Numer Algor (2017) 74:137–152
DOI 10.1007/s11075-016-0142-7

ORIGINAL PAPER

A relaxation modulus-based matrix splitting iteration method for solving linear complementarity problems

Hua Zheng[1] · **Wen Li**[2,3] · **Seakweng Vong**[4]

Received: 30 March 2015 / Accepted: 6 May 2016 / Published online: 18 May 2016

Abstract In this paper, a relaxation modulus-based matrix splitting iteration method is established, which covers the known general modulus-based matrix splitting iteration methods. The convergence analysis and the strategy of the choice of the parameters are given. Numerical examples show that the proposed methods are efficient and accelerate the convergence performance with less iteration steps and CPU times.

1 Introduction

The linear complementarity problem (LCP(q, A)) consists of finding vectors $z \in \mathbf{R}^n$ such that

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

where $A \in \mathbf{R}^{n\times n}$ and $q \in \mathbf{R}^n$.

✉ Hua Zheng
hzhengjlfdu@gmail.com

✉ Wen Li
liwen@scnu.edu.cn

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

2 School of Mathematical Sciences, South China Normal University, Guangzhou, People's Republic of China

3 Guangdong Engineering Technical Research Center for Data Sciences, South China Normal University, Guangzhou, People's Republic of China

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

The LCP(q, A) has many applications, e.g., the economies with institutional restrictions upon prices, the linear and quadratic programming, the free boundary problems, and the optimal stopping in Markov chain; see [8, 16] for details.

Recently, many articles gave some solvers of LCP(q, A) based on the modulus iteration method presented by van Bokhoven in [18]. In particular, Bai presented a modulus-based matrix splitting method in [3] which not only includes the modified modulus method [9] and the nonstationary extrapolated modulus algorithms [12] as its special cases but also yields a series of iteration methods, such as modulus-based Jacobi, Gauss-Seidel, SOR, and AOR iteration methods, which were extended to more general cases by Li [14]. In addition, Hadjidimos et al. [11] and Zhang [19] proposed scaled extrapolated modulus algorithms and two-step modulus-based matrix splitting iteration methods, respectively. Moreover, the modulus-based synchronous multisplitting iteration methods and modulus-based synchronous two-stage multisplitting iteration methods were established in [6, 7], while in [21], Zheng and Yin proposed a class of accelerated modulus-based matrix splitting iteration methods, which can be discussed in [15]. The global convergence conditions are discussed when the system matrix is either a positive definite matrix or an H_+-matrix; see the references mentioned above for details.

In this paper, we propose a relaxation modulus-based matrix splitting iteration method for solving LCP, and give its theoretical analysis. The main idea of the proposed method is to combine a relaxation technique with the fixed-point iteration formula for updating the iteration vector. Hence, the main contributions of this paper are given below:

- Input a parameter matrix into the modulus-based matrix splitting iteration method given in [14], which can accelerate the convergence performance of the method;
- Propose the strategy for choosing the parameter matrix in each iteration step.

Numerical experiments are given to show that the proposed method is efficient.

In order to present our method, first we introduce some notations and definitions.

For two $m \times n$ real matrices $B = (b_{ij})$ and $C = (c_{ij})$, the order $B \geq (>)C$ means $b_{ij} \geq (>)c_{ij}$ for any i and j. Let e be an $n \times 1$ vector whose elements are all equal to 1, $A = (a_{ij}) \in \mathbf{R}^{n\times n}$ and let $A = D_A - L_A - U_A = D_A - B_A$, where D_A, $-L_A$, and $-U_A$ are the diagonal, the strictly lower-triangular, and the strictly upper-triangular matrices of A, respectively. By $|A|$, we denote $|A| = (|a_{ij}|)$ and the comparison matrix of A is $\langle A\rangle = (\langle a_{ij}\rangle)$, defined by $\langle a_{ij}\rangle = |a_{ij}|$ if $i = j$ and $\langle a_{ij}\rangle = -|a_{ij}|$ if $i \neq j$. The matrix A is called (e.g., see [2]) a Z-matrix if all of its off-diagonal entries are non-positive, an M-matrix if it is a Z-matrix with $A^{-1} \geq 0$, and an H-matrix if its comparison matrix $\langle A\rangle$ is an M-matrix. Specially, an H-matrix with positive diagonal entries is called an H_+-matrix (e.g., see [4]). The splitting $A = M - N$ is called an M-splitting if M is a nonsingular M-matrix and $N \geq 0$; an H-splitting if $\langle M\rangle - |N|$ is an M-matrix; and an H-compatible splitting if $\langle A\rangle = \langle M\rangle - |N|$ (e.g., see [20]). Note that if $A = M - N$ is an M-splitting and A is a nonsingular M-matrix, then $\rho(M^{-1}N) < 1$, and an H-compatible splitting of an H-matrix is an H-splitting, but not vice versa.

The rest of this paper is organized as follows. In Section 2, we propose the relaxation modulus-based matrix splitting iteration method for solving LCP(q, A). In Section 3, we give the convergence analysis of the proposed method. The discussion of the choice of the parameter is given in Section 4. In Section 5, we give some numerical examples. A conclusion remark is given in the final section.

2 The relaxation modulus-based matrix splitting iteration method

Let $A = M - N$ be a splitting of A, and Γ, Ω, Ω_1, and Ω_2 be nonnegative diagonal matrices with $\Omega = \Omega_1 + \Omega_2$. It is known from [3] that the linear complementarity problem LCP(q, A) is completely equivalent to solving the fixed-point equations:

$$(\Omega_2 + M\Gamma)x = (N\Gamma - \Omega_1)x + (\Omega - A\Gamma)|x| - q. \tag{1}$$

In particular, taking

$$\Omega = \Omega_2, \Omega_1 = 0, \text{and } \Gamma = \frac{1}{\gamma}I,$$

where γ is a positive constant, (1) is simplified as

$$(\Omega + M)x = Nx + (\Omega - A)|x| - \gamma q, \tag{2}$$

which leads to the modulus-based matrix splitting iteration method:

Method 2.1 [3] *Let $A = M-N$ be a splitting of A. Given an initial vector $x^{(0)} \in \mathbf{R}^n$, for $k = 1, 2, \cdots$ until the iteration sequence $\{z^{(k)}\}_{k=1}^{+\infty} \subset \mathbf{R}^n$ is convergent, compute $x^{(k)} \in \mathbf{R}^n$ by solving the linear system*

$$(\Omega + M)x^{(k)} = Nx^{(k-1)} + (\Omega - A)|x^{(k-1)}| - \gamma q, \tag{3}$$

and set

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

Here, Ω is an $n \times n$ positive diagonal matrix and γ is a positive constant.

By different choices of the splitting $A = M - N$, from the equation (2), one may deduce a series of modulus-based matrix splitting iteration methods; see [3] for details.

Taking $\Omega_2 = \Omega$ and $\Gamma = \Omega_1$ in (1), Li [14] proposed a general modulus-based matrix splitting iteration method as follows:

Method 2.2 *For any given positive diagonal matrices Ω_1 and Ω_2, let $A\Omega_1 = M_{\Omega_1} - N_{\Omega_1}$ be a splitting of the matrix $A\Omega_1 \in \mathbf{R}^{n\times n}$. Given an initial vector $x^{(0)} \in R^n$, compute $x^{(k)} \in R^n$ by solving the linear system*

$$(\Omega_2 + M_{\Omega_1})x^{(k)} = N_{\Omega_1}x^{(k-1)} + (\Omega_2 - A\Omega_1)|x^{(k-1)}| - q. \tag{4}$$

Then set

$$z^{(k)} = \Omega_1(|x^{(k)}| + x^{(k)})$$

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

At the kth step in (4), consider mixing the new approach vector and the old approach vector before the next iteration, by introducing a nonsingular parameter matrix $P \in \mathbf{R}^{n\times n}$, we compute $x^{(k)}$ by solving the next linear system:

$$\begin{cases} (\Omega_2 + M_{\Omega_1})x^{(k-\frac{1}{2})} = N_{\Omega_1}x^{(k-1)} + (\Omega_2 - A\Omega_1)|x^{(k-1)}| - q, \\ x^{(k)} = (I-P)x^{(k-1)} + Px^{(k-\frac{1}{2})}. \end{cases} \tag{5}$$

In fact, (5) can be written as

$$x^{(k)} = (I-P)x^{(k-1)} + P(\Omega_2 + M_{\Omega_1})^{-1}[N_{\Omega_1}x^{(k-1)} + (\Omega_2 - A\Omega_1)|x^{(k-1)}| - q]. \tag{6}$$

Remark 2.3 The parameter matrix P can vary for a given k during the iteration. However, it is difficult to choose such a parameter matrix P so that the convergence rate of (6) is optimal.

Now, we consider the special cases, e.g.,

$$P = \theta I,$$

where $\theta \in \mathbf{R}\backslash\{0\}$ and θ can be vary with respect to k. Then based on (5), we have

Method 2.4 (**The relaxation modulus-based matrix splitting iteration method**) *For any given positive diagonal matrices Ω_1 and Ω_2, let $A\Omega_1 = M_{\Omega_1} - N_{\Omega_1}$ be a splitting of the matrix $A\Omega_1 \in \mathbf{R}^{n\times n}$ and $\theta \in \mathbf{R}\backslash\{0\}$. Given an initial vector $x^{(0)} \in \mathbf{R}^n$, for $k = 1, 2, \cdots$ until the iteration sequence $\{z^{(k)}\}_{k=1}^{+\infty} \subset \mathbf{R}^n$ is convergent, compute $x^{(k)} \in \mathbf{R}^n$ by solving the linear system*

$$\begin{cases} (\Omega_2 + M_{\Omega_1})x^{(k-\frac{1}{2})} = N_{\Omega_1}x^{(k-1)} + (\Omega_2 - A\Omega_1)|x^{(k-1)}| - q, \\ x^{(k)} = (1-\theta)x^{(k-1)} + \theta x^{(k-\frac{1}{2})} \end{cases} \tag{7}$$

and set

$$z^{(k)} = \Omega_1(|x^{(k)}| + x^{(k)}).$$

Method 2.4 provides a general framework of modulus-based matrix splitting iteration methods for solving LCP(q, A), from which a series of relaxation modulus-based matrix splitting iteration methods can be derived. For example, when taking

$$M_{\Omega_1} = \frac{1}{\alpha}(D_{A\Omega_1} - \beta L_{A\Omega_1}), N_{\Omega_1} = \frac{1}{\alpha}[(1-\alpha)D_{A\Omega_1} + (\alpha - \beta)L_{A\Omega_1} + \alpha U_{A\Omega_1}],$$

we can obtain the relaxation modulus-based accelerated overrelaxation (RMAOR) iteration method, which extends the relaxation modulus-based successive overrelaxation (RMSOR) iteration method, the relaxation modulus-based Gauss-Seidel (RMGS) iteration method, and the relaxation modulus-based Jacobi (RMJ) iteration method when $\alpha = \beta$, $\alpha = \beta = 1$ and $\alpha = 1, \beta = 0$, respectively.

Remark 2.5 If we take $\theta = 1$, Method 2.4 reduces to Methods 2.2. Hence, the proposed method provides a more general framework for the existing modulus-based methods.

3 Convergence analysis

In this section, the convergence analysis for Method 2.4 is presented when the system matrix A of LCP(q, A) is an H_+-matrix.

The following lemmas are useful in the sequel of the paper.

Lemma 3.6 [13] *Let $B \in \mathbf{R}^{n\times n}$ be a strictly diagonal dominant matrix. Then $\forall C \in \mathbf{R}^{n\times n}$,*

$$||B^{-1}C||_\infty \le \max_{1\le i\le n} \frac{(|C|e)_i}{(\langle B\rangle e)_i},$$

holds, where $e = (1, 1, \cdots, 1)^T$.

Lemma 3.7 [10] *Let A be an H-matrix. Then $|A^{-1}| \le \langle A\rangle^{-1}$.*

Lemma 3.8 [2] *Let A be a Z-matrix with positive diagonal entries. Then A is an M-matrix if and only if there exists a positive diagonal matrix D, such that AD is a strictly diagonal dominant matrix with positive diagonal entries.*

Lemma 3.9 [2] *Let A, B be two Z-matrices, A be an M-matrix, and $B \ge A$. Then B is an M-matrix.*

Lemma 3.10 *Let $A \in \mathbf{R}^{n\times n}$ be an H_+-matrix and $A\Omega_1 = M_{\Omega_1} - N_{\Omega_1}$ be an H-splitting of $A\Omega_1$. Then there exists a positive diagonal matrix D such that $(\langle M_{\Omega_1}\rangle - |N_{\Omega_1}|)D$ and $(\Omega_2 + \langle M_{\Omega_1}\rangle)D$ are two strictly diagonal dominant matrices, and*

$$[(\langle A\Omega_1\rangle + \langle M_{\Omega_1}\rangle - |N_{\Omega_1}|)De]_i > 0, i = 1, 2, \cdots, n. \tag{8}$$

Proof Since $A\Omega_1 = M_{\Omega_1} - N_{\Omega_1}$ is an H-splitting of $A\Omega_1$, $\langle M_{\Omega_1}\rangle - |N_{\Omega_1}|$ and $\langle A\Omega_1\rangle$ are two M-matrices. From Lemma 3.8, there exists a positive diagonal matrix D, such that $(\langle M_{\Omega_1}\rangle - |N_{\Omega_1}|)D$ is a strictly diagonal dominant matrix. For $\langle M_{\Omega_1}\rangle \ge \langle M_{\Omega_1}\rangle - |N_{\Omega_1}|$, $(\Omega_2 + \langle M_{\Omega_1}\rangle)D$ is strictly diagonal dominant too. Since

$$\begin{aligned}
&\langle A\Omega_1\rangle + \langle M_{\Omega_1}\rangle - |N_{\Omega_1}| \\
&\quad = |D_{M_{\Omega_1}} - D_{N_{\Omega_1}}| - |B_{M_{\Omega_1}} - B_{N_{\Omega_1}}| + \langle M_{\Omega_1}\rangle - |N_{\Omega_1}| \\
&\quad \ge |D_{M_{\Omega_1}}| - |D_{N_{\Omega_1}}| - |B_{M_{\Omega_1}}| - |B_{N_{\Omega_1}}| + \langle M_{\Omega_1}\rangle - |N_{\Omega_1}| \\
&\quad = 2(\langle M_{\Omega_1}\rangle - |N_{\Omega_1}|),
\end{aligned}$$

$(\langle A\Omega_1\rangle + \langle M_{\Omega_1}\rangle - |N_{\Omega_1}|)D$ is also a strictly diagonal dominant matrix, and then (8) holds. □

Let (z^*, r^*) be the solution of LCP(q, A). By (6) we can easily get that $x^* = \frac{1}{2}(\Omega_1^{-1}z^* - \Omega_2^{-1}r^*)$ satisfies

$$x^* = (I - P)x^* + P(\Omega_2 + M_{\Omega_1})^{-1}[N_{\Omega_1}x^* + (\Omega_2 - A\Omega_1)|x^*| - q]. \tag{9}$$

By Lemma 3.7, we have

$$0 \le |(\Omega_2 + M_{\Omega_1})^{-1}| \le (\Omega_2 + \langle M_{\Omega_1}\rangle)^{-1}.$$

Subtracting (9) from (6) gives

$$
\begin{aligned}
&|x^{(k)} - x^*| \\
&\quad \le P|(\Omega_2 + M_{\Omega_1})^{-1}|[|(\Omega_2 + M_{\Omega_1})(P^{-1} - I) + N_{\Omega_1}| \\
&\qquad +|\Omega_2 - A\Omega_1|]|x^{(k-1)} - x^*| \\
&\quad \le P(\Omega_2 + \langle M_{\Omega_1}\rangle)^{-1}[|(\Omega_2 + M_{\Omega_1})(P^{-1} - I) + N_{\Omega_1}| \\
&\qquad +|\Omega_2 - A\Omega_1|]|x^{(k-1)} - x^*| \\
&\quad = \mathcal{L}|x^{(k-1)} - x^*|,
\end{aligned}
$$

where

$$\mathcal{L} = P\tilde{M}^{-1}\tilde{N}P^{-1}, \tilde{M} = \Omega_2 + \langle M_{\Omega_1}\rangle, \tag{10}$$

$$\tilde{N} = |(\Omega_2 + M_{\Omega_1})(I - P) + N_{\Omega_1}P| + |\Omega_2 - A\Omega_1|P. \tag{11}$$

Lemma 3.11 *With the same assumptions and notations as in Lemma 3.10, let P be positive diagonal matrices and let matrices $\tilde{M}$ and $\tilde{N}$ be given by (10) and (11) respectively. Then $\rho(\mathcal{L}) < 1$ if*

$$
\begin{cases}
(\Omega_2 e)_i \ge (D_{A\Omega_1}e)_i, \\
\{[2\Omega_2 + 2|D_{M_{\Omega_1}}| - (2\Omega_2 + |M_{\Omega_1}| + |N_{\Omega_1}| - \langle A\Omega_1\rangle)P]De\}_i > 0, & \textit{when } (Pe)_i \ge 1, \\
\{[(\langle A\Omega_1\rangle + |M_{\Omega_1}| - |N_{\Omega_1}|)P - 2|B_{M_{\Omega_1}}|]De\}_i > 0, & \textit{when } 0 < (Pe)_i < 1,
\end{cases} \tag{12}
$$

or

$$
\begin{cases}
[\frac{1}{2}(|A\Omega_1| - \langle M_{\Omega_1}\rangle + |N_{\Omega_1}|)De]_i < (\Omega_2 De)_i < (D_{A\Omega_1}De)_i, \\
\{[2\Omega_2 + 2|D_{M_{\Omega_1}}| - (|M_{\Omega_1}| + |N_{\Omega_1}| + |A\Omega_1|)P]De\}_i > 0, & \textit{when } (Pe)_i \ge 1, \\
\{[(2\Omega_2 - |A\Omega_1| + |M_{\Omega_1}| - |N_{\Omega_1}|)P - 2|B_{M_{\Omega_1}}|]De\}_i > 0, & \textit{when } 0 < (Pe)_i < 1,
\end{cases} \tag{13}
$$

where D is given by Lemma 3.10, $i = 1, 2, \cdots, n$.

Proof From Lemma 3.6 and Lemma 3.10, we have

$$||D^{-1}P^{-1}\mathcal{L}PD||_\infty = ||(\tilde{M}D)^{-1}(\tilde{N}D)||_\infty \le \max_{1\le i\le n} \frac{(\tilde{N}De)_i}{(\tilde{M}De)_i}. \tag{14}$$

Next, we show that

$$(\tilde{M}De)_i > (\tilde{N}De)_i, \tag{15}$$

for any $1 \le i \le n$.

By (11), we have

$$D_{\tilde{N}} \le (\Omega_2 + D_{M_{\Omega_1}})|P - I| + |D_{N_{\Omega_1}}|P + |\Omega_2 - D_{A\Omega_1}|P, \tag{16}$$

and

$$
\begin{aligned}
B_{\tilde{N}} &= -|B_{M_{\Omega_1}}(P - I) + B_{N_{\Omega_1}}P| - |B_{A\Omega_1}|P \\
&\ge -|B_{M_{\Omega_1}}||P - I| - |B_{N_{\Omega_1}}|P - |B_{A\Omega_1}|P.
\end{aligned} \tag{17}
$$

By (10), (16) and (17), we have

$$\begin{aligned}(\tilde{M}De)_i - (\tilde{N}De)_i &= [(\tilde{M} - D_{\tilde{N}} + B_{\tilde{N}})De]_i \\ &\geq \{[\Omega_2 + \langle M_{\Omega_1}\rangle - (\Omega_2 + D_{M_{\Omega_1}})|P - I| - |D_{N_{\Omega_1}}|P \\ &\quad -|\Omega_2 - D_{A\Omega_1}|P - |B_{M_{\Omega_1}}||P - I| - |B_{N_{\Omega_1}}|P - |B_{A\Omega_1}|P]De\}_i. \quad (18)\end{aligned}$$

Case 1: If $(\Omega_2 e)_i \geq (D_{A\Omega_1}e)_i$, then by (18) we have

$$(\tilde{M}De)_i - (\tilde{N}De)_i \geq \begin{cases} \{[2\Omega_2 + 2|D_{M_{\Omega_1}}| - (2\Omega_2 + |M_{\Omega_1}| + |N_{\Omega_1}| - \langle A\Omega_1\rangle)P]De\}_i > 0, & \text{when } (Pe)_i \geq 1; \\ \{[(\langle A\Omega_1\rangle + |M_{\Omega_1}| - |N_{\Omega_1}|)P - 2|B_{M_{\Omega_1}}|]De\}_i > 0, & \text{when } 0 < (Pe)_i < 1. \end{cases}$$

Case 2: If $[\frac{1}{2}(|A\Omega_1| - \langle M_{\Omega_1}\rangle + |N_{\Omega_1}|)De]_i < (\Omega_2 De)_i < (D_{A\Omega_1}De)_i$, then one may derive from (18) that

$$(\tilde{M}De)_i - (\tilde{N}De)_i \geq \begin{cases} \{[2\Omega_2 + 2|D_{M_{\Omega_1}}| - (|M_{\Omega_1}| + |N_{\Omega_1}| + |A\Omega_1|)P]De\}_i > 0, & \text{when } (Pe)_i \geq 1; \\ \{[(2\Omega_2 - |A\Omega_1| + |M_{\Omega_1}| - |N_{\Omega_1}|)P - 2|B_{M_{\Omega_1}}|]De\}_i > 0, & \text{when } 0 < (Pe)_i < 1. \end{cases}$$

By Cases 1 and 2, it is easy to see that the assertion (15) holds, which together with (14) gives $\rho(\mathcal{L}) < 1$. This proves the lemma.

□

Furthermore, if $A\Omega_1 = M_{\Omega_1} - N_{\Omega_1}$ is an H-compatible splitting, then $\langle A\Omega_1\rangle = \langle M_{\Omega_1}\rangle - |N_{\Omega_1}|$. Hence Lemma 3.11 can be simplified to the following results.

Corollary 3.12 *With the same assumptions and notations as in Lemma 3.11, if* $A\Omega_1 = M_{\Omega_1} - N_{\Omega_1}$ *is an* H*-compatible splitting, then* $\rho(\mathcal{L}) < 1$ *if*

$$\begin{cases} (\Omega_2 e)_i \geq (D_{A\Omega_1}e)_i, \\ \{[\Omega_2 + |D_{M_{\Omega_1}}| - (\Omega_2 + |D_{M_{\Omega_1}}| - \langle A\Omega_1\rangle)P]De\}_i > 0 & \text{when } (Pe)_i \geq 1, \\ \{[(|D_{M_{\Omega_1}}| - |N_{\Omega_1}|)P - |B_{M_{\Omega_1}}|]De\}_i > 0 & \text{when } 0 < (Pe)_i < 1, \end{cases}$$

and

$$\begin{cases} [|B_{A\Omega_1}|De]_i < (\Omega_2 De)_i < (D_{A\Omega_1}De)_i, \\ \{[\Omega_2 + |D_{M_{\Omega_1}}| - (|D_{M_{\Omega_1}}| + |B_{A\Omega_1}|)P]De\}_i > 0 & \text{when } (Pe)_i \geq 1, \\ \{[(\Omega_2 - |B_{A\Omega_1}| + |B_{M_{\Omega_1}}|)P - |B_{M_{\Omega_1}}|]De\}_i > 0 & \text{when } 0 < (Pe)_i < 1. \end{cases}$$

Now, we give the convergence result for Method 2.4.

Theorem 3.13 *With the same assumptions and notations as in Lemmas 3.10 and 3.11, then for any given* $x^{(0)} \in \mathbf{R}^n$, $\{z^{(k)}\}_{k=1}^{+\infty}$, *generated by Method 2.4, converges to the exact solution* $z^* \in \mathbf{R}^n$ *of* LCP(q, A) *provided*

$$\begin{cases} (\Omega_2 e)_i \geq (D_{A\Omega_1}e)_i, \\ \delta_i^{(1)} < \theta < \delta_i^{(2)} \end{cases}, \quad (19)$$

or

$$\begin{cases} [\frac{1}{2}(|A\Omega_1| - \langle M_{\Omega_1}\rangle + |N_{\Omega_1}|)De]_i < (\Omega_2 De)_i < (D_{A\Omega_1} De)_i, \\ \delta_i^{(3)} < \theta < \delta_i^{(4)} \end{cases}, \tag{20}$$

where D is given by Lemma 3.10,

$$\delta_i^{(1)} = \frac{(2|B_{M_{\Omega_1}}|De)_i}{[(\langle A\Omega_1\rangle + |M_{\Omega_1}| - |N_{\Omega_1}|)De]_i}, \tag{21}$$

$$\delta_i^{(2)} = \frac{[(2\Omega_2 + 2|D_{M_{\Omega_1}}|)De]_i}{[(2\Omega_2 + |M_{\Omega_1}| + |N_{\Omega_1}| - \langle A\Omega_1\rangle)De]_i}, \tag{22}$$

$$\delta_i^{(3)} = \frac{(2|B_{M_{\Omega_1}}|De)_i}{[(2\Omega_2 - |A\Omega_1| + |M_{\Omega_1}| - |N_{\Omega_1}|)De]_i}, \tag{23}$$

and

$$\delta_i^{(4)} = \frac{[(2\Omega_2 + 2|D_{M_{\Omega_1}}|)De]_i}{[(|A\Omega_1| + |M_{\Omega_1}| + |N_{\Omega_1}|)De]_i}, \tag{24}$$

$i = 1, 2, \cdots, n.$

Proof If $(\Omega_2 e)_i \geq (D_{A\Omega_1} e)_i$, by (8), we have

$$\begin{aligned} &[(2\Omega_2 + 2|D_{M_{\Omega_1}}|)De]_i - [(2\Omega_2 + |M_{\Omega_1}| + |N_{\Omega_1}| - \langle A\Omega_1\rangle)De]_i \\ &\quad = [(\langle A\Omega_1\rangle + \langle M_{\Omega_1}\rangle - |N_{\Omega_1}|)De]_i > 0, \end{aligned}$$

and

$$\begin{aligned} &[(\langle A\Omega_1\rangle + |M_{\Omega_1}| - |N_{\Omega_1}|)De]_i - 2(|B_{M_{\Omega_1}}|De)_i \\ &\quad = [(\langle A\Omega_1\rangle + \langle M_{\Omega_1}\rangle - |N_{\Omega_1}|)De]_i > 0. \end{aligned}$$

Then by (21) and (22), we have $0 < \delta_i^{(1)} < 1 < \delta_i^{(2)}$.

If $[\frac{1}{2}(|A\Omega_1| - \langle M_{\Omega_1}\rangle + |N_{\Omega_1}|)De]_i < (\Omega_2 De)_i < (D_{A\Omega_1} De)_i$, we have

$$\begin{aligned} &[(2\Omega_2 + 2|D_{M_{\Omega_1}}|)De]_i - [(|A\Omega_1| + |M_{\Omega_1}| + |N_{\Omega_1}|)De]_i \\ &= [(2\Omega_2 - |A\Omega_1| + \langle M_{\Omega_1}\rangle - |N_{\Omega_1}|)De]_i > 0, \end{aligned}$$

and

$$\begin{aligned} &[(2\Omega_2 - |A\Omega_1| + |M_{\Omega_1}| - |N_{\Omega_1}|)De]_i - (2|B_{M_{\Omega_1}}|De)_i \\ &\quad = [(2\Omega_2 - |A\Omega_1| + \langle M_{\Omega_1}\rangle - |N_{\Omega_1}|)De]_i > 0. \end{aligned}$$

Then by (23) and (24), we have $0 < \delta_i^{(3)} < 1 < \delta_i^{(4)}$.

Taking $P = \theta I$ in the proof of Lemma 3.11, it is easy to see that, if the assumption (19) or (20) holds, $\rho(\mathcal{L}) < 1$, which implies that Method 2.4 converges. □

If the splitting $A\Omega_1 = M_{\Omega_1} - N_{\Omega_1}$ is assumed to be an H-compatible splitting, then the convergence condition can be simplified as follows:

Corollary 3.14 *With the same assumptions and notations as in Theorem 3.13, furthermore, if $A\Omega_1 = M_{\Omega_1} - N_{\Omega_1}$ is an H-compatible splitting, then the sufficient conditions (19)–(24) for convergence are simplified to*

$$\begin{cases} (\Omega_2 e)_i \geq (D_{A\Omega_1} e)_i, \\ \delta_i^{(1)} < \theta < \delta_i^{(2)} \end{cases} \text{and} \begin{cases} (|B_{A\Omega_1}|De)_i < (\Omega_2 De)_i < (D_{A\Omega_1} De)_i, \\ \delta_i^{(3)} < \theta < \delta_i^{(4)} \end{cases},$$

where

$$\delta_i^{(1)} = \frac{(|B_{M_{\Omega_1}}|De)_i}{[(|D_{M_{\Omega_1}}| - |N_{\Omega_1}|)De]_i}, \delta_i^{(2)} = \frac{[(\Omega_2 + |D_{M_{\Omega_1}}|)De]_i}{[(\Omega_2 + |D_{M_{\Omega_1}}| - \langle A\Omega_1 \rangle)De]_i},$$

$$\delta_i^{(3)} = \frac{(|B_{M_{\Omega_1}}|De)_i}{[(\Omega_2 - |B_{A\Omega_1}| + |B_{M_{\Omega_1}}|)De]_i}, \delta_i^{(4)} = \frac{[(\Omega_2 + |D_{M_{\Omega_1}}|)De]_i}{[(|D_{M_{\Omega_1}}| + |B_{A\Omega_1}|)De]_i},$$

$i = 1, 2, \cdots, n.$

Remark 3.15 In the applications, we can get a positive diagonal matrix D such that $(\langle M_{\Omega_1} \rangle - |N_{\Omega_1}|)D$ is a strictly diagonal dominant matrix as in [1, 5, 14].

Remark 3.16 It is known from the above discussion that (19) and (20) are the sufficient conditions for which Method 2.4 converges. In the applications, it is needed to choose a suitable θ such that Method 2.4 converges faster than Method 2.2, which will be discussed in the next section.

4 The optimal parameters

In this section, we give the strategy how to choose the parameter θ for a given k in Method 2.4.

Obviously, we have

$$x^* = (1-\theta)x^* + \theta x^*,$$

combining with (7) together yields

$$d^{(k)} = (1-\theta)d^{(k-1)} + \theta d^{(k-\frac{1}{2})}, \tag{25}$$

where

$$d^{(k-1)} = x^{(k-1)} - x^*, d^{(k)} = x^{(k)} - x^* \text{and}\, d^{(k-\frac{1}{2})} = x^{(k-\frac{1}{2})} - x^*.$$

Let

$$\varphi(\theta) = ||d^{(k)}||_2^2 - ||d^{(k-\frac{1}{2})}||_2^2.$$

Theorem 4.17 *With the same notations as above, suppose $\sigma^{(k)} \doteq (d^{(k-1)} - d^{(k-\frac{1}{2})})^T d^{(k-\frac{1}{2})} \neq 0$, then*

$$\varphi(\theta) < 0 \tag{26}$$

if

$$\sigma^{(k)} > 0 \, and \, \theta \in (1, 1 + \frac{2\sigma^{(k)}}{||d^{(k-1)} - d^{(k-\frac{1}{2})}||_2^2}) \tag{27}$$

or

$$\sigma^{(k)} < 0 \, and \, \theta \in (1 + \frac{2\sigma^{(k)}}{||d^{(k-1)} - d^{(k-\frac{1}{2})}||_2^2}, 1). \tag{28}$$

Furthermore, the minimum point of $\varphi(\theta)$*, is given by*

$$\begin{aligned} \theta_{\min} &= 1 + \frac{\sigma^{(k)}}{||d^{(k-1)} - d^{(k-\frac{1}{2})}||_2^2} \\ &= 1 + \frac{(x^{(k-\frac{1}{2})} - x^*)^T (x^{(k-1)} - x^{(k-\frac{1}{2})})}{||x^{(k-1)} - x^{(k-\frac{1}{2})}||_2^2}. \end{aligned} \tag{29}$$

Proof From (25), we have

$$\begin{aligned} \varphi(\theta) &= ||d^{(k)}||_2^2 - ||d^{(k-\frac{1}{2})}||_2^2 \\ &= ||(1-\theta)(d^{(k-1)} - d^{(k-\frac{1}{2})}) + d^{(k-\frac{1}{2})}||_2^2 - ||d^{(k-\frac{1}{2})}||_2^2 \\ &= (1-\theta)^2 ||d^{(k-1)} - d^{(k-\frac{1}{2})}||_2^2 \\ &\quad +2(1-\theta)(d^{(k-1)} - d^{(k-\frac{1}{2})})^T d^{(k-\frac{1}{2})}. \end{aligned}$$

Obviously, $\varphi(\theta)$ is a quadratic function with respect to θ. It is easy to see that, when (27) or (28) is satisfied, (26) holds.

By a straightforward computation, we get

$$\frac{\mathrm{d}\varphi(\theta)}{\mathrm{d}\theta} = -2||d^{(k-1)} - d^{(k-\frac{1}{2})}||_2^2(1-\theta) - 2\sigma^{(k)}.$$

Let $\frac{\mathrm{d}\varphi(\theta)}{\mathrm{d}\theta} = 0$. Then (29) is derived. □

Remark 4.18 Although the various entities involved depend on k in Theorem 4.17, the parameter θ is taken to be a constant in a predefined interval unless otherwise stated (see (32)–(34) below).

It is known from Theorem 4.17 that the inequality $||d^{(k)}||_2 < ||d^{(k-\frac{1}{2})}||_2$ holds under the assumption of (27) or (28). This implies that $x^{(k)}$ is closer to x^* than $x^{(k-\frac{1}{2})}$, i.e., the relaxation technique in Method 2.4 is useful. From the Contraction Mapping Theorem (see [17]), we may obtain an error bound as follows:

$$|x^{(k)} - x^*| \leq \mathcal{L}|x^{(k-1)} - x^*| \Rightarrow ||x^{(k)} - x^*|| \leq \frac{\rho^k(\mathcal{L})}{1-\rho(\mathcal{L})}||x^{(1)} - x^{(0)}||.$$

Although (29) is only of a theoretical significant for choosing θ because x^* is unknown, in the applications, if $k > 1$, replacing k and x^* by $k-1$ and $x^{(k-\frac{1}{2})}$ in (29), a reasonable choice may be

$$\theta^{(k)} = 1 + \frac{\left(x^{(k-\frac{3}{2})} - x^{(k-\frac{1}{2})}\right)^T \left(x^{(k-2)} - x^{(k-\frac{3}{2})}\right)}{||x^{(k-2)} - x^{(k-\frac{3}{2})}||_2^2}. \tag{30}$$

Notice that if

$$x^* \not\approx x^{(k-\frac{1}{2})}, \tag{31}$$

that is to say $x^{(k-\frac{1}{2})}$ is not close to x^*, $\theta^{(k)}$ given in (30) can not guarantee that Method 2.4 converges faster. To avoid this, we consider checking the parameter in each step using an interval given in advance. If $\theta^{(k)}$ given in (30) is out of such interval, set $\theta^{(k)}$ to be the corresponding endpoint.

Summarizing the above discussion, the choice of the parameter θ in Method 2.4 is suggested below:

$$\theta = \begin{cases} 1, & \text{if } k = 1; \\ \theta^{(k)}, & \text{if } k > 1 \text{ and } \theta^{(k)} \in (a, b); \\ a, & \text{if } k > 1 \text{ and } \theta^{(k)} \leq a; \\ b, & \text{if } k > 1 \text{ and } \theta^{(k)} \geq b, \end{cases} \tag{32}$$

where $\theta^{(k)}$ is given in (30), and a and b are two constants given as follows.

Remark 4.19 In the applications, to guarantee that Method 2.4 converges, the parameters a, b in (32) can be computed from (19) or (20), e.g., for the case $(\Omega_2 e)_i \geq (D_{A\Omega_1} e)_i$ we have

$$a = \max_{1 \leq i \leq n} \delta_i^{(1)}, b = \min_{1 \leq i \leq n} \delta_i^{(2)} \tag{33}$$

or

$$a = \min_{1 \leq i \leq n} \delta_i^{(1)}, b = \max_{1 \leq i \leq n} \delta_i^{(2)}, \tag{34}$$

where $\delta_i^{(1)}$ and $\delta_i^{(2)}$ are given by (21) and (22) respectively.

5 Numerical examples

In this section, two examples are given to examine the efficiency of our algorithms from the aspects of the number of iteration steps (denoted by 'IT') and the elapsed CPU time in seconds (denoted by 'CPU'). All numerical tests were run on an Intel(R) Core(TM), where the CPU is 2.50 GHz and the memory is 4.00 GB, and the programming language is MATLAB 7.11.

Since Method 2.4 is globally convergent with the assumptions in Theorem 3.13, the initial vector can be chosen arbitrarily. In the following numerical examples, all initial vectors are chosen to be $x^{(0)} = e$ and all iterations are terminated once

$$\frac{||\min(Az^{(k)} + q, z^{(k)})||_2}{||\min(Az^{(0)} + q, z^{(0)})||_2} \leq 10^{-6},$$

where the minimum is taken componentwise.

Let $\Omega_1 = I$. We compare Method 2.4 (RMJ, RMGS and RMSOR) with Method 2.2 (when $\theta = 1$ in Method 2.4). In all numerical examples, we take $\Omega_2 = 2D_A$ and $n = 2500$. For MSOR and RMSOR, let the overrelaxation parameter be $\alpha = 1.2$.

To observe the behavior of Method 2.4, we run Method 2.4 with five strategies of θ:

Strategy 1: Taking θ to be a constant during the iteration. For this case, let θ change from 0.5 to 1.6 with interval 0.1.
Strategy 2: Taking $\theta = 1$ when $k = 1$ and θ be given by (30) when $k > 1$.
Strategy 3: Taking θ to be given by (32) and (33).
Strategy 4: Taking θ to be given by (32) and (34).
Strategy 5: Taking θ to be given by (32) and $a = 0$, $b = 2.3$.

The following two examples are taken from [3, 7], respectively.

Example 5.20 [3] Let LCP(q, A) be given by $A = \hat{A} + 4I$ where

$$\hat{A} = \begin{pmatrix} S & -I & & & & \\ -I & S & -I & & & \\ & -I & S & \ddots & & \\ & & \ddots & \ddots & -I & \\ & & & -I & S & -I \\ & & & & -I & S \end{pmatrix} \in R^{n\times n} \text{ and } q = \begin{pmatrix} -1 \\ 1 \\ -1 \\ \vdots \\ (-1)^{n-1} \\ (-1)^n \end{pmatrix} \in R^n,$$

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

Clearly, A is strictly diagonal dominant with positive diagonal entries. We take $D = I$ in (21) and (22). Then (a, b) is given by

$$(0, 1.2), (0.3333, 1.2), (0.4286, 1.1333),$$

in (33) and

$$(0, 1.3333), (0, 1.3333), (0, 1.2593)$$

in (34) for RMJ, RMGS and RMSOR, respectively.

Example 5.21 [7] Let LCP(q, A) be given by

$$A = \begin{pmatrix} S & -I & -I & & & \\ & S & -I & -I & & \\ & & \ddots & \ddots & \ddots & \\ & & & S & -I & -I \\ & & & & S & -I \\ & & & & & S \end{pmatrix} \in R^{n\times n} \text{ and } q = \begin{pmatrix} -1 \\ 1 \\ -1 \\ \vdots \\ (-1)^{n-1} \\ (-1)^n \end{pmatrix} \in R^n,$$

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

Then A is not a strictly diagonal dominant matrix. We take $D = \text{diag}[(\langle M_{\Omega_1}\rangle - |N_{\Omega_1}|)^{-1}e]$ in (21) and (22). Then (a, b) is given by

$$(0,\ 1.0050),\ (0.9437,\ 1.0050),\ (1 - 6 \times 10^{-7},\ 1 + 5 \times 10^{-8})$$

in (33) and

$$(0,\ 1.2948),\ (0,\ 1.2948),\ (0,\ 1.2143)$$

in (34) for RMJ, RMGS and RMSOR, respectively.

The results of all methods are shown in Tables 1 and 2, where '–' denotes the iteration is divergent. It is shown that Method 2.4 is more efficient than Method 2.2 for both examples in some cases. Furthermore, we have

- For Strategy 1, Method 2.4 converges slower than Method 2.2 when $\theta < 1$. As θ increases, the iteration steps of Method 2.4 first decrease then increase. When θ is out of the convergence range, Method 2.4 diverges. The real convergence range of Method 2.4 is larger than (34) in most cases, which means that (19) and (20) may be improved.
- For Strategy 2, Method 2.4 converges much faster than Method 2.2 for Example 5.20, while it requires more CPU times than the one by Method 2.2 for Example 5.21 for RMJ and RMSOR. The reason is that (31) may happen during the iteration.
- For Strategy 3, by using an interval (a, b) to bound θ, Method 2.4 also converges faster than Method 2.2 for Example 5.20. For Example 5.21, Method 2.4 requires

Table 1 Results of Example 5.20

	RMJ		RMGS		RMSOR	
Strategy	IT	CPU	IT	CPU	IT	CPU
Strategy 1 ($\theta = 0.5$)	72	1.3635	67	1.2578	63	1.1798
Strategy 1 ($\theta = 0.6$)	59	1.1223	55	1.0432	51	0.9557
Strategy 1 ($\theta = 0.7$)	49	0.9243	45	0.8406	42	0.7545
Strategy 1 ($\theta = 0.8$)	42	0.7944	39	0.7315	36	0.6669
Strategy 1 ($\theta = 0.9$)	36	0.6867	33	0.6166	31	0.5672
Strategy 1 ($\theta = 1.0$)	31	0.5548	29	0.5540	27	0.5312
Strategy 1 ($\theta = 1.1$)	27	0.5176	25	0.4766	31	0.5801
Strategy 1 ($\theta = 1.2$)	24	0.4875	34	0.6440	45	0.8422
Strategy 1 ($\theta = 1.3$)	22	0.4165	55	1.0779	130	2.5101
Strategy 1 ($\theta = 1.4$)	19	0.3675	179	3.4875	–	–
Strategy 1 ($\theta = 1.5$)	17	0.3334	–	–	–	–
Strategy 1 ($\theta = 1.6$)	–	–	–	–	–	–
Strategy 2	21	0.3899	19	0.3806	19	0.3591
Strategy 3	26	0.4894	26	0.4895	24	0.4789
Strategy 4	24	0.4544	22	0.4319	22	0.4207
Strategy 5	20	0.3711	19	0.3454	19	0.3618

Table 2 Results of Example 5.21

Strategy	RMJ		RMGS		RMSOR	
	IT	CPU	IT	CPU	IT	CPU
Strategy 1 ($\theta = 0.5$)	411	7.8219	375	7.1993	351	6.6893
Strategy 1 ($\theta = 0.6$)	339	6.3717	309	5.8829	289	5.5191
Strategy 1 ($\theta = 0.7$)	288	5.4652	263	5.0228	245	4.7423
Strategy 1 ($\theta = 0.8$)	250	4.7808	227	4.3191	212	4.0718
Strategy 1 ($\theta = 0.9$)	220	4.3769	200	3.8010	187	3.5603
Strategy 1 ($\theta = 1.0$)	196	3.9300	178	3.3808	166	3.4898
Strategy 1 ($\theta = 1.1$)	176	3.3711	160	3.0502	149	2.8495
Strategy 1 ($\theta = 1.2$)	160	3.0507	145	2.7480	135	2.5567
Strategy 1 ($\theta = 1.3$)	146	2.7809	132	2.5109	123	2.3304
Strategy 1 ($\theta = 1.4$)	134	2.5423	121	2.3101	164	3.1109
Strategy 1 ($\theta = 1.5$)	124	2.3680	341	7.1056	–	–
Strategy 1 ($\theta = 1.6$)	–	–	–	–	–	–
Strategy 2	189	4.0334	155	3.2913	189	4.0256
Strategy 3	195	3.9207	177	3.5254	166	3.1600
Strategy 4	147	3.2221	133	2.7106	133	2.7877
Strategy 5	161	3.2503	139	2.7825	154	3.1427

less CPU times than the one by Strategy 2. But the convergence rate of Method 2.4 is almost the same as Method 2.2. Because b, the right endpoint of (a, b) given in (33), is just a little larger than 1, which deteriorates the convergence rate of Method 2.4.

- For Strategy 4, by enlarging the convergence range, the faster convergence of Method 2.4 is also guaranteed like Strategy 3. Although Method 2.3 with Strategy 2 outperforms Strategy 4 with less iterations and CPU time for Example 5.20, as the comments for Strategy 2 given above, Strategy 2 works less stably than Strategy 4. Consequently, with Strategy 4 Method 2.4 can work more efficient and stable than Method 2.2.

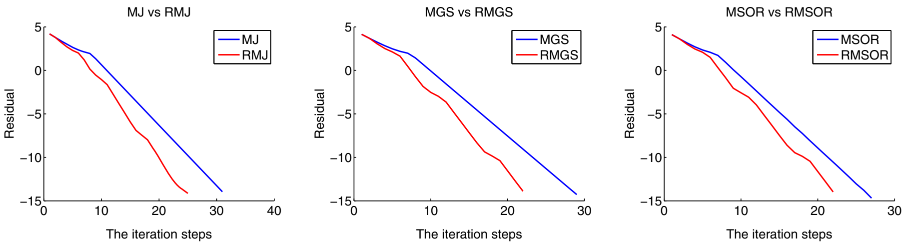

Fig. 1 Residual comparison with Strategy 4 for Example 5.20

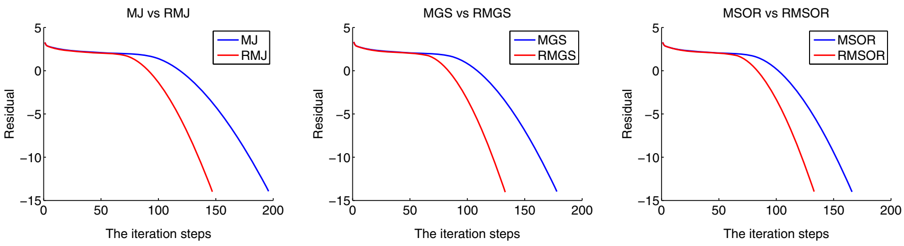

Fig. 2 Residual comparison with Strategy 4 for Example 5.21

- For Strategy 5, where b is larger than that in (34). Comparing with Strategy 4, Method 2.4 converges faster for Example 5.20, while slower for Example 5.21. Although Method 2.4 can be accelerated by enlarging the length of (a, b), this strategy can not be guaranteed as being effective.

In summary, Strategy 4 is recommended for Method 2.4. With Strategy 4, the residuals of Method 2.4 and Method 2.2 are shown in Figs. 1 and 2, where the longitudinal coordinates of all figures denote the natural logarithm of the residuals. It is learnt from Figs. 1 and 2 that the more $x^{(k)}$ approximates x^*, the faster Method 2.4 converges.

6 Conclusions

In this paper, a relaxation modulus-based matrix splitting iteration method for solving the linear complementarity problem is proposed. The convergence analysis of the proposed method is established. We also discuss the choice of the parameter θ. Numerical results show that the relaxation method with Strategy 4 is more efficient and stable. More specifically, comparing with Strategy 4, Strategy 1 and Strategy 5 are without theoretical results, Strategy 3 has slower convergence, and Strategy 2 may lead to divergence. It is also shown that convergence ranges (19) and (20) may be improved. The further theoretical analysis for finding a better parameter θ is worth studying in the future. The relaxation technique can be also applied to the other existing modulus methods.

Acknowledgments The authors would like to thank the referees for their helpful comments. The work was supported by the National Natural Science Foundation of China (Grant No. 11271144, 61305036), Project of Department of Education of Guangdong Province (Grant No. 2013KJCX0053), University of Macau (Grant No. MYRG2015-00064-FST), the Opening Project of Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University(Grant No. 2016005), the Opening Project of Guangdong Provincial Engineering Technology Research Center for Data Sciences and the Science Foundation of Shaoguan University (Grant No. SY2014KJ01).

References

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

2. Berman, A., Plemmons, R.J.: Nonnegative matrix in the mathematical sciences. SIAM Publisher, Philadelphia (1994)
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.: On the convergence of the multisplitting methods for the linear complementarity problem. SIAM J. Matrix Anal. Appl. **21**, 67–78 (1999)
5. Bru Garcia, R., Giménez, I., Hadjidimos, A.: Is $A \in C^{n\times n}$ a general H-matrix? Linear Algebra Appl. **436**, 364–380 (2012)
6. Bai, Z.-Z., Zhang, L.-L.: Modulus-based synchronous multisplitting iteration methods for linear complementarity problems. Numer. Linear Algebra Appl. **20**, 425–439 (2013)
7. Bai, Z.-Z., Zhang, L.-L.: Modulus-based synchronous two-stage multisplitting iteration methods for linear complementarity problems. Numer. Algorithms **62**, 59–77 (2013)
8. Cottle, R.W., Pang, J.-S., Stone, R.E.: The Linear Complementarity Problem. Academic, SanDiego (1992)
9. Dong, J.-L., Jiang, M.-Q.: A modified modulus method for symmetric positive-definite linear complementarity problems. Numer. Linear Algebra Appl. **16**, 129–143 (2009)
10. Frommer, A., Mayer, G.: Convergence of relaxed parallel multisplitting methods. Linear Algebra Appl. **119**, 141–152 (1989)
11. Hadjidimos, A., Lapidakis, M., Tzoumas, M.: On iterative solution for linear complementarity problem with an H-matrix. SIAM J. Matrix Anal. Appl. **33**, 97–110 (2011)
12. Hadjidimos, A., Tzoumas, M.: Nonstationary extrapolated modulus algorithms for the solution of the linear complementarity problem. Linear Algebra Appl. **431**, 197–210 (2009)
13. Hu, J.-G.: Estimates of $||B^{-1}C||_\infty$ and their applications. Mathematica Numerica Sinica **4**, 272–282 (1982)
14. Li, W.: A general modulus-based matrix splitting method for linear complementarity problems of H-matrices. Appl. Math. Lett. **26**, 1159–1164 (2013)
15. Liu, S.-M., Zheng, H., Li, W.: A general accelerated modulus-based matrix splitting iteration method for solving linear complementarity problems. Calcolo, published online (2015)
16. Murty, K.G.: Linear Complementarity, Linear and nonlinear programming. Heldermann Verlag, Berlin (1988)
17. Ortega, J.M., Rheinboldt, W.C.: Iterative solution of nonlinear equations in several variables. SIAM, Philadelphia (2000)
18. van Bokhoven, W.M.G.: Piecewise-linear modelling and analysis. Proefschrift, Eindhoven (1981)
19. Zhang, L.-L.: Two-step modulus based matrix splitting iteration for linear complementarity problems. Numer. Algorithms **57**, 83–99 (2011)
20. 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)
21. Zheng, N., Yin, J.-F.: Accelerated modulus-based matrix splitting iteration methods for linear complementarity problems. Numer. Algorithms **64**, 245–262 (2013)