Adv Syst Sci Appl 2023; **03**:16–26
Published online at https://ijassa.ipu.ru.

Newton Method vs. Semismooth Newton Method for Singular Solutions of Nonlinear Complementarity Problems

Alexey Izmailov[1*], Evgeniy Uskov[2], Yan Zhibai[1]

[1] *Lomonosov Moscow State University, Moscow, Russia*
[2] *Derzhavin Tambov State University, Tambov, Russia*

Abstract: Among the most successful techniques for solving nonlinear complementarity problems is the one consisting of reformulation of the problem in question as a system of nonlinear equations, by means of the so-called complementarity functions. Different complementarity functions lead to nonlinear systems with different smoothness and regularity properties, thus allowing for application of different classed of numerical methods. In this paper we compare the Newton method for the smooth reformulation with the semismooth Newton method for the reformulation relying on the nonsmooth Fischer–Burmeister complementarity function, with a special emphasis on the cases when the solution in question violates the strict complementarity condition.

Keywords: nonlinear complementarity problem, complementarity function, strict complementarity, singular solution, Newton method, semismooth Newton method, extrapolation

1. INTRODUCTION

We consider the nonlinear complementarity problem (NCP)

$$u \geq 0, \quad F(u) \geq 0, \quad \langle u,\, F(u)\rangle = 0, \tag{1.1}$$

where $F : \mathbb{R}^p \to \mathbb{R}^p$ is a given smooth mapping. Here and throughout the paper, $\langle \cdot,\, \cdot \rangle$ stands for the Euclidian inner product. Accordingly, by $\| \cdot \|$ we will denote the Euclidian norm.

Among the most successful approaches to numerical solution of NCP (1.1) relies on a reformulation of this problem as an equivalent system of equations. Recall that a function $\psi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is referred to as a complementarity function if

$$\psi(a,\, b) = 0 \quad \Leftrightarrow \quad a \geq 0, \quad b \geq 0, \quad ab = 0.$$

With any such function at hand, one can readily observe that (1.1) is equivalent to the equation

$$\Phi(u) = 0 \tag{1.2}$$

with $\Phi : \mathbb{R}^p \times \mathbb{R}^p$,

$$\Phi(u) = \psi(u,\, F(u)), \tag{1.3}$$

where ψ is applied componentwise.

Known complementarity functions are plenty [17, 18], and their instances may possess different smoothness properties. Therefore, the specific choice of such a function naturally

*Corresponding author: izmaf@cs.msu.ru

restricts the class of numerical methods applicable to the corresponding instance of equation (1.2).

The rest of the paper is organized as follows. In Section 2, we consider a smooth complementarity function, allowing for the basic Newton method to be applied to the corresponding equation (1.2). We discuss the local convergence and rate-of-convergence properties, as well the linesearch technique for globalization of convergence, and the extrapolation technique for potential acceleration of convergence to solutions violating strict complementarity. Section 3 deals with the Fischer–Burmeister complementarity function, in which case the corresponding equation (1.2) can be tackled by the semismooth Newton method. Local convergence, rate-of-convergence, and globalization of convergence are discussed. Section 4 presents the results of a numerical comparison of the two methods on a collection of NCPs with solutions violating strict complementarity.

2. SMOOTH COMPLEMENTARITY FUNCTION AND NEWTON METHOD

Consider, for example, a complementarity function defined by

$$\psi(a,\, b) = 2ab - (\min\{0,\, a+b\})^2. \tag{2.1}$$

This function is smooth, and assuming that F is differentiable at $u \in \mathbb{R}^p$, the corresponding mapping Φ defined in (1.3) is also differentiable at u, with the rows of the Jacobian $\Phi'(u)$ given by

$$\Phi_i'(u) = 2u_iF_i'(u) + 2F_i(u)e^i - 2\min\{0,\, u_i + F_i(u)\}(F_i'(u) + e^i), \quad i = 1,\, \ldots,\, p, \tag{2.2}$$

where $e^1,\, \ldots,\, e^p$ is the standard basis in $\mathbb{R}^p$. Moreover, assuming that F is differentiable on a neighborhood of some point in $\mathbb{R}^p$, with its derivative F' being Lipschitz-continuous on this neighborhood, the same property is possessed by Φ'.

Under these smoothness assumptions, one can apply the basic Newton method to equation (1.2) with Φ defined according to (1.3), (2.1): For a given iterate $u^k \in \mathbb{R}^p$, if F is differentiable at u^k, one can compute the next iterate as $u^{k+1} = u^k + v^k$, where v^k is a solution of the linear equation

$$\Phi(u^k) + \Phi'(u^k)v = 0; \tag{2.3}$$

see, e.g., [15, Section 2.1.1].

Let $\bar{u}$ be a solution of NCP (1.1). Assume that F is differentiable near $\bar{u}$, with its derivative F' being continuous at $\bar{u}$, and that $\Phi'(\bar{u})$ with rows defined in (2.2) is nonsingular. Then, according to [15, Theorem 2.2], for any $u^0 \in \mathbb{R}^p$ close enough to $\bar{u}$, the Newton method specified above uniquely defines a sequence $\{u^k\}$, and this sequence converges to $\bar{u}$ superlinearly, or even quadratically if F' is Lipschitz-continuous near $\bar{u}$.

Convergence of the Newton method can be globalized, e.g., by linesearch techniques. The following is essentially [7, Algorithm 2.1].

Algorithm 2.1:
Define the mapping Φ according to (1.3), (2.1). Choose the parameters $\sigma \in (0,\, 1)$ and $\theta \in (0,\, 1)$. Choose $u^0 \in \mathbb{R}^p$, and set $k = 0$.

1. If $\Phi(u^k) = 0$, stop with success.
2. Compute $v^k \in \mathbb{R}^p$ as a solution of (2.3). If such v^k cannot be found, stop with a failure.
3. Set $\tau = 1$. If the inequality

$$\|\Phi(u^k + \tau v^k)\| \le (1 - \sigma\tau)\|\Phi(u^k)\| \tag{2.4}$$

is satisfied, set $\tau_k = \tau$. Otherwise, replace τ by $\theta\tau$, check inequality (2.4) again, etc., until (2.4) becomes valid.

4. Set $u^{k+1} = u^k + \tau_k v^k$.
5. Increase k by 1 and go to Step 1.

Observe that if $\Phi(u^k) \neq 0$, then the function $\|\Phi(\cdot)\|$ is differentiable at u^k, and its gradient is $(\Phi'(u^k))^\top \Phi(u^k)\|\Phi(u^k)\|$. Then the inner product of this gradient and v^k solving (2.3) evidently equals $-\|\Phi(u^k)\| < 0$, while the stepsize test (2.4) is nothing else but the classical Armijo inequality; see, e.g., [15, (2.62)], and Step 3 of the algorithm is the standard Armijo backtracking procedure.

In practical implementations, and for global convergence analysis, instead of declaring failure when v^k cannot be found in Step 2, one can supply Algorithm 2.1 with some safeguarding tools; see, for example, [7, Algorithm 3.1]. According to [7, Theorem 3.1], any accumulation point $\bar{u}$ of any sequence $\{u^k\}$ generated by such enhanced Algorithm 2.1 satisfies $(\Phi'(\bar{u}))^\top \Phi(\bar{u}) = 0$, and in particular, can be guaranteed to be a solution of NCP (1.1) under some reasonable additional assumptions. However, in the numerical experiments reported in Section 4, we employ Algorithm 2.1 as it is, without any additional safeguards, and this indeed leads to some failures on Step 2.

Moreover, assuming that $\Phi'(\bar{u})$ is nonsingular at a solution $\bar{u}$ of NCP (1.1), and employing the argument in [15, Theorem 5.4], one can easily see that (2.4) is satisfied with $\alpha = 1$ provided u^k is close enough to $\bar{u}$. Therefore, the algorithm asymptotically accepts the full step, and inherits the local convergence and rate of convergence properties of the basic Newton method.

Observe, however, that if $\bar{u}$ violates that strict complementarity condition $\bar{u} + F(\bar{u}) > 0$, which means that there exist $i \in \{1, \ldots, p\}$ such that $\bar{u}_i = F_i(\bar{u}) = 0$. Then (2.2) yields $\Phi'_i(\bar{u}) = 0$, and hence, $\Phi'(\bar{u})$ is necessarily singular, not allowing to apply the theory outlined above. The strict complementarity condition is regarded as rather restrictive in the literature, and therefore, one should be ready to face singularity of a solution $\bar{u}$ of (1.2) when the latter is obtained as a reformulation of NCP (1.1), employing the smooth complementarity function (2.1) (and actually, any other smooth complementarity function).

There exists quite a rich literature on the behavior of Newton-type methods near singular solutions of nonlinear equations; we mention only some references most tightly related to the contents of this work: [7–11, 13]. Apart from characterizing local linear convergence from asymptotically dense domains starlike with respect to such solution, some acceleration techniques were developed, such as the so-called extrapolation [9, 11]. In its simplest form, this procedure amounts to generating, in parallel with the main sequence $\{u^k\}$ of the Newton method, an auxiliary sequence $\{\widehat{u}^k\}$ obtained by doubling the Newton step: for each k set

$$\widehat{u}^{k+1} = u^k + 2v^k. \tag{2.5}$$

This specific procedure is suggested by the convergence pattern of the Newton method to singular solutions of certain wide classes. According to [11, Theorem 4.1], under the appropriate assumptions, the sequence $\{\widehat{u}^k\}$ generated this way converges to $\bar{u}$ linearly with the asymptotic ratio of $1/4$ (instead of $1/2$ for the sequence $\{u^k\}$ generated by the Newton method), from all points in the domain of convergence of Newtonian sequences. The main iterative sequences $\{u^k\}$ are not affected by this acceleration technique in any way, and generating the auxiliary sequence $\{\widehat{u}^k\}$ is very cheap. Therefore, this acceleration technique is is fully practical, and can be easily incorporated into the globalized Algorithm 2.1 [7].

That said, in all the references cited above in connection with Newton-type methods near singular solutions it is assumed that Φ is al least twice differentiable, which is not the case for the mapping defined by (1.3), (2.1). To the best of our knowledge, the current paper is the first one where the properties in question will be assessed for equations with locally Lipchitzian first derivatives but perhaps without second derivatives.

3. FISCHER–BURMEISTER COMPLEMENTARITY FUNCTION AND SEMISMOOTH NEWTON METHOD

If one does not want to deal with potential singularity of solutions (as in Section 2), a natural thing to do is to employ different (nonsmooth) complementarity functions. One of the most successful and widely used is the so-called Fischer–Burmeister complementarity function defined by

$$\psi(a,\, b) = a + b - \sqrt{a^2 + b^2}. \tag{3.1}$$

This function is not differentiable at $(0,\, 0)$, and the corresponding mapping Φ defined in (1.3) need not be differentiable at $u \in \mathbb{R}^p$ with $u_i = F_i(u) = 0$ for at least one $i \in \{1,\, \ldots,\, p\}$, no matter how smooth is F. In particular, Φ need not be differentiable at a solution violating strict complementarity. At the same time, if F is differentiable near $u \in \mathbb{R}^p$, with F' being continuous at u, then Φ is Lipschitz continuous at u, and even semismooth at u; see [15, Proposition 3.8]

Therefore, even though the Newton method is in general not applicable to equation (1.2) with Φ defined according to (1.3), (3.1), one can make use of the so-called semismooth Newton method [15, Section 2.4.1], which is the iterative procedure specified above for Newton method, but with the iteration equation (2.3) replaced by

$$\Phi(u^k) + J_k v = 0, \tag{3.2}$$

where J_k is some element of $\partial_B \Phi(u^k)$, Here, for any $u \in \mathbb{R}^p$,

$$\partial_B \Phi(u) = \{J \in \mathbb{R}^{p \times p} \mid \exists\, \{u^j\} \subset \mathcal{S}_\Phi :\, \{u^j\} \to u,\, \{\Phi(u^j)\} \to J\},$$

is the so-called B-differential of Φ at u, with $\mathcal{S}_\Phi$ standing for the set of all points where Φ is differentiable; see, e.g., [15, Section 1.4.1]. If $\bar{u}$ is a solution of NCP (1.1), and all matrices in $\partial_B \Phi(\bar{u})$ are nonsingular, then, according to [15, Theorem 2.42], for any $u^0 \in \mathbb{R}^p$ close enough to $\bar{u}$, the semismooth Newton method uniquely defines a sequence $\{u^k\}$, and this sequence converges to $\bar{u}$ superlinearly, or even quadratically under further smoothness assumptions on F.

The upper estimate of $\partial_B \Phi(u)$ for Φ defined according to (1.3), (3.1) is given in [15, Proposition 3.11]: the rows of any matrix $J \in \partial_B \Phi(u)$ satisfy the equalities

$$J_i = \begin{cases} \alpha_i F_i'(u) + \beta_i e^i & \text{if } u_i = F_i(u) = 0, \\ F_i'(u) + e^i - \dfrac{F_i(u) F_i'(u) + u_i e^i}{\sqrt{u_i^2 + (F_i(u))^2}} & \text{if } u_i \neq 0 \text{ or } F_i(u) \neq 0, \end{cases} \quad i = 1,\, \ldots,\, p, \tag{3.3}$$

with some $(\alpha_i,\, \beta_i) \in S$ for $i = 1,\, \ldots,\, p$ such that $u_i = F_i(u) = 0$, where

$$S = \{(a,\, b) \in \mathbb{R}^2 \mid (a-1)^2 + (b-1)^2 = 1\}.$$

This upper estimate serves as the basis for practical procedures allowing to compute an element $J_k \in \partial_B \Phi(u^k)$, to be used in (3.2); see, e.g., [15, p. 161]. However, in Section 4, we employ the upper estimate above directly, i.e., we pick up J_k in (3.2) as some matrix with rows satisfying (3.3) for $u = u^k$. This choice is not fully clean theoretically, but it is somehow practically justified, in particular, by the natural expectation that the case when $u_i^k = F_i(u^k) = 0$ for some $i \in \{1,\, \ldots,\, p\}$ will not be encountered along the iteration sequences $\{u^k\}$, and hence, Φ will differentiable at u^k, and the upper estimate of $\partial_B \Phi(u^k)$ given be (3.3) will reduce to $\{\Phi'(u^k)\}$, for all k.

As for globalization of convergence, we next present a counterpart of Algorithm 2.1 for the semismooth Newton method.

Algorithm 3.1:
Define the mapping Φ according to (1.3), (3.1). Choose the parameters $\sigma \in (0,\, 1)$ and $\theta \in (0,\, 1)$. Choose $u^0 \in \mathbb{R}^p$, and set $k = 0$.

1. If $\Phi(u^k) = 0$, stop with success.
2. Compute some $J_k \in \partial_B \Phi(u^k)$. Compute $v^k \in \mathbb{R}^p$ as a solution of (3.2). If such v^k cannot be found, stop with a failure.
3. Set $\tau = 1$. If the inequality (2.4) is satisfied, set $\tau_k = \tau$. Otherwise, replace τ by $\theta\tau$, check inequality (2.4) again, etc., until (2.4) becomes valid.
4. Set $u^{k+1} = u^k + \tau_k v^k$.
5. Increase k by 1 and go to Step 1.

The use of the stepsize test (2.4) in this context can be explained as follows. The function $\|\Phi(\cdot)\|$ with Φ defined according to (1.3), (3.1) need not be differential at u^k, even if $\Phi(u^k) \neq 0$. That said, according to [15, Proposition 5.5], the function $\varphi : \mathbb{R}^p \to \mathbb{R}$, $\varphi(u) = \|\Phi(u)\|^2$, is differentiable at every point $u \in \mathbb{R}^p$ where F is differentiable, and its gradient satisfies

$$\varphi'(u) = 2J^\top \Phi(u) \quad \forall\, J \in \partial\Phi(u), \tag{3.4}$$

where $\partial\Phi(u)$ is Clarke's generalized Jacobian of Φ at $\bar{u}$, defined as the convex hull of $\partial_B\Phi(u)$; see, e.g., [15, Section 1.4.1]. Therefore, for v^k solving (3.2) it holds that

$$\langle \varphi'(u^k),\, v^k\rangle = 2\langle \Phi(u^k),\, J_k v^k\rangle = -2\|\Phi(u^k)\|^2 = -2\varphi(u^k). \tag{3.5}$$

Hence,

$$\begin{aligned}
\|\Phi(u^k + \tau v^k)\|^2 &= \varphi(u^k + tv^k) \\
&= \varphi(u^k) + \tau\langle \varphi'(u^k),\, v^k\rangle + o(\tau) \\
&= \|\Phi(u^k)\|^2 - 2\tau\|\Phi(u^k)\|^2 + o(\tau) \\
&= (1 - 2\tau + o(\tau))\|\Phi(u^k)\|^2
\end{aligned}$$

as $t \to 0$, implying that (2.4) is satisfied for all $\tau > 0$ small enough, for any fixed $\sigma \in (0,\, 1)$. Observe further that employing (3.5), condition (2.4) can be equivalently written in the form

$$\varphi(u^k + \tau v^k) \leq (1 - 2\sigma\tau + \sigma^2\tau^2)\varphi(u^k) = \varphi(u^k) + \sigma\tau(1 - \sigma\tau/2)\langle \varphi'(u^k),\, v^k\rangle. \tag{3.6}$$

Since we are not aware of any existing global convergence theories for the linesearch semismooth Newton method with the specific stepsize test (2.4), we next address this issue for the following safeguarded version of Algorithm 3.1.

Algorithm 3.2:
Define the mapping Φ according to (1.3), (3.1). Choose the parameters $C > 0$, $q > 0$, $\sigma \in (0,\, 1)$ and $\theta \in (0,\, 1)$. Choose $u^0 \in \mathbb{R}^p$, and set $k = 0$.

1. If $\Phi(u^k) = 0$, stop with success.
2. Compute some $J_k \in \partial_B \Phi(u^k)$. Compute $v^k \in \mathbb{R}^p$ as a solution of (3.2). If such v^k cannot be found, or violates
$$\|v^k\| \leq \max\{C,\, 1/\|\Phi(u^k)\|^q\}, \tag{3.7}$$
go to Step 4.
3. Set $\tau = 1$. If the inequality (2.4) is satisfied, set $\tau_k = \tau$. Otherwise, replace τ by $\theta\tau$, check inequality (2.4) again, etc., until (2.4) becomes valid. Go to Step 6
4. Set $v^k = -2J_k^\top \Phi(u^k)$. If $v^k = 0$, stop.

5. Set $\tau = 1$. If the inequality

$$\varphi(u^k + \tau v^k) \leq \varphi(u^k) - \sigma\tau\|v^k\|^2 \tag{3.8}$$

is satisfied, set $\tau_k = \tau$. Otherwise, replace τ by $\theta\tau$, check (3.8), etc., until (3.8) becomes valid.
6. Set $u^{k+1} = u^k + \tau_k v^k$.
7. Increase k by 1 and go to Step 1.

Theorem 3.1:
Let $F : \mathbb{R}^p \to \mathbb{R}^p$ be continuously differentiable.

Then, for any starting point $u^0 \in \mathbb{R}^p$, Algorithm 3.2 either terminates with some iterate u^k satisfying

$$J^\top \Phi(u^k) = 0 \quad \forall\, J \in \partial\Phi(u^k), \tag{3.9}$$

or generates an infinite sequence $\{u^k\}$ such that every accumulation point $\bar{u}$ of this sequence satisfies

$$J^\top \Phi(\bar{u}) = 0 \quad \forall\, J \in \partial\Phi(\bar{u}), \tag{3.10}$$

where Φ according to (1.3)*,* (3.1)*.*

Moreover, if for some $\bar{u}$ there exists an infinite subsequence $\{u^{k_j}\}$ convergent to $\bar{u}$ such that the semismooth Newton direction is used by Algorithm 3.2 for all indices $k = k_j$ (i.e., for all j, for $k = k_j$ equation (3.2) *is solvable, and the computed solution satisfies* (3.7)*), then*

$$\{\Phi(u^k)\} \to 0 \tag{3.11}$$

as $k \to \infty$, and in particular, all accumulation points of $\{u^k\}$ are solutions of the NCP (1.1)*.*

Proof
If for a current iterate u^k it holds that $\Phi(u^k) = 0$, the algorithm terminates, and (3.9) is evidently satisfied in this case. Otherwise, if the algorithm accepts the semismooth Newton direction v^k, then the discussion before Algorithm 3.1 implies that (2.4) is always satisfied after a finite number of backtrackings, i.e., the linesearch procedure at Step 3 of Algorithm 3.2 terminates with some $\tau_k > 0$. If the Newton direction is not used, then, according to (3.4), at Step 4 the algorithm picks up $v^k = -\varphi'(u^k)$. In this case, the linesearch procedure at Step 5 of the algorithm with the stepsize test (3.8) terminates finitely with some $\tau_k > 0$ (as it is the standard Armijo linesearch procedure for a function which is smooth at u^k, and in the direction the direction of the negative gradient of this function at u^k) if $\varphi'(u^k) \neq 0$, while otherwise, the algorithm terminates with (3.9) being satisfied again, the latter following from (3.4).

Therefore, Algorithm 3.2 either terminates with some iterate u^k satisfying (3.9), or generates an infinite sequence $\{u^k\}$, and it remains to analyze the latter case. Observe that the sequence $\{\|\Phi(u^k)\|\}$ is always monotonically non-increasing, no matter which search directions are used. But this sequence is bounded below (by zero), and hence converges.

Consider first the case when there exists an infinite subsequence $\{u^{k_j}\}$ convergent to some $\bar{u}$, and such that the semismooth Newton direction is accepted by Algorithm 3.2 at every point of this subsequence.

If

$$\bar{\tau} = \limsup_{j\to\infty} \tau_{k_j} > 0, \tag{3.12}$$

then (2.4) implies that for infinitely many indices k_j, the residual $\|\Phi(u^{k_j})\|$ is reduced at least linearly (by a factor of at least $(1 - \sigma\bar{\tau}/2)$), and hence, $\{\Phi(u^{k_j})\} \to 0$ as $j \to \infty$. Since $\{\|\Phi(u^k)\|\}$ is convergent, we conclude that (3.11) holds.

On the other hand, if (3.12) is not valid, then

$$\lim_{j\to\infty} \tau_{k_j} = 0, \tag{3.13}$$

implying that for each j large enough, the initial trial stepsize value had been reduced at least once, i.e., the value $\tau = \tau_{k_j}/\theta > \tau_{k_j}$ does not satisfy (2.4). Since (2.4) is equivalent to (3.6), it then holds that

$$\frac{\varphi(u^{k_j} + \tau_{k_j} v^{k_j}/\theta) - \varphi(u^{k_j})}{\tau_{k_j}/\theta} > \sigma(1 - \sigma\tau_{k_j}/2)\langle\varphi'(u^k),\, v^k\rangle. \tag{3.14}$$

Observe that according to [15, Proposition 5.5], φ is continuously differentiable on $\mathbb{R}^p$, and employing the mean-value theorem [15, Theorem A.10(a)], we then obtain from (3.14) that for each j there exists $\omega_{k_j} \in [0,\, 1]$ such that

$$\langle\varphi(u^{k_j} + \omega_{k_j}\tau_{k_j} v^{k_j}/\theta),\, v^{k_j}\rangle > \sigma(1 - \sigma\tau_{k_j}/2)\langle\varphi'(u^k),\, v^k\rangle. \tag{3.15}$$

If the sequence $\{v^{k_j}\}$ of the semismooth Newton directions is unbounded, condition (3.7) implies that

$$\liminf_{j\to\infty} \|\Phi(u^{k_j})\| = 0.$$

In view of convergence of the sequence $\{\|\Phi(u^k)\|\}$, this again implies (3.11). Therefore, it remains to consider the case when $\{v^{k_j}\}$ is bounded. Taking a further subsequence, if necessary, we may assume that $\{v^{k_j}\}$ converges to some $\widetilde{v}$, and therefore, by (3.5),

$$\langle\varphi'(\bar{u}),\, \widetilde{v}\rangle = -2\varphi'(\bar{u}).$$

Then by (3.13), passing onto the limit in (3.15) as $j \to \infty$, we obtain that

$$-\varphi(\bar{u}) \geq -\sigma\varphi(\bar{u}),$$

which is only possible when $\Phi(\bar{u}) = 0$.

It remains to consider the case when there exists an infinite subsequence $\{u^{k_j}\}$ convergent to some $\bar{u}$, and such that the semismooth Newton direction is not accepted by Algorithm 3.2 at every point of this subsequence. In this case, the iterates $\{u^{k_j+1}\}$ are generated by the gradient steps with Armijo linesearch for the continuously differentiable merit function φ. Then by standard argument (see, e.g., [2, Proposition 1.2.3]), it follows that $\varphi'(\bar{u}) = 0$, and according to (3.4), this yields (3.10). □

4. NUMERICAL RESULTS

In this section we provide some numerical comparisons of the algorithms discussed above for a collection of small NCPs taken from [19], and for some other examples of NCPs with solutions violating strict complementarity, taken from various sources. For the former, we use the identifiers of the problems adopted in [19], while the latter are cited as they appear in the related references. For each of these test problems, a solution of interest $\bar{u}$, violating strict complementarity, is known; this information is provided in Table 3.1.

The algorithms being tested are abbreviated as follows:

- NM is Algorithm 2.1.
- NM-EP (for "ExtraPolation") is Algorithm 2.1, generating also an auxiliary sequence $\{\widehat{u}^k\}$ according to (2.5).

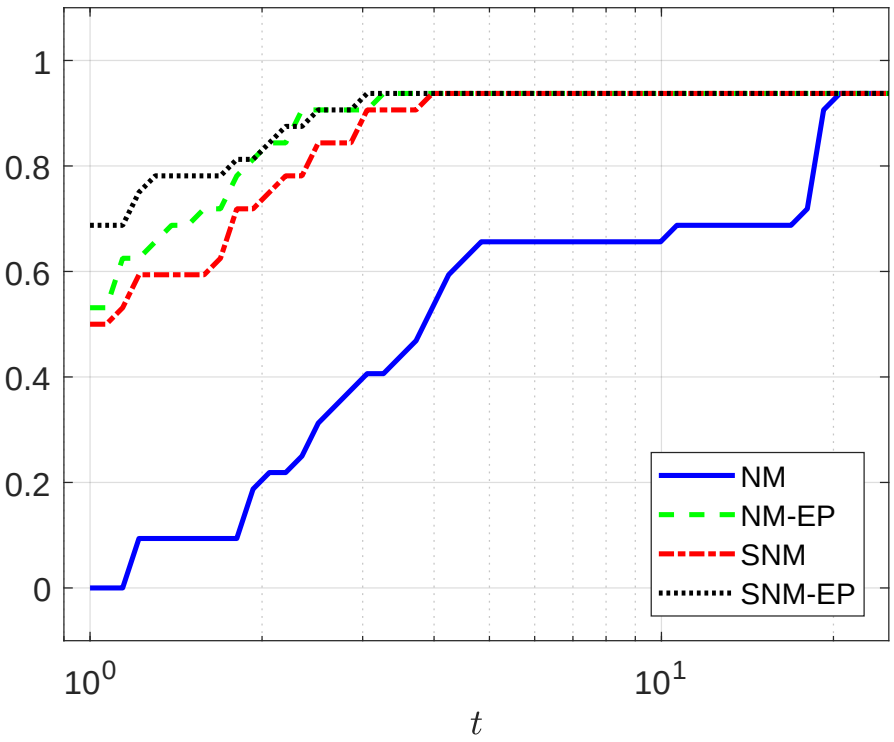

Fig. 3.1. Single runs from "standard" starting points.

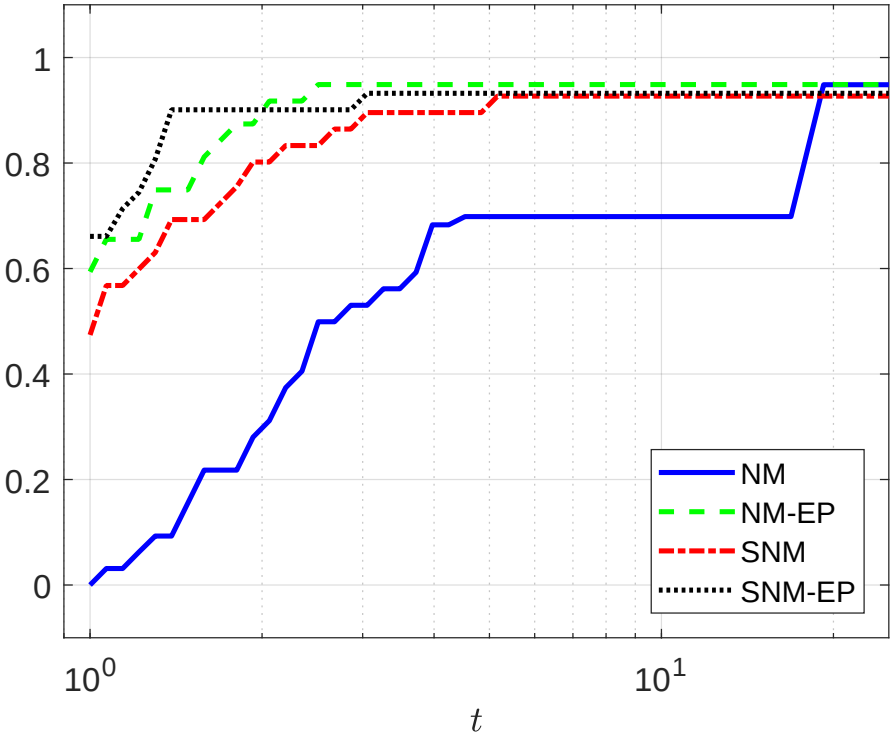

Fig. 3.2. Multiple runs from random starting points.

- SNM is Algorithm 3.1.
- SNM-EP is Algorithm 3.1, generating also an auxiliary sequence $\{\widehat{u}^k\}$ according to (2.5).

All algorithms were run with with parameter values $\sigma = 0.01$ and $\theta = 0.5$. Runs of NM and SNM were terminate with success declared once an iterate u^k was generated satisfying

$$\|\Phi(u^k)\| \leq 10^{-11}.$$

For NM-EP and SNM-EP, for $k = 1, 2, \ldots$, we first generated $\widehat{u}^k$ according to (2.5), and terminated the run with success if

$$\|\Phi(\widehat{u}^k)\| \leq 10^{-11}. \tag{4.1}$$

Convergence to the solution of interest $\bar{u}$ was declared when, at successful termination, we had

$$\|u^k - \bar{u}\| \leq 10^{-3}$$

(with u^k replaced by $\widehat{u}^k$ for NM-EP and SNM-EP when termination happened because of (4.1)). If successful termination did not occur after 50 iterations, or the backtracking procedures in Steps 3 of Algorithms 2.1 and 3.1 produced a trial value τ such that $\tau\|v^k\| \leq 10^{-10}$, the process was terminated with failure declaring that the step became too short to proceed.

All problems in [19] are supplied with "recommended" starting points. For each of the other test problems being used, we have also selected some starting point, trying to keep our choices reasonably generic. Information on "standard" starting points u^0 and solutions of interest $\bar{u}$ is collected in Table 3.1.

In the first part of the experiments, for each test problem, we have performed a single run of each algorithm using the specified "standard" starting point. The results (iteration counts until successful termination) are reported in Table 3.2; cases of failure are shown as "–". Information in Table 3.2 is further summarized in Figure 3.1 in the form of performance profile, proposed in [4]. The value of any function whose graph is presented in the figure at a point t of the horizontal axis corresponds to the portion of test problems for with the result (in our case, the iteration count) of the algorithm associated with this graph was no more than t times worse than the best result among all the algorithms being tested (the result of a failure is regarded infinitely many times worse than any result).

In the second part of the experiments, for each test problem, we have performed 1000 runs of each algorithm from randomly generated starting points with uniform distribution in the cubic neighborhood of the solution of interest, with the half-edge of the cube equal to 1. In Table 3.3, we report the percentage of successful runs, the average iteration count over successful runs, and the average distance to the solution of interest over cases when convergence to such solution was detected (separated by slashes). The cases when there were no successful runs are still marked by "–". The corresponding performance profile in Figure 3.2 is an adaptation of the original proposal in [4] to the case of multiple runs for each test problem, suggested in [16].

The failures for DIS64 and [1, Example 3.1] happen at Steps 2 of both Algorithms 2.1 and 3.1 because the Jacobians of the corresponding Φ at u^0 are singular, and the iteration equation cannot be solved. Generally, this is a typical reason of failures encountered in these experiments. There are also some rare cases when failure was declared because of too short steps (for quarquad and doubleknot), or when the iteration limit was achieved.

Another observation is that the full step of NM was always asymptotically accepted in these experiments, while for SNM, there were runs with the last step before successful termination using $\tau_k < 1$.

Looking at the results of the experiments, one can conclude that NM is definitely outperformed in terms of efficiency by any of the concurrent algorithms, and the reason for

that must be precisely the slow (linear, at best) convergence to so-called critical solutions, established in [9–11, 13] for Newton-type methods in the twice differentiable case. This might give an impression that the use of the Fischer–Burmeister complementarity function and semismooth Newton-type methods is definitely preferable with respect to the smooth complementarity function and usual Newton-type methods. Indeed, it appears that attraction to any special ("critical") singular solutions and the related special convergence pattern show up in a much less persistent way for nonsmooth equations. Nevertheless, passing from SNM to SNM-EP makes the efficiency somewhat higher, even though there currently exists no any theory supporting this behavior. Observe, however, that the impact of extrapolation on the Newton method is even more evident: NM-EP outperforms SNM in terms of efficiency, and is demonstrated performance similar to that of SNM-EP. All in all, NM-EP demonstrates a concurrent behavior when compared with SNM and SNM-EP, even though NM evidently suffers from the negative effect of critical solutions on its convergence rate.

5. CONCLUDING REMARKS

In this paper, we provided a numerical comparison of two approaches to numerical solution of nonlinear complementarity problems, one employing a smooth complementarity function, while another one making use of the (nonsmooth) Fischer–Burmeister complementarity function. The results obtained demonstrate that even in the presence of solutions violating strict complementarity, the first approach may still be fully competitive if equipped with the simplest extrapolation procedure intended for acceleration of convergence to such solutions. Theoretical justification of the effects demonstrated in this work will be a subject of future research.

ACKNOWLEDGEMENTS

This research was supported by the Russian Science Foundation Grant 23-11-20020 (https://rscf.ru/en/project/23-11-20020/).

REFERENCES

1. Arutyunov, A.V. & Izmailov, A.F. (2018) Stability of possibly nonisolated solutions of constrained equations with applications to complementarity and equilibrium problems. *Set-Valued Var. Anal.*, **26**, 327–352.
2. Bertsekas, D.P. (1999) *Nonlinear Programming*. 2nd edition. Belmont: Athena.
3. Daryina, A.N. & Izmailov, A.F. (2004) On identification of active indices in mixed complementarity problems. In: Bereznev, V.A. (ed.), Questions of Modelling and Analysis in Decision Making Problems. CCAS, Moscow, 72–87.
4. Dolan, E.D. & More, J.J. (2002) Benchmarking optimization software with performance profiles. *Math. Program.*, **91**, 201–213.
5. Fischer, A., Izmailov, A.F. & Jelitte, M. (2021) Newton-type methods near critical solutions of piecewise smooth nonlinear equations. *Comput. Optim. Appl.*, **80**, 587–615.
6. Fischer, A., Izmailov, A.F. & Solodov, M.V. (2019) Local attractors of Newton-type methods for constrained equations and complementarity problems with nonisolated solutions. *J. Optim. Theory Appl.*, **180**, 140–169.
7. Fischer, A., Izmailov, A.F. & Solodov, M.V. (2021) Accelerating convergence of the globalized Newton method to critical solutions of nonlinear equations. *Comput. Optim. Appl.*, **78**, 273–286.

8. Fischer, A., Izmailov, A.F. & Solodov, M.V. (2021) Unit stepsize for the Newton method close to critical solutions. *Math. Program.*, **187**, 697–721.
9. Griewank, A. (1980) *Analysis and modification of Mewton's method at singularities*. PhD Thesis. Australian National University, Canberra.
10. Griewank, A. (1980) Starlike domains of convergence for Newton's method at singularities. *Numer. Math.*, **35**, 95–111.
11. Griewank, A. (1985) On solving nonlinear equations with simple singularities or nearly singular solutions. *SIAM Rev.*, **27**, 537–563.
12. Izmailov, A.F. & Kurennoy, A.S. (2014) On regularity conditions for complementarity problems. *Comput. Optim. Appl.*, **57**, 667–684.
13. Izmailov, A.F. Kurennoy, A.S. & Solodov, M.V. (2018) Critical solutions of nonlinear equations: Local attraction for Newton-type methods. *Math. Program.*, **167**, 355–379.
14. Izmailov, A.F. & Solodov, M.V. (2001) Error bounds for 2-regular mappings with Lipschitzian derivatives and their applications. *Math. Program.*, **89**, 413–435.
15. Izmailov, A.F. & Solodov, M.V. (2014) *Newton-type methods for optimization and variational problems*. Springer Series in Operations Research and Financial Engineering. Cham: Springer.
16. Izmailov, A.F., Solodov, M.V. & Uskov, E.I. (2015) Combining stabilized SQP with the augmented Lagrangian algorithm *Comput. Optim. Appl.*, **62**, 405–429.
17. Kanzow, C. (1994) Some equation-based methods for the nonlinear complementarity problem. *Optim. Methods Software*, **3**, 327–340.
18. Mangasarian, O.L. (1976) Equivalence of the complementarity problem to a system of nonlinear equations. *SIAM J. Appl. Math.*, **31**, 89–92.
19. Oberlin, C. & Wright, S.J. (2009) An accelerated Newton method for equations with semismooth Jacobians and nonlinear complementarity problems. *Math. Program.*, **117**, 355–386.

Test problems	u^0	$\bar{u}$
quarp	0.9	1
DIS61	(0.2, 0.85)	$(0, (\sqrt{5}-1)/2)$
quarquad, 1	(0.1, 0.9)	(0, 1)
quarquad, 2	(0.9, 0.9)	(1, 0)
affknot1	(0.9, 0.1)	(0, 1)
affknot2	(0.5, 0.5)	(0, 1)
quadknot	(0.5, 0.5)	(0, 1)
munson4	(0, 0)	(1, 1)
DIS64	(2, 4)	(0, 0)
ne-hard	(10, 1, 10)	$(0, 0, \sqrt{200})$
doubleknot	(0.5, 0.5, 0.5, 0.5)	(1, 0, 0, 1)
quad1, 1	(0.9, -0.1)	(1, 0)
quad1, 2	(0.9, 0.1)	(1, 0)
quad2, 1	(-1, -1)	(0, 0)
quad2, 2	(1, 1)	(0, 0)
quarn	0.9	1
[1, Example 3.1]	(-0.5, -0.9)	(0, 0)
[1, Example 3.2]	(-0.5, -0.5)	(0, 0)
[1, Example 3.3], 1	(-0.9, 0.5)	(0, 0)
[1, Example 3.3], 2	(0.9, -0.9)	(1, 0)
[1, Example 3.4]	(1.9, -0.5)	(1, 0)
[3, Example 6]	(0.9, 0.1)	(0, 0)
[3, Example 7]	(0.9, 0.1)	(0, 0)
[3, Example 8]	(1.9, 0.9)	(0, 1)
[3, Example 9]	(0.9, 0.9)	(0, 0)
[5, Example 1]	-0.9	0
[5, Example 3], 1	(0.9, 0.1)	(0, 1)
[5, Example 3], 2	(-0.9, 0.5)	(0, 0)
[6, Example 4.2]	(0.9, 0.9)	(0, 0)
[6, Example 4.3]	(0.9, 0.9)	(0, 0)
[12, Example 3.4]	0.9	0
[12, Example 3.5]	(0.9, 0.9)	(0, 0)
[14, Example 1]	(0.9, 0.9)	(0, 1)
[14, Example 2]	(0.9, 0.1)	(0, 0)

Table 3.1. "Standard" starting points and solutions.

Test problems	NM	NM-EP	SNM	SNM-EP
quarp	15	14	15	13
DIS61	19	10	18	12
quarquad, 1	16	8	4	4
quarquad, 2	18	17	17	15
affknot1	20	2	6	6
affknot2	18	1	1	1
quadknot	18	9	16	11
munson4	19	10	20	8
DIS64	–	–	–	–
ne-hard	25	16	7	7
doubleknot	21	11	8	8
quad1	15	5	12	12
quad2	20	9	18	18
quarn	15	14	15	13
[1, Example 3.1]	–	–	–	–
[1, Example 3.2]	18	9	4	4
[1, Example 3.3], 1	20	11	7	7
[1, Example 3.3], 2	19	5	5	5
[1, Example 3.4]	20	11	18	6
[3, Example 6]	19	1	1	1
[3, Example 7]	19	1	1	1
[3, Example 8]	20	11	18	13
[3, Example 9]	19	1	1	1
[5, Example 1]	19	8	17	17
[5, Example 3], 1	21	16	5	5
[5, Example 3], 2	19	10	18	18
[6, Example 4.2]	19	1	1	1
[6, Example 4.3]	19	9	5	5
[12, Example 3.4]	19	1	1	1
[12, Example 3.5]	20	1	1	1
[14, Example 1]	19	7	15	4
[14, Example 2]	19	1	1	1

Table 3.2. Single runs from “standard” starting points.

Test problems	NM	NM-EP	SNM	SNM-EP
quarp	100/18/5.1e-04	100/16/6.7e-04	100/16/4.9e-04	100/15/6.9e-04
DIS61	100/17/1.7e-06	100/9/1.2e-06	100/15/2.4e-06	100/11/5.9e-09
quarquad, 1	100/20/6.7e-07	100/13/4.0e-07	100/10/1.7e-13	100/10/1.2e-13
quarquad, 2	97/20/5.6e-04	97/17/6.2e-04	100/17/5.6e-04	100/16/7.1e-04
affknot1	100/11/4.9e-06	100/5/2.8e-07	94/5/5.1e-05	94/5/5.2e-05
affknot2	100/19/4.5e-06	100/9/5.7e-13	100/5/2.7e-13	100/5/3.3e-13
quadknot	100/19/2.1e-06	100/9/1.3e-06	49/17/2.2e-06	68/10/1.3e-07
munson4	100/19/1.7e-06	100/9/1.5e-06	100/18/2.5e-06	100/7/1.4e-06
DIS64	100/19/1.7e-06	100/1/5.8e-17	100/1/2.5e-16	100/1/2.9e-16
ne-hard	100/22/2.1e-06	100/15/2.4e-07	100/6/5.6e-06	100/6/7.7e-06
doubleknot	88/19/2.6e-06	88/10/6.6e-13	70/5/4.4e-13	71/5/4.3e-13
quad1	100/19/7.2e-05	100/13/7.0e-05	100/16/2.3e-06	100/16/2.2e-06
quad2	100/19/6.3e-05	100/12/7.0e-05	100/16/2.3e-06	100/16/2.2e-06
quarn	100/19/1.0e-04	100/18/7.3e-04	100/19/6.2e-04	100/18/5.3e-04
[1, Example 3.1]	–	–	–	–
[1, Example 3.2]	50/18/2.3e-06	51/8/8.4e-13	53/4/5.7e-13	50/4/5.5e-13
[1, Example 3.3], 1	100/10/2.8e-06	100/4/3.4e-7	100/7/1.0e-10	100/5/8.9e-07
[1, Example 3.3], 2	100/5/2.3e-06	100/2/3.0e-05	100/10/7.3e-06	100/6/8.2e-06
[1, Example 3.4]	100/17/7.0e-05	100/11/6.0e-05	100/15/3.3e-06	100/7/3.9e-06
[3, Example 6]	100/19/1.9e-06	100/1/2.2e-16	100/1/1.8e-16	100/1/1.7e-16
[3, Example 7]	100/18/2.8e-06	100/1/2.7e-16	100/1/2.1e-16	100/1/2.0e-16
[3, Example 8]	100/17/1.9e-06	100/8/1.3e-06	100/15/2.5e-06	100/11/2.2e-12
[3, Example 9]	100/19/2.2e-06	100/1/6.0e-16	100/1/1.7e-16	100/1/1.7e-16
[5, Example 1]	100/19/7.2e-05	100/12/7.0e-05	100/16/2.3e-06	100/16/2.2e-06
[5, Example 3], 1	100/19/6.9e-05	100/13/3.8e-09	100/10/4.3e-13	100/10/5.4e-13
[5, Example 3], 2	100/19/6.9e-05	100/13/6.2e-05	100/15/2.2e-06	100/15/2.2e-06
[6, Example 4.2]	100/18/2.1e-06	100/1/1.4e-16	100/1/6.5e-15	100/1/7.4e-15
[6, Example 4.3]	100/19/1.6e-06	100/8/1.3e-10	100/5/5.3e-13	100/5/4.7e-13
[12, Example 3.4]	100/18/1.6e-06	100/1/6.3e-18	100/1/6.6e-17	100/1/6.6e-17
[12, Example 3.5]	100/19/1.6e-06	100/1/8.4e-16	100/1/3.5e-15	100/1/3.6e-15
[14, Example 1]	100/19/1.7e-06	100/9/1.2e-06	100/18/2.3e-06	100/6/1.3e-06
[14, Example 2]	100/18/2.0e-06	100/1/7.2e-16	100/1/9.5e-16	100/1/5.3e-16

Table 3.3. Multiple runs from random starting points.

