

## RESEARCH

## Open Access

# Modified Newton-type methods for the NCP by using a class of one-parametric NCP-functions

Weisong Xie and Zijun Deng\*

\*Correspondence: [dzjxsd@126.com](mailto:dzjxsd@126.com)  
Department of Mathematics, Tianjin University, Tianjin, 300072, China**Abstract**

In this paper, we propose a new Newton-type method for solving the nonlinear complementarity problem (NCP) based on a class of one-parametric NCP-functions, where an approximate Newton direction can be obtained by solving a modified Newton equation in each iteration. The method is shown to be globally convergent without any additional assumption. To investigate the fast convergence of this class of methods, we propose a modified version of the proposed method and show the method is globally and locally superlinearly convergent. The preliminary numerical results show the effectiveness of the modified method.

**Keywords:** nonlinear complementarity problem; NCP-function; generalized Newton method; global convergence; superlinear convergence

## 1 Introduction

Consider the nonlinear complementarity problem  $\text{NCP}(F)$

$$x \geq 0, F(x) \geq 0, \quad x^T F(x) = 0,$$

where  $F : \mathcal{R}^n \rightarrow \mathcal{R}^n$  is a continuously differentiable function. We assume that  $F$  is a  $P_0$ -function throughout this paper. It is well known that  $\text{NCP}(F)$  can be reformulated as a system of nonsmooth equations, where the so-called NCP-function plays an important role in this class of methods.

**Definition 1** A function  $\phi : \mathcal{R}^2 \rightarrow \mathcal{R}$  is called an NCP-function if it satisfies

$$\phi(a, b) = 0 \iff a \geq 0, \quad b \geq 0, \quad ab = 0.$$

Over the past two decades, a variety of NCP-functions have been studied (see, for example, [1–7]). Among them, a popular NCP-function is the well-known Fischer-Burmeister NCP-function [3] defined as

$$\phi_{\text{FB}}(a, b) = \sqrt{a^2 + b^2} - a - b.$$

In this paper, we use a family of NCP-functions based on the FB function, which was introduced by Kanzow and Kleinmichel [6],

$$\phi_\lambda(a, b) = \sqrt{(a - b)^2 + \lambda ab} - a - b, \tag{1}$$

where  $\lambda$  is a fixed parameter such that  $\lambda \in (0, 4)$ . In the case of  $\lambda = 2$ , the NCP-function  $\phi_\lambda$  obviously reduces to the Fischer-Burmeister function.

By using  $\phi_\lambda$  defined by (1), the NCP is equivalent to a system of nonsmooth equations

$$\Phi_\lambda(x) = \begin{bmatrix} \phi_\lambda(x_1, F_1(x)) \\ \vdots \\ \phi_\lambda(x_n, F_n(x)) \end{bmatrix} = 0.$$

Let  $\theta_\lambda(x) = \frac{1}{2} \|\Phi_\lambda(x)\|^2$ . Then solving NCP( $F$ ) is equivalent to solving the unconstrained minimization  $\min_{x \in \mathbb{R}^n} \theta_\lambda(x)$  with the optimal value 0.

Kanzow and Kleinmichel [6] studied the properties of  $\Phi_\lambda$  and  $\theta_\lambda$  and proposed the corresponding semismooth Newton method. Their method first attempted to use the Newton direction, but if the Newton equation is unsolvable or the Newton direction is not a direction of sufficient decrease for  $\theta_\lambda$ , then it switches to the steepest descent direction. In this paper, we propose a Newton-type method for the  $P_0$ -NCP( $F$ ), where, in each iteration, we need to construct an approximation of  $\partial\Phi_\lambda(x)$  (the Clarke subdifferential of  $\Phi_\lambda$  at  $x$ , which is defined in the next section), which is nonsingular, and hence the direction-finding problem can be solved only by solving a system of perturbed Newton equations. We show that the proposed method is globally convergent without any additional assumption. The proposed method is similar to the one discussed by Yamashita and Fukushima [8], where the NCP-function  $\phi_{FB}$  was used. Since  $\phi_{FB}$  is a special case of  $\phi_\lambda$ , the proposed method can be used more widely. However, it is hard for us to discuss the locally fast convergence of the proposed method. In order to investigate the locally fast convergence of this class of methods, we revise the proposed method. We show that the modified method is globally and locally superlinearly convergent. The preliminary numerical results show the effectiveness of the modified method.

## 2 Preliminaries

In this section, we recall some basic concepts and known results.

**Definition 2**  $F : \mathcal{R}^n \rightarrow \mathcal{R}^n$  is called a  $P_0$ -function if

$$\max_{\substack{1 \leq i \leq n \\ x_i \neq y_i}} (x_i - y_i)(F_i(x) - F_i(y)) \geq 0, \quad \forall x, y \in \mathcal{R}^n, x \neq y.$$

**Definition 3** A matrix  $M \in \mathcal{R}^{n \times n}$  is a  $P_0$ -matrix if each of its principal minors is nonnegative.

It is known that the Jacobian of every continuously differentiable  $P_0$ -function is a  $P_0$ -matrix. The following theorem will play an important role in our analysis. Notice that, for a vector  $a$ ,  $D_a$  denotes the diagonal matrix with the  $i$ th diagonal element being  $a_i$ .

**Theorem 4** (see [8]) *Let  $M$  be a  $P_0$ -matrix,  $D_a$  and  $D_b$  be negative definite diagonal matrices. Then  $D_a + D_bM$  is nonsingular.*

Let  $\Phi : \mathcal{R}^n \rightarrow \mathcal{R}^n$  be locally Lipschitz continuous; by Rademacher's theorem,  $\Phi$  is differentiable almost everywhere.

**Definition 5** Let  $D_\Phi$  denote the set  $\{x \in \mathcal{R}^n | \Phi \text{ is differentiable at } x\}$ , then the  $B$ -subdifferential of  $\Phi$  at  $x$  is defined as

$$\partial_B \Phi(x) = \left\{ v \in \mathcal{R}^{n \times n} | v = \lim_{\substack{x^k \in D_\Phi \\ x^k \rightarrow x}} \Phi'(x^k) \right\}.$$

The Clarke subdifferential of  $\Phi$  at  $x$  is defined as

$$\partial \Phi(x) = co \partial_B \Phi(x),$$

where  $co$  denotes the convex hull of a set.

By the definition of  $\Phi_\lambda$ , we know that  $\Phi_\lambda$  is not differentiable at  $x$  if  $x_i = 0 = F_i(x)$  for some  $i$ . However, since  $\Phi_\lambda$  is locally Lipschitz continuous [6, Lemma 2.1],  $\partial_B \Phi_\lambda(x)$  is nonempty at every  $x \in \mathcal{R}^n$ . But how to specify the set  $\partial_B \Phi_\lambda(x)$  exactly at  $x$  where  $\nabla \Phi_\lambda(x)$  does not exist?

To solve this problem, we construct two mappings  $\tilde{\mathcal{H}}$  and  $\hat{\mathcal{H}}$  which approximate  $\partial_B \Phi_\lambda$ . For a set  $X$ , we denote the power set of  $X$  by  $P(X)$ .

Define the mapping  $\tilde{\mathcal{H}} : \mathcal{R}^n \rightarrow P(\mathcal{R}^{n \times n})$  as

$$\tilde{\mathcal{H}}(x) = \{ \tilde{H} \in \mathcal{R}^{n \times n} | \tilde{H} = D_{\tilde{a}} + D_{\tilde{b}} F'(x), (\tilde{a}, \tilde{b}) \in \tilde{\Omega}(x) \},$$

where  $\tilde{\Omega} : \mathcal{R}^n \rightarrow P(\mathcal{R}^{2n})$  is given by

$$\tilde{\Omega}(x) = \{ (\tilde{a}, \tilde{b}) \in \mathcal{R}^{2n} | (\tilde{a}_i, \tilde{b}_i) \in \tilde{\Omega}_i(x), i = 1, 2, \dots, n \}$$

with

$$\tilde{\Omega}_i(x) = \begin{cases} \{ (\tilde{a}_i, \tilde{b}_i) \in \mathcal{R}^2 | (\tilde{a}_i + 1)^2 + (\tilde{b}_i + 1)^2 \leq C_\lambda \}, & \text{if } x_i = 0 = F_i(x), \\ \{ (\tilde{a}_i, \tilde{b}_i) \in \mathcal{R}^2 | \tilde{a}_i = \hat{a}_i, \tilde{b}_i = \hat{b}_i \}, & \text{otherwise.} \end{cases} \quad (2)$$

Here,  $C_\lambda$  denotes the constant  $2 - \frac{\lambda(4-\lambda)}{8}$ , and

$$\hat{a}_i = \frac{2(x_i - F_i(x)) + \lambda F_i(x)}{2\sqrt{(x_i - F_i(x))^2 + \lambda x_i F_i(x)}} - 1,$$

$$\hat{b}_i = \frac{-2(x_i - F_i(x)) + \lambda x_i}{2\sqrt{(x_i - F_i(x))^2 + \lambda x_i F_i(x)}} - 1.$$

In the following, we define  $\hat{\mathcal{H}}$  similarly to  $\tilde{\mathcal{H}}$ , which is a subset of  $\tilde{\mathcal{H}}$ .

The mapping  $\hat{\mathcal{H}} : \mathcal{R}^n \rightarrow P(\mathcal{R}^{n \times n})$  is defined by

$$\hat{\mathcal{H}}(x) = \{ \hat{H} \in \mathcal{R}^{n \times n} | \hat{H} = D_{\hat{a}} + D_{\hat{b}} F'(x), (\hat{a}, \hat{b}) \in \hat{\Omega}(x) \},$$

where  $\hat{\Omega} : \mathcal{R}^n \rightarrow P(\mathcal{R}^{2n})$  is defined by

$$\hat{\Omega}(x) = \{(\hat{a}, \hat{b}) \in \mathcal{R}^{2n} \mid (\hat{a}, \hat{b}) = (g(x, z), h(x, z)), z \in Z(x)\}.$$

Here  $Z(x) = \{z \in \mathcal{R}^n \mid z_i \neq 0, \text{ if } i \in \beta\}$ , and  $\beta$  denotes the set  $\{i \mid x_i = 0 = F_i(x)\}$ . The components of a vector  $g(x, z)$  are given by

$$g_i(x, z) = \begin{cases} \frac{2(z_i - \nabla F_i^T(x)z) + \lambda \nabla F_i^T(x)z}{2\sqrt{(z_i - \nabla F_i^T(x)z)^2 + \lambda z_i \nabla F_i^T(x)z}} - 1, & \text{if } x_i = 0 = F_i(x), \\ \frac{2(x_i - F_i(x)) + \lambda F_i(x)}{2\sqrt{(x_i - F_i(x))^2 + \lambda x_i F_i(x)}} - 1, & \text{otherwise;} \end{cases}$$

and the components of a vector  $h(x, z)$  are given by

$$h_i(x, z) = \begin{cases} \frac{-2(z_i - \nabla F_i^T(x)z) + \lambda z_i}{2\sqrt{(z_i - \nabla F_i^T(x)z)^2 + \lambda z_i \nabla F_i^T(x)z}} - 1, & \text{if } x_i = 0 = F_i(x), \\ \frac{-2(x_i - F_i(x)) + \lambda x_i}{2\sqrt{(x_i - F_i(x))^2 + \lambda x_i F_i(x)}} - 1, & \text{otherwise.} \end{cases}$$

**Remark** From (2), we find that, for every  $x \in \mathcal{R}^n$ ,  $(\tilde{a}, \tilde{b}) \in \tilde{\Omega}(x)$  satisfies  $-\sqrt{C_\lambda} - 1 \leq \tilde{a}_i, \tilde{b}_i \leq 0$  (see [6, Proposition 2.6]), and  $\tilde{a}_i, \tilde{b}_i$  do not vanish simultaneously. It is the same with elements in  $\hat{\mathcal{H}}$ .

The mappings  $\tilde{\mathcal{H}}$  and  $\hat{\mathcal{H}}$  have the following property which will play an important role in our analysis.

**Theorem 6** For an arbitrary  $x \in \mathcal{R}^n$ , we have  $\hat{\mathcal{H}}(x) \subseteq \partial_B \Phi_\lambda(x) \subseteq \tilde{\mathcal{H}}(x)$ .

*Proof*  $\partial_B \Phi_\lambda(x) \subseteq \tilde{\mathcal{H}}(x)$  was shown in [6, Proposition 2.5]. Hence, we prove  $\hat{\mathcal{H}}(x) \subseteq \partial_B \Phi_\lambda(x)$  in the following.

For an arbitrary  $\hat{H} \in \hat{\mathcal{H}}(x)$ , we shall build a sequence of points  $\{y^k\}$  where  $\Phi_\lambda$  is differentiable at every  $y^k$  and such that  $\nabla \Phi_\lambda(y^k)^T$  tends to  $\hat{H}$ ; then the theorem will be obtained by the definition of  $B$ -subdifferential.

Let  $y^k = x + \varepsilon^k z$ , where  $z \in Z(x)$  and  $\{\varepsilon^k\}$  is a sequence of positive numbers converging to 0. If  $i \notin \beta$ , either  $x_i \neq 0$  or  $F_i(x) \neq 0$ , and  $z_i \neq 0$  for all  $i \in \beta$ .

We can see, by continuity, that if  $\varepsilon^k$  is small enough, then for each  $i$ , either  $y_i^k \neq 0$  or  $F_i(y^k) \neq 0$ , so  $\Phi_\lambda$  is differentiable at  $y^k$ . If  $i \notin \beta$ , by continuity, the  $i$ th row of  $\nabla \Phi_\lambda(y^k)^T$  tends to the  $i$ th row of  $\hat{H}$ . So, we only concern the case of  $i \in \beta$ .

From [6, Proposition 2.5], we know that the  $i$ th row of  $\nabla \Phi_\lambda(y^k)^T$  is

$$(a_i(y^k) - 1)e_i^T + (b_i(y^k) - 1)\nabla F_i(y^k)^T, \tag{3}$$

where

$$a_i(y^k) = \frac{2(\varepsilon^k z_i - F_i(y^k)) + \lambda F_i(y^k)}{2\sqrt{(\varepsilon^k z_i - F_i(y^k))^2 + \lambda \varepsilon^k z_i F_i(y^k)}},$$

$$b_i(y^k) = \frac{-2(\varepsilon^k z_i - F_i(y^k)) + \lambda \varepsilon^k z_i}{2\sqrt{(\varepsilon^k z_i - F_i(y^k))^2 + \lambda \varepsilon^k z_i F_i(y^k)}}.$$

By Taylor-expansion, we have, for each  $i \in \beta$ ,

$$F_i(y^k) = F_i(x) + \varepsilon^k \nabla F_i(\xi^k)^T z = \varepsilon^k \nabla F_i(\xi^k)^T z, \quad \text{with } \xi^k \rightarrow x. \tag{4}$$

Substituting (4) into (3) and passing to limit, we have, by the continuity of  $\nabla F$ , that the rows of  $\nabla \Phi_\lambda(y^k)^T$  tend to the corresponding rows of  $\hat{H}$  when  $i \in \beta$ . Hence,  $\nabla \Phi_\lambda(y^k)^T$  tends to  $\hat{H}$ .  $\square$

In this paper, we present two algorithms. The first one, which is presented in Section 3, uses matrices obtained by perturbing  $\tilde{H} \in \tilde{\mathcal{H}}$ . We will establish its global convergence. While in Section 4, we present another algorithm based on  $\hat{H} \in \hat{\mathcal{H}}$ , which is a restricted version of the first one. The second algorithm can be superlinearly convergent.

### 3 Algorithm and global convergence

Considering the Newton-type method, the direction-finding problem is solved by  $\tilde{H}d = -\Phi_\lambda(x^k)$ , where  $\tilde{H} \in \tilde{\mathcal{H}}(x^k)$ . However,  $\tilde{H}$  is not necessarily nonsingular. In this section, we will perturb  $\tilde{H}$  to  $\tilde{G}$ , which is nonsingular. Then a search direction can be obtained by solving  $\tilde{G}d = -\Phi_\lambda(x^k)$ . Now, let us construct  $\tilde{G}$  as follows.

First, mapping  $\Lambda_i : \mathcal{R}^{n+2} \rightarrow P(\mathcal{R}^2)$ ,  $i = 1, 2, \dots, n$  are defined by

$$\Lambda_i(x, a_i, b_i) = \begin{cases} \left\{ (\bar{a}_i, \bar{b}_i) \in \mathcal{R}^2 \left| \begin{array}{l} \bar{a}_i = \frac{\sigma(\theta_\lambda(x))}{b_i} \\ \bar{b}_i = 0 \end{array} \right. \right\}, & \text{if } -\varepsilon < a_i \text{ and } b_i \leq -\varepsilon, \\ \left\{ (\bar{a}_i, \bar{b}_i) \in \mathcal{R}^2 \left| \begin{array}{l} \bar{a}_i = \tau \frac{\sigma(\theta_\lambda(x))}{b_i} \\ \bar{b}_i = (1 - \tau) \frac{\sigma(\theta_\lambda(x))}{a_i} \\ \tau \in [0, 1] \end{array} \right. \right\}, & \text{if } a_i \leq -\varepsilon \text{ and } b_i \leq -\varepsilon, \\ \left\{ (\bar{a}_i, \bar{b}_i) \in \mathcal{R}^2 \left| \begin{array}{l} \bar{a}_i = 0 \\ \bar{b}_i = \frac{\sigma(\theta_\lambda(x))}{a_i} \end{array} \right. \right\}, & \text{if } a_i \leq -\varepsilon \text{ and } -\varepsilon < b_i, \end{cases}$$

where  $\varepsilon \in (0, 1 - \sqrt{C_\lambda/2})$ , and  $\sigma : \mathcal{R}^+ \rightarrow \mathcal{R}^+$  is a nondecreasing continuous function such that  $\sigma(0) = 0$  and  $\sigma(t) > 0$  for all  $t > 0$ .

Because  $\varepsilon \in (0, 1 - \sqrt{C_\lambda/2})$ , it is obvious that for  $(a, b) \in \tilde{\Omega}(x)$ , the case of  $-\varepsilon < a_i$  and  $-\varepsilon < b_i$  will not happen.

In the following, we construct  $\tilde{G}$  as

$$\tilde{G} = D_{\tilde{p}} + D_{\tilde{q}}F'(x),$$

where  $\tilde{p}$  and  $\tilde{q}$  are vectors such that

$$(\tilde{p}_i, \tilde{q}_i) = (\bar{a}_i + \tilde{a}_i, \bar{b}_i + \tilde{b}_i), \quad i = 1, 2, \dots, n, \tag{5}$$

with  $(\tilde{a}, \tilde{b}) \in \tilde{\Omega}(x)$ , and  $(\bar{a}_i, \bar{b}_i) \in \Lambda_i(x, \tilde{a}_i, \tilde{b}_i)$ ,  $i = 1, 2, \dots, n$ .

If  $\theta_\lambda(x) > 0$ , the definition of  $\Lambda_i$  and (5) imply that both  $D_{\tilde{p}}$ ,  $D_{\tilde{q}}$  are negative definite matrices. Furthermore, we define  $\tilde{G} : \mathcal{R}^n \rightarrow P(\mathcal{R}^{n \times n})$  as follows:

$$\tilde{G}(x) = \left\{ \tilde{G} \in \mathcal{R}^{n \times n} \left| \begin{array}{l} \tilde{G} = D_{\tilde{p}} + D_{\tilde{q}}F'(x), (\tilde{p}, \tilde{q}) \text{ is defined by (5)} \\ \text{with } (\tilde{a}, \tilde{b}) \in \tilde{\Omega}(x) \text{ and } (\bar{a}_i, \bar{b}_i) \in \Lambda_i(x, \tilde{a}_i, \tilde{b}_i) \\ \text{for } i = 1, 2, \dots, n \end{array} \right. \right\}.$$

It is obvious that  $\tilde{G} = D_{\tilde{p}} + D_{\tilde{q}}F'(x)$  and  $\tilde{H} = D_{\tilde{a}} + D_{\tilde{b}}F'(x)$  are closely related.  $\tilde{G}$  is nonsingular under proper conditions.

**Theorem 7** *If  $x$  is not a solution of NCP( $F$ ), i.e.,  $\theta_\lambda(x) > 0$ , then every  $\tilde{G} \in \tilde{\mathcal{G}}(x)$  is nonsingular.*

*Proof* For every  $\tilde{G} \in \tilde{\mathcal{G}}(x)$ , if  $\theta_\lambda(x) > 0$ , then it follows from the definition of  $\tilde{G}$  that  $D_{\tilde{p}}$  and  $D_{\tilde{q}}$  are negative definite matrices.

Since  $F$  is a  $P_0$ -function, the Jacobian of  $F$  is a  $P_0$ -matrix. So,  $F'(x^k)$  is a  $P_0$ -matrix. Hence, by Theorem 4,  $\tilde{G}$  is nonsingular.  $\square$

By the mapping  $\tilde{\mathcal{G}}$ , we define  $\tilde{\Delta} : \mathcal{R}^n \rightarrow P(\mathcal{R}^n)$  as

$$\tilde{\Delta}(x) = \{d \in \mathcal{R}^n \mid \tilde{G}d = -\Phi_\lambda(x), \tilde{G} \in \tilde{\mathcal{G}}(x)\}.$$

It is easy to see that  $\tilde{\Delta}(x)$  is nonempty for every  $x$  such that  $\theta_\lambda(x) > 0$ . Now we give the first algorithm.

**Algorithm 1**

- Step 1. Initialization: choose  $\lambda \in (0, 4)$ ,  $x^0 \in \mathcal{R}^n$ ,  $\rho \in (0, 0.5)$ ,  $\beta \in (0, 1)$ , and set  $k := 0$ .
- Step 2. Termination criterion: if  $\theta_\lambda(x) = 0$ , stop. Otherwise, go to Step 3.
- Step 3. Search direction calculation: find a vector  $d^k \in \tilde{\Delta}(x^k)$ .
- Step 4. Line search: let  $m$  be the smallest nonnegative integer such that

$$\theta_\lambda(x^k + \beta^m d^k) - \theta_\lambda(x^k) \leq \beta^m \rho \nabla \theta_\lambda(x^k)^T d^k.$$

- Step 5. Update: set  $x^{k+1} := x^k + t_k d^k$ , where  $t_k = \beta^m$ ,  $k := k + 1$ , and go to Step 2.

It is obvious that if  $\theta_\lambda(x^k) = 0$ , then  $x^k$  is a solution of NCP( $F$ ). Next, we will prove the global convergence of Algorithm 1. First, we show that every  $d \in \tilde{\Delta}(x)$  is a descent direction of  $\theta_\lambda$  at  $x$ .

**Lemma 8** (see [8, Lemma 3.2]) *If  $x$  is not a solution of NCP( $F$ ), i.e.,  $\theta_\lambda(x) > 0$ , then every  $d \in \tilde{\Delta}(x)$  satisfies the descent condition for  $\theta_\lambda$ , i.e.,  $\nabla \theta_\lambda(x)^T d < 0$ .*

**Theorem 9** *Every accumulation point of a sequence  $\{x^k\}$  generated by Algorithm 1 is a solution of NCP( $F$ ).*

*Proof* Owing to Step 4,  $\{\theta_\lambda(x^k)\}$  is decreasing monotonically and nonnegative. It must converge to some  $\theta_\lambda^* \geq 0$ . We assume  $\theta_\lambda^* > 0$ . Let  $x^*$  be an accumulation point of  $\{x^k\}$  and  $\{x^k\}_{k \in \mathcal{K}}$  be a subsequence converging to  $x^*$ .

$\tilde{\Delta}$  is uniformly compact near  $x^*$  and closed at  $x^*$  (see [8, Lemma 3.4]), we assume, without loss of generality, that  $\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}}} d^k = d^* \in \tilde{\Delta}(x^*)$ . From Lemma 8, we will get the contradiction if we can prove  $\nabla \theta_\lambda(x^*)^T d^* = 0$ . This can be obtained by considering the following two cases:

- Suppose that  $\inf\{t_k\} \geq t > 0$ . Then we have

$$\theta_\lambda(x^k + t_k d^k) - \theta_\lambda(x^k) \leq t_k \rho \nabla \theta_\lambda(x^k)^T d^k \leq 0.$$

It is obvious that  $\nabla\theta_\lambda(x^*)^T d^* = 0$  is satisfied.

- Suppose that  $\inf\{t_k\} = 0$ . In this case, we assume  $\lim_{k \in \mathcal{K}} t_k = 0$  without loss of generality. By line search, we have

$$\frac{\theta_\lambda(x^k + \frac{t_k}{\beta}d^k) - \theta_\lambda(x^k)}{\frac{t_k}{\beta}} > \rho \nabla\theta_\lambda(x^k)^T d^k,$$

taking the limit yields  $\nabla\theta_\lambda(x^*)^T d^* \geq \rho \nabla\theta_\lambda(x^*)^T d^*$ . Since  $\rho \in (0, 0.5)$ , we have  $\nabla\theta_\lambda(x^*)^T d^* \geq 0$ . Hence,  $\nabla\theta_\lambda(x^*)^T d^* = 0$ .

We get the contradiction. The proof is complete. □

#### 4 Modified algorithm and fast convergence

In the above section, we established global convergence of Algorithm 1. It determines a search direction based on  $\tilde{\mathcal{H}}$  which contains the generalized Jacobian  $\partial_B \Phi_\lambda(x)$ . However, it is hard for us to show the superlinear convergence of Algorithm 1. In the following, we should modify the search direction properly to accelerate the convergence of algorithm. By the definition of  $\hat{\mathcal{H}}$ , we know that  $\hat{H} \in \hat{\mathcal{H}}$  is not necessarily nonsingular. Can we perturb  $\hat{H}$  similar to  $\tilde{H}$ ? Next, we give a positive answer to this question.

Define  $\hat{G}$  as

$$\hat{G} = D_{\hat{p}} + D_{\hat{q}}F'(x),$$

where  $\hat{p}$  and  $\hat{q}$  are vectors such that

$$(\hat{p}_i, \hat{q}_i) = (\hat{a}_i + \bar{a}_i, \hat{b}_i + \bar{b}_i), \quad i = 1, 2, \dots, n, \tag{6}$$

with  $(\hat{a}, \hat{b}) \in \hat{\Omega}(x)$ , and  $(\bar{a}_i, \bar{b}_i) \in \Lambda_i(x, \hat{a}_i, \hat{b}_i)$ ,  $i = 1, 2, \dots, n$ .

If  $\theta_\lambda(x) > 0$ , the definition of  $\Lambda_i$  and (6) imply that both  $D_{\hat{p}}$ ,  $D_{\hat{q}}$  are negative definite matrices.

Mapping  $\hat{\mathcal{G}} : \mathcal{R}^n \rightarrow P(\mathcal{R}^{n \times n})$  is defined by

$$\hat{\mathcal{G}}(x) = \left\{ \hat{G} \in \mathcal{R}^{n \times n} \left| \begin{array}{l} \hat{G} = D_{\hat{p}} + D_{\hat{q}}F'(x), (\hat{p}, \hat{q}) \text{ is defined by (6)} \\ \text{with } (\hat{a}, \hat{b}) \in \hat{\Omega}(x) \text{ and } (\bar{a}_i, \bar{b}_i) \in \Lambda_i(x, \hat{a}_i, \hat{b}_i) \\ \text{for } i = 1, 2, \dots, n \end{array} \right. \right\}.$$

From Theorem 6,  $\hat{\mathcal{H}} \subseteq \tilde{\mathcal{H}}$ . It is obvious that  $\hat{\mathcal{G}} \subseteq \tilde{\mathcal{G}}$ . And from Theorem 7, every  $\hat{G} \in \hat{\mathcal{G}}(x)$  is nonsingular if  $\theta_\lambda(x) > 0$ .

Define  $\hat{\Delta} : \mathcal{R}^n \rightarrow P(\mathcal{R}^n)$  as

$$\hat{\Delta}(x) = \{d \in \mathcal{R}^n \mid \hat{G}d = -\Phi_\lambda(x), \hat{G} \in \hat{\mathcal{G}}(x)\}.$$

For any  $x$ , since  $\hat{\mathcal{G}}(x) \subseteq \tilde{\mathcal{G}}(x)$ , we obtain  $\hat{\Delta}(x) \subseteq \tilde{\Delta}(x)$ . Hence,  $\hat{\Delta}(x)$  is nonempty for every  $x$  such that  $\theta_\lambda(x) > 0$ . Next, we will give the second algorithm. The search direction is chosen from  $\hat{\Delta}(x)$ . The only difference from Algorithm 1 is the search direction.

**Algorithm 2**

- Step 1. Initialization: choose  $\lambda \in (0, 4)$ ,  $x^0 \in \mathcal{R}^n$ ,  $\rho \in (0, 0.5)$ ,  $\beta \in (0, 1)$ , and set  $k := 0$ .
- Step 2. Termination criterion: if  $\theta_\lambda(x) = 0$ , stop. Otherwise, go to Step 3.
- Step 3. Search direction calculation: find a vector  $d^k \in \hat{\Delta}(x^k)$ .
- Step 4. Line search: let  $m$  be the smallest nonnegative integer such that

$$\theta_\lambda(x^k + \beta^m d^k) - \theta_\lambda(x^k) \leq \beta^m \rho \nabla \theta_\lambda(x^k)^T d^k.$$

- Step 5. Update: set  $x^{k+1} := x^k + t_k d^k$ , where  $t_k = \beta^m$ ,  $k := k + 1$ , and go to Step 2.

Since  $\hat{\Delta}(x^k) \subseteq \tilde{\Delta}(x^k)$  at each  $x^k$  as mentioned above, global convergence of Algorithm 2 is directly obtained from Theorem 9. We state the following theorem without proof.

**Theorem 10** *Every accumulation point of a sequence  $\{x^k\}$  generated by Algorithm 2 is a solution of NCP(F).*

In the following, we focus our attention on the superlinear convergence rate of Algorithm 2. To begin with, we assume that the sequence  $\{x^k\}$  generated by Algorithm 2 has a unique limit point  $x^*$ .

**Lemma 11** *We have  $\|x^k + d^k - x^*\| = o(\|x^k - x^*\|)$ .*

*Proof* For each  $k$ , we have

$$\begin{aligned} & \|x^k + d^k - x^*\| \\ &= \|x^k - \hat{G}_k^{-1} \Phi_\lambda(x^k) - x^*\| \\ &= \|\hat{G}_k^{-1}(\Phi_\lambda(x^*) - \Phi_\lambda(x^k)) + \hat{H}_k(x^k - x^*) + (\hat{G}_k - \hat{H}_k)(x^k - x^*)\| \\ &\leq \|\hat{G}_k^{-1}\|(\|\Phi_\lambda(x^*) - \Phi_\lambda(x^k) + \hat{H}_k(x^k - x^*)\| + \|\hat{G}_k - \hat{H}_k\| \|x^k - x^*\|), \end{aligned}$$

where  $\hat{H}_k \in \hat{\mathcal{H}}(x^k)$  is the matrix corresponding to  $\hat{G}_k$ . Since  $\Phi_\lambda$  is semismooth [6, Lemma 2.2] and  $\hat{\mathcal{H}}(x^k) \in \partial_B \Phi_\lambda(x^k)$  for each  $k$ , by Theorem 6, we have

$$\|\Phi_\lambda(x^*) - \Phi_\lambda(x^k) + \hat{H}_k(x^k - x^*)\| = o(\|x^k - x^*\|)$$

(see the proof of [9, Theorem 3.1]). Moreover, by the definition of  $\Lambda_i$  and (6), we have

$$\|\hat{G}_k - \hat{H}_k\| = O(\sigma(\theta_\lambda(x^k))).$$

Consequently, it follows that

$$\|x^k + d^k - x^*\| \leq \|\hat{G}_k^{-1}\| (o(\|x^k - x^*\|) + O(\sigma(\theta_\lambda(x^k))) \|x^k - x^*\|).$$

Since  $\sigma(\theta_\lambda(x^k)) \rightarrow 0$  and  $\{\|\hat{G}_k^{-1}\|\}$  is bounded (see the proof of [8, Lemma 4.2]), we obtain the desired result. □

Now, we prove the superlinear convergence of Algorithm 2.



**Theorem 12** *Algorithm 2 has a superlinear rate of convergence.*

*Proof* We have  $x^{k+1} = x^k + d^k$  for all  $k$  sufficiently large (see the proof of [8, Lemma 4.6]). It then follows from Lemma 11 that

$$\lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|} = 0.$$

The proof is complete. □

### 5 Numerical results

In this section, we do some preliminary numerical experiments to test Algorithm 2 and compare its performance with that of the algorithms proposed in Chen and Pan [2] and Sun and Zeng [10].

First, we set  $\beta = 0.5$ ,  $\rho = 10^{-4}$ ,  $\varepsilon = 0.05$  and  $\sigma(t) = 0.1 \min\{1, t\}$ .

For  $z \in Z(x)$ , we define

$$z_i = \begin{cases} 1, & \text{if } x_i^k = F_i(x^k) = 0, \\ 0, & \text{otherwise,} \end{cases} \quad i = 1, 2, \dots, n.$$

The stopping criterion for Algorithm 2 is  $\theta_\lambda(x^k) \leq 10^{-8}$ . The programs are coded in MATLAB and run on a personal computer with a 2.1 GHZ CPU processor.

The meaning of the columns in the tables are stated as follows:

iter: the total iteration number, resi: the value of  $\theta_\lambda(x)$ .

**Problem 1** Let  $F(x) = Ax + q$ , where

$$A = \begin{bmatrix} 3 & -1 & 0 & \cdots & 0 \\ -1 & 3 & -1 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 3 \end{bmatrix}, \quad q = (-1, \dots, -1)^T.$$

The corresponding complementarity problem has the unique solution. Table 1 lists the test results for Problem 1 with different  $n$ ,  $\lambda$  and initial points  $a = (-5, \dots, -5)^T$ ,  $b = (0, \dots, 0)^T$ ,  $c = (1, \dots, 1)^T$ ,  $d = (8, \dots, 8)^T$ .

From Table 1, we see that the test results for  $\lambda \in (0, 1)$  are better than for other cases. Especially, the good numerical results are obtained when  $\lambda$  closes to 0. Then we compare the test results with Chen and Pan [2], where we set  $p = 2$ ,  $\varepsilon = 1.0e-08$ ,  $\sigma = 1.0e-10$ ,  $\beta = 0.2$  for convenience. Table 2 lists the test results for [2].

Tables 1 and 2 indicate that Algorithm 2 performed much better than Chen and Pan [2] did on Problem 1.

**Problem 2** Free boundary problems can also be solved by the method we presented. The following problem arises from the discretization of a free boundary problem (see [10]). Let  $\Omega = (0, 1) \times (0, 1)$  and a function  $g$  satisfy  $g(x, 0) = x(1 - x)$ ,  $g(x, y) = 0$  on  $x = 0, 1$  or  $y = 1$ .

**Table 1 Test results for Problem 1**

Initials		<b>a</b>		<b>b</b>		<b>c</b>		<b>d</b>	
<b>n</b>	$\lambda$	iter	resi	iter	resi	iter	resi	iter	resi
4	0.0001	3	1.61e-021	2	4.71e-012	3	1.42e-018	3	2.36e-016
	0.001	3	4.84e-020	2	6.26e-012	3	1.37e-017	3	1.22e-015
	0.03	3	3.81e-012	2	3.74e-010	3	2.11e-014	3	7.61e-013
	0.05	3	3.15e-010	2	1.54e-009	3	1.76e-013	3	2.52e-012
	0.1	4	1.21e-012	3	3.90e-012	3	1.20e-012	4	6.59e-010
	0.5	4	8.78e-011	3	2.70e-010	3	8.18e-010	5	7.32e-013
	1	4	3.31e-010	4	1.86e-017	4	1.66e-016	5	1.05e-012
	1.3	4	5.23e-009	4	3.78e-013	4	1.92e-014	5	3.20e-011
	1.5	5	1.36e-017	4	1.57e-012	4	2.07e-014	6	7.86e-016
	2	5	2.22e-013	4	9.67e-011	4	3.05e-012	7	1.64e-016
	3.2	6	1.77e-015	5	2.28e-011	4	1.36e-010	13	1.72e-011
	3.5	6	4.69e-012	5	5.76e-009	4	5.97e-009	16	1.06e-010
	3.8	7	1.62e-013	6	5.16e-011	4	6.87e-010	8	5.69e-013
10	0.0001	3	2.56e-017	2	4.37e-010	3	1.49e-018	3	5.27e-015
	0.001	3	1.04e-016	2	4.84e-010	3	3.18e-017	3	5.15e-015
	0.03	3	7.55e-011	2	5.42e-009	3	2.90e-014	3	7.16e-009
	0.05	3	3.80e-009	3	3.01e-021	3	2.05e-013	3	5.87e-012
	0.1	3	6.69e-009	3	1.03e-010	3	3.16e-012	4	7.08e-010
	0.5	4	2.86e-010	3	3.91e-010	3	2.30e-009	5	3.87e-014
	1	4	2.98e-009	4	1.47e-016	3	1.24e-009	5	7.37e-018
	1.3	5	9.65e-018	4	1.25e-014	4	2.27e-013	5	1.49e-011
	1.5	5	6.52e-017	4	2.08e-009	4	1.45e-013	6	8.19e-016
	2	5	5.67e-014	5	3.91e-017	4	2.22e-014	6	7.76e-009
	3.2	6	5.53e-014	5	5.85e-010	4	7.08e-009	13	2.23e-009
	3.5	6	3.03e-011	6	7.13e-015	4	6.83e-010	20	1.12e-010
	3.8	7	2.93e-013	6	2.81e-010	4	3.62e-010	8	8.00e-014
100	0.0001	3	9.91e-012	3	6.59e-022	3	2.43e-018	3	2.25e-011
	0.001	3	1.41e-011	3	1.97e-021	3	3.31e-017	3	1.68e-011
	0.03	4	7.16e-021	3	8.29e-018	3	3.25e-014	4	3.76e-019
	0.05	4	6.93e-020	3	1.27e-016	3	1.13e-013	4	9.21e-017
	0.1	4	8.22e-012	4	3.17e-021	3	3.05e-012	4	4.96e-010
	0.5	5	8.60e-020	4	6.17e-019	3	3.00e-009	5	4.00e-011
	1	5	2.57e-017	4	1.03e-014	3	2.55e-009	5	7.21e-018
	1.3	5	1.11e-015	4	4.37e-011	4	1.00e-014	5	2.87e-009
	1.5	5	1.63e-014	5	7.45e-016	4	1.42e-012	6	4.25e-017
	2	5	8.25e-012	5	4.76e-014	4	4.63e-011	7	1.16e-013
	3.2	6	7.28e-010	6	5.92e-011	4	2.88e-011	17	1.14e-014
	3.5	7	2.78e-015	6	3.85e-009	4	3.11e-009	27	1.02e-009
	3.8	7	7.03e-009	7	4.89e-013	4	3.62e-010	7	8.13e-012

**Table 2 Test results for Chen and Pan [2] on Problem 1**

Initials		<b>a</b>		<b>b</b>		<b>c</b>		<b>d</b>	
<b>n</b>		iter	resi	iter	resi	iter	resi	iter	resi
4		34	4.92e-009	18	7.17e-009	14	5.90e-009	35	5.90e-009
10		48	6.79e-009	20	3.76e-009	19	5.57e-009	48	4.35e-009
100		97	9.58e-009	23	6.47e-009	18	9.34e-009	48	8.69e-009

Consider the following problem: find  $u$  such that

$$\begin{cases} u \geq 0 & \text{in } \Omega, \\ -\Delta u + f(u, x, y) - 8(y - 0.5) \geq 0 & \text{in } \Omega, \\ u(-\Delta u + f(u, x, y) - 8(y - 0.5)) = 0 & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega, \end{cases}$$

where  $f(u, x, y)$  is a continuously differentiable  $P_0$ -function. We discretize the problem by the five-point difference scheme with mesh-step  $h$ . Then we get the following complementarity problem: find  $x \in \mathcal{R}^n$  such that

$$x \geq 0, \quad Ax + \Psi(x) \geq 0 \quad \text{and} \quad x^T(Ax + \Psi(x)) = 0.$$

Set initial point as  $x_0 = (0, \dots, 0)^T$ . Table 3 lists the test results with different functions  $f$ ,  $\lambda$ , and  $h$ .

From Table 3, we have the following observations.

- Our test results become better when  $\lambda$  decreases. It is obvious that when  $\lambda = 2$  the result is not good enough. That is to say, Algorithm 2 with the NCP-function  $\phi_\lambda$

**Table 3 Test results for Problem 2**

$h$	$f(u, x, y)$	$\lambda$	$1/2^3$		$1/2^4$		$1/2^5$	
			iter	resi	iter	resi	iter	resi
0		0.0001	3	2.68e-009	7	4.15e-014	13	1.87e-015
		0.001	3	5.30e-009	7	2.44e-012	13	1.79e-013
		0.01	4	5.05e-018	7	2.30e-010	13	1.78e-011
		0.05	4	5.24e-014	7	5.58e-009	13	4.33e-010
		0.5	5	1.02e-009	8	7.08e-011	14	5.49e-012
		0.8	6	1.60e-014	8	8.17e-010	14	6.35e-011
		1	6	2.54e-013	8	2.45e-009	14	1.91e-010
		1.5	6	3.17e-011	9	2.21e-012	14	1.22e-009
		2	6	8.05e-010	9	5.63e-011	15	3.94e-009
		2.2	6	2.23e-009	10	1.56e-010	15	5.59e-009
$u^3$		0.0001	3	2.19e-009	7	4.12e-014	13	1.86e-015
		0.001	3	4.68e-009	7	2.43e-012	13	1.79e-013
		0.01	4	4.52e-018	7	2.30e-010	13	1.77e-011
		0.05	4	5.13e-014	7	5.57e-009	13	4.31e-010
		0.5	5	1.04e-009	8	7.07e-011	14	5.47e-012
		0.8	6	1.62e-014	8	8.16e-010	14	6.33e-011
		1	6	2.52e-013	8	2.45e-009	14	1.90e-010
		1.5	6	3.17e-011	9	2.21e-012	14	1.22e-009
		2	6	8.04e-010	9	5.62e-011	15	3.92e-009
		2.2	6	2.23e-009	10	7.82e-011	15	5.57e-009
$\frac{1}{1+u}$		0.0001	2	4.46e-010	5	9.42e-014	11	1.33e-014
		0.001	2	7.90e-010	5	1.14e-011	11	1.34e-012
		0.01	3	9.27e-019	5	1.18e-009	11	1.33e-010
		0.05	3	1.27e-014	6	6.77e-016	11	3.21e-009
		0.5	3	5.89e-009	6	3.67e-010	12	3.06e-009
		0.8	4	6.77e-014	6	5.99e-009	12	6.87e-009
		1	4	9.83e-013	7	7.14e-010	12	9.11e-009
		1.5	4	1.13e-010	7	4.59e-009	12	1.00e-009
		2	5	1.23e-010	8	8.62e-011	13	3.70e-010
		2.2	5	6.28e-010	8	2.39e-010	13	5.53e-010

**Table 4 Test results for Sun and Zeng [10] on Problem 2**

$h$	$1/2^3$		$1/2^4$		$1/2^5$	
	iter	resi	iter	resi	iter	resi
0	5	6.45e-016	8	3.11e-015	14	5.03e-011
$u^3$	5	8.12e-009	9	1.20e-012	15	3.86e-008
$\frac{1}{1+u}$	5	1.89e-009	8	7.92e-010	14	1.13e-007

**Table 5 Test results for MCPLIB problems**

$\lambda$	1.5		2		3	
	iter	time	iter	time	iter	time
mathinum(1)	4	0.048	4	0.050	5	0.050
mathinum(2)	4	0.031	5	0.028	6	0.036
mathinum(3)	7	0.030	7	0.030	8	0.032
mathinum(4)	6	0.076	7	0.074	7	0.080
nash(1)	7	2.536	8	2.538	11	2.529
nash(2)	8	2.284	9	2.301	13	2.334
tobin(1)	7	16.878	9	16.729	10	17.321
tobin(2)	12	17.389	14	17.342	16	17.358

$(0 < \lambda < 2)$  is better than the one discussed in [8] where the Fischer-Burmeister function was used.

- Whether the function  $f(u, x, y)$  is linear or nonlinear, the test results are good. The results are especially better when  $\lambda$  closes to 0.

We compare the test results with Sun and Zeng [10] where we set  $\beta = 0.5, c = 0.5^4$ . Table 4 lists the test results for [10] with different functions  $f$  and  $h$ .

Tables 3 and 4 indicate that Algorithm 2 performed as well as Sun and Zeng [10] did on Problem 2.

**Problem 3** We implemented Algorithm 2 for some test problems with all available starting points in MCPLIB [11]. The results are reported in Table 5 with seconds for unit of time.

The above examples indicate that the results are better when  $\lambda$  closes to 0. A reasonable interpretation for this is that the values of  $g_i(x, z)$  and  $h_i(x, z)$  become smaller when  $\lambda$  increases and hence causes some difficulty for Algorithm 2. This also implies that the performance of Algorithm 2 will become worse when  $p$  increases. When  $\lambda \rightarrow 0$ , the NCP obviously reduces to  $\min\{x, F(x)\} = 0$ . But it is a nonsmooth equation so we cannot use this method.

## 6 Concluding remarks

In this paper, we have studied a class of one-parametric NCP-functions  $\phi_\lambda(\cdot, \cdot)$  which include the well-known Fischer-Burmeister function as a special case and proposed modified Newton-type algorithms for solving  $P_0$  complementarity problems.

Numerical results for the test problems have shown that this method is promising when  $\lambda \in (0, 4)$ . Moreover, our numerical results indicated that the performance of the modified Newton-type method becomes better when  $\lambda$  decreases, which is a new and important numerical result. We believe that Algorithm 2 can effectively solve more practical problems if they can be reformulated as an NCP( $F$ ). We leave this as a future research topic.

#### Competing interests

The authors declare that they have no competing interests.

#### Authors' contributions

WX participated in the design of the algorithm. ZD performed the numerical experiment and statistical analysis. All authors read and approved the final manuscript.

#### Acknowledgements

This work was supported by the NSFC (50975200).

Received: 24 February 2012 Accepted: 12 November 2012 Published: 7 December 2012

#### References

1. Chen, J-S: On some NCP-functions based on the generalized Fischer-Burmeister function. *Asia-Pac. J. Oper. Res.* **24**, 401-420 (2007)
2. Chen, J-S, Pan, S: A family of NCP functions and a descent method for the nonlinear complementarity problem. *Comput. Optim. Appl.* **40**, 389-404 (2008)
3. Fischer, A: A special Newton-type optimization methods. *Optimization* **24**, 269-284 (1992)
4. Hu, SL, Huang, ZH, Chen, J-S: Properties of a family of generalized NCP-functions and a derivative free algorithm for complementarity problems. *J. Comput. Appl. Math.* **230**(1), 69-82 (2009)
5. Hu, SL, Huang, ZH, Lu, N: Smoothness of a class of merit functions for the second-order cone complementarity problem. *Pac. J. Optim.* **6**(3), 551-571 (2010)
6. Kanzow, C, Kleinmichel, H: A new class of semismooth Newton-type methods for nonlinear complementarity problems. *Comput. Optim. Appl.* **11**, 227-251 (1998)
7. Lu, LY, Huang, ZH, Hu, SL: Properties of a family of merit functions and a derivative-free method for the NCP. *Appl. Math. J. Chin. Univ. Ser. A* **25**(4), 379-390 (2010)
8. Yamashita, N, Fukushima, M: Modified Newton methods for solving a semismooth reformulation of monotone complementarity problems. *Math. Program.* **76**, 469-491 (1997)
9. Qi, L: Convergence analysis of some algorithms for solving nonsmooth equations. *Math. Oper. Res.* **18**, 227-244 (1993)
10. Sun, Z, Zeng, JP: A monotone semismooth Newton type method for a class of complementarity problem. *J. Comput. Appl. Math.* **235**, 1261-1274 (2011)
11. Billups, SC, Dirkse, SP, Soares, MC: A comparison of algorithms for large scale mixed complementarity problems. *Comput. Optim. Appl.* **7**, 3-25 (1997)

doi:10.1186/1029-242X-2012-286

**Cite this article as:** Xie and Deng: Modified Newton-type methods for the NCP by using a class of one-parametric NCP-functions. *Journal of Inequalities and Applications* 2012 **2012**:286.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---