Filomat 37:2 (2023), 467–476
https://doi.org/10.2298/FIL2302467F

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

The Square-Newton iteration for linear complementarity problem

Mei Feng[a], Xiang Wang[a], Teng Wang[a,]

[a] *School of Mathematics and Computer Sciences, Nanchang University*

Abstract. For getting the numerical solution of the linear complementary problem (LCP), there are many methods such as the modulus-based matrix splitting iteration and the modulus-based nonsmooth Newton's method. We proposed the Square-Newton method to solve the LCP. This method could solve LCP efficiently. We gave the theoretical analysis and numerical experiments in the paper.

1. Introduction

In this paper, we focus on the solution of LCP, which is to find a pair of real vectors $(\hat{\mathbf{x}}, \hat{\mathbf{z}}) \in \mathbb{R}^n \times \mathbb{R}^n$, such that

$$\hat{\mathbf{z}} := A\hat{\mathbf{x}} + \mathbf{q} \geq 0,\ \hat{\mathbf{x}} \geq 0, \quad \textit{where}\ \hat{\mathbf{x}}^{\mathrm{T}}\hat{\mathbf{z}} = 0, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is a given large, sparse matrix, and $\mathbf{q} = (q_1, q_2, ..., q_n)^{\mathrm{T}} \in \mathbb{R}^n$ is a given vector.

LCP was raised by Lemke in 1964, and Cottle and Dantzig formally defined the linear complementarity problem and called it the fundamental problem in [1]. A wide variety of applications require the solution of LCP, such as the economies with institutional restrictions upon prices, the linear and quadratic programming, the free boundary problems and the optimal stopping in Markov chain [2, 3].

There are many efficient iteration methods to solve LCP by using matrix splitting, such as the projected relaxation iteration methods [4, 5], the general fixed-point iteration methods [7, 9, 10] and the matrix multi-splitting iteration methods [11–13]. These methods mainly convert LCP into a implicit fixed-point equation [7], that is,

$$(I + A)\mathbf{y} = (I - A)|\mathbf{y}| - \mathbf{q},$$

where $\mathbf{y} = (\hat{\mathbf{x}} - \hat{\mathbf{z}})/2$, $|\mathbf{y}| = (\hat{\mathbf{x}} + \hat{\mathbf{z}})/2$.

Based on this previous work, a modulus-based matrix splitting iteration method was presented by Bai in [14]. Bai construct a general framework of the modulus-based matrix splitting iteration methods to obtain the solution of LCP. By choosing different matrix splitting, a series of modulus-based matrix splitting iteration methods are given, such as the modulus-based Jacobi, Gauss-Seidel, SOR and AOR iteration methods [14]. More discussions and further generalization of modulus-based matrix splitting iteration

2020 *Mathematics Subject Classification*. Primary 98C53.
Keywords. Square-Newton iteration; Linear complementarity problem; Quadratic convergence.
Received: 05 June 2020; Accepted: 06 March 2022
Communicated by Predrag Stanimirović
Corresponding author: Teng Wang
Email addresses: 604152276@qq.com (Mei Feng), wangxiang49@ncu.edu.cn (Xiang Wang), spring0920@163.com (Teng Wang)

methods have been extensively studied; see [15–20, 22] for details. On the other hand, Zheng gave some Newton methods to solve LCP in [23–25].

In this paper, we consider solving LCP without splitting of coefficient matrix A. We observe that $\hat{\mathbf{x}},\ \hat{\mathbf{z}} \geq 0$ and any square number is non-negative. Let $\mathbf{x}^2 := (x_1^2, x_2^2, ..., x_n^2)^{\mathrm{T}}$, $\mathbf{z}^2 := (z_1^2, z_2^2, ..., z_n^2)^{\mathrm{T}}$ replace $\hat{\mathbf{x}},\ \hat{\mathbf{z}}$, respectively. Obviously, $\hat{\mathbf{x}}^{\mathrm{T}}\hat{\mathbf{z}} = 0$ is equivalent to $x_i z_i = 0, i = 0, 1, ..., n$. Then the problem (1) can be transformed into the following system of nonlinear equations:

$$\begin{cases} \frac{1}{2}A\mathbf{x}^2 + \frac{1}{2}\mathbf{q} = \frac{1}{2}\mathbf{z}^2, \\ x_1 z_1 = 0, \\ x_2 z_2 = 0, \\ \vdots \\ x_n z_n = 0. \end{cases} \tag{2}$$

For the nonlinear problems, the Newton method is efficient and has the property of quadratic convergence. Then we propose a Square-Newton iteration method by combining the Newton method with the non-negative of square number to solve LCP. The structure of our paper is organized as follows. In Section 2, we give some necessary notations, definitions and lemmas, and the MSOR method is introduced in this section. In Section 3, the Square-Newton iteration for solving LCP is established and the convergence properties of our method are discussed. Meanwhile, Some numerical experiments are proposed to prove the effectiveness of our method in Section 4. Finally, our conclusions are shown in Section 5.

2. Preliminaries

In this section, we first introduce some necessary notations, definitions and lemmas. And then we show the basic modulus-based matrix splitting iteration method and give the expression of 'MSOR' method.

2.1. *Definition and notation*

For $\mathbf{x} = (x_1, x_2, ..., x_n)^{\mathrm{T}} \in \mathbb{R}^n$, we define the Euclidean norm of x as $\|\mathbf{x}\| = \sqrt{x_1^2 + ... + x_n^2}$, the infinity norm $\mathbf{x}$ is defined as $\|\mathbf{x}\|_\infty = \max\limits_{1\leq i\leq n} |x_i|$. For any $A \in \mathbb{R}^{n\times n}$, $\|A\|$ is denoted as the spectral norm, that is $\|A\| = \max\{\|A\mathbf{x}\| : \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\| = 1\}$, which $\|\mathbf{x}\|$ is the Euclidean norm. And the vector $\mathbf{x} * \mathbf{z}$ is defined by

$$\mathbf{x} * \mathbf{z} = \begin{bmatrix} x_1 z_1 \\ x_2 z_2 \\ \vdots \\ x_n z_n \end{bmatrix}.$$

Definition 2.1 ([25]). *Let $f : \mathbb{D} \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be a given function, and let $\mathbf{x}$ be a given point in $\mathbb{R}^n$. The function f is said to be Lipschitz near $\mathbf{x}$, if there exist a scalar L and a positive number ϵ, s.t*

$$\|f(\mathbf{x}'') - f(\mathbf{x}')\| \leq L\|\mathbf{x}'' - \mathbf{x}'\|,$$

for all $\mathbf{x}''$, $\mathbf{x}' \in \mathbf{x} + \epsilon B$, where B signifies the open ball in $\mathbb{R}^n$, so that $\mathbf{x} + \epsilon B$ is the unit open ball of radius ϵ centred at $\mathbf{x}$.

Lemma 2.2 ([21]). *Let $\|\cdot\|$ satisfy $\|AB\| \leq \|A\|\|B\|$. Then $\|X\| \leq 1$ implies that $I - X$ is invertible, and*

$$\|(I - X)^{-1}\| \leq \frac{1}{1 - \|X\|}.$$

Corollary 2.3. *For the matrix $A,\ C \in \mathbb{R}^{n\times n}$, if A is nonsingular, $\|A^{-1}\| \le \alpha$ and $\|A - C\| \le \beta$, $\alpha\beta < 1$, then C is nonsingular, and*

$$\|C^{-1}\| \le \frac{\alpha}{1-\alpha\beta}.$$

Proof. Let $B = A^{-1}(A - C) = I - A^{-1}C$, then

$$\|B\| \le \|A^{-1}\|\|(A - C)\| \le \alpha\beta < 1.$$

According Lemma 2.2, it is obvious that $I - B = A^{-1}C$ is nonsingular, then C is nonsingular, and $C^{-1} = (I - B)^{-1}A^{-1}$, so

$$\begin{aligned}\|C^{-1}\| &\le \|(I - B)^{-1}\| \cdot \|A^{-1}\| \\ &\le \frac{\alpha}{1-\|B\|} \\ &\le \frac{\alpha}{1-\alpha\beta},\end{aligned}$$

which completes the proof. □

Lemma 2.4. *Let $\mathbb{D}$ be the convex domain on $\mathbb{R}^n$, $F(\mathbf{x}) : \mathbb{D} \to \mathbb{R}^n$. F is differentiable on $\mathbb{D}$. If $\forall \mathbf{x},\ \mathbf{y} \in \mathbb{D}$, there exists $\gamma > 0$, s.t $\|F'(\mathbf{x}) - F'(\mathbf{y})\| \le \gamma\|\mathbf{x} - \mathbf{y}\|$. Then*

$$\|F(\mathbf{x}) - F(\mathbf{y}) - F'(\mathbf{y})(\mathbf{x} - \mathbf{y})\| \le \frac{\gamma\|\mathbf{x} - \mathbf{y}\|^2}{2},\ \forall \mathbf{x},\ \mathbf{y} \in \mathbb{D}.$$

Proof. Let $h(\tau) = F(\mathbf{y} + \tau(\mathbf{x} - \mathbf{y}))$, $\tau \in (0, 1)$. According to the definition of derivative, we know $h(\tau)$ is derivable, and $h'(\tau) = F'(\mathbf{y} + \tau(\mathbf{x} - \mathbf{y}))(\mathbf{x} - \mathbf{y})$.

For $\tau \in (0, 1)$, we have

$$\begin{aligned}\|h'(\tau) - h'(0)\| &\le \|F'(\mathbf{y} + \tau(\mathbf{x} - \mathbf{y})) - F'(\mathbf{y})\| \cdot \|\mathbf{x} - \mathbf{y}\| \\ &\le \gamma\tau\|\mathbf{x} - \mathbf{y}\|^2,\end{aligned}$$

then

$$\begin{aligned}\|F(\mathbf{x}) - F(\mathbf{y}) - F'(\mathbf{y})(\mathbf{x} - \mathbf{y})\| &= \|h(1) - h(0) - h'(0)\| \\ &= \|\int_0^1 (h'(\tau) - h'(0))d\tau\| \\ &\le \int_0^1 \|(h'(\tau) - h'(0))\|d\tau \\ &\le \gamma\|\mathbf{x} - \mathbf{y}\|^2 \int_0^1 \tau d\tau \\ &= \frac{\gamma\|\mathbf{x} - \mathbf{y}\|^2}{2},\end{aligned}$$

which completes the proof. □

2.2. *The MSOR method*

The following Theorem establishes an equivalent expression of LCP, and it is useful to propose matrix splitting iteration methods for solving LCP.

Theorem 2.5 ([14]). *Let $A = M - N$ be a splitting of the matrix $A \in \mathbb{R}^{n\times n}$, Ω_1 and Ω_2 be $n \times n$ nonnegative diagonal matrices, and Ω and Γ be $n \times n$ positive diagonal matrices such that $\Omega = \Omega_1 + \Omega_2$. For the LCP, the following statements hold true:*

(i) If (ω, z) is a solution of LCP, then $\mathbf{x} = \frac{1}{2}(\Gamma^{-1}z - \Omega^{-1}\omega)$ satisfied the implicit fixed-point equation

$$(M\Gamma + \Omega_1)\mathbf{x} = (N\Gamma - \Omega_2)\mathbf{x} + (\Omega - A\Gamma)|\mathbf{x}| - \mathbf{q}. \tag{3}$$

(ii) If x satisfied the implicit fixed-point equation (3), then

$$z = \Gamma(|\mathbf{x}| + \mathbf{x}), \omega = \Omega(|\mathbf{x}| - \mathbf{x});$$

is a solution of LCP.

Setting $\Omega_1 = \Omega$, $\Omega_2 = 0$ and $\Gamma = \frac{1}{\gamma}I$, then the implicit fixed-point equation can be simplified, that is,

$$(M + \Omega)\mathbf{x} = (N)\mathbf{x} + (\Omega - A)|\mathbf{x}| - \gamma\mathbf{q}.$$

Based on the simplified implicit fixed-point equation, the modulus-based matrix splitting iteration(MS) method for LCP was proposed by Bai [14].

Method (MS)[14]: *Let $A = M - N$ be a splitting of the matrix $A \in \mathbb{R}^{n\times n}$. Given an initial vector $x^{(0)} \in \mathbb{R}^n$, for $k = 0, 1, 2, \ldots$ until the iteration sequence $\{z^{(k)}\}_{k=0}^{+\infty} \subset \mathbb{R}^n$ is convergent, compute $x^{(k+1)} \in \mathbb{R}^n$ by solving the linear system*

$$(M + \Omega)\mathbf{x}^{(k+1)} = (N - \Omega)\mathbf{x}^k + (\Omega - A)|\mathbf{x}^k| - \gamma\mathbf{q};$$

where $\Omega \in \mathbb{R}^{n\times n}$ is an positive diagonal matrix, $\gamma > 0$. Then set

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

the value of $(\hat{\mathbf{x}}, \hat{\mathbf{z}})$ are obtained.

When $M = (1/\alpha)D - L$, $N = (1/\alpha - 1)D + U$, and $\gamma = 2$, MS method is transformed into the modulus-based SOR(MSOR) iteration method, that is,

$$(D + \Omega - \alpha L)\mathbf{x}^{k+1} = [(1 - \alpha)D + \alpha U]\mathbf{x}^k + (\Omega - \alpha M)|\mathbf{x}^k| - 2\alpha\mathbf{q}; \tag{4}$$

with $\mathbf{z}^{(k+1)} = \frac{1}{2}(|\mathbf{x}^{k+1}| + \mathbf{x}^{k+1})$.

3. The proposed method

We now discuss how to use Newton method to solve LCP by combining the non-negative of square number. And the convergence of the proposed method are introduced in this section.

3.1. *The Square-Newton iteration method*

In order to establish the Square-Newton iteration, based on (2), we give the following function $G(\mathbf{y})$:

$$G(\mathbf{y}) = \begin{bmatrix} \frac{1}{2}A\mathbf{x}^2 + \frac{1}{2}\mathbf{q} - \frac{1}{2}\mathbf{z}^2 \\ \mathbf{x} * \mathbf{z} \end{bmatrix}, \tag{5}$$

where $\mathbf{y} = [\mathbf{x}^T, \mathbf{z}^T]^T \in \mathbb{R}^{2n}$, $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ and $\mathbf{z} = (z_1, \ldots, z_n)^T \in \mathbb{R}^n$.

From (5), we obtain the Jacobian matrix $G'(\mathbf{y})$:

$$G'(\mathbf{y}) = \begin{bmatrix} AD & -W \\ W & D \end{bmatrix}, \tag{6}$$

where $D = diag(x_1, x_2, ..., x_n)$, $W = diag(z_1, z_2, ..., z_n)$.

It is clear that the Newton iteration for solving the equation $G(\mathbf{y}) = 0$ is defined by

$$G(\mathbf{y}_k) + G'(\mathbf{y}_k)(\mathbf{y}_{k+1} - \mathbf{y}_k) = 0. \tag{7}$$

Based on (5) and (6), the Newton iteration (7) simplifies to the following form:

$$\begin{bmatrix} AD_k & -W_k \\ W_k & D_k \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{z}_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}A\mathbf{x}_k^2 - \frac{1}{2}\mathbf{q} - \frac{1}{2}\mathbf{z}_k^2 \\ \mathbf{x}_k * \mathbf{z}_k \end{bmatrix}. \tag{8}$$

Now for the linear complementarity problem, we propose the Square-Newton iteration as follows:

Algorithm: The Square-Newton iteration(SN)

Input: Given matrix $A \in \mathbb{R}^{n\times n}$, vector $\mathbf{q} \in \mathbb{R}^n$ and a tolerance $\varepsilon > 0$.
The initial value $\mathbf{y}_0 = [1, \cdots, 1]^{\mathrm{T}}$.
Output: The desired vector $\mathbf{y}_*$.
1. Set $k := 0$.
2. Compute $G_k = \begin{bmatrix} \frac{1}{2}A\mathbf{x}_k^2 - \frac{1}{2}\mathbf{q} - \frac{1}{2}\mathbf{z}_k^2 \\ \mathbf{x}_k * \mathbf{z}_k \end{bmatrix}$ and $G'_k = \begin{bmatrix} AD_k & -W_k \\ W_k & D_k \end{bmatrix}$.
3. Solve the linear system $G'_k \cdot \mathbf{y}_{k+1} = G_k$,
to get $\mathbf{y}_{k+1}$ (here we use the left division).
4. If $\|\mathbf{y}_{k+1} - \mathbf{y}_k\| \le \varepsilon$, the solution $\mathbf{y}_* = \mathbf{y}_{k+1}$.
Otherwise, Set $k = k + 1$ and go to step 1.

3.2. *Convergence analysis*

Here, we discuss the convergence of Square-Newton iteration method and prove the quadratic convergence of our method.

Theorem 3.1. *Assume that the function $F(\mathbf{x}) : D \to \mathbb{R}^n$ and $\exists \mathbf{x}_* \in D$ s.t. $F(\mathbf{x}_*) = 0$. If $F'(\mathbf{x})$ is continuous in the open neighborhood $S \subset D$ of $\mathbf{x}_*$ and $F'(\mathbf{x}_*)$ is nonsingular, then the sequence $\{\mathbf{x}_k\}$ by the Newton iteration superlinearly converges to $\mathbf{x}_*$. Specially, if there exists a constant $L > 0$, satisfying*

$$\|F'(\mathbf{x}) - F'(\mathbf{x}_*)\| \le L\|\mathbf{x} - \mathbf{x}_*\|, \forall \mathbf{x} \in S, \tag{9}$$

then $\{\mathbf{x}_k\}$ at least quadratic convergence.

Proof. Since $F'(\mathbf{x}_*)$ is nonsingular, set $\|F'(\mathbf{x}_*)^{-1}\| = \alpha > 0$. $F'(\mathbf{x})$ is continuous at $\mathbf{x}_*$, so for $\forall \varepsilon \in (0, \frac{1}{2\alpha})$, $\exists \delta > 0$, s.t $\forall \mathbf{x} \in S = S(\mathbf{x}_*, \delta)$, there is

$$\|F'(\mathbf{x}) - F'(\mathbf{x}_*)\| \le \varepsilon. \tag{10}$$

According to Corollary 2.3, we know for $\alpha\varepsilon < \frac{1}{2}$, $F'(\mathbf{x})$ is nonsingular and

$$\|F'(\mathbf{x})^{-1}\| \le \frac{\alpha}{1 - \alpha\varepsilon} \le 2\alpha, \forall \mathbf{x} \in S. \tag{11}$$

Then

$$\begin{aligned} \|\mathbf{x}_{k+1} - \mathbf{x}_*\| =& \| - F'(\mathbf{x}_k)^{-1}F(\mathbf{x}_k) + \mathbf{x}_k - \mathbf{x}_*\| \\ =& \| - F'(\mathbf{x}_k)^{-1}[F(\mathbf{x}_k) - F'(\mathbf{x}_k)(\mathbf{x}_k - \mathbf{x}_*)]\| \\ =& \| - F'(\mathbf{x}_k)^{-1}[F(\mathbf{x}_k) - F'(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*) \\ &+ (F'(\mathbf{x}_*) - F'(\mathbf{x}_k))(\mathbf{x}_k - \mathbf{x}_*)]\| \\ \le& 2\alpha[\|F(\mathbf{x}_k) - F(\mathbf{x}_*) - F'(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*)\| \\ &+ \|(F'(\mathbf{x}_*) - F'(\mathbf{x}_k))(\mathbf{x}_k - \mathbf{x}_*)\|]. \end{aligned} \tag{12}$$

Since $F'(\mathbf{x})$ exists and is continuous, so

$$\|F(\mathbf{x}_k) - F(\mathbf{x}_*) - F'(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*)\| \leq \varepsilon\|\mathbf{x}_k - \mathbf{x}_*\|, \ \forall \mathbf{x}_k \in S. \tag{13}$$

From (10) and (13), we have

$$\begin{aligned} \|\mathbf{x}_{k+1} - \mathbf{x}_*\| &\leq 2\alpha(\varepsilon + \varepsilon)\|(\mathbf{x}_k - \mathbf{x}_*)\| \\ &= 4\alpha\varepsilon\|(\mathbf{x}_k - \mathbf{x}_*)\|. \end{aligned} \tag{14}$$

When $\mathbf{x}_k \neq \mathbf{x}_*$, then

$$\lim_{k\to\infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}_*\|}{\|\mathbf{x}_k - \mathbf{x}_*\|} = 0,$$

which shows $\{\mathbf{x}_k\}$ superlinear convergence.

According to lemma 2.4, if (9) is satisfied, then the equation (12) transforms into

$$\begin{aligned} \|\mathbf{x}_{k+1} - \mathbf{x}_*\| \leq & 2\alpha\|F(\mathbf{x}_k) - F(\mathbf{x}_*) - F'(\mathbf{x}_*)(\mathbf{x}_k - \mathbf{x}_*)\| \cdot \|(\mathbf{x}_k - \mathbf{x}_*)\| \\ & + 2\alpha\|(F'(\mathbf{x}_*) - F'(\mathbf{x}))\| \cdot \|(\mathbf{x}_k - \mathbf{x}_*)\| \\ \leq & \alpha L\|(\mathbf{x}_k - \mathbf{x}_*)\|^2 + 2\alpha L\|(\mathbf{x}_k - \mathbf{x}_*)\|^2 \\ \leq & 3\alpha L\|(\mathbf{x}_k - \mathbf{x}_*)\|^2. \end{aligned} \tag{15}$$

It is clear that $\{\mathbf{x}_k\}$ at least quadratic convergence. □

Theorem 3.2. *Suppose that there exists $\mathbf{y}_* \in \mathbb{D}$ satisfying $G(\mathbf{y}_*) = 0$ and the corresponding Jacobian matrix $G'(\mathbf{y}_*)$ is nonsingular. Let the sequence $\{\mathbf{y}_k\}$ be generated by Square-Newton iteration. If $G'(\mathbf{y}_*)$ is nonsingular, then the sequence $\{\mathbf{y}_k\}$ is at least quadratic convergence to the unique solution $\mathbf{y}_*$.*

Proof. It is obvious that $G'(\mathbf{y})$ exists and $\mathbf{y}_*$ is the unique solution of LCP, so $G(\mathbf{y}_*) = 0$.

Now, we will prove that $G(\mathbf{y})$ satisfied (9).

$$\begin{aligned} \|G'(\mathbf{y}_k) - G'(\mathbf{y}_*)\| &= \left\|\begin{bmatrix} AD_k & -W_k \\ W_k & D_k \end{bmatrix} - \begin{bmatrix} AD_* & -W_* \\ W_* & D_* \end{bmatrix}\right\| \\ &= \left\|\begin{bmatrix} A(D_k - D_*) & -W_k + W_* \\ W_k - W_* & D_k - D_* \end{bmatrix}\right\| \\ &\leq \left\|\begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}\right\| \cdot \left\|\begin{bmatrix} D_k - D_* & 0 \\ 0 & D_k - D_* \end{bmatrix}\right\| \\ &\quad + \left\|\begin{bmatrix} 0 & -(W_k - W_*) \\ W_k - W_* & 0 \end{bmatrix}\right\|. \end{aligned} \tag{16}$$

Let $B = diag(M, I)$, due to D_k, W_k are diagonal matrix, so

$$\left\|\begin{bmatrix} D_k - D_* & 0 \\ 0 & D_k - D_* \end{bmatrix}\right\| \leqslant \max_{1\leq i\leq n} |x_i^{(k)} - x_i^*|;$$

the same is true for W_k, then

$$\begin{aligned} \|G'(\mathbf{y}_k) - G'(\mathbf{y}_*)\| \leq & \|B\| \max_{1\leq i\leq n} |x_i^{(k)} - x_i^*| + \max_{1\leq i\leq n} |z_i^{(k)} - z_i^*| \\ \leq & (\|B\| + 1) \max_{1\leq i\leq n} \{|x_i^{(k)} - x_i^*|, |z_i^{(k)} - z_i^*|\} \\ = & (\|B\| + 1)\|\mathbf{y}_k - \mathbf{y}_*\|_\infty \\ \leq & (\|B\| + 1)\|\mathbf{y}_k - \mathbf{y}_*\|. \end{aligned} \tag{17}$$

From Definition 2.1, we know that $G'(\mathbf{y})$ is Lipschitz continuous. So according to Theorem 2.5, the sequence $\{\mathbf{y}_k\}$ which is generated by Square-Newton iteration converges to the real solution $\mathbf{y}_*$ at least quadratic rate. □

4. Numerical experiments

In this section, we use some numerical experiments to verify the efficiency of our algorithm in terms of iteration steps (denoted as IT), computing time in seconds (denoted as CPU) and the norm of residual vectors (denoted as RES). Here, we define the RES as

$$\text{RES:=}\|\min(\mathbf{x}_k, \mathbf{z}_k)\|_2,$$

where $\mathbf{x}_k, \mathbf{z}_k$ is the approximate solution of LCP (1).

The Square-Newton iteration method is proposed by combining Newton iteration and the non-negativity of square number. We compare the Square-Newton iteration method with 'MSOR' iteration (4) which was presented in [14] in those experiments.

All experiments are performed in MATLAB(R2016b) with machine precision 10^{-16}, and all experiments are implemented on a personal computer with 2.00G memory and Win10 operating system. Here we take the initial value $\mathbf{y}_0 = [1, 1, ..., 1]^{\mathrm{T}}$.

Example 4.1 ([14]). *Let m be a prescribed positive integer and $n = m^2$. Consider the LCP (1), in which $A \in \mathbb{R}^{n\times n}$ is given by $A = \hat{A} + \mu I$ and $\mathbf{q} \in \mathbb{R}^n$ is given by $\mathbf{q} = -(\frac{1}{\alpha}D - L)\mathbf{z}_*$, where*

$$\hat{A} = tridiag(-I, S, -I) = \begin{bmatrix} S & -I & & & & \\ -I & S & -I & & & \\ & -I & S & \ddots & & \\ & & \ddots & \ddots & -I & \\ & & & -I & S & -I \\ & & & & -I & S \end{bmatrix} \in \mathbb{R}^{n\times n},$$

is a block-tridiagonal matrix,

$$S = \begin{bmatrix} 4 & -1 & & & & \\ -1 & 4 & -1 & & & \\ & -1 & 4 & \ddots & & \\ & & \ddots & \ddots & -1 & \\ & & & -1 & 4 & -1 \\ & & & & -1 & 4 \end{bmatrix} \in \mathbb{R}^{m\times m},$$

is a tridiagonal matrix, and $\mathbf{z}_* = [1, 2, ..., 1, 2]^{\mathrm{T}} \in \mathbb{R}^n$. In this example, $\alpha = 1$.

We should attention that when $\mu \geq 0$, the system matrix $A \in \mathbb{R}^{m\times m}$ is strictly diagonally dominant. Then the LCP (1) has one unique solution.

Example 4.2 ([14]). *Let m be a prescribed positive integer and $n = m^2$. Consider the LCP (1), in which $A \in \mathbb{R}^{n\times n}$ is given by $A = \hat{A} + \mu I$ and $\mathbf{q} \in \mathbb{R}^n$ is given by $\mathbf{q} = -A\mathbf{z}_*$, where*

$$\begin{aligned} \hat{A} &= tridiag(-1.5I, S, -0.5I) \\ &= \begin{bmatrix} S & -0.5I & & & & \\ -1.5I & S & -0.5I & & & \\ & -1.5I & S & \ddots & & \\ & & \ddots & \ddots & -0.5I & \\ & & & -1.5I & S & -0.5I \\ & & & & -1.5I & S \end{bmatrix} \in \mathbb{R}^{n\times n}, \end{aligned}$$

m	Method	IT	CPU	RES
10	MSOR	35	0.1637	7.1345e-06
	SN	5	0.1250	1.4296e-17
20	MSOR	37	0.3055	8.3960e-06
	SN	5	0.1563	9.3952e-17
30	MSOR	38	0.8966	8.1090e-06
	SN	5	0.3594	9.9269e-17
40	MSOR	38	2.2490	9.8112e-06
	SN	5	1.1873	1.1336e-16
50	MSOR	39	6.3224	7.9998e-06
	SN	5	2.0313	1.2965e-16

Table 1: Numerical results for $\mu = 4$ for Example 1(Symmetric matrix).

m		IT	CPU	RES
10	MSOR	22	0.1083	7.4094e-06
	SN	5	0.0781	6.8370e-15
20	MSOR	26	0.2429	8.2821e-06
	SN	5	0.1875	1.7799e-14
30	MSOR	27	0.6576	9.4333e-06
	SN	5	0.5625	2.0279e-15
40	MSOR	28	2.4853	9.8149e-06
	SN	5	2.8281	1.6786e-15
50	MSOR	30	5.2891	9.7299e-06
	SN	6	3.6719	3.2384e-14

Table 2: Numerical results for $\mu = 2$ for Example 2(Nonsymmetric matrix).

is a block-tridiagonal matrix,

$$S = \begin{bmatrix} 4 & -0.5 & & & & \\ -1.5 & 4 & -0.5 & & & \\ & -1.5 & 4 & \ddots & & \\ & & \ddots & \ddots & -0.5 & \\ & & & -1.5 & 4 & -0.5 \\ & & & & -1.5 & 4 \end{bmatrix} \in \mathbb{R}^{m\times m},$$

is a tridiagonal matrix, and $\mathbf{z}_* = [1, 2, ..., 1, 2]^{\mathrm{T}} \in \mathbb{R}^n$. *In this example,* $\alpha = 0.8$.

Table 1 and Table 2 show that the iteration steps, computing time and residual generated by the Square-Newton and 'MSOR' method to solve LCP. We can find when A is Symmetric matrix, our iteration steps are much less than the steps of 'MSOR', and our accuracy is much higher than that of 'MSOR'. Meanwhile, the computing time of Square-Newton method is nearly half of the computing time of 'MSOR'. When A is nonsymmetric matrix, the iteration steps and the accuracy of our method are much less than that of 'MSOR', but we attention that the computing time of both methods is nearly the same, because the computation of Jacobian matrix. In Fig.1-2, we find the convergence rate of our method is faster than that of 'MSOR', and it's clear that Square-Newton method has quadratic convergence.

5. Conclusion

In this paper, we propose a Square-Newton method for solving LCP. And we analyze the convergence of our method, the quadratic convergence of our method is proved. In the numerical experiments, we

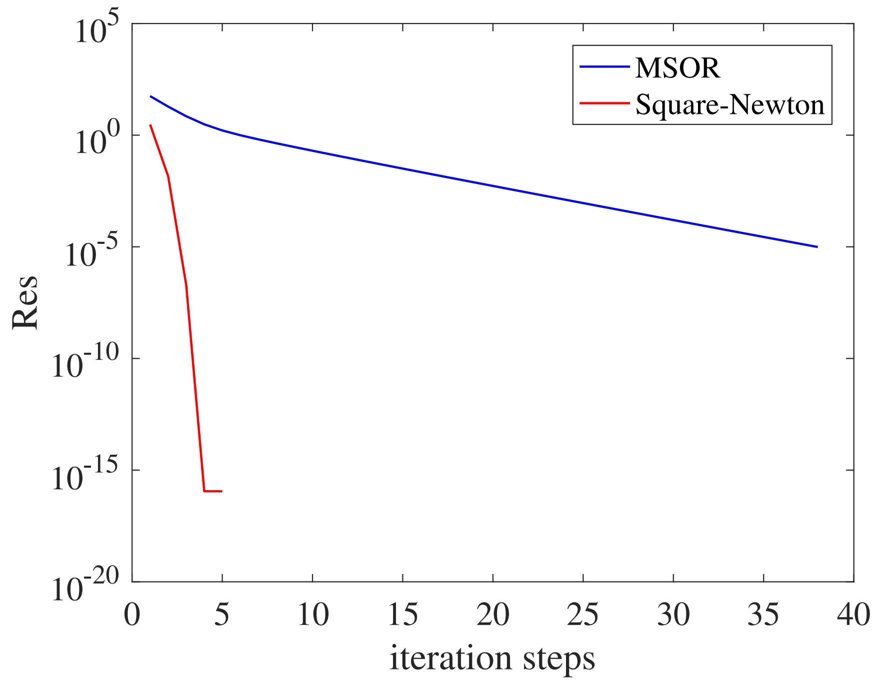

Figure 1: The Res comparison of Example 1, when $m = 40$.

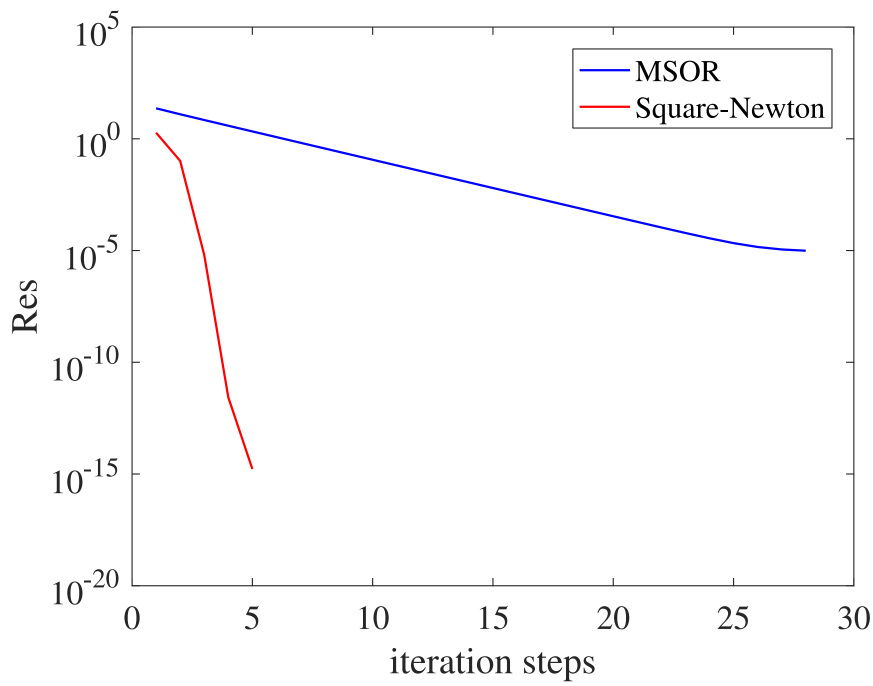

Figure 2: The Res comparison of Example 2, when $m = 40$

verify that our method can effectively solve LCP. By comparing our method with 'MSOR' method, it can be proved that the convergence rate of our method is faster than that of 'MSOR'.

References

[1] Cottle, Richard W and Dantzig, George B, Complementary pivot theory of mathematical programming, Linear Algebra and Its Applications 1 (1968) 103-125.
[2] Pardalos, Panos M, The Linear Complementarity Problem, Advances in Optimization and Numerical Analysis 7 (1994) 39-49.
[3] Schäfer, Uwe, A Linear Complementarity Problem with a P-Matrix, Siam Review 46 (2004) 189-201.
[4] Cryer, Colin W, The solution of a quadratic programming using systematic overrelaxation, SIAM Journal on Control 9 (1971) 385-392.
[5] Bai, Zhong Zhi, On the monotone convergence of the projected iteration methods for linear complementarity problem, Numerical Mathematics A Journal of Chinese Universities 5 (1996) 228-233.
[6] Leenaerts, Domine M W and Bokhoven, Wim M Van, Piecewise Linear Modeling and Analysis, Technische Hogeschoolndhoven 88 (1981) 578-578.
[7] Mangasarian, O L, Solution of symmetric linear complementarity problems by iterative methods, Journal of Optimization Theory and Applications 22 (1977) 465-485.
[8] Pang, J S, Necessary and sufficient conditions for the convergence of iterative methods for the linear complementarity problem, Journal of Optimization Theory and Applications 42 (1984) 1-17.
[9] B, H Ahn, Solution of nonsymmetric linear complementarity problems by iterative methods, Journal of Optimization Theory and Applications 33 (1981) 175-185.
[10] Pang, J S, Necessary and sufficient conditions for the convergence of iterative methods for the linear complementarity problem, Journal of Optimization Theory and Applications 42 (1984) 1-17.
[11] Bai, Zhong Zhi and Evans, D J, Matrix multisplitting relaxation methods for linear complementarity problems, International Journal of Computer Mathematics 63 (1997) 18.
[12] Bai, Zhong Zhi, On the Convergence of the Multisplitting Methods for the Linear Complementarity Problem, Siam Journal on Matrix Analysis and Applications 21 (1999) 67-78.
[13] Bai, Zhong Zhi and Evans, D J, Matrix Multisplitting Methods with Applications to Linear Complementarity Problems: Parallel Asynchronous Methods, International Journal of Computer Mathematics 79 (2002) 205-232.
[14] Bai, Zhong Zhi, Modulus-based matrix splitting iteration methods for linear complementarity problems, Numerical Linear Algebra with Applications 17 (2010) 917-933.
[15] Zhang, Li Li, Two-step modulus-based matrix splitting iteration method for linear complementarity problems, Numerical Algorithms 57 (2011) 83-99.
[16] Zheng, Hua and Li, Wen, The modulus-based nonsmooth Newton's method for solving linear complementarity problems, Journal of Computational and Applied Mathematics 288 (2015) 116-126.
[17] Xu, Wei Wei and Liu, Hao, A modified general modulus-based matrix splitting method for linear complementarity problems of H-matrices, Linear Algebra and Its Applications 458 (2014) 626-637.
[18] Zhang, Li Li and Ren, Zhi Ru, Improved convergence theorems of modulus-based matrix splitting iteration methods for linear complementarity problems, Applied Mathematics Letters 26 (2013) 638-642.
[19] Zheng, Ning and Yin, Jun Feng, Accelerated modulus-based matrix splitting iteration methods for linear complementarity problem, Numerical Algorithms 64 (2013) 245-262.
[20] Zheng, Hua and Li, Wen and Vong, Seakweng, A relaxation modulus-based matrix splitting iteration method for solving linear complementarity problems, Numerical Algorithms 74 (2017) 137-152.
[21] Demmel, James W, Applied numerical linear algebra, Siam (2011).
[22] Zheng, Hua and Vong, Seakweng, A two-step modulus-based matrix splitting iteration method for horizontal linear complementarity problems, Numerical Algorithms 86 (2021) 1791-1810.
[23] Zheng, Hua and Vong, Seakweng and Guo, W X, Newton-type methods for solving quasi-complementarity problems via sign-based equation, Calcolo 56 (2019) 20.
[24] Zheng, Hua and Liu, L, The sign-based methods for solving a class of nonlinear complementarity problems, J. Optim. Theory Appl. 180 (2019) 480-499.
[25] Zheng, Hua and Vong, Seakweng, The modulus-based nonsmooth Newton's method for solving a class of nonlinear complementarity problems of P-matrices, Calcolo 55 (2018) 37.