beta\lambda$, then $K = \frac{2-2|1-\alpha|}{\alpha}D - \frac{4\beta}{\alpha}|B|$, and K is an M-matrix.
- If $0 < \alpha \leqslant \beta,\ \frac{1}{2\alpha}D \leqslant \Lambda < \frac{1}{\alpha}D,\ \lambda < \frac{1}{4\beta},\ \frac{4\beta\lambda+1}{2} < \alpha < \frac{3-4\beta\lambda}{2}$, then $K = \frac{1-2|1-\alpha|}{\alpha}D - \frac{4\beta}{\alpha}|B|$, and K is an M-matrix.

Since in each of the mentioned cases, the matrix K is an M-matrix, then for each of the conditions 1-4, by Lemma 2.3, there is a positive vector v such that $Kv > 0$. Thus, we have $Q_i v < v$ $(i = 1,2)$.
Now, we define

$$\begin{aligned} R &= (\Lambda + \langle F_2 \rangle)^{-1}[|G_2|(\Lambda + \langle F_1 \rangle)^{-1}(|G_1| + |\Lambda - A|) + |\Lambda - A|] \\ &= I - (\tilde{F_2})^{-1}[|G_2|Q_1 + |\Lambda - A|]. \end{aligned} \tag{25}$$

In each of conditions 1-4, we demonstrated that for matrix Q_1, there exists a positive vector v such that $Q_1 v < v$. Which implies,

$$\begin{aligned} Rv &= v - (\tilde{F_2})^{-1}[|G_2|Q_1 + |\Lambda - A|]v \\ &= v - (\tilde{F_2})^{-1}[|G_2|Q_1 v + |\Lambda - A|v] \\ &\leqslant v - (\tilde{F_2})^{-1}[|G_2|v + |\Lambda - A|v] \\ &= v - (\tilde{F_2})^{-1}[|G_2| + |\Lambda - A|]v \\ &= v - Q_2 v < v. \end{aligned} \tag{26}$$

Based on the proof of Theorem 4.1, it is sufficient to confirm the validity of $\rho(E) < 1$, where E is defined in (9). Let's suppose $X = \begin{bmatrix} v \\ v \end{bmatrix}$. Consequently, $EX = \begin{bmatrix} Q_1 v \\ Rv \end{bmatrix} < \begin{bmatrix} v \\ v \end{bmatrix} = X$. In other words, $EX < X$. As E is a nonnegative matrix based on the Lemma 2.5, $\rho(E) < 1$. Thus, the proof is completed. □

Theorem 4.4. *Let $A = D - B$ be an H_+-matrix ($D = diag(A)$), $A = F_1 - G_1 = F_2 - G_2$ be two splittings of A, $\lambda = \rho(D^{-1}|B|)$, and Λ be a positive diagonal matrix that met $\Lambda \geq \frac{1}{2\alpha}D$. The MAOR II iteration method is convergent when one of the following conditions is satisfied.*

1. $0 \leqslant \beta \leqslant \alpha,\ \frac{1}{2\alpha}D \leqslant \Lambda < \frac{1}{\alpha}D,\ \lambda < \frac{1}{2},\ \frac{1}{2(1-\lambda)} < \alpha < \frac{3}{2(1+\lambda)}$.
2. $0 \leqslant \beta \leqslant \alpha,\ \ \Lambda \geq \frac{1}{\alpha}D,\ \lambda < 1,\ 0 < \alpha < \frac{2}{1+\lambda}$.
3. $0 < \alpha \leqslant \beta,\ \Lambda \geq \frac{1}{\alpha}D,\ \ \lambda < \frac{1}{2\beta},\ 2\beta\lambda < \alpha < 2 - 2\beta\lambda$.
4. $0 < \alpha \leqslant \beta,\ \frac{1}{2\alpha}D \leqslant \Lambda < \frac{1}{\alpha}D,\ \lambda < \frac{1}{4\beta},\ \frac{4\beta\lambda + 1}{2} < \alpha < \frac{3 - 4\beta\lambda}{2}$.

Proof. In the proof of Theorem 4.3, we defined $Q_i\ \ (i = 1, 2)$ and demonstrated that $Q_i\ \ (i = 1, 2)$ are M-matrices.
Let $P = Q_2 Q_1 = (I - (\tilde{F_2})^{-1}S_2)(I - (\tilde{F_1})^{-1}S_1)$, where $\tilde{F_i},\ Q_i$, and $S_i\ \ (i = 1, 2)$ are defined in (22), (23), and (24). According to the proof of Theorem 4.2, it is sufficient to demonstrate that $\rho(\hat{E}) < 1$ ($\hat{E}$ is defined in (19)). From the proof of Theorem 4.3, for conditions 1-4, a positive vector v related to matrix K exists such that $Q_i v < v\ \ (i = 1, 2)$. Thus $Pv = (Q_2 Q_1)v < v$. Let $X = \begin{bmatrix} v \\ v \end{bmatrix}$ then $\hat{E}X = \begin{bmatrix} Q_1 v \\ Pv \end{bmatrix} < \begin{bmatrix} v \\ v \end{bmatrix} = X$. Hence $\hat{E}X < X$.
Based on Lemma 2.5, since $\hat{E}$ is a nonnegative matrix, we have $\rho(\hat{E}) < 1$. The proof is finished. □

5. Numerical experiments

In order to check the adequacy of the proposed numerical methods, this section reports the three items, the CPU time, iteration steps (IT), and residual norm for the new methods, the modulus-based method (MSOR) [2], and the two-step modulus-based method (TSMSOR) [42]. In fact, this section numerically compares the new modulus schemes with the previous two efficient methods by presenting two examples related to the LCP. Numerical calculations of this section are performed in Matlab R2018a on a PC with an Intel(R), Core(TM), 2.80 GHz CPU, and 16.00 GB memory. The norm of absolute residual vectors (RES) is calculated from the following equation

$$RES(z^{(k)}) :=\| min(Az^{(k)} + q, z^{(k)}) \|_2 .$$

In our numerical computations, $\Lambda = \frac{1}{2\alpha}D$ and initial vectors are $x^{(0)} = y^{(0)} = [1, 0, 1, 0, ...]^T$. The condition for terminating the operation is that $RES(z^{(k)}) \leqslant 10^{-5}$ or $IT > 10^4$. Tables 2-13 show that the efficiency of the presented methods in terms of CPU time and iteration steps (IT).

Table 1: Methods.

Abbreviation	Description
MSOR	The modulus-based successive overrelaxation method
TSMSOR	The two-step modulus-based successive overrelaxation method
MSOR I	The modulus-based successive overrelaxation method I
MSOR II	The modulus-based successive overrelaxation method II

Example 5.1. *[2] Consider the LCP(q, A), such that $q = -Az^* \in R^n, A \in R^{n\times n}$ is given by $A = \hat{A} + \eta I$ $(\eta \geq 0)$, where $\hat{A} = Tridiag(-I, T, -I) \in R^{n\times n}$ is a block-tridiagonal matrix and $T = tridiag(-1, 4, -1) \in R^{m\times m}$ is a tridiagonal matrix. Let $I \in R^{m\times m}$ be the identity matrix, m be a prescribed positive integer $(n = m^2)$, and $z^* = (1, 2, 1, 2, ...)^T \in R^n$ be the unique solution of the LCP(q, A).*

Example 5.2. *[2] Assume LCP(q, A), in which $A = \hat{A} + \eta I$ $(\eta \geq 0) \in R^{n\times n}$, $q = -Az^* \in R^n$, and*

$$\hat{A} = Tridiag(-0.5I, T, -1.5I) \in R^{n\times n}, \quad z^* = (1, 2, 1, 2, ...)^T \in R^n.$$

Note that, $I \in R^{m\times m}$ is a identity matrix, $\hat{A}$ is a block-tridiagonal matrix, and $T = tridiag(-0.5, 4, -1.5) \in R^{m\times m}$ $(n = m^2)$ is a tridiagonal matrix.

Table 2: Numerical results for Example 5.1 with $\eta = 0.3$, $\alpha = 1$

	m=40			m=60			m=80		
method	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
MSOR	0.014203	202	9.9533e-06	0.041059	265	9.6548e-06	0.087162	313	9.3994e-06
TSMSOR	0.014991	145	9.557e-06	0.038901	149	9.9922e-06	0.062515	152	9.635e-06
MSOR I	0.0074389	69	9.6383e-06	0.019085	73	8.7244e-06	0.032044	75	8.8927e-06
MSOR II	0.0060319	63	8.5995e-06	0.017948	66	8.9132e-06	0.029976	68	8.7172e-06

Table 3: Numerical results for Example 5.1 with $\eta = 0.5$, $\alpha = 1$

	m=40			m=60			m=80		
method	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
MSOR	0.0088783	175	8.9822e-06	0.036766	206	9.7829e-06	0.068805	216	9.8297e-06
TSMSOR	0.0086499	93	9.355e-06	0.024855	95	9.7163e-06	0.045385	97	9.0636e-06
MSOR I	0.0060772	45	9.4156e-06	0.012775	47	9.1266e-06	0.021676	48	9.7012e-06
MSOR II	0.0076159	41	9.2514e-06	0.011992	43	8.4389e-06	0.019876	44	8.7074e-06

Table 4: Numerical results for Example 5.1 with $\eta = 0.7$, $\alpha = 1$

	m=40			m=60			m=80		
method	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
MSOR	0.010206	147	9.229e-06	0.027551	158	9.4626e-06	0.064813	162	9.5505e-06
TSMSOR	0.0079852	69	9.8907e-06	0.019922	71	9.1928e-06	0.041699	72	9.2264e-06
MSOR I	0.0036563	34	9.2702e-06	0.011246	36	6.8984e-06	0.01888	36	9.8932e-06
MSOR II	0.0034142	32	6.4082e-06	0.0098037	33	7.0732e-06	0.015924	34	6.3822e-06

Table 5: Numerical results for Example 5.1 with $\eta = 3.5,\ \alpha = 1$

	m=40			m=60			m=80		
method	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
MSOR	0.0027169	44	8.6675e-06	0.0058596	45	8.1691e-06	0.0081555	45	9.6728e-06
TSMSOR	0.0018238	21	5.8533e-06	0.0040521	21	7.384e-06	0.0062207	21	8.6479e-06
MSOR I	0.0015698	19	4.8774e-06	0.0034413	19	6.0469e-06	0.0056848	19	7.0267e-06
MSOR II	0.00095067	12	4.6499e-06	0.002375	12	5.8758e-06	0.0038594	12	6.898e-06

Table 6: Numerical results for Example 5.1 with $\eta = 4,\ \alpha = 1$

	m=40			m=60			m=80		
method	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
MSOR	0.0017749	40	9.754e-06	0.0044807	41	8.8566e-06	0.0073955	42	7.4072e-06
TSMSOR	0.0013835	19	7.3784e-06	0.0035522	19	9.2882e-06	0.0057737	20	5.3299e-06
MSOR I	0.001333	18	9.9184e-06	0.0035424	19	5.4304e-06	0.0055199	19	6.3119e-06
MSOR II	0.0009714	12	2.6845e-06	0.0024541	12	3.3819e-06	0.0037494	12	3.9594e-06

Table 7: Numerical results for Example 5.2 with $\eta = 0,\ \alpha = 1$

	m=40			m=60			m=80		
method	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
MSOR	0.013199	362	9.2868e-06	0.049045	534	8.2085e-06	0.1297	706	8.2807e-06
TSMSOR	0.015807	260	9.7408e-06	0.054206	359	9.7217e-06	0.1254	455	9.3162e-06
MSOR I	0.0082556	102	9.5735e-06	0.020914	141	9.3453e-06	0.049532	179	8.1879e-06
MSOR II	0.0062241	98	9.7146e-06	0.020326	137	7.5949e-06	0.048455	173	9.5097e-06

Table 8: Numerical results for Example 5.2 with $\eta = 1.5,\ \alpha = 1$

	m=40			m=60			m=80		
method	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
MSOR	0.0079819	221	9.9758e-06	0.02866	318	8.4362e-06	0.074263	414	8.3525e-06
TSMSOR	0.0026056	40	9.8988e-06	0.0064847	41	9.401e-06	0.012097	42	8.0885e-06
MSOR I	0.0014357	20	9.0126e-06	0.0037352	21	7.2587e-06	0.0068463	22	4.8954e-06
MSOR II	0.0015195	21	5.5619e-06	0.0035743	21	9.9306e-06	0.0066503	22	6.9076e-06

Table 9: Numerical results for Example 5.2 with $\eta = 2,\ \alpha = 1$

	m=40			m=60			m=80		
method	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
MSOR	0.0056108	190	9.5561e-06	0.028226	267	9.9078e-06	0.05659	343	8.8948e-06
TSMSOR	0.0021659	32	9.7419e-06	0.0060596	33	8.3303e-06	0.0099434	33	9.8878e-06
MSOR I	0.0014027	19	5.4095e-06	0.0033454	19	6.7773e-06	0.0064628	19	7.9697e-06
MSOR II	0.0012554	17	7.7318e-06	0.0031051	18	5.4474e-06	0.0057595	18	7.7406e-06

Table 12: Numerical results for Example 5.1 with $\eta = 3.5$

	α		0.8	0.9	0.95	1.1	1.15	1.2
m	method							
		CPU	0.0010915	0.0014209	0.0017426	0.0061626	0.021346	0.093148
	MSOR	IT	18	24	32	137	487	2082
		RES	8.0541e-06	8.6027e-06	7.8723e-06	9.77e-06	9.6719e-06	9.6719e-06

50	TSMSOR	CPU	0.0010162	0.0012625	0.0014888	0.0054068	0.080815	0.78486
		IT	9	12	15	62	1030	10000
		RES	8.1703e-06	3.5731e-06	7.5305e-06	9.3028e-06	9.9506e-06	530.3494
	MSOR I	CPU	0.0016954	0.0018042	0.00176	0.001688	0.0017189	0.0018935
		IT	16	18	19	18	18	20
		RES	6.6153e-06	7.185e-06	5.0884e-06	7.9623e-06	9.211e-06	2.6353e-06
	MSOR II	CPU	0.0015877	0.0015133	0.0012532	0.0015623	0.0016836	0.0020135
		IT	15	13	12	15	18	21
		RES	4.7957e-06	2.8448e-06	2.7972e-06	7.6473e-06	4.837e-06	6.8795e-06
70	MSOR	CPU	0.0034036	0.0034782	0.0084237	0.015862	0.069159	0.29493
		IT	19	25	32	140	636	2896
		RES	4.6438e-06	5.7715e-06	9.5697e-06	9.7618e-06	9.5786e-06	9.7433e-06
	TSMSOR	CPU	0.0024455	0.0028201	0.0038403	0.012015	0.20916	1.7876
		IT	10	12	15	63	1172	10000
		RES	1.846e-06	4.2932e-06	9.0608e-06	9.3241e-06	9.9496e-06	755.8766
	MSOR I	CPU	0.003621	0.0039744	0.0042917	0.0041269	0.0040158	0.0041891
		IT	16	18	19	19	19	20
		RES	8.0885e-06	8.5842e-06	6.0751e-06	3.9848e-06	5.1501e-06	3.2168e-06
	MSOR II	CPU	0.0034044	0.002949	0.0028256	0.0034181	0.0039029	0.0044569
		IT	15	13	12	15	18	21
		RES	6.9885e-06	4.1617e-06	4.0076e-06	9.2558e-06	5.8741e-06	8.3775e-06
90	MSOR	CPU	0.0044449	0.0061094	0.0070129	0.026873	0.13217	0.61288
		IT	19	25	33	142	780	3711
		RES	6.1297e-06	6.6259e-06	7.0732e-06	9.6644e-06	9.4569e-06	9.7062e-06
	TSMSOR	CPU	0.0040929	0.0046067	0.0058149	0.020064	0.36385	2.9039
		IT	10	12	16	64	1242	10000
		RES	2.4323e-06	4.9088e-06	4.1577e-06	8.8064e-06	9.9897e-06	981.2911
	MSOR I	CPU	0.0058035	0.0064307	0.0066506	0.006735	0.0066679	0.0071236
		IT	16	18	19	19	19	20
		RES	9.4233e-06	9.7853e-06	6.9226e-06	4.8531e-06	6.3728e-06	3.7436e-06
	MSOR II	CPU	0.0055862	0.0050182	0.0046303	0.0059046	0.0064255	0.0074793
		IT	15	13	12	16	18	21
		RES	9.1813e-06	5.4787e-06	5.2177e-06	3.8897e-06	6.7538e-06	9.6457e-06

Table 13: Numerical results for Example 5.2 with $\eta = 3.5$

	α		0.8	0.9	0.95	1.1	1.15	1.2
m	method							
	MSOR	CPU	0.0011779	0.0020656	0.0026346	0.01968	0.42468	0.44467
		IT	22	38	56	396	10000	10000
		RES	4.8084e-06	6.926e-06	9.3849e-06	8.2941e-06	6.0353	0.021466

50	TSMSOR	CPU	0.0010359	0.0012458	0.0015832	0.0059311	0.029062	0.80006
		IT	9	12	16	68	371	10000
		RES	4.3112e-06	6.7146e-06	6.2631e-06	9.5973e-06	9.7555e-06	499.5819
	MSOR I	CPU	0.0016805	0.0017601	0.0017686	0.0017679	0.0019381	0.002216
		IT	17	19	19	19	21	24
		RES	7.8918e-06	6.7311e-06	8.4167e-06	7.7204e-06	8.6589e-06	7.9348e-06
	MSOR II	CPU	0.0016289	0.0016188	0.0013688	0.0012033	0.0012854	0.0014046
		IT	17	15	14	11	12	14
		RES	6.3169e-06	3.7295e-06	3.2904e-06	4.3575e-06	5.25e-06	4.4195e-06
70	MSOR	CPU	0.0032136	0.0048193	0.0072019	0.058198	1.0172	1.0152
		IT	22	38	57	538	10000	10000
		RES	6.7985e-06	8.6173e-06	9.3184e-06	9.5697e-06	319.7292	221.9894
	TSMSOR	CPU	0.0024762	0.0027987	0.0035058	0.013215	0.085615	1.7646
		IT	9	12	16	70	480	10000
		RES	6.3853e-06	8.0803e-06	7.562e-06	8.2442e-06	9.7698e-06	709.7413
	MSOR I	CPU	0.0040719	0.0040687	0.0041507	0.0041543	0.0046666	0.0048573
		IT	18	19	20	19	22	24
		RES	4.2258e-06	8.051e-06	4.6503e-06	9.3375e-06	4.1914e-06	9.5505e-06
	MSOR II	CPU	0.0039097	0.0033813	0.0030813	0.0025702	0.003116	0.0031086
		IT	17	15	14	11	12	14
		RES	9.33e-06	5.5377e-06	4.8988e-06	5.8849e-06	6.2969e-06	5.3062e-06
90	MSOR	CPU	0.0054165	0.008449	0.011466	0.11478	1.6477	1.652
		IT	22	39	58	681	10000	10000
		RES	8.7769e-06	6.8554e-06	8.5487e-06	8.892e-06	634.992	609.0909
	TSMSOR	CPU	0.0037938	0.0046769	0.0059422	0.021642	0.16976	2.9319
		IT	9	12	16	70	584	10000
		RES	8.4592e-06	9.2464e-06	8.6684e-06	9.7124e-06	9.6765e-06	919.4969
	MSOR I	CPU	0.0064268	0.0066812	0.0070438	0.006983	0.0074798	0.0083884
		IT	18	17	20	20	22	25
		RES	5.2588e-06	9.1831e-06	5.2987e-06	4.358e-06	4.7879e-06	4.1849e-06
	MSOR II	CPU	0.0064258	0.0055416	0.005317	0.0043567	0.0046607	0.0052042
		IT	18	15	14	11	12	14
		RES	4.7993e-06	7.3457e-06	6.5071e-06	7.4054e-06	7.1935e-06	6.0646e-06

Tables 2 to 11 report three parameters: CPU time (CPU), iteration steps (IT), and residual (RES), for symmetric and nonsymmetric Examples 5.1, 5.2, and $\alpha = 1$. The following results can be seen:

1. As the size of the problem increases with a constant η, the CPU time and IT for all the mentioned methods increase. In essence, the superiority of the newly proposed methods is maintained with an increase in the problem size.
2. With an increase in the value of η, it can be seen that the CPU and IT decrease.
3. The proposed methods exhibit nearly three times better performance than the MSOR method and 1.5 times better performance than the TSMSOR method.

Table 10: Numerical results for Example 5.2 with $\eta = 3.5,\ \alpha = 1$

	m=40			m=60			m=80		
method	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
MSOR	0.0037872	99	9.6179e-06	0.0099387	103	9.4803e-06	0.016687	105	9.375e-06
TSMSOR	0.0016322	22	7.0793e-06	0.0038406	22	9.0081e-06	0.006843	23	5.615e-06
MSOR I	0.0013923	19	4.7351e-06	0.003565	19	5.8404e-06	0.0058009	19	6.77e-06
MSOR II	0.00095573	12	8.1282e-06	0.0025592	13	3.911e-06	0.0043146	13	5.4558e-06

Table 11: Numerical results for Example 5.2 with $\eta = 4,\ \alpha = 1$

	m=40			m=60			m=80		
method	CPU	IT	RES	CPU	IT	RES	CPU	IT	RES
MSOR	0.0031098	80	8.6388e-06	0.0083771	82	9.0801e-06	0.013386	83	9.5347e-06
TSMSOR	0.0016004	20	8.0737e-06	0.0035757	21	5.0998e-06	0.0067784	21	5.9901e-06
MSOR I	0.0013663	19	4.6341e-06	0.0032135	19	5.7216e-06	0.0063392	19	6.6359e-06
MSOR II	0.0008787	11	8.7877e-06	0.002270	12	3.8597e-06	0.004298	12	5.3642e-06

4. It's noteworthy that for $\alpha = 1$, the modulus-based SOR method reduces to the modulus-based Gauss-Seidel method.

From Tables 12 and 13 for different values of $\alpha \neq 1$, constant η, and different sizes of the problem, it can be seen that:

1. The proposed methods for $\alpha > 1$ demonstrate significantly better performance compared to previous methods in terms of CPU time and IT. This superiority remains valid with an increase in problem sizes.
2. Moreover, for some α-values, MSOR and TSMSOR methods exhibit divergence, while the proposed methods demonstrate convergence. In other words, the new methods have a wider range of convergence.

Overall, the numerical results indicate that the proposed methods (MSOR I, MSOR II) outperform other mentioned methods in terms of CPU time and iteration steps (IT).

6. Conclusions

This paper presented two efficient methods for solving the linear complementarity problem. The convergence of these methods was discussed when the system matrix was an H_+-matrix. The numerical results confirmed the efficiency of the new methods by comparing them with the modulus-based method [2] and two-step modulus-based method [42].

Acknowledgements

The authors are deeply indebted to the editor and an anonymous reviewer for their constructive comments and suggestions.

References

[1] Z.-Z. Bai. On the convergence of the multisplitting methods for the linear complementarity problem. *SIAM Journal on Matrix Analysis and Applications*, 21(1):67–78, 1999.

[2] Z.-Z. Bai. Modulus-based matrix splitting iteration methods for linear complementarity problems. *Numerical Linear Algebra with Applications*, 17(6):917–933, 2010.

[3] Z.-Z. Bai, A. Buccini, K. Hayami, L. Reichel, J.-F. Yin, and N. Zheng. Modulus-based iterative methods for constrained tikhonov regularization. *Journal of Computational and Applied Mathematics*, 319:1–13, 2017.

[4] Z.-Z. Bai and D. Evans. Matrix multisplitting methods with applications to linear complementarity problems: parallel synchronous and chaotic methods. *Calculateurs Parallèles Réseaux et Systèmes Répartis*, 13(1):125–141, 2001.

[5] Z.-Z. Bai and D. J. Evans. Matrix multisplitting relaxation methods for linear complementarity problems. *International Journal of Computer Mathematics*, 63(3-4):309–326, 1997.

[6] Z.-Z. Bai and L.-L. Zhang. Modulus-based synchronous multisplitting iteration methods for linear complementarity problems. *Numerical Linear Algebra with Applications*, 20(3):425–439, 2013.

[7] M. Bashirizadeh and M. Hajarian. Two-step two-sweep modulus-based matrix splitting iteration method for linear complementarity problems. *Numerical Mathematics: Theory, Methods & Applications*, 15(3), 2022.

[8] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*, volume 9. SIAM, 1994.

[9] R. W. Cottle, J.-S. Pang, and R. E. Stone. *The linear complementarity problem*. SIAM, 2009.

[10] C. W. Cryer. The solution of a quadratic programming problem using systematic overrelaxation. *SIAM Journal on Control*, 9(3):385–392, 1971.

[11] M. Dehghan and M. Hajarian. Convergence of SSOR methods for linear complementarity problems. *Operations Research Letters*, 37(3):219–223, 2009.

[12] M. Dehghan and M. Hajarian. Two class of synchronous matrix multisplitting schemes for solving linear complementarity problems. *Journal of Computational and Applied Mathematics*, 235(15):4325–4336, 2011.

[13] J.-L. Dong and M.-Q. Jiang. A modified modulus method for symmetric positive-definite linear complementarity problems. *Numerical Linear Algebra with Applications*, 16(2):129–143, 2009.

[14] K. Fan. Topological proofs for certain theorems on matrices with non-negative elements. *Monatshefte für Mathematik*, 62(3):219–237, 1958.

[15] X. Fang. The convergence of the modulus-based jacobi (MJ) iteration method for solving horizontal linear complementarity problems. *Computational and Applied Mathematics*, 41(4):1–16, 2022.

[16] M. C. Ferris and J.-S. Pang. Engineering and economic applications of complementarity problems. *SIAM Review*, 39(4):669–713, 1997.

[17] A. Frommer and G. Mayer. Convergence of relaxed parallel multisplitting methods. *Linear Algebra and its Applications*, 119:141–152, 1989.

[18] L. Gao, Y. Wang, C. Li, and Y. Li. Error bounds for linear complementarity problems of S-Nekrasov matrices and B-S-Nekrasov matrices. *Journal of computational and Applied Mathematics*, 336:147–159, 2018.

[19] A. Hadjidimos and M. Tzoumas. Nonstationary extrapolated modulus algorithms for the solution of the linear complementarity problem. *Linear Algebra and its Applications*, 431(1-2):197–210, 2009.

[20] M. Hajarian. The PMCGAOR and PMCSSOR methods for solving linear complementarity problems. *Computational and Applied Mathematics*, 34(1):251–264, 2015.

[21] B. Huang and C. Ma. Accelerated modulus-based matrix splitting iteration method for a class of nonlinear complementarity problems. *Computational and Applied Mathematics*, 37(3):3053–3076, 2018.

[22] Z.-G. Huang and J.-J. Cui. Accelerated relaxation modulus-based matrix splitting iteration method for linear complementarity problems. *Bulletin of the Malaysian Mathematical Sciences Society*, 44(4):2175–2213, 2021.

[23] Y.-F. Ke and C.-F. Ma. On the convergence analysis of two-step modulus-based matrix splitting iteration method for linear complementarity problems. *Applied Mathematics and Computation*, 243:413–418, 2014.

[24] C. E. Lemke and J. T. Howson, Jr. Equilibrium points of bimatrix games. *Journal of the Society for industrial and Applied Mathematics*, 12(2):413–423, 1964.

[25] C.-L. Li and J.-T. Hong. Modulus-based synchronous multisplitting iteration methods for an implicit complementarity problem. *EAST Asian Journal on Applied Mathematics*, 7(2):363–375, 2017.

[26] D.-K. Li, L. Wang, and Y.-Y. Liu. A relaxation general two-sweep modulus-based matrix splitting iteration method for solving linear complementarity problems. *Journal of Computational and Applied Mathematics*, 409:114140, 2022.

[27] W. Li. A general modulus-based matrix splitting method for linear complementarity problems of H-matrices. *Applied Mathematics Letters*, 26(12):1159–1164, 2013.

[28] Z. Li, Y. Ke, H. Zhang, and R. Chu. SOR-like iteration methods for second-order cone linear complementarity problems. *East Asian Journal on Applied Mathematics*, 10(2):295–315, 2020.

[29] F. Mezzadri. On the equivalence between some projected and modulus-based splitting methods for linear complementarity problems. *Calcolo*, 56(4):1–20, 2019.

[30] F. Mezzadri and E. Galligani. Modulus-based matrix splitting methods for horizontal linear complementarity problems. *Numerical Algorithms*, 83(1):201–219, 2020.

[31] H. Ren, X. Wang, X.-B. Tang, and T. Wang. The general two-sweep modulus-based matrix splitting iteration method for solving linear complementarity problems. *Computers & Mathematics with Applications*, 77(4):1071–1081, 2019.

[32] H. Ren, X. Wang, X.-B. Tang, and T. Wang. A preconditioned general two-step modulus-based matrix splitting iteration method for linear complementarity problems of H_+-matrices. *Numerical Algorithms*, 82(3):969–986, 2019.

[33] Q. Shi, Q.-Q. Shen, and T.-P. Tang. A class of two-step modulus-based matrix splitting iteration methods for quasi-complementarity problems. *Computational and Applied Mathematics*, 39(1):1–23, 2020.

[34] R. S.Varga. *Matrix Iterative Analysis*. Prentice-Hall, 1962.

[35] W. M. G. Van Bokhoven. Piecewise-linear modelling and analysis. *Proefschrift: Eindhoven*, 1981.

[36] B. Wang, X. Wu, and F. Meng. Trigonometric collocation methods based on lagrange basis polynomials for multi-frequency oscillatory second-order differential equations. *Journal of Computational and Applied Mathematics*, 313:185–201, 2017.

[37] Z. I. Woznicki. Nonnegative splitting theory. *Japan Journal of Industrial and Applied Mathematics*, 11(2):289–342, 1994.

[38] S.-L. Wu and C.-X. Li. Two-sweep modulus based matrix splitting iteration methods for linear complementarity problems. *Journal of Computational and Applied Mathematics*, 302:327–339, 2016.

[39] Y.-J. Wu, G.-L. Yan, and A.-L. Yang. Modulus-based synchronous multisplitting iteration methods for a restricted class of nonlinear complementarity problems. *Numerical Mathematics: Theory, Methods & Applications*, 12(3), 2019.

[40] Z. Xia and C. Li. Modulus-based matrix splitting iteration methods for a class of nonlinear complementarity problem. *Applied Mathematics and Computation*, 271:34–42, 2015.

[41] D. M. Young. *Iterative Solution of Large Linear Systems*. Academic Press, New York, 1971.

[42] L.-L. Zhang. Two-step modulus-based matrix splitting iteration method for linear complementarity problems. *Numerical Algorithms*, 57(1):83–99, 2011.

[43] L.-L. Zhang and Z.-R. Ren. Improved convergence theorems of modulus-based matrix splitting iteration methods for linear complementarity problems. *Applied Mathematics Letters*, 26(6):638–642, 2013.

[44] L.-T. Zhang, Y.-X. Zhang, T.-X. Gu, X.-P. Liu, and L.-W. Zhang. New convergence of modulus-based synchronous block multisplitting multi-parameter methods for linear complementarity problems. *Computational and Applied Mathematics*, 36(1):481–492, 2017.

[45] H. Zheng and L. Liu. A two-step modulus-based matrix splitting iteration method for solving nonlinear complementarity problems of H_+-matrices. *Computational and Applied Mathematics*, 37(4):5410–5423, 2018.

[46] H. Zheng and S. Vong. On the modulus-based successive overrelaxation iteration method for horizontal linear complementarity problems arising from hydrodynamic lubrication. *Applied Mathematics and Computation*, 402:126165, 2021.

[47] H. Zheng, S. Vong, and L. Liu. The relaxation modulus-based matrix splitting iteration method for solving a class of nonlinear complementarity problems. *International Journal of Computer Mathematics*, 96(8):1648–1667, 2019.