Computational and Applied Mathematics (2024) 43:33
https://doi.org/10.1007/s40314-023-02563-9

A relaxed two-step modulus-based matrix synchronous multisplitting iteration method for linear complementarity problems

Yongxiong Zhang[1] · Wenxiu Guo[2,3] · Hua Zheng[3] · Seakweng Vong[4]

Received: 9 October 2023 / Revised: 24 November 2023 / Accepted: 28 November 2023 /
Published online: 30 December 2023

Abstract
In this paper, a relaxed two-step modulus-based matrix synchronous multisplitting iteration method for solving the linear complementarity problems is constructed. The convergence conditions of the proposed method are analyzed with the convergence range of the relaxation parameters. Some parallel numerical experiments under OpenACC framework are given to show that the proposed method can accelerate the convergence rate of the existing method significantly.

Keywords Linear complementarity problem · Relaxation · Parallel · Modulus equation

Mathematics Subject Classification 65F10 · 65Y05 · 90C33

Communicated by Jinyun Yuan.

✉ Hua Zheng
hzheng@sgu.edu.cn

Yongxiong Zhang
zhangyx@gzgs.edu.cn

Wenxiu Guo
wxguo@sgu.edu.cn

Seakweng Vong
swvong@um.edu.mo

1 School of Engineering and Technology, Guangzhou College of Technology and Business, Guangzhou, People's Republic of China

2 School of Computer Science and Engineering, Macau University of Science and Technology, Macau, People's Republic of China

3 School of Mathematics and Statistics, Shaoguan University, Shaoguan, People's Republic of China

4 Department of Mathematics, University of Macau, Macau, People's Republic of China

1 Introduction

Focus on the numerical algorithms for linear complementarity problems (LCP). The definition of LCP is as follows: given matrix $M \in \mathbb{R}^{n\times n}$ and vector $q \in \mathbb{R}^n$, finding $z \in \mathbb{R}^n$, satisfying the following constraints

$$r = Mz + q \geq 0, z \geq 0, \text{ and } z^T r = 0,$$

where for two $s \times t$ matrices $K = (k_{ij})$ and $T = (t_{ij})$ the order $K \geq (>)T$ means $k_{ij} \geq (>)t_{ij}$ for any i and j. The LCP has extremely wide applications in fields such as contact problems, network equilibrium problems, free boundary problems and so on; see Cottle and Pang (1992), Murty (1988) and the references therein.

The modulus-based matrix splitting (MMS) iterative method constructed based on LCP's equivalent modulus equation by introducing matrix splitting technology had become a research hotspot in recent years. In 2010, Bai first proposed the MMS iteration method in Bai (2010), which involves equivalently transforming LCP into the following modulus equations:

$$(\Omega + M)x = (\Omega - M)|x| - \gamma q, \tag{1}$$

where Ω is a positive diagonal matrix, $\gamma > 0$ and the absolute operation is taken componentwise. Once the solution $x \in \mathbb{R}^n$ of (1) is obtained, the solution of LCP is computed by $z = \frac{1}{\gamma}(x + |x|)$. The type of MMS iteration methods is more convenient and efficient than the projected relaxation (Cryer 1971) and the modified modulus methods (Dong and Jiang 2009). Furthermore, researchers had promoted and accelerated the MMS method in different ways, such as Fang (2022), Fang et al. (2023), Fang and Zhu (2019), Huang and Ma (2018), Ke and Ma (2014), Ke et al. (2018), Ren et al. (2019), Song et al. (2022), Wu et al. (2018), Zhang (2011), Zheng et al. (2021), Zheng et al. (2017), Zheng et al. (xxxx), Zheng and Vong (2021), Zheng et al. (2019). On the other hand, by high-performance computers, the parallel methods had been studied by introducing synchronous multisplitting techniques into (1). In Bai and Zhang (2013b), the modulus-based synchronous multisplitting (MSM) iteration method was first introduced. The idea of the MSM method is to solve a simple linear equation system by evenly distributing workload, combining different matrix splittings at every iteration in each worker. By combining different acceleration techniques, the MSM method had also been improved in subsequent researches, such as Bai and Zhang (2013a), Wu and Li (2019), Xu et al. (2020), Zhang (2014), Zhang (2015).

Among numerous acceleration technologies of MMS and MSM, the two-step method is an important one, whose idea is to use two different splittings of the system matrix M to construct two linear systems which need to be solved in each iteration. The advantage of this approach is that it can fully utilize the elements' information of the system matrix to achieve acceleration. The two-step methods for solving LCP can be mainly divided into two categories: serial and parallel, see Zhang (2011) and Zhang (2015) for details, respectively. The two-step method also has important applications in the iterative solution of linear equations. In a recent literature (Bai xxxx), Bai had discussed in detail the the two-step MMS paradigm for linear equations, where a TMSI paragigm was introduced, which can be seen as a result of combining the techniques of two-step splitting and relaxation.

In this paper, we further focuses on the acceleration of two-step parallel methods for solving LCP. Inspired by the TMSI paradigm in Bai (xxxx), we will introduce relaxed technology on the two-step MSM (TMSM) given in Zhang (2015) and construct a relaxed two-step parallel method in Sect. 2. Next, we provide convergence analysis of the proposed method to obtain

the selection range of relaxed parameters in Sect. 3, where the given convergence results can generalize and improve the corresponding ones in Zhang (2015). In Sect. 4, numerical examples are given to show that the proposed method can converge faster than the existing methods. Finally, the conclusions are presented in the last section.

Next, we provide some notations, definitions, and results that need to be used in subsequent discussions.

Use e to denote an $n \times 1$ vector whose entries are all equal to 1 and $\mathrm{diag}(x)$ to denote an $n \times n$ diagonal matrix whose main diagonal is $x \in \mathbb{R}^n$. Let $M = (m_{ij}) \in \mathbb{R}^{n\times n}$. The absolute of M is denoted as $|M| = (|m_{ij}|)$. The comparison matrix of M is denoted as $\langle M\rangle = (m'_{ij})$, defined as

$$m'_{ij} = \begin{cases} |m_{ii}|, & \text{if } i = j, \\ -|m_{ij}|, & \text{if } i \neq j. \end{cases}$$

The spectral radius of M is denoted as $\rho(M)$. The diagonal matrix formed by the diagonal elements of M is denoted as D_M, the upper triangular matrix formed by the strictly upper triangular part is denoted as $-U_M$, and the lower triangular matrix formed by the strictly lower triangular part is denoted as $-L_M$. Note that we have $M = D_M - L_M - U_M$. The following are the definitions of some special matrices and the definitions of matrix splittings (Bai 1999; Berman and Plemmons 1994; Frommer and Szyld 1992):

- if $m_{ij} \leq 0$ for $i \neq j$, M is called a Z-matrix;
- if M is a nonsingular Z-matrix and $M^{-1} \geq 0$, M is called a nonsingular M-matrix;
- if $|m_{ii}| > \sum_{j\neq i}|m_{ij}|$ for any $i \in \{1, \ldots, n\}$, M is called a strictly diagonal dominant (s.d.d.) matrix;
- if $\langle M\rangle$ is a nonsingular M-matrix, M is called an H-matrix;
- if M is an H-matrix with positive diagonal entries, M is called an H_+-matrix;
- if F is nonsingular, $M = F - G$ is called a splitting of M;
- if $\langle F\rangle - |G|$ is a nonsingular M-matrix, $M = F - G$ is called an H-splitting of M;
- if $\langle F\rangle - |G| = \langle M\rangle$, $M = F - G$ is called an H-compatible splitting of M;
- if $M = F_i - G_i$ is a splitting of M for $1 \leq i \leq \ell$, $\{F_i, G_i, E_i\}_{i=1}^{\ell}$ is called a multisplitting of M, where $E_i \in \mathbb{R}^{n\times n}$ is a nonnegative diagonal weight matrix satisfying $\sum_{i=1}^{\ell} E_i = I$;
- If $M = F_i^{(1)} - G_i^{(1)} = F_i^{(2)} - G_i^{(2)}$ are two splitting of M for $1 \leq i \leq \ell$, $\{F_i^{(1)}, G_i^{(1)}, F_i^{(2)}, G_i^{(2)}, E_i\}_{i=1}^{\ell}$ is called a two-step multisplitting of M, where the definition of E_i is the same as the one given above.

2 New method

We review the TMSM method given in Zhang (2015) first. Let $M = F^{(1)} - G^{(1)} = F^{(2)} - G^{(2)}$ be two splittings of M. Then, the LCP can be equivalently transformed into a system of fixed-point equations

$$\begin{cases} (\Omega + F^{(1)})x = G^{(1)}x + (\Omega - M)|x| - \gamma q, \\ (\Omega + F^{(2)})x = G^{(2)}x + (\Omega - M)|x| - \gamma q, \end{cases} \tag{2}$$

where Ω is a positive diagonal parameter matrix and γ is a positive constant; see Zhang (2011) for more details. Let $\{F_i^{(1)}, G_i^{(1)}, F_i^{(2)}, G_i^{(2)}, E_i\}_{i=1}^{\ell}$ be a two-step multisplittings of M. Based on (2), in order to reduce communication cost between processors in a high-performance computer environment with multiple processors and make full use of the information of previous iteration, the TMSM iteration method was proposed in Zhang (2015) as follows:

Method 1 (Zhang 2015) Given $x^{(0)} \in \mathbb{R}^n$ and $\epsilon > 0$, for $i = 1, 2, \ldots, \ell$, compute $x^{(k+1,i)} \in \mathbb{R}^n$ by

$$\begin{cases} (\Omega + F_i^{(1)})x^{(k+\frac{1}{2},i)} = G_i^{(1)}x^{(k)} + (\Omega - M)|x^{(k)}| - \gamma q, \\ (\Omega + F_i^{(2)})x^{(k+1,i)} = G_i^{(2)}x^{(k+\frac{1}{2},i)} + (\Omega - M)|x^{(k+\frac{1}{2},i)}| - \gamma q. \end{cases}$$

Then, calculate the weight combination of the updates in ℓ processors

$$x^{(k+1)} = \sum_{i=1}^{\ell} E_i x^{(k+1,i)} \text{ and } z^{(k+1)} = \frac{1}{\gamma}(|x^{(k+1)}| + x^{(k+1)})$$

for $k = 0, 1, 2, \ldots$ until $\| \min\{z^{(k+1)}, Mz^{(k+1)} + q\}\| < \epsilon$.

To obtain fast convergence, in the ith worker, consider mixing the local update $x^{(k+1,i)}$ with the old approach vector $x^{(k)}$ before the combination. By introducing relaxed parameters, we proposed the relaxed TMSM method as follows:

Method 2 Given an initial vector $x^{(0)} \in \mathbb{R}^n$ and $\epsilon > 0$, for $i = 1, 2, \ldots, \ell$, compute $x^{(k+1,i)} \in \mathbb{R}^n$ by

$$\begin{cases} (\Omega + F_i^{(1)})x^{(k+\frac{1}{3},i)} = G_i^{(1)}x^{(k)} + (\Omega - M)|x^{(k)}| - \gamma q, \\ (\Omega + F_i^{(2)})x^{(k+\frac{2}{3},i)} = G_i^{(2)}x^{(k+\frac{1}{3},i)} + (\Omega - M)|x^{(k+\frac{1}{3},i)}| - \gamma q, \\ x^{(k+1,i)} = \lambda^{(k,i)}x^{(k+\frac{2}{3},i)} + (1 - \lambda^{(k,i)})x^{(k)}, \end{cases} \tag{3}$$

where $\lambda^{(k,i)}$ is a scalar. Then, calculate the weight combination of the updates in ℓ processors

$$x^{(k+1)} = \sum_{i=1}^{\ell} E_i x^{(k+1,i)} \text{ and } z^{(k+1)} = \frac{1}{\gamma}(|x^{(k+1)}| + x^{(k+1)})$$

for $k = 0, 1, 2, \ldots$ until $\| \min\{z^{(k+1)}, Mz^{(k+1)} + q\}\| < \epsilon$.

For Method 2, we have the next comments.

- If we take $\lambda^{(k,i)} \equiv 1$, then Method 2 reduces to Method 1, which implies that the scope of the TMSM method is extended. Furthermore, one may chose some relaxed parameters $\lambda^{(k,i)}$ to make Method 2 converge faster than Method 1.
- In applications, one can specially chose the splittings as those in Zhang (2015) to obtain the classes of relaxed symmetric MSM accelerated overrelaxation (RSMSMAOR), MSM overrelaxation (RSMSMSOR) and MSM Gauss-Seidel (RSMSMGS) iteration methods.

3 Convergence analysis

Some useful lemmas are presented first.

Lemma 3 (Frommer and Mayer 1989) *If M is an H-matrix, $|M^{-1}| \leq \langle M \rangle^{-1}$.*

Lemma 4 (Hu 1982) *Let $F \in \mathbb{R}^{n\times n}$ be an s.d.d. matrix. Then, $\forall G \in \mathbb{R}^{n\times n}$,*

$$\|F^{-1}G\|_\infty \leq \max_{1\leq j\leq n} \frac{(|G|e)_j}{(\langle F \rangle e)_j}.$$

Lemma 5 (Berman and Plemmons 1994) *Let M be a Z-matrix. Then, the following statements are equivalent:*

- *M is a nonsingular M-matrix;*
- *There exists a positive diagonal matrix Π, such that $M\Pi$ is an s.d.d. matrix with positive diagonal entries;*
- *For any splitting $M = F - G$ satisfying $F^{-1} \geq 0$ and $G \geq 0$, it holds $\rho(F^{-1}G) < 1$.*

Next, the main convergence theorem of the Method 2 is given.

Theorem 6 *For $i = 1, 2, \ldots, \ell$, assume that:*

(I) *$A = F_i^{(1)} - G_i^{(1)} = F_i^{(2)} - G_i^{(2)}$ are H-splittings of M;*

(II) *there exists a positive diagonal matrix Π such that $\langle M\rangle\Pi$, $(\langle F_i^{(1)}\rangle - |G_i^{(1)}|)\Pi$ and $(\langle F_i^{(2)}\rangle - |G_i^{(2)}|)\Pi$ are s.d.d. matrices.*

Then, Method 2 is convergent for any initial vector $x^{(0)} \in \mathbb{R}^n$ provided

$$\Omega e > D_M e - \frac{1}{2} \min_{\substack{1\leq i\leq \ell \\ s=1,2}} \left[\Pi^{-1}(\langle M\rangle + \langle F_i^{(s)}\rangle - |G_i^{(s)}|)\Pi e\right], \tag{4}$$

and

$$0 < \lambda^{(k,i)} < \frac{2}{1 + \|\Pi^{-1}\mathcal{P}_i^{(2)}\mathcal{P}_i^{(1)}\Pi\|_\infty}, \tag{5}$$

where

$$\mathcal{P}_i^{(s)} = (\Omega + \langle F_i^{(s)}\rangle)^{-1}(|G_i^{(s)}| + |\Omega - M|), s = 1, 2. \tag{6}$$

Proof Let z^* be the solution of the LCP. By Zhang (2015), $x^* = \frac{\gamma}{2}(z^* - \Omega^{-1}r)$ satisfies the implicit fixed-point equations

$$\begin{cases} (\Omega + F_i^{(1)})x^* = G_i^{(1)}x^* + (\Omega - M)|x^*| - \gamma q, \\ (\Omega + F_i^{(2)})x^* = G_i^{(2)}x^* + (\Omega - M|x^*| - \gamma q. \end{cases} \tag{7}$$

Denote the errors in the kth step by

$$\delta^{(k)} = x^{(k)} - x^*, \delta^{(k,i)} = x^{(k,i)} - x^*.$$

For $s = 1, 2$, since $(\langle F_i^{(s)}\rangle - |G_i^{(s)}|)\Pi$ are s.d.d. matrices, we have

$$(\Omega + \langle F_i^{(s)}\rangle)\Pi e \geq \langle F_i^{(s)}\rangle\Pi e \geq (\langle F_i^{(s)}\rangle - |G_i^{(s)}|)\Pi e > 0,$$

which implies that $(\Omega + \langle F_i^{(s)}\rangle)D$ are s.d.d. matrices and $\Omega + F_i^{(s)}$ are H-matrices. By subtracting (7) from the first and second equations of (3), with Lemma 3 we have

$$|\delta^{(k+\frac{1}{3},i)}| \leq |(\Omega + F_i^{(s)})^{-1}|(|G_i^{(s)}| + |\Omega - M|)|\delta^{(k)}| \leq \mathcal{P}_i^{(1)}|\delta^{(k)}|.$$

Similarly, we can get $|\delta^{(k+\frac{2}{3},i)}| \leq \mathcal{P}_i^{(2)}|\delta^{(k+\frac{1}{3},i)}|$. Then, with the third equation of (3), we have

$$|\delta^{(k+1,i)}| \leq |\lambda^{(k,i)}||\delta^{(k+\frac{2}{3},i)}| + |1 - \lambda^{(k,i)}||\delta^{(k)}| \leq (\lambda^{(k,i)}\mathcal{P}_i^{(2)}\mathcal{P}_i^{(1)} + |1 - \lambda^{(k,i)}|I)|\delta^{(k)}|.$$

Note that $x^{(k+1)} = \sum_{i=1}^{\ell} E_i x^{(k+1,i)}$. We obtain

$$|\delta^{(k+1)}| \le \sum_{i=1}^{\ell} E_i(\lambda^{(k,i)}\mathcal{P}_i^{(2)}\mathcal{P}_i^{(1)} + |1-\lambda^{(k,i)}|I)|\delta^{(k)}|. \tag{8}$$

Now we estimate the infinite norm of $\Pi^{-1}\mathcal{P}_i^{(s)}\Pi$. By Lemma 4, we have

$$\|\Pi^{-1}\mathcal{P}_i^{(s)}\Pi\|_\infty \le \max_{1\le j\le n} \frac{[(\Omega + \langle F_i^{(s)}\rangle)\Pi e]_j}{[(|G_i^{(s)}| + |\Omega - M|)\Pi e]_j}. \tag{9}$$

Since $\langle M\rangle\Pi$ is an s.d.d. matrix, it holds that $\Pi^{-1}(\langle M\rangle + \langle F_i^{(s)}\rangle - |G_i^{(s)}|)\Pi e > 0$.
Case 1: If $\Omega \ge D_M$, we have

$$\begin{aligned}
&(\Omega + \langle F_i^{(s)}\rangle)\Pi e - (|G_i^{(s)}| + |\Omega - M|)\Pi e\\
&= (\Omega + \langle F_i^{(s)}\rangle - |G_i^{(s)}| - |\Omega - M|)\Pi e\\
&\ge (\langle M\rangle + \langle F_i^{(s)}\rangle - |G_i^{(s)}|)\Pi e\\
&> 0.
\end{aligned}$$

Case 2: If $D_M e > \Omega e > D_M e - \frac{1}{2}\min_{\substack{1\le i\le \ell\\ s=1,2}}\left[\Pi^{-1}(\langle M\rangle + \langle F_i^{(s)}\rangle - |G_i^{(s)}|)\Pi e\right]$, we have

$$D_M\Pi e > \Omega\Pi e > \left[\frac{1}{2}(|M| - \langle F_i^{(s)}\rangle + |G_i^{(s)}|)\right]\Pi e$$

and

$$\begin{aligned}
&(\Omega + \langle F_i^{(s)}\rangle)\Pi e - (|G_i^{(s)}| + |\Omega - M|)\Pi e\\
&= (\Omega + \langle F_i^{(s)}\rangle - |G_i^{(s)}| - |\Omega - M|)\Pi e\\
&\ge (2\Omega - |M| + \langle F_i^{(s)}\rangle - |G_i^{(s)}|)\Pi e\\
&> 0.
\end{aligned}$$

Then by (9), we have $\|\Pi^{-1}\mathcal{P}_i^{(s)}\Pi\|_\infty < 1$, which implies that

$$\|\Pi^{-1}\mathcal{P}_i^{(2)}\mathcal{P}_i^{(1)}\Pi\|_\infty \le \|\Pi^{-1}\mathcal{P}_i^{(2)}\Pi\|_\infty\|\Pi^{-1}\mathcal{P}_i^{(1)}\Pi\|_\infty < 1.$$

By direct computation, we have

$$\lambda^{(k,i)}\|\Pi^{-1}\mathcal{P}_i^{(2)}\mathcal{P}_i^{(1)}\Pi\|_\infty + |1-\lambda^{(k,i)}| < 1$$

for $\lambda^{(k,i)}$ satisfying (5). Then, we have the next inequality:

$$\begin{aligned}
&\rho\Big(\sum_{i=1}^{\hbar} E_i(\lambda^{(k,i)}\mathcal{P}_i^{(2)}\mathcal{P}_i^{(1)} + |1-\lambda^{(k,i)}|I)\Big)\\
&= \rho\big(\Pi^{-1}\sum_{i=1}^{\hbar} E_i(\lambda^{(k,i)}\mathcal{P}_i^{(2)}\mathcal{P}_i^{(1)} + |1-\lambda^{(k,i)}|I)\Pi\big)\\
&= \rho\Big(\sum_{i=1}^{\hbar} E_i(\lambda^{(k,i)}\Pi^{-1}\mathcal{P}_i^{(2)}\mathcal{P}_i^{(1)}D + |1-\lambda^{(k,i)}|I)\Big)
\end{aligned}$$

$$\begin{aligned}
&\leq \max_{1\leq l\leq n}\sum_{i=1}^{\hbar} e_l^{(s)}\|\lambda^{(k,i)}\Pi^{-1}\mathcal{P}_i^{(2)}\mathcal{P}_i^{(1)}\Pi + |1-\lambda^{(k,i)}|I\|_\infty \\
&\leq \max_{1\leq l\leq n}\sum_{i=1}^{\hbar} e_l^{(s)}(\lambda^{(k,i)}\|\Pi^{-1}\mathcal{P}_i^{(2)}\mathcal{P}_i^{(1)}\Pi\|_\infty + |1-\lambda^{(k,i)}|) \\
&< 1,
\end{aligned}$$

where $e_l^{(s)}$ is the lth diagonal entry of E_i. Hence, we can have the conclusion that $\{x^{(k)}\}_{k=1}^{\infty}$ converges by (8), proving the claim. □

Remark 1 In the convergence theorems of Zhang (2015), all splittings of M are assumed to be H-compatible splittings, and the positive diagonal matrix Ω in should satisfy

$$\Omega \geq D_M. \tag{10}$$

By Theorem 6, we have the next comments:

- It is well known that an H-compatible splitting is an H-splitting but not vice versus, which implies that the assumption on the matrix splittings in Theorem 6 is weaker than that of Zhang (2015).
- Note that by the proof of Theorem 6, we have $\Pi^{-1}(\langle M\rangle + \langle F_i^{(s)}\rangle - |G_i^{(s)}|)\Pi e > 0$. So (4) provides a larger convergence domain than (10).
- Compared to Zhang (2015), an extra assumption besides (5) is Assumption **(I)**. In fact, for the commonly used multisplittings, Assumption **(II)** is easy to be satisfied. Consider the AOR multisplittings in the numerical examples of Zhang (2015):

$$\begin{cases} F_i^{(1)} = \frac{1}{\alpha}(D_M - \beta L_{M_i}),\ G_i^{(1)} = F_i^{(1)} - M, \\ F_i^{(2)} = \frac{1}{\alpha}(D_M - \beta U_{M_i}),\ G_i^{(2)} = F_i^{(2)} - M, \end{cases}$$

where $-L_{M_i}$ and $-U_{M_i}$ are the strictly low-triangular and the strictly upper-triangular parts of $D_M - E_i M E_i$, respectively,

$$E_i = \mathrm{Diag}(0, 0, \ldots, I_{t_i}, 0, 0, \ldots, 0) \in \mathbb{R}^{n\times n},$$

the size $t_i = \phi_q + 1$ if $i \leq \phi_r$, and $t_i = \phi_q$ otherwise, with ϕ_q and ϕ_r being two nonnegative integers satisfying $n = \phi_q \ell + \phi_r$ and $0 \leq \phi_r \leq \ell$. The type of multisplittings can make the computational workload be almost evenly distributed to the ℓ processors; see Bai and Zhang (2013b), Zhang (2015) for more details. By simple computation, we have

$$\langle F_i^{(1)}\rangle - |G_i^{(1)}| = \langle F_i^{(2)}\rangle - |G_i^{(2)}| = \frac{1-|1-\alpha|}{\alpha} D_M - |L_M + U_M|.$$

Then, we can get that, if

$$0 < \beta \leq \alpha < \frac{2}{1+\rho(D_M^{-1}|L_M + U_M|)}, \tag{11}$$

Π given by

$$\Pi = \mathrm{diag}\Big[\Big(\frac{1-|1-\alpha|}{\alpha} D_M - |L_M + U_M|\Big)^{-1} e\Big] \tag{12}$$

can satisfy the Assumption **(II)** of Theorem 6.

Remark 2 In a recent work (Bai xxxx), a paradigm of two-step matrix splitting iteration methods for solving linear equations was constructed. If we take $\ell = 1$, Method 1 reduces to its serial version, whose idea is similar to a special case of the TMSI paradigm given in Bai (xxxx), with the mathematical problem being changed to the LCP. It had been shown in Bai (xxxx) that to find the theoretical optimal relaxation parameter is very difficult even in the case of linear equations. Note that in Theorem 6, the convergence range of $\lambda^{(k,i)}$ is given by (5) without analyzing the optimal one theoretically.

4 Numerical examples

In this section, three examples are given to illustrate the efficiency of Method 2.

Example 1 (Dong and Jiang 2009) Consider the LCP arising from finite difference discretization on the equidistant grid of a free boundary value problem about the flow of water through a porous dam, where

$$M = \begin{pmatrix} S & -I & & & \\ -I & S & -I & & \\ & -I & S & -I & \\ & & \ddots & \ddots & \ddots \\ & & & -I & S & -I \\ & & & & -I & S \end{pmatrix} \in \mathbb{R}^{n\times n},$$

$n = m^2$, $S = \text{tridiag}(-1, 4, -1) \in \mathbb{R}^{m\times m}$ and $I \in \mathbb{R}^{m\times m}$ is the identity matrix.

Example 2 (Bai 2010) Consider the LCP, where

$$M = \begin{pmatrix} S & -I & -I & & \\ & S & -I & -I & \\ & & \ddots & \ddots & \ddots \\ & & & S & -I & -I \\ & & & & S & -I \\ & & & & & S \end{pmatrix} \in \mathbb{R}^{n\times n},$$

where $n = m^2$, $S = \text{tridiag}(-1, 4, -1) \in \mathbb{R}^{m\times m}$ and $I \in \mathbb{R}^{m\times m}$ is the identity matrix.

Example 3 (Bai 2010) Consider the LCP, where

$$M = \begin{pmatrix} T & -0.5I & & & & \\ -1.5I & T & -0.5I & & & \\ & -1.5I & T & -0.5I & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1.5I & T & -0.5I \\ & & & & -1.5I & T \end{pmatrix} \in \mathbb{R}^{n\times n},$$

where $n = m^2$, $T = \text{tridiag}(-1.5, 4, -0.5) \in \mathbb{R}^{m\times m}$ and $I \in \mathbb{R}^{m\times m}$ is the identity matrix.

The numerical tests are performed on a Dell PowerEdge R740, using one NVIDIA Tesla V100S-32GB GPU. The programming language is C, accelerated using OpenACC (NVIDIA

2023) with CUDA Driver Version 12, where the parallel levels are set to "worker=1", "vector=1", and "gang=$2^0, 2^1, \ldots, 2^9$", respectively. With ℓ workers, we use the notations T_ℓ, $\mathcal{E}_\ell$ and IT_ℓ to represent the total computation time (in seconds), the speedup and the iteration steps, respectively, where $\mathcal{E}_\ell = T_1/(\ell T_\ell)$. Let $x^{(0)} = e$ and $\epsilon = 10^{-6}$. With SOR splitting, Method 2 is compared with Method 1, that is the RSMSMSOR versus the SMSMSOR. The positive diagonal matrices Ω are chosen as $\Omega = D_M$ in both two methods. For each case, the parameter α used in the SOR splittings is chosen as the experimentally optimal one satisfying (11), while the positive diagonal matrix Π is always set as I, which can satisfy Assumption **(II)** of Theorem 6.

The numerical results of three examples are shown in Tables 1, 2 and 3, respectively. As Example 1 is a ill-conditioned problem that would cost a lot of computational time, the size is set to $m = 128$ and $m = 256$, while the sizes of Examples 2 and 3, well-conditioned, are set to be $m = 1024$ and $m = 2048$. For the choice of the relaxation parameters, $\lambda^{(k,i)}$ is set to be $\lambda^{(k,i)} \equiv \lambda$, where λ is the optimal parameter obtained experimentally, where $\lambda = 1.9, 1.2, 1.5$ for Examples 1, 2, 3, respectively.

In terms of iteration steps, for Example 2 and Example 3, when the $\ell < 128$, the iteration steps of each case are relatively stable and not significantly different, while when $\ell \geq 128$ there is a significant increase in iteration steps, which is related to the number of workers exceeding the maximum parallel efficiency limit of 80. On the other hand, for the ill-conditioned problem Example 1, the threshold for significant changes in iteration steps is reduced to $\ell = 32$. From the comparisons between the SMSMSOR and RSMSMSOR, it can be seen that with relaxation technology, the iteration steps of the RSMSMSOR are significantly less than the SMSMSOR. The specific comparisons of iteration steps are shown in Fig. 1.

In terms of computational time, as ℓ increases, the computational time becomes less and less due to the effects of parallel computing. By the comparisons between the SMSMSOR and RSMSMSOR, it can be seen that the computational time of the RSMSMSOR is significantly less than those of the SMSMSOR. The percentages of the time saved by the RSMSMSOR compared to the SMSMSOR are also shown in Tables 1, 2, 3, where "SAVE" is defined by

$$SAVE = \frac{T_\ell^{SMSMSOR} - T_\ell^{RSMSMSOR}}{T_\ell^{SMSMSOR}} \times 100\%.$$

The specific comparisons of the computational time are shown in Fig. 2, which implies that the introduction of relaxation technology has a significant acceleration effect on the SMSMSOR.

In terms of parallel efficiency, when $\ell < 128$, all the values of the speedup exceed 0.9, while when $\ell \geq 128$, the low bound of the values of the speedup is larger than 0.6, which is also a satisfied result. This indicates that the numerical experiments based on the OpenACC framework fully utilize the hardware resources of GPU devices during the implementation process to achieve high parallel efficiency.

5 Conclusions

By introducing the relaxation technology, a relaxed two-step modulus-based synchronous multisplitting iteration method has been constructed for solving the LCP. We also provide the convergence analysis and demonstrate the selection range of the relaxation parameter. Parallelly numerical experiments based on OpenACC framework have showed that the proposed method can effectively accelerate the existing methods. Specially, for different examples, the

Table 1 Numerical results of Example 1

m	Method	ℓ	1	2	4	8	16	32	64	128	256	512
128	1	T_ℓ	768.17	385.55	196.01	98.78	50.15	25.92	13.24	7.42	4.04	2.26
		$\mathcal{E}_\ell$	1.00	1.00	0.98	0.97	0.96	0.93	0.91	0.81	0.74	0.66
		IT_ℓ	13971	14015	14054	14135	14300	14628	15305	16298	16338	16379
	2	T_ℓ	404.04	202.90	103.17	51.86	26.42	13.52	6.91	3.82	2.10	1.16
		$\mathcal{E}_\ell$	1.00	1.00	0.98	0.97	0.96	0.93	0.91	0.83	0.75	0.68
		IT_ℓ	7351	7374	7394	7437	7524	7697	8053	8575	8596	8618
		SAVE	47.40%	47.37%	47.36%	47.50 %	47.32%	47.85%	47.82%	48.56 %	48.07 %	48.76%
256	1	T_ℓ	14293.92	7171.36	3636.85	1833.36	926.72	472.67	245.62	133.78	76.75	45.10
		$\mathcal{E}_\ell$	1.00	1.00	0.98	0.97	0.96	0.95	0.91	0.83	0.73	0.62
		IT_ℓ	55203	55286	55366	55529	55854	56504	57803	60479	64403	64481
	2	T_ℓ	7527.18	3771.87	1911.73	963.56	486.84	248.31	129.12	70.13	40.47	23.84
		$\mathcal{E}_\ell$	1.00	1.00	0.98	0.98	0.97	0.95	0.91	0.84	0.73	0.62
		IT_ℓ	29052	29096	29138	29223	29394	29737	30420	31828	33894	33935
		SAVE	47.34%	47.40%	47.43%	47.44%	47.47%	47.47%	47.43%	47.58%	47.27 %	47.14%

Table 2 Numerical results of Example 2

m	Method	ℓ	1	2	4	8	16	32	64	128	256	512
1024	1	T_ℓ	2247.51	1124.97	574.79	289.78	145.73	74.42	38.96	21.14	12.00	7.04
		$\mathcal{E}_\ell$	1.00	1.00	0.98	0.97	0.96	0.94	0.90	0.83	0.73	0.62
		IT_ℓ	501	501	501	502	503	506	511	521	542	575
	2	T_ℓ	1791.56	896.45	457.49	230.19	115.58	59.35	30.96	16.79	9.59	5.55
		$\mathcal{E}_\ell$	1.00	1.00	0.98	0.97	0.97	0.94	0.90	0.83	0.73	0.63
		IT_ℓ	399	399	399	399	400	402	406	414	429	453
		SAVE	20.29 %	20.31%	20.41 %	20.57 %	20.69 %	20.24 %	20.54 %	20.56%	20.13%	21.21%
2048	1	T_ℓ	17249.83	8633.55	4412.29	2221.38	1134.52	573.01	298.90	159.16	88.59	51.95
		$\mathcal{E}_\ell$	1.00	1.00	0.98	0.97	0.95	0.94	0.90	0.85	0.76	0.65
		IT_ℓ	951	951	951	952	953	956	961	972	993	1036
	2	T_ℓ	13942.20	6979.36	3563.01	1793.46	916.70	463.70	240.57	129.51	71.62	42.69
		$\mathcal{E}_\ell$	1.00	1.00	0.98	0.97	0.95	0.94	0.91	0.84	0.76	0.64
		IT_ℓ	768	768	768	769	770	772	776	784	801	833
		SAVE	19.17%	19.16%	19.25%	19.26%	19.20%	19.08%	19.52%	18.63%	19.16%	17.83%

Table 3 Numerical results of Example 3

m	Method	ℓ	1	2	4	8	16	32	64	128	256	512
1024	1	T_ℓ	7433.60	3727.44	1885.94	949.50	480.04	247.39	125.73	68.94	39.12	21.73
		$\mathcal{E}_\ell$	1.00	1.00	0.99	0.98	0.97	0.94	0.92	0.84	0.74	0.67
		IT_ℓ	1622	1622	1623	1624	1627	1633	1644	1667	1711	1805
	2	T_ℓ	4814.10	2419.07	1221.54	614.11	309.79	159.78	81.63	44.53	24.97	14.23
		$\mathcal{E}_\ell$	1.00	1.00	0.99	0.98	0.97	0.94	0.92	0.84	0.75	0.66
		IT_ℓ	1052	1052	1053	1054	1055	1059	1067	1081	1111	1173
		SAVE	35.24%	35.10%	35.23%	35.32%	35.47 %	35.42 %	35.07%	35.41%	36.17 %	34.54%
2048	1	T_ℓ	58191.61	29212.15	14758.99	7421.51	3787.78	1926.10	990.97	530.23	289.93	165.92
		$\mathcal{E}_\ell$	1.00	1.00	0.99	0.98	0.96	0.94	0.92	0.86	0.78	0.68
		IT_ℓ	3127	3127	3128	3129	3132	3137	3148	3170	3214	3302
	2	T_ℓ	38032.54	19087.17	9637.32	4848.40	2472.27	1260.64	645.80	349.70	188.51	107.42
		$\mathcal{E}_\ell$	1.00	1.00	0.99	0.98	0.96	0.94	0.92	0.85	0.79	0.69
		IT_ℓ	2043	2043	2044	2045	2046	2050	2057	2072	2101	2159
		SAVE	34.64%	34.66%	34.70 %	34.67%	34.73%	34.55%	34.83%	34.05%	34.98%	35.26%

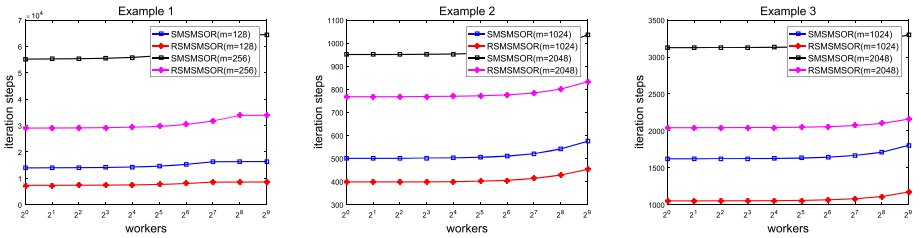

Fig. 1 The comparisons of the iterations steps

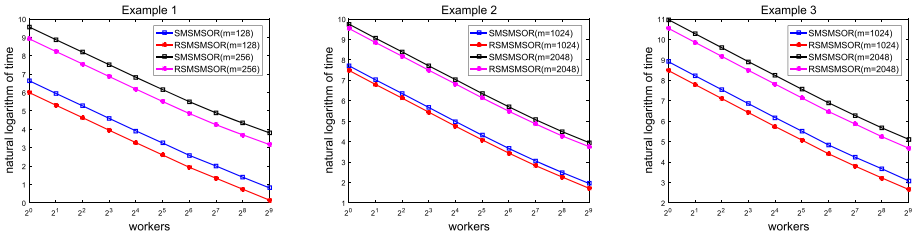

Fig. 2 The comparisons of the computational time

computational time can be reduced by a range from 17% to 48%. However, how to select the optimal relaxation parameter $\lambda^{(k,i)}$ is still a challenging problem, which is a very meaningful research work in the future.

Acknowledgements The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported by Scientific Computing Research Innovation Team of Guangdong Province (No. 2021KCXTD052), Science and Technology Development Fund, Macau SAR (No. 0151/2022/A), University of Macau (No. MYRG2022-00076-FST, MYRG-GRG2023-00037-FST-UMDF), Guangdong Key Construction Discipline Research Capacity Enhancement Project (No. 2022ZDJS049), Characteristic innovation project of Guangdong Provincial Department of Education (No. 2023KTSCX195) and Technology Planning Project of Shaoguan (No. 230330108034184).

Data availability statement Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Bai Z-Z (1999) On the convergence of the multisplitting methods for the linear complementarity problem. SIAM J. Matrix Anal. Appl. 21:67–78

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

Bai Z-Z A two-step matrix splitting iteration paradigm based on one single splitting for solving systems of linear equations, Numer. Linear Algebra Appl. (In press) https://doi.org/10.1002/nla.2510

Bai Z-Z, Zhang L-L (2013) Modulus-based synchronous two-stage multisplitting iteration methods for linear complementarity problems. Numer. Algorithms 62:59–77

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

Berman A, Plemmons RJ (1994) Nonnegative Matrix in the Mathematical Sciences. SIAM Publisher, Philadelphia

Cottle R-W, Pang J-S (1992) Stone RE. Academic, The Linear Complementarity Problem. SanDiego
Cryer C (1971) The solution of a quadratic programming using systematic overrelaxation. SIAM J. Control Opt. 9:385–392
Dong J-L, Jiang M-Q (2009) A modified modulus method for symmetric positive-definite linear complementarity problems. Numer. Linear Algebra Appl. 16:129–143
Fang X-M (2022) The convergence of the modulus-based Jacobi (MJ) iteration method for solving horizontal linear complementarity problems. Comp. Appl. Math. 41:134
Fang X-M, Gu Z, Qiao Z-J (2023) Convergence of the two-point modulus-based matrix splitting iteration method. J. Appl. Anal. Comput. 13(5):2504–2521
Fang X-M, Zhu Z-W (2019) The modulus-based matrix double splitting iteration method for linear complementarity problems. Comput. Math. Appl. 78:3633–3643
Frommer A, Szyld DB (1992) H-splittings and two-stage iterative methods. Numer. Math. 63:345–356
Frommer A, Mayer G (1989) Convergence of relaxed parallel multisplitting methods. Linear Algebra Appl. 119:141–152
Huang B-H, Ma C-F (2018) Accelerated modulus-based matrix splitting iteration method for a class of nonlinear complementarity problems. Comp. Appl. Math. 37:3053–3076
Hu J-G (1982) Estimates of $\|B^{-1}C\|_\infty$ and their applications. Math. Numer. Sin. 4:272–282
Ke Y-F, Ma C-F (2014) On the convergence analysis of two-step modulus-based matrix splitting iteration method for linear complementarity problems. Appl. Math. Comput. 243:413–418
Ke Y-F, Ma C-F, Zhang H (2018) The relaxation modulus-based matrix splitting iteration methods for circular cone nonlinear complementarity problems. Comp. Appl. Math. 37:6795–6820
Murty KG (1988) Linear Complementarity, Linear and Nonlinear Programming. Heldermann Verlag, Berlin
NVIDIA HPC SDK Version 23.5 Documentation, URL https://docs.nvidia.com/hpc-sdk/index.html
Ren H, Wang X, Tang X-B, Wang T (2019) The general two-sweep modulus-based matrix splitting iteration method for solving linear complementarity problems. Comput. Math. Appl. 77:1071–1081
Song Y-L, Zheng H, Lu X-P, Vong S (1882) A two-step iteration method for vertical linear complementarity problems. Symmetry 14(9)(2022)
Wu M-H, Li C-L (2019) A preconditioned modulus-based matrix multisplitting block iteration method for the linear complementarity problems with Toeplitz matrix. Calcolo 56:13
Wu X-P, Peng X-F, Li W (2018) A preconditioned general modulus-based matrix splitting iteration method for linear complementarity problems of H-matrices. Numer Algorithms 79:1131–1146
Xu W-W, Zhu L, Peng X-F, Liu H, Yin J-F (2020) A class of modified modulus-based synchronous multisplitting iteration methods for linear complementarity problems. Numer. Algorithms 85:1–21
Zhang L-L (2011) Two-step modulus-based matrix splitting iteration for linear complementarity problems. Numer. Algorithms 57:83–99
Zhang L-L (2014) Two-stage multisplitting iteration method using modulus-based matrix splitting as inner iteration for linear complementarity problems. J. Opt. Theory Appl. 160:189–203
Zhang L-L (2015) Two-step modulus-based synchronous multisplitting iteration methods for linear complementarity problems. J. Comput. Math. 33:100–112
Zhang Y-X, Zheng H, Lu X-P, Vong S (2023) A two-step parallel iteration method for large sparse horizontal linear complementarity problems. Appl. Math. Comput. 438:127609
Zheng H, Luo L, Li S-Y (2021) A two-step iteration method for the horizontal nonlinear complementarity problem. JPN. J. Ind. Appl. Math. 38:1023–1036
Zheng H, Vong S (2021) A two-step modulus-based matrix splitting iteration method for horizontal linear complementarity problems. Numer. Algorithms 86:1791–1810
Zheng H, Vong S, Liu L (2019) The relaxation modulus-based matrix splitting iteration method for solving a class of nonlinear complementarity problems. Intern. J. Comput. Math. 96(8):1648–1667
Zheng H, Li W, Vong S (2017) A relaxation modulus-based matrix splitting iteration method for solving linear complementarity problems. Numer. Algorithms 74:137–152
Zheng H, Lu X-P, Vong S A two-step modulus-based matrix splitting iteration method without auxiliary variable for solving vertical linear complementarity problems, Commun. Appl. Math. Comput. (In press) https://doi.org/10.1007/s42967-023-00280-y