Xie and Ma *Journal of Inequalities and Applications* (2015) 2015:410
DOI 10.1186/s13660-015-0937-4

Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

A generalized Newton method of high-order convergence for solving the large-scale linear complementarity problem

Yajun Xie[1,2] and Changfeng Ma[1,2*]

*Correspondence:
macf@fjnu.edu.cn
[1]School of Mathematics and Computer Science, Fujian Normal University, Fuzhou, 350117, China
[2]Department of Mathematics and Physics, Fujian Jiangxia University, Fuzhou, 350108, China

Abstract

In this paper, by extending the classical Newton method, we present the generalized Newton method (GNM) with high-order convergence for solving a class of large-scale linear complementarity problems, which is based on an additional parameter and a modulus-based nonlinear function. Theoretically, the performance of high-order convergence is analyzed in detail. Some numerical experiments further demonstrate the efficiency of the proposed new method.

MSC: 15A24; 65H10; 65F10; 65F30

Keywords: linear complementarity problem; generalized Newton method; high-order convergence; the modulus-based nonlinear function; additional parameter; numerical test

1 Introduction

We consider the linear complementarity problem, abbreviated as $LCP(q,A)$, to find a vector $u \in \mathbb{R}^n$ such that

$$\begin{cases} u \geq 0, \\ w := Au + q \geq 0, \\ w^T u = 0, \end{cases} \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}$ and $q \in \mathbb{R}^n$ are a given real matrix and a real vector, respectively.

The linear complementarity problem frequently arises in various scientific and engineering applications, such as Nash equilibrium point of a bimatrix game, the contract problem, and the free boundary problem for journal bearings; see [1–3].

In [3], Lemke proposed first a solution for linear complementarity problem. Along these ideas, Scarf has given the approximation of fixed-points of a continuous mapping [4]. The relationship between the linear complementarity problem and the fixed-points problem is well described by Eaves *et al.* [5, 6].

Many efficient methods were developed to solve linear complementarity problem. Especially, when the system matrix A is large and sparse. For instance, we have the projected successive overrelaxation iteration [7] and the general fixed-point iterations [8]. On the matrix splitting iterations approaches, Bai *et al.* have derived some fruitful research results [9–12], especially, in [9], Bai proposed the modulus-based matrix splitting iteration scheme which is a powerful method for solving the linear complementarity problem. Matrix multisplitting iteration aspects, also, in many works were considered by Bai *et al.* to solve the linear complementarity problem in [13–17]. A variety of accelerated modulus-based matrix splitting iteration versions were also established; see [18, 19]. Furthermore, the modulus-based synchronous multisplitting iteration methods for large sparse linear complementarity problems are introduced in [20]. On the basis of [20], Ljiljana *et al.* avoided the assumption of the parameter constraint and improved the convergence area [21]. Recently, in [22], by the vector divisions and the secant method, Foutayeni *et al.* investigated an efficient hybrid method for solving the linear complementarity problem.

As is well known the semismooth (or smooth) Newton method is very efficient for some nonsmooth (or smooth) equations, which arise from the complementarity problem, the nonlinear programming problem, the variational inequality problem, the discretization problem of partial differential equations, the maximal monotone operator problem, *etc.*; see [23–25] for a detailed discussion. These methods are competitive since they converge rapidly for any sufficiently right initial guess. In order to ensure that the global convergence of semismooth Newton methods, some merit functions, such as squared norm merit functions, are often exploited; see [26] and the references therein. However, just as the statement in [27], a globalization of the semismooth Newton method for nonsmooth equations is very hard because the corresponding merit function is nondifferentiable in most cases.

In view of this, by introducing a smooth equation and some reasonable equivalent reformulations, we investigate a generalized Newton iteration method with high-order convergence rate for solving a class of large-scale linear complementarity problem, which make full use of the superiority of the second-order convergence rate of the classical Newton method. In this article, we suppose that the matrix A of the linear complementarity problem (1.1) is a P-matrix, *i.e.*, the determinants of all principal submatrices are positive. Under this assumption, as is well known, the linear complementarity problem (1.1) has a unique solution for every q.

For simplicity of the presentation, we use the following notations throughout the paper: Let $\mathbb{N}_k = \{1, 2, \dots, k\}$ denote the set of first k positive integers. For $x \in \mathbb{R}^n$, $\|x\|$ stands for the Euclidean norm. Given two real $n \times m$ matrices $A = (a_{ij})$ and $B = (b_{ij})$, we write $A \geq B$ (or $A > B$) if $a_{ij} \geq b_{ij}$ (or $a_{ij} > b_{ij}$) hold for all $i \in \mathbb{N}_n$ and $j \in \mathbb{N}_m$. $|A|$ and $\rho(A)$ denote the absolute value and spectral radius of the matrix $A \in \mathbb{R}^{n\times m}$, respectively. For a differential function $F(x)$, $F'(x)$ is referred to as the Jacobi matrix of the function $F(x)$. For an invertible matrix A, A^{-1} denotes the inverse matrix of A. The matrix $\mathrm{Diag}\{a_1, a_2, \dots, a_n\}$ denotes the diagonal matrix, where a_i ($i \in \mathbb{N}_n$) are the elements of the principal diagonal.

The outline of the paper is organized as follows. In Section 2, we first consider a generalized Newton method (GNM) with high-order convergence rate for solving a class of the linear complementarity problems (1.1). In Section 3, we analyze the performance and rate of convergence of the GNIM in detail. Some numerical experiments are given to illustrate that the GNM is efficient in Section 4. At last, we end the paper with some conclusions in Section 5.

2 The generalized Newton method

We first introduce some useful results which are vital for the equivalent reformulation as regards the $LCP(q, A)$. Furthermore, these conclusions contribute significantly to the analysis of the convergence rate of the generalized Newton iteration method which will be presented in the following.

Lemma 2.1 *The LCP*(q,A) (1.1) *is equivalent to the system of nonlinear equations*

$$(I+A)x+(I-A)|x|-q=0, \tag{2.1}$$

where A, q *are given matrix and vector in* (1.1), *respectively. I is an identity matrix with appropriate dimension,* $x\in\mathbb{R}^n$ *is a vector to be determined.*

Proof Let x be the solution of equation (2.1). Then it yields

$$|x|+x=A\bigl(|x|-x\bigr)+q. \tag{2.2}$$

Set

$$w:=|x|+x, \qquad u:=|x|-x. \tag{2.3}$$

It is apparent from (2.2) and (2.3) that

$$w=Au+q, \quad w\geq 0, u\geq 0, \quad \text{and} \quad w^Tu=0,$$

which means that u is the solution of *LCP*(q,A) (1.1).

On the other hand, we assume u is the solution of *LCP*(q,A) (1.1). It is obvious that

$$w\geq 0, \qquad u\geq 0, \qquad w^Tu=0, \quad \text{where } w:=Au+q.$$

We also set

$$w:=|x|+x, \qquad u:=|x|-x,$$

it turns out that

$$x=\frac{1}{2}(w-u)$$

satisfies equation (2.1). □

Now, we introduce the smooth function $F:\mathbb{R}^{n+1}\to\mathbb{R}^n$ by

$$F(z):=(I-A)\sqrt{x^2+\varepsilon^2\mathbf{e}}+(I+A)x-q, \tag{2.4}$$

where $z=(\varepsilon,x^T)^T\in\mathbb{R}^{n+1}$, ε is a positive variable, $\mathbf{e}$ is a vector with all elements equal 1, *i.e.*, $\mathbf{e}=(1,1,\dots,1)^T\in\mathbb{R}^n$,

$$\sqrt{x^2+\varepsilon^2\mathbf{e}}:=\Bigl[\sqrt{x_1^2+\varepsilon^2},\sqrt{x_2^2+\varepsilon^2},\dots,\sqrt{x_n^2+\varepsilon^2}\Bigr]^T\in\mathbb{R}^n. \tag{2.5}$$

We further define the nonlinear and differentiable function

$$\Psi(z) := \begin{pmatrix} \varepsilon \\ F(z) \end{pmatrix} \in \mathbb{R}^{n+1}. \tag{2.6}$$

Lemma 2.2 *LCP*(q, A) (1.1) *is equivalent to the nonlinear system*

$$\Psi(z) = 0, \tag{2.7}$$

where $\Psi(z)$ *is defined by* (2.6), $z = (\varepsilon, x^T)^T \in \mathbb{R}^{n+1}$.

Proof The conclusion can easily be drawn by equation (2.4) and Lemma 2.1. □

Lemma 2.3 *The Jacobi matrix of equation* (2.6) *is*

$$\Psi'(z) = \begin{pmatrix} 1 & \mathbf{0} \\ (I-A)g_\varepsilon & (I-A)D_x + A + I \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \tag{2.8}$$

where $g(\varepsilon) = (g_1, g_2, \ldots, g_n)^T \in \mathbb{R}^n$, $g_i = \frac{\varepsilon}{\sqrt{x_i^2+\varepsilon^2}}$, $D_x = \mathrm{Diag}(d_1, d_2, \ldots, d_n) \in \mathbb{R}^{n\times n}$, $d_i = \frac{x_i}{\sqrt{x_i^2+\varepsilon^2}}$, $i \in \mathbb{N}_n$, *and* $\mathbf{0} \in \mathbb{R}^{1\times n}$ *is a zero vector.*

Proof It follows from the derivatives for the variable z on both sides of equation (2.6) that

$$\Psi'(z) = \begin{pmatrix} 1 & \mathbf{0} \\ F'_\varepsilon & F'_x \end{pmatrix}. \tag{2.9}$$

According to equations (2.4) and (2.5), we immediately have

$$F'_\varepsilon = (I-A)\left(\frac{\varepsilon}{\sqrt{x_1^2+\varepsilon^2}}, \frac{\varepsilon}{\sqrt{x_2^2+\varepsilon^2}}, \ldots, \frac{\varepsilon}{\sqrt{x_n^2+\varepsilon^2}}\right)^T \in \mathbb{R}^n \tag{2.10}$$

and

$$F'_x = (I-A)\,\mathrm{Diag}\left(\frac{x_1}{\sqrt{x_1^2+\varepsilon^2}}, \frac{x_2}{\sqrt{x_2^2+\varepsilon^2}}, \ldots, \frac{x_n}{\sqrt{x_n^2+\varepsilon^2}}\right) + A + I \in \mathbb{R}^{n\times n}. \tag{2.11}$$

This completes the proof. □

Now, we show the generalized Newton iteration method for the nonlinear smooth system (2.7). A detailed description follows.

Algorithm 2.1 (The generalized Newton method)

Step 1 Input the initial guess $z^0 = (\varepsilon_0, (x^0)^T)^T$, give the matrix A and vector q and any small positive numbers $\sigma_1, \sigma_2 \in (0,1)$, preset a positive integer $m \geq 2$. Set $k := 0$.

Step 2 Compute $\Psi(z^k)$, the Jacobi matrix $\Psi'(z^k)$, and its inverse matrix $A_k := (\Psi'(z^k))^{-1}$.

Step 3 Set $z^{k,1} = z^k$, $j := 1$.

Step 4 Evaluate $\Psi(z^{k,j})$, update the vector sequence

$$z^{k,j+1} = z^{k,j} - A_k b,$$

then calculate $\Psi(z^{k,j+1})$, where $b = \Psi(z^{k,j})$.

Step 5 Set $j := j + 1$, $z^{k,j} = z^{k,j+1}$, $\Psi(z^{k,j}) = \Psi(z^{k,j+1})$, $y := A_k b$. If $j = m$, return to Step 6, otherwise go to Step 4.

Step 6 If $\|y\| < \sigma_1$ or $\|\Psi(z^{k,m})\| < \sigma_2$, let $z^{k,m} = z^*$, otherwise $k := k + 1$, return to Step 3.

The generalized Newton iteration method (GNIM) also can be written with the following iterative scheme:

$$\begin{cases} z^{k,0} = z^k, \\ z^{k,j} = z^{k,j-1} - (\Psi'(z^k))^{-1}\Psi(z^{k,j-1}), \quad j = 1, 2, \ldots, m, \\ x^{k+1} = x^{k,m}, \quad k = 0, 1, 2, \ldots. \end{cases} \tag{2.12}$$

Remark 2.1 From Lemma 2.2, we know that the iterative solution z^* generated by Algorithm 2.1 is also the solution u^* of $LCP(q, A)$ (1.1).

Remark 2.2 The update of parameter ε_k can be chosen with $\varepsilon_k = \varepsilon_{k-1}^m$. Since the positive integer m is selected at least greater than or equal to 2 in Algorithm 2.1, the positive sequence $\{\varepsilon_k\}_0^\infty$ declines monotonically and tends to zero.

Remark 2.3 Once we set $m = 1$, then the GNIM reduces to the classical Newton iterative method.

3 The analysis of convergence

Definition 3.1 ([28]) Let $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$, $x^* \in D$ is the solution of system $F(x) = 0$. There is a region $S \subset D$ for the point x^*, for any initial approximation $x^0 \in S$, if the iteration sequence $\{x^k, k = 0, 1, \ldots\}$ is always well defined and converges to x^*, we call it the attractive point of the iteration sequence.

The classical Newton iteration features a convergence rate of at least order two. We have the following results; for more details see [28] and the references therein.

Lemma 3.1 ([28]) *Let $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ be Fréchet differentiable on the open interval of $S_0 \in D$, and $F'(x^*)$ be nonsingular, where x^* is the solution of system $F(x) = 0$. Then the mapping $G(x) = x - F'(x)^{-1}F(x)$ is well defined on S for the closed sphere $S = \bar{S}(x^*, \delta) \subset S_0$. Moreover, if the inequality*

$$\|F'(x) - F'(x^*)\| \le \beta\|x - x^*\| \tag{3.1}$$

holds, where β is a constant, $x \in S$, then the classical Newton iteration method has at least a convergence rate of order two.

Lemma 3.2 ([28]) *Let $F : D \subset \mathbb{R}^n \to \mathbb{R}^n$ be Fréchet differentiable on the fixed-point $x^* \in$* int(D), *and the spectral radius of $F'(x^*)$*

$$\rho(F'(x^*)) = \sigma \le 1. \tag{3.2}$$

Then x^* is the attractive point of the iterative sequence $x^{k+1} = F(x^k)$ $(k = 0, 1, \ldots)$ for the open sphere $S = S(x^*, \delta) \subset D$ and any initial guess $x^0 \in S$.

Lemma 3.3 [28] *Let $F : D \subset \mathbb{R}^n \to \mathbb{R}^m$ be continuous and differentiable on the convex set $D_0 \subset D$, and satisfy*

$$\|F'(u) - F'(v)\| \leq \beta \|u - v\|^p, \quad u, v \in D_0. \tag{3.3}$$

Then we have

$$\|F(y) - F(x) - F'(x)(y - x)\| \leq \frac{\beta}{1 + p}\|y - x\|^{1+p}, \quad x, y \in D_0, \tag{3.4}$$

where the constants $\beta \geq 0$, $p \geq 0$.

We are now in a position to derive the main convergence result of the generalized Newton iteration method.

Theorem 3.1 *Let $\Psi : D \subset \mathbb{R}^{n+1} \to \mathbb{R}^{n+1}$ be Fréchet differentiable on the circle region of $S(z^*, \delta) \in D$, and $\Psi'(z^*)$ be nonsingular, where z^* is the solution of $\Psi(z) = 0$. Assume that for arbitrary $z \in S$, there is a constant $\beta > 0$ such that*

$$\|\Psi'(z) - \Psi'(z^*)\| \leq \beta \|z - z^*\| \tag{3.5}$$

holds. Then z^* is the attractive point of the iterative sequence $\{z^k\}_0^\infty$ generated by Algorithm 2.1, and

$$\|z^{k+1} - z^*\| \leq L_2 \|z^k - z^*\|^{m+1}, \tag{3.6}$$

where L_2 is a constant independent of iteration number k.

Proof To begin with, we consider the iterative equation (2.12) with the case $m = 2$, *i.e.*,

$$z^{k+1} = z^k - (\Psi'(z^k))^{-1}[\Psi(z^k) - \Psi(z^k - (\Psi'(z^k))^{-1}\Psi(z^k))], \quad k = 0, 1, \ldots. \tag{3.7}$$

By applying Lemma 3.1, $P(z) := z - (\Psi'(z))^{-1}\Psi(z)$ is well defined on $S_1 := \bar{S}(z^*, \delta_1) \subset S$ and

$$\|P(z) - z^*\| \leq \xi \|z^k - z^*\|^2, \quad z \in S_1, \tag{3.8}$$

where ξ is a positive constant. Hence, the mapping $M(z) = P(z) - (\Psi'(z))^{-1}\Psi(P(z))$ is well defined on the closed sphere $S_2 = \bar{S}(z^*, \delta_2) \subset S_1$, where $\delta_2 \leq \frac{\delta_1}{\xi}$ and

$$M'(z^*) = I - (\Psi'(z^*))^{-1}\Psi(P(z^*)) = 0.$$

So, $\rho(M'(z^*)) = 0 < 1$. It follows from Lemma 3.2 that z^* is the attractive point of equation (3.7).

On the other hand, note the nonsingularity of $\Psi'(z^*)$, hence $\|(\Psi'(z))^{-1}\| \leq \zeta$ for $z \in S_2$. Then it follows from Lemma 3.3 and the assumptions that

$$\begin{aligned}\|M(z) - z^*\| &\leq \|(\Psi'(z))^{-1}\| \|\Psi'(z)[P(z) - z^*] - \Psi(P(z))\| \\ &\leq \zeta [\|\Psi(P(z)) - \Psi(z^*) - \Psi'(z^*)(P(z) - z^*)\|\end{aligned}$$

$$\begin{aligned}
&+ \|(\Psi'(z^*) - \Psi'(z))(P(z) - z^*)\|] \\
&\le \zeta\left[\frac{1}{2}\beta\|P(z) - z^*\|^2 + \beta\|z - z^*\|\|P(z) - z^*\|\right] \\
&\le \beta\zeta\left[\frac{1}{2}\xi^2\|z - z^*\| + \xi\right]\|z - z^*\|^3 \\
&\le \beta\zeta\xi\left[\frac{1}{2}\xi\delta_2 + 1\right]\|z - z^*\|^3.
\end{aligned} \tag{3.9}$$

Let $L_0 := \beta\zeta\xi[\frac{1}{2}\xi\delta_2 + 1]$, evidently, it is a constant independent of the iteration number k.

When we choose $z^{k+1} = M(z^k)$ for the left of the inequality (3.9), it yields

$$\|z^{k+1} - z^*\| \le L_0\|z^k - z^*\|^3, \tag{3.10}$$

which implies (3.6) holds for $m = 2$.

Now, we state that the iterative scheme (2.12) has at least a convergence rate of order $m + 1$. The result will be shown by mathematical induction.

Noting that (3.9), one has the argument

$$\|z^{k,2} - z^*\| \le L_0\|z^k - z^*\|^3 = O(\|z^k - z^*\|^2). \tag{3.11}$$

Next, we assume that

$$\|z^{k,m-1} - z^*\| \le L_1\|z^k - z^*\|^m = O(\|z^k - z^*\|^m) \tag{3.12}$$

holds. Now, we will verify the statement

$$\|z^{k,m} - z^*\| \le L_2\|z^k - z^*\|^{m+1} = O(\|z^k - z^*\|^{m+1}), \tag{3.13}$$

where L_1, L_2 are constants independent of the iteration number k.

As a matter of fact, observing (2.12), also by Lemma 3.3 and (3.12), one obtains

$$\begin{aligned}
\|z^{k,m} - z^*\| &\le \|(\Psi'(z^k))^{-1}\|[\|\Psi(z^{k,m-1}) - \Psi(z^*) - \Psi'(z^*)(z^{k,m-1} - z^*)\| \\
&\quad + \|(\Psi'(z^*) - \Psi'(z^k))(z^{k,m-1} - z^*)\|] \\
&\le \zeta\left[\frac{1}{2}\beta\|z^{k,m-1} - z^*\|^2 + \beta\|z^k - z^*\|\|z^{k,m-1} - z^*\|\right] \\
&\le \zeta\beta\left[\frac{1}{2}L_1^2\|z^k - z^*\|^{m-1} + L_1\right]\|z^k - z^*\|^{m+1} \\
&\le \zeta\beta\left[\frac{1}{2}L_1^2\delta_2^{m-1} + L_1\right]\|z^k - z^*\|^{m+1} \\
&= L_2\|z^k - z^*\|^{m+1} = O(\|z^k - z^*\|^{m+1}),
\end{aligned} \tag{3.14}$$

where $L_2 := \zeta\beta[\frac{1}{2}L_1^2\delta_2^{m-1} + L_1]$, which completes the proof. □

4 Numerical experiments

In this section, we report some numerical results to illustrate the effectiveness of the GNIM for solving the linear complementarity problem. We compare the GNIM with the Fischer-based semismooth Newton method which we call FBSN (see [29, 30]), and the cosh-based smoothing Newton method which we call CBSN (see [31]) by the iteration step (denoted as 'IT'), the elapsed CPU time in seconds (denoted as 'CPU'), and the residual error (denoted as 'RES'). In actual computations, all runs are terminated if the current iterations satisfy

$$\mathrm{RES} := \left\| \Psi\left(z^k\right) \right\| < 10^{-14},$$

or if the number of iterations exceeds the prescribed number of iteration steps $k_{\max}$. The numerical experiments have been carried out by MATLAB R2011b (7.13), Intel(R) Core(TM) i7-2670QM, CPU 2.20GHZ, RAM 8.GB PC Environment, and Windows 7 operating system.

In fact, our approach can be considered as the version of speeding up on the basis of the smoothing Newton method. Thereby, high precision is the advantage of the GNIM. Once we set $m = 1$, the GNIM will reduce to the classical Newton method. But we know that the larger m may lead to the consumption of more CPU time since there is an increased number of inner iterations. We usually choose $m = 2$ in concrete tests, which also can ensure the rapid convergence rate. To confirm this judgment, we can observe the following examples.

Example 4.1 ([27]) We consider the linear complementarity problem (1.1) with

$$A = \mathrm{tridiag}(-1, 4, -1), \qquad q = (-1, \ldots, -1)^T.$$

Example 4.2 We consider the linear complementarity problem (1.1) with $A = (a_{ij})$, $i, j \in \mathbb{N}_n$, where $a_{ij} = \frac{i\delta_{ij}}{n}$, δ is the Kronecker delta ($\delta_{ii} = 1$, $\delta_{ij} = 0$, when $i \neq j$), and $q = (q_i)$, $i \in \mathbb{N}_n$ such that $q_i = -1$.

In these examples, we choose $m = 2$ for the GNIM, and $\rho = 0.15, \beta = 0.25, p = 3.0$ for the FBSN (for more details, see [29]). Especially, the RES will be regarded as 0 if RES $< 10^{-17}$ in the numerical results. The initial guess will be selected with $z^0_{(1)}$ or $z^0_{(2)}$ by $[\varepsilon_0, q^T]^T$ or the zero vector, respectively. First, we compare the performance of the numerical results of the GNIM with the CBSN by arranging different n. From Tables 1 and 2, we find the GNIM and the CBSN are all effective methods whose solution pairs (u^*, w^*) are also displayed in the last column of tables. However, from the aspects of the iteration number or CPU, the CBSN does not stand comparison with our method. In Table 3, we provide the numerical results for the GNIM and the CBSN with larger dimensions. Furthermore, the merit of the GNIM is reflected clearly and incisively.

It is observed from Table 4 that the GNIM is much more competitive than the FBSN for the large-scale linear complementarity problem. The numerical results of Example 4.2 is shown by Tables 5-7. Especially, from Table 7, we find that the FBSN is unable to obtain the convergence result in less iteration steps, such as within 11, for Example 4.2.

To sum up, by all numerical results, we illustrate the effectiveness of the GNIM.

Table 1 The GNIM numerical results for Example 4.1 with initial $\boldsymbol{z^0_{(1)}}$

		The performance of numerical results	The solution pairs ($\boldsymbol{u^*}, \boldsymbol{w^*}$)
$n=3$	It	3	$u* = (0.3571, 0.4286, 0.3571)$
	CPU	0.0068	
	RES	0	$w* = (0, 0, 0)$
$n=5$	It	3	$u* = (0.3654, 0.4615, 0.4808, 0.4615, 0.3654)$
	CPU	0.0073	
	RES	0	$w* = 1.0e{-}015 * (0.4441, 0, 0, 0, 0)$
$n=8$	It	3	$u* = (0.3660, 0.4641, 0.4902, 0.4967, 0.4967, 0.4902, 0.4641, 0.3660)$
	CPU	0.0074	
	RES	0	$w* = (0, 0, 0, 0, 0, 0, 0, 0)$
$n=10$	It	3	$u* = (0.3660, 0.4641, 0.4904, 0.4974, 0.4991, 0.4991, 0.4974, 0.4904, 0.4641, 0.3660)$
	CPU	0.0075	
	RES	0	$w* = 1.0e{-}015 * (-0.1110, 0.4441, -0.4441, 0, 0, 0, 0.2220, 0, 0, 0)$

Table 2 The CBSN numerical results for Example 4.1 with initial $\boldsymbol{z^0_{(1)}}$

		The performance of numerical results	The solution pairs ($\boldsymbol{u^*}, \boldsymbol{w^*}$)
$n=3$	It	4	$u* = (0.3571, 0.4286, 0.3571)$
	CPU	0.0104	
	RES	0	$w* = 1.0e{-}011 * (0.8880, -0.5869, 0.8880)$
$n=5$	It	4	$u* = (0.3654, 0.4615, 0.4808, 0.4615, 0.3654)$
	CPU	0.0106	
	RES	$6.1063e{-}016$	$w* = 1.0e{-}015 * (0, 0, 0, 0, -0.1110)$
$n=8$	It	4	$u* = (0.3660, 0.4641, 0.4902, 0.4967, 0.4967, 0.4902, 0.4641, 0.3660)$
	CPU	0.0113	
	RES	$6.2042e{-}016$	$w* = 1.0e{-}011 * (0.4839, -0.1609, -0.0001, 0, 0, -0.0001, -0.1608, 0.4839)$
$n=10$	It	3	$u* = (0.3660, 0.4641, 0.4904, 0.4974, 0.4991, 0.4991, 0.4974, 0.4904, 0.4641, 0.3660)$
	CPU	0.0115	
	RES	$6.2232e{-}016$	$w* = 1.0e{-}011 * (0.4835, -0.1607, -0.0001, 0, 0, 0, 0, -0.0001, -0.1607, 0.4835)$

Table 3 Numerical results for Example 4.1

Initials		$\boldsymbol{z^0_{(1)}}$	$\boldsymbol{z^0_{(1)}}$	$\boldsymbol{z^0_{(2)}}$	$\boldsymbol{z^0_{(2)}}$
Methods		**GNIM**	**CBSN**	**GNIM**	**CBSN**
$n=50$	It	2	3	3	4
	CPU	0.0084	0.0250	0.0119	0.0447
	RES	$5.7787e{-}016$	$1.7554e{-}016$	$5.3787e{-}016$	$1.8619e{-}016$
$n=100$	It	2	3	3	4
	CPU	0.0135	0.0230	0.0142	0.0364
	RES	$6.1815e{-}016$	$2.1678e{-}016$	$5.6610e{-}016$	$2.7616e{-}016$
$n=300$	It	2	3	3	4
	CPU	0.0376	0.0547	0.0562	0.0905
	RES	$6.2315e{-}016$	$2.2648e{-}016$	$5.7610e{-}016$	$2.9616e{-}016$

Table 4 Numerical results for Example 4.1

Initials		$\boldsymbol{z^0_{(1)}}$	$\boldsymbol{z^0_{(1)}}$	$\boldsymbol{z^0_{(2)}}$	$\boldsymbol{z^0_{(2)}}$
Methods		**GNIM**	**FBSN**	**GNIM**	**FBSN**
$n = 500$	It	2	11	3	11
	CPU	0.0886	0.3229	0.1366	0.3831
	RES	$6.1815e{-}016$	$6.6613e{-}016$	$5.6610e{-}016$	$1.1047e{-}015$
$n = 1000$	It	2	11	3	11
	CPU	0.5114	1.0758	0.7728	1.3069
	RES	$6.2113e{-}016$	$6.6723e{-}016$	$5.6720e{-}016$	$1.3027e{-}015$
$n = 3000$	It	2	11	3	11
	CPU	9.1418	26.6155	11.1484	11.5429
	RES	$6.2812e{-}016$	$6.6921e{-}016$	$5.5610e{-}016$	$1.2123e{-}015$
$n = 5000$	It	2	11	3	11
	CPU	42.9507	71.3675	66.0831	79.3341
	RES	$6.4615e{-}016$	$6.6990e{-}016$	$5.7810e{-}016$	$1.3056e{-}015$

Table 5 The GNIM numerical results for Example 4.2 with initial $\boldsymbol{z^0_{(1)}}$

		The performance of numerical results	**The solution pairs ($\boldsymbol{u^*}, \boldsymbol{w^*}$)**
$n = 3$	It	2	$u* = (3, 1.5, 1)$
	CPU	0.0067	
	RES	0	$w* = (0, 0, 0)$
$n = 5$	It	2	$u* = (5, 2.5, 1.6667, 1.25, 1)$
	CPU	0.0070	
	RES	0	$w* = 1.0e{-}015 * (0.2220, 0, 0, 0, 0)$
$n = 8$	It	2	$u* = (8, 4, 2.6667, 2, 1.6, 1.3333, 1.1429, 1)$
	CPU	0.0072	
	RES	0	$w* = (0, 0, 0, 0, 0, 0, 0, 0)$
$n = 10$	It	2	$u* = (10, 5, 3.3333, 2.5, 2, 1.6667, 1.4286, 1.25, 1.1111, 1)$
	CPU	0.0075	
	RES	0	$w* = 1.0e{-}015 * (0, 0.2220, -0.1110, 0, 0, 0, 0.2220, 0, 0, 0)$

Table 6 The CBSN numerical results for Example 4.2 with initial $\boldsymbol{z^0_{(1)}}$

		The performance of numerical results	**The solution pairs ($\boldsymbol{u^*}, \boldsymbol{w^*}$)**
$n = 3$	It	4	$u* = (3, 1.5, 1)$
	CPU	0.0155	
	RES	0	$w* = (0, 0, 0)$
$n = 5$	It	4	$u* = (5, 2.5, 1.6667, 1.25, 1)$
	CPU	0.0161	
	RES	0	$w* = 1.0e{-}015 * (0.4441, 0, 0, 0, 0)$
$n = 8$	It	4	$u* = (8, 4, 2.6667, 2, 1.6, 1.3333, 1.1429, 1)$
	CPU	0.0163	
	RES	$2.2204e{-}016$	$w* = 1.0e{-}011 * (-0.1110, 0.2220, 0, 0, 0.2220, 0, 0, 0)$
$n = 10$	It	4	$u* = (10, 5, 3.3333, 2.5, 2, 1.6667, 1.4286, 1.25, 1.1111, 1)$
	CPU	0.0147	
	RES	$8.8818e{-}016$	$w* = 1.0e{-}014 * (-0.1221, 0.0444, -0.0111, 0.0222, 0, -0.0111, 0.0222, 0, 0, 0)$

Table 7 Numerical results for Example 4.2

Initials		$z^0_{(1)}$	$z^0_{(1)}$	$z^0_{(2)}$	$z^0_{(2)}$
Methods		**GNIM**	**FBSN**	**GNIM**	**FBSN**
$n = 500$	It	2	11	3	11
	CPU	0.0957	0.3025	0.1421	0.3308
	RES	2.3182*e*–015	1.7073*e*+001	2.1006*e*–015	1.8073*e*+001
$n = 1000$	It	2	11	3	11
	CPU	0.5016	0.9461	0.6327	1.1754
	RES	6.6438*e*–014	2.4129*e*+001	6.6351*e*–014	2.6249*e*+001

5 Conclusion

In this paper, we establish the generalized Newton method (GNM) for solving the large-scale linear complementarity problem. The GNM features the high-order convergence rate, at least convergence order $m + 1$, which has been verified from theoretical analysis section in detail. The new strategy will increase the efficiency remarkably. In fact, it can be regarded as an accelerated process for the classical Newton approach. Experimental tests provide the comparison of the numerical performance for some existing efficient methods, which testify to the viability of the proposed approach.

Competing interests

The authors declare that there is no conflict of interests regarding the publication of this article.

Authors' contributions

All authors contributed equally and significantly in writing this article. All authors read and approved the final manuscript.

Acknowledgements

The Project Supported by National Natural Science Foundation of China (Grant Nos. 11071041, 11201074), Fujian Natural Science Foundation (Grant Nos. 2015J01578, 2013J01006) and The Outstanding Young Training Plan of Fujian Province universities (Grant No. 15kxtz13, 2015).

Received: 8 October 2015 Accepted: 8 December 2015 Published online: 23 December 2015

References

1. Cottle, RW, Pang, J-S, Stone, RE: The Linear Complementarity Problem. Academic Press, San Diego (1992)
2. Murty, KG: Linear Complementarity, Linear and Nonlinear Programming. Heldermann, Berlin (1988)
3. Lemke, CE: Bimatrix equilibrium points and mathematical programming. Manag. Sci. **11**, 681-689 (1965)
4. Scarf, H: The approximation of fixed points of a continuous mapping. SIAM J. Appl. Math. **15**, 1328-1342 (1967)
5. Eaves, BC, Scarf, H: The solution of systems of piecewise linear equations. Math. Oper. Res. **1**, 1-27 (1976)
6. Eaves, BC, Lemke, CE: Equivalence of LCP and PLS. Math. Oper. Res. **6**, 475-484 (1981)
7. Cryer, C: The solution of a quadratic programming using systematic overrelaxation. SIAM J. Control Optim. **9**, 385-392 (1971)
8. Tseng, P: On linear convergence of iterative method for the variational inequality problem. J. Comput. Appl. Math. **60**, 237-252 (1995)
9. Bai, Z-Z: Modulus-based matrix splitting iteration methods for linear complementarity problems. Numer. Linear Algebra Appl. **17**, 917-933 (2010)
10. Bai, Z-Z, Evans, DJ: Matrix multisplitting methods with applications to linear complementarity problems: parallel synchronous and chaotic methods. In: Calculateurs Parallèles Réseaux et Systèmes Répartis, vol. 13, pp. 125-141 (2001)
11. Bai, Z-Z, Evans, DJ: Matrix multisplitting methods with applications to linear complementarity problems: parallel asynchronous methods. Int. J. Comput. Math. **79**, 205-232 (2002)
12. Bai, Z-Z, Zhang, L-L: Modulus-based synchronous multisplitting iteration methods for linear complementarity problems. Numer. Linear Algebra Appl. **20**, 425-439 (2013)
13. Bai, Z-Z: On the convergence of the multisplitting methods for the linear complementarity problem. SIAM J. Matrix Anal. Appl. **21**, 67-78 (1999)
14. Bai, Z-Z, Evans, DJ: Matrix multisplitting relaxtion methods for linear complementarity problem. Int. J. Comput. Math. **63**, 309-326 (1997)
15. Bai, Z-Z: Experimental study of the asynchronous multisplitting relaxtation methods for linear complementarity problems. J. Comput. Math. **20**, 561-574 (2002)
16. Bai, Z-Z, Huang, Y-G: A class of asynchronous iterations for the linear complementarity problem. J. Comput. Math. **21**, 773-790 (2003)
17. Bai, Z-Z, Evans, DJ: Parallel chaotic multisplitting iterative methods for the large sparse linear complementarity problem. J. Comput. Math. **19**, 281-292 (2001)

18. Zhang, L-L: Two-step modulus based matrix splitting iteration for linear complementarity problems. Numer. Algorithms **57**, 83-99 (2011)
19. Zheng, N, Yin, J-F: Accelerated modulus-based matrix splitting iteration methods for linear complementarity problem. Numer. Algorithms **64**, 245-262 (2013)
20. Bai, Z-Z, Zhang, L-L: Modulus-based synchronous multisplitting iteration methods for large sparse linear complementarity problem. Numer. Linear Algebra Appl. **20**, 425-439 (2013)
21. Ljiljana, C, Vladimir, K: A note on the convergence of the MSMAOR method for linear complementarity problems. Numer. Linear Algebra Appl. **21**, 534-539 (2014)
22. Foutayeni, YEL, Khaladi, M: Using vector divisions in solving the linear complementarity problem. J. Comput. Appl. Math. **236**, 1919-1925 (2012)
23. Qi, L, Sun, J: A nonsmooth version of Newton's method. Math. Program. **58**, 353-368 (1993)
24. Jiang, H-Y, Qi, L: A new nonsmooth equations approach to nonlinear complementarity problems. SIAM J. Control Optim. **35**, 178-193 (1997)
25. Chen, X, Nashed, Z, Qi, L: Smoothing methods and semismooth methods for nondifferentiable operator equations. SIAM J. Numer. Anal. **38**, 1200-1216 (2000)
26. Pang, J-S: Newton's method for *B*-differentiable equations. Math. Oper. Res. **15**, 311-341 (1990)
27. Sun, Z, Zeng, J-P: A monotone semismooth Newton type method for a class of complementarity problems. J. Comput. Appl. Math. **235**, 1261-1274 (2011)
28. Huang, X-D, Zeng, Z-G, Ma, Y-N: The Theory and Methods for Nonlinear Numerical Analysis, pp. 57-59. Wuhan University Press, Wuhan (2004)
29. Luca, D, Fancchinei, F, Kanzow, C: A semismooth equation approach to the solution of nonlinear complementarity problems. Math. Program. **75**, 407-439 (1996)
30. Fischer, A: A special Newton-type optimization method. Optimization **24**, 269-284 (1992)
31. Yu, Z, Qin, Y: A cosh-based smoothing Newton method for P0 nonlinear complementarity problem. Nonlinear Anal., Real World Appl. **12**, 875-884 (2011)