A novel generalization of the natural residual function and a neural network approach for the NCP

Jan Harold Alcantara, Jein-Shan Chen[1],*

Department of Mathematics, National Taiwan Normal University

Abstract

The natural residual (NR) function is a mapping often used to solve nonlinear complementarity problems (NCPs). Recently, three discrete-type families of complementarity functions with parameter $p \geq 3$ (where p is odd) based on the NR function were proposed. Using a neural network approach based on these families, it was observed from some preliminary numerical experiments that lower values of p provide better convergence rates. Moreover, higher values of p require larger computational time for the test problems considered. Hence, the value $p = 3$ is recommended for numerical simulations, which is rather unfortunate since we cannot exploit the wide range of values for the parameter p of the family of NCP functions. This paper is a follow-up study on the aforementioned results. Motivated by previously reported numerical results, we formulate a continuous-type generalization of the NR function and two corresponding symmetrizations. The new families admit a continuous parameter $p > 0$, giving us a wider range of choices for p and smooth NCP functions when $p > 1$. Moreover, the generalization subsumes the discrete-type generalization initially proposed. The numerical simulations show that in general, increased stability and better numerical performance can be achieved by taking values of p in the interval $(1, 3)$. This is indeed a significant improvement of preceding studies.

Keywords: Complementarity Functions, Natural Residual Function, Nonlinear Complementarity Problem

2010 MSC: 37-N40, 65-K10, 65-K15

*Corresponding author

Email addresses: 80640005s@ntnu.edu.tw (Jan Harold Alcantara), jschen@math.ntnu.edu.tw (Jein-Shan Chen)

[1] The author's work is supported by Ministry of Science and Technology, Taiwan.

1. Motivation

The nonlinear complementarity problem (NCP) is very important in engineering and economic applications [11], as well as in operations research [8]. In particular, given a mapping $F : \mathbb{R}^n \to \mathbb{R}^n$, the problem consists of finding a vector $x \in \mathbb{R}^n$ satisfying the conditions

$$x \geq 0, \quad F(x) \geq 0 \quad \text{and} \quad \langle x, F(x) \rangle = 0.$$

This problem will be denoted by NCP(F). The solution set of this problem is denoted by SOL(F) and the feasible region is denoted by $\Omega_{\rm F} := \{x \in \mathbb{R}^n \mid x \geq 0, F(x) \geq 0\}$. Some solution methods for NCP(F) can be found in [1, 6, 9, 10, 13, 14, 21, 17, 18, 25, 27]. A natural reformulation of NCP(F) is to consider the fixed-point problem

$$x = P_K(x - F(x)),$$

where P_K denotes the projection onto K with $K = \mathbb{R}^n_+$. Consequently, NCP(F) is equivalent to solving the equation

$$\begin{pmatrix} \phi_{\rm NR}(x_1, F_1(x)) \\ \vdots \\ \phi_{\rm NR}(x_n, F_n(x)) \end{pmatrix} = 0,$$

where

$$\phi_{\rm NR}(a, b) = a - (a - b)_+, \tag{1}$$

and $t_+ := \max\{t, 0\}$. The function $\phi_{\rm NR}$ is called the natural residual (NR) function. In fact, $\phi_{\rm NR}$ can be replaced by any other function $\phi : \mathbb{R}^2 \to \mathbb{R}$ with the property that

$$\phi(a, b) = 0 \quad \Longleftrightarrow \quad a \geq 0, \quad b \geq 0, \quad ab = 0, \tag{2}$$

that is, NCP(F) and the system

$$\Phi_{\rm F}(x) := \begin{pmatrix} \phi(x_1, F_1(x)) \\ \vdots \\ \phi(x_n, F_n(x)) \end{pmatrix} = 0 \tag{3}$$

are equivalent. A function satisfying (2) is known in the literature as an NCP-function. Other than the NR function, the generalized Fischer-Burmeister (GFB) function

$$\phi^p_{\rm FB}(a, b) = \|(a, b)\|_p - (a + b), \quad p > 1 \tag{4}$$

is another popular NCP function used in dealing with the complementarity problem. The GFB function is known as a “continuous” extension of the famous Fischer-Burmeister (FB) function given by

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

which can be obtained by taking $p = 2$ in expression (4). The generalization is considered continuous since p can take on any value in the interval $(1,\infty)$. Motivated by this extension, a generalization of the NR function (1) was formulated in [5] which is given by

$$\phi^p_{\text{NR}}(a,b) = a^p - [(a-b)_+]^p, \tag{5}$$

where p is an odd integer. Indeed, taking $p = 1$ yields the NR function (1). This generalization is considered to be of “discrete”' type since p can only take odd integral values. Note that ϕ^p_{NR} is a twice continuously differentiable function for $p \geq 3$ but its surface is not symmetric. To resolve this, two symmetrizations were proposed in [3], which are given by

$$\phi^p_{\text{S-NR}}(a,b) = \begin{cases} a^p - (a-b)^p & \text{if} \quad a \geq b, \\ b^p - (b-a)^p & \text{if} \quad a < b, \end{cases} \tag{6}$$

and

$$\psi^p_{\text{S-NR}}(a,b) = \begin{cases} a^p b^p - (a-b)^p b^p & \text{if} \quad a \geq b, \\ a^p b^p - (b-a)^p a^p & \text{if} \quad a < b, \end{cases} \tag{7}$$

where $p \geq 3$ is an odd integer. Properties of these three discrete-type families are elaborated in [2, 15].

The first attempt to use the above three discrete-type functions in designing solution methods for NCP was a neural network approach, which was presented in our previous work [2]. To construct the neural network, note that by taking Φ_{F} as defined in (3), the unconstrained minimization problem $\min_{x\in\mathbb{R}^n} \Psi_{\text{F}}(x)$, where

$$\Psi_{\text{F}}(x) = \frac{1}{2}\|\Phi_{\text{F}}(x)\|^2 = \frac{1}{2}\sum_{j=1}^{n} \phi(x_j, F_j(x))^2, \tag{8}$$

is equivalent to NCP(F). Then the gradient dynamical system

$$\frac{dx}{dt} = -\rho\nabla\Psi_{\text{F}}(x(t)), \quad x(0) = x^0 \tag{9}$$

is a natural neural network to be considered to deal with NCP(F). In [2], the discrete-type functions (5), (6), and (7) were used to form the merit function Ψ_{F}. Preliminary numerical experiments conducted in [2] showed that lower values of the parameter p result to faster convergence, although theoretical evidence for this phenomenon is yet to be verified. Moreover, longer computation time is usually required when a higher value of p is used. There are also test instances when larger values of p lead to ill-conditioning problems. In turn, the choice $p = 3$ may seem optimal in practice. In other words, the results suggest that choosing higher values of p need not be done. Consequently, this seems to suggest that the discrete-type generalization appears to be not very useful in the sense that only one member of each of the families is useful for numerical purposes. This motivates us to explore if there exists a continuous generalization of the NR function, i.e. a generalization parametrized by p which assumes values on some interval. This will provide us more values to consider for the tunable parameter, instead of just the odd integers with value at least 3.

We provide an affirmative answer to this problem. More precisely, the main contributions of this paper are as follows:

(i) We propose a continuous-type generalization of the NR function. The proposed function does not have a symmetric surface, but we provide two symmetrizations which also admit a continuous parameter p. This generalizes the results in [3, 5].

(ii) We establish several properties of these newly formulated NCP functions which are prerequisite to designing solution methods for the complementarity problem, which are not limited to the neural network approach. These properties extend the results in [15].

(iii) Stability properties of the neural network (9) will be established as important extensions of the results in [2].

More importantly,

(iv) We illustrate that the proposed continuous generalization is meaningful. In particular, it provides a wider range of values of p which offer better convergence rates than the ones based on the discrete-type generalization and their symmetrizations illustrated in [2].

(v) We provide theoretical evidence for the performance dependence on p of the gradient dynamical systems based on the three new families of NCP functions. This was not accomplished in [2].

(vi) This work is a significant improvement of the numerical results that were initially presented in [2], since the proposed families not only provide faster convergence rates but also higher stability. That is, the proposed generalizations yield neural networks which are less sensitive to initial conditions, which is one of the main issues encountered in [2].

In summary, this paper can be viewed as an important extension of the works presented in [2, 3, 5, 15] where the discrete-type generalization and two discrete-type symmetrizations of the NR function were studied.

This paper is organized as follows: In Section 2, we present our proposed continuous generalization of the NR function. We also prove important properties of the obtained generalization, which are extensions of the results given in [2, 3, 5, 15]. The theoretical properties proved in this section will later be used in the analysis of the neural network, which will be presented in Section 3. Results of several numerical experiments are presented in Section 4 and elaborately discussed in Section 5. Concluding remarks are presented in Section 6.

2. Continuous Generalization

Our proposed generalization of the NR function is defined as

$$\widetilde{\phi}^{p}_{\text{NR}}(a,b) = \operatorname{sgn}(a)|a|^p - [(a-b)_+]^p. \tag{10}$$

Here, we assume that p is any number in $(0,\infty)$ and

$$\operatorname{sgn}(t) := \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t = 0 \\ -1 & \text{if } t < 0 \end{cases}.$$

Observe that $\widetilde{\phi}^{p}_{\text{NR}}$ is an NCP function. Indeed, note that $\widetilde{\phi}^{p}_{\text{NR}}(a,b) = f(a) - f((a-b)_+)$, where $f(t) = \operatorname{sgn}(t)|t|^p$ which is a bijective function. It follows that

$$\widetilde{\phi}^{p}_{\text{NR}}(a,b) = 0 \Longleftrightarrow f(a) = f((a-b)_+) \Longleftrightarrow a = (a-b)_+ \Longleftrightarrow \phi_{\text{NR}}(a,b) = 0.$$

Note that if p is odd, then $\widetilde{\phi}^{p}_{\text{NR}} = \phi^{p}_{\text{NR}}$ and so the above generalization subsumes the discrete-type extension given by (5). We note herein that the transformation employed on ϕ_{NR} via the monotonic function f can always be applied to any NCP function of the form $\phi = \phi_1 - \phi_2$. This fact has also been noted in [12].

It is easy to see that the function (10) does not have a symmetric surface. Employing the same strategy as in [3], we propose two symmetrizations of $\widetilde{\phi}^p_{\rm NR}$ as

$$\widetilde{\phi}^p_{\rm S-NR}(a,b) = \begin{cases} \mathrm{sgn}(a)|a|^p - (a-b)^p & \text{if} \quad a \geq b, \\ \mathrm{sgn}(b)|b|^p - (b-a)^p & \text{if} \quad a < b, \end{cases} \tag{11}$$

and

$$\widetilde{\psi}^p_{\rm S-NR}(a,b) = \begin{cases} \mathrm{sgn}(a)\mathrm{sgn}(b)|a|^p|b|^p - \mathrm{sgn}(b)(a-b)^p|b|^p & \text{if} \quad a \geq b, \\ \mathrm{sgn}(a)\mathrm{sgn}(b)|a|^p|b|^p - \mathrm{sgn}(a)(b-a)^p|a|^p & \text{if} \quad a < b, \end{cases} \tag{12}$$

where $p > 0$. Notice that $\widetilde{\phi}^p_{\rm S-NR} = \phi^p_{\rm S-NR}$ and $\widetilde{\psi}^p_{\rm S-NR} = \psi^p_{\rm S-NR}$ whenever p is odd.

Proposition 2.1. *For any $p > 0$, the functions $\widetilde{\phi}^p_{\rm NR}$, $\widetilde{\phi}^p_{\rm S-NR}$, and $\widetilde{\psi}^p_{\rm S-NR}$ are NCP functions. Moreover, $\widetilde{\phi}^p_{\rm NR}(a,b) > 0$ ($\widetilde{\phi}^p_{\rm S-NR}(a,b) > 0$) if and only if $a > 0$ and $b > 0$, while $\widetilde{\psi}^p_{\rm S-NR}(a,b) \geq 0$ for all $(a,b) \in \mathbb{R}^2$.*

Proof. That $\widetilde{\phi}^p_{\rm NR}$ is an NCP function follows from the above discussion. Moreover, note that $a > 0$ and $b > 0$ if and only if $a > (a-b)_+$. Since $f(t) = \mathrm{sgn}(t)|t|^p$ is strictly increasing, we see that $a > 0$ and $b > 0$ if and only if $\mathrm{sgn}(a)|a|^p > \mathrm{sgn}((a-b)_+)|(a-b)_+|^p$, i.e. $\widetilde{\phi}^p_{\rm NR}(a,b) > 0$. On the other hand, observe that

$$\widetilde{\phi}^p_{\rm S-NR}(a,b) = \begin{cases} \widetilde{\phi}^p_{\rm NR}(a,b) & \text{if} \quad a \geq b, \\ \widetilde{\phi}^p_{\rm NR}(b,a) & \text{if} \quad a < b, \end{cases}, \tag{13}$$

and

$$\widetilde{\psi}^p_{\rm S-NR}(a,b) = \begin{cases} \mathrm{sgn}(b)|b|^p \widetilde{\phi}^p_{\rm NR}(a,b) & \text{if} \quad a \geq b, \\ \mathrm{sgn}(a)|a|^p \widetilde{\phi}^p_{\rm NR}(b,a) & \text{if} \quad a < b. \end{cases} \tag{14}$$

Using above identities and the fact that $\widetilde{\phi}^p_{\rm NR}$ is an NCP function, then $\widetilde{\phi}^p_{\rm S-NR}$ and $\widetilde{\psi}^p_{\rm S-NR}$ are also NCP functions with algebraic signs as specified in the proposition. □

In view of the above proposition, we may then view the functions $\widetilde{\phi}^p_{\rm NR}$, $\widetilde{\phi}^p_{\rm S-NR}$ and $\widetilde{\psi}^p_{\rm S-NR}$ as continuous generalizations of the functions $\phi^p_{\rm NR}$, $\phi^p_{\rm S-NR}$ and $\psi^p_{\rm S-NR}$. Now, we establish some properties of the above functions which will later be used in the neural network approach. We begin with smoothness properties. By $C^1(\Omega)$ and $C^2(\Omega)$, we mean the class of continuously differentiable and twice continuously differentiable functions defined on $\Omega \subset \mathbb{R}^n$, respectively.

Proposition 2.2. *The following result holds:*

(a) *If $p > 1$, the function $\widetilde{\phi}^p_{\rm NR} \in C^1(\mathbb{R}^2)$ whose gradient is given by*

$$\nabla \widetilde{\phi}^p_{\rm NR}(a,b) = p \left[\begin{array}{c} |a|^{p-1} - (a-b)^{p-1}{\rm sgn}((a-b)_+) \\ (a-b)^{p-1}{\rm sgn}((a-b)_+) \end{array} \right].$$

If $p > 2$, then $\widetilde{\phi}^p_{\rm NR} \in C^2(\mathbb{R}^2)$ whose Hessian is given by

$$\nabla^2 \widetilde{\phi}^p_{\rm NR}(a,b) = p(p-1) \left[\begin{array}{cc} {\rm sgn}(a)|a|^{p-2} - (a-b)^{p-2}{\rm sgn}((a-b)_+) & (a-b)^{p-2}{\rm sgn}((a-b)_+) \\ (a-b)^{p-2}{\rm sgn}((a-b)_+) & -(a-b)^{p-2}{\rm sgn}((a-b)_+) \end{array} \right].$$

(b) *If $p > 1$, the function $\widetilde{\phi}^p_{\rm S-NR} \in C^1(\Omega)$ where $\Omega := \{(a,b) \,|\, a \neq b\}$. In this case, the gradient of $\widetilde{\phi}^p_{\rm S-NR}$ is given by*

$$\nabla \widetilde{\phi}^p_{\rm S-NR}(a,b) = \begin{cases} p\,[\,|a|^{p-1} - (a-b)^{p-1},\ (a-b)^{p-1}\,]^T & \text{if} \quad a > b, \\ p\,[\,(b-a)^{p-1},\ |b|^{p-1} - (b-a)^{p-1}\,]^T & \text{if} \quad a < b. \end{cases}$$

Further, $\widetilde{\phi}^p_{\rm S-NR}$ is differentiable at $(0,0)$ with $\nabla \widetilde{\phi}^p_{\rm S-NR}(0,0) = [0,0]^T$. If $p > 2$, then $\widetilde{\phi}^p_{\rm S-NR} \in C^2(\Omega)$ with Hessian given by

$$\nabla^2 \widetilde{\phi}^p_{\rm S-NR}(a,b) = \begin{cases} p(p-1) \left[\begin{array}{cc} {\rm sgn}(a)|a|^{p-2} - (a-b)^{p-2} & (a-b)^{p-2} \\ (a-b)^{p-2} & -(a-b)^{p-2} \end{array} \right] & \text{if} \quad a > b, \\ p(p-1) \left[\begin{array}{cc} -(b-a)^{p-2} & (b-a)^{p-2} \\ (b-a)^{p-2} & {\rm sgn}(b)|b|^{p-2} - (b-a)^{p-2} \end{array} \right] & \text{if} \quad a < b. \end{cases}$$

(c) *If $p > 1$, then $\widetilde{\psi}^p_{\rm S-NR} \in C^1(\mathbb{R}^2)$ whose gradient is given by*

$$\nabla \widetilde{\psi}^p_{\rm S-NR}(a,b) = \begin{cases} p \left[\begin{array}{c} {\rm sgn}(b)|b|^p(|a|^{p-1} - (a-b)^{p-1}) \\ {\rm sgn}(a)|a|^p|b|^{p-1} - (a-b)^p|b|^{p-1} + {\rm sgn}(b)(a-b)^{p-1}|b|^p \end{array} \right] & \text{if} \quad a > b, \\ p|a|^{2p-1} \left[\begin{array}{c} 1 \\ 1 \end{array} \right] & \text{if} \quad a = b, \\ p \left[\begin{array}{c} {\rm sgn}(b)|a|^{p-1}|b|^p - (b-a)^p|a|^{p-1} + {\rm sgn}(a)(b-a)^{p-1}|a|^p \\ {\rm sgn}(a)|a|^p(|b|^{p-1} - (b-a)^{p-1}) \end{array} \right] & \text{if} \quad a < b, \end{cases}$$

If $p > 2$, then $\widetilde{\psi}^p_{\text{S-NR}} \in C^2(\mathbb{R}^2)$ whose Hessian is given by

$$\nabla^2 \widetilde{\psi}^p_{\text{S-NR}}(a,b) = \begin{cases} p \begin{bmatrix} \begin{array}{c} (p-1)[\text{sgn}(a)\text{sgn}(b)|a|^{p-2}|b|^p] \\ -(p-1)(a-b)^{p-2}\text{sgn}(b)|b|^p \end{array} & \begin{array}{c} (p-1)(a-b)^{p-2}\text{sgn}(b)|b|^p \\ +p[|a|^{p-1}-(a-b)^{p-1}]|b|^{p-1} \end{array} \\ & \\ \begin{array}{c} (p-1)(a-b)^{p-2}\text{sgn}(b)|b|^p \\ +p[|a|^{p-1}-(a-b)^{p-1}]|b|^{p-1} \end{array} & \begin{array}{c} (p-1)[\text{sgn}(a)\text{sgn}(b)|a|^p|b|^{p-2}] \\ -(p-1)(a-b)^p\text{sgn}(b)|b|^{p-2} \\ +2p(a-b)^{p-1}|b|^{p-1} \\ -(p-1)(a-b)^{p-2}\text{sgn}(b)|b|^p \end{array} \end{bmatrix} & \\ & \text{if } a > b, \\ p \begin{bmatrix} (p-1)\text{sgn}(a)\text{sgn}(b)|a|^{p-2}|b|^p & p|a|^{p-1}|b|^{p-1} \\ p|a|^{p-1}|b|^{p-1} & (p-1)\text{sgn}(a)\text{sgn}(b)|a|^p|b|^{p-2} \end{bmatrix} & \\ & \text{if } a = b, \\ p \begin{bmatrix} \begin{array}{c} (p-1)[\text{sgn}(a)\text{sgn}(b)|a|^{p-2}|b|^p] \\ -(p-1)(b-a)^p\text{sgn}(a)|a|^{p-2} \\ +2p(b-a)^{p-1}|a|^{p-1} \\ -(p-1)(b-a)^{p-2}\text{sgn}(a)|a|^p \end{array} & \begin{array}{c} (p-1)(b-a)^{p-2}\text{sgn}(a)|a|^p \\ +p[|b|^{p-1}-(b-a)^{p-1}]|a|^{p-1} \end{array} \\ & \\ \begin{array}{c} (p-1)(b-a)^{p-2}\text{sgn}(a)|a|^p \\ +p[|b|^{p-1}-(b-a)^{p-1}]|a|^{p-1} \end{array} & \begin{array}{c} (p-1)[\text{sgn}(a)\text{sgn}(b)|a|^p|b|^{p-2}] \\ -(p-1)(b-a)^{p-2}\text{sgn}(a)|a|^p \end{array} \end{bmatrix} & \\ & \text{if } a < b. \end{cases}$$

Proof. Note that $f(t) = \text{sgn}(t)|t|^p$ is continuously differentiable when $p > 1$ with $f'(t) = p|t|^{p-1}$. Moreover, f is twice continuously differentiable when $p > 2$ with $f''(t) = p(p-1)\text{sgn}(t)|t|^{p-2}$. Using these and the alternative formulas given in (13) and (14), the gradients and Hessians can be easily obtained. The calculations are omitted. □

The above proposition is a generalization of [5, Proposition 2.2], [3, Proposition 2.2 and Proposition 3.2], and [15, Proposition 4.3]. On the other hand, the following result is an extension of [15, Proposition 3.4, Proposition 4.5, and Proposition 5.4].

Proposition 2.3. *Let $p > 1$. Then, the following hold:*

(a) $$\nabla_a \widetilde{\phi}^p_{\text{NR}}(a,b) \cdot \nabla_b \widetilde{\phi}^p_{\text{NR}}(a,b) \begin{cases} > 0 & \text{on } \{(a,b) \mid a > b > 0 \text{ or } a > b > 2a\}, \\ = 0 & \text{on } \{(a,b) \mid a \leq b \text{ or } a > b = 2a \text{ or } a > b = 0\}, \\ < 0 & \text{otherwise.} \end{cases}$$

(b) $$\nabla_a \widetilde{\phi}^p_{\text{S-NR}}(a,b) \cdot \nabla_b \widetilde{\phi}^p_{\text{S-NR}}(a,b) \begin{cases} > 0 & \text{on } \{(a,b) \mid a > b > 0 \text{ or } a > b > 2a\} \\ & \quad \text{and on } \{(a,b) \mid b > a > 0 \text{ or } b > a > 2b\}, \\ = 0 & \text{on } \{(a,b) \mid \widetilde{\phi}^p_{\text{S-NR}}(a,b) = 0 \text{ or } a > b = 2a \text{ or } b > a = 2b\}, \\ < 0 & \text{otherwise.} \end{cases}$$

(c) $\nabla_a \widetilde{\psi}^p_{\text{S-NR}}(a,b) \cdot \nabla_b \widetilde{\psi}^p_{\text{S-NR}}(a,b) > 0$ *on the first quadrant* $\mathbb{R}^2_{++}$, *and* $\widetilde{\psi}^p_{\text{S-NR}}(a,b) = 0 \Longleftrightarrow \nabla \widetilde{\psi}^p_{\text{S-NR}}(a,b) = 0$.

Proof. Using Proposition 2.2(a),

$$\begin{aligned} \nabla_a \widetilde{\phi}^p_{\text{NR}}(a,b) \cdot \nabla_b \widetilde{\phi}^p_{\text{NR}}(a,b) &= p^2[|a|^{p-1} - (a-b)^{p-1}\text{sgn}((a-b)_+)](a-b)^{p-1}\text{sgn}((a-b)_+) \\ & \begin{cases} p^2[|a|^{p-1} - (a-b)^{p-1}](a-b)^{p-1} & \text{if } a > b \\ 0 & \text{if } a \leq b \end{cases}. \end{aligned}$$

Suppose now that $a > b$. Since $g(t) := t^{p-1}$ is a strictly increasing function on $[0, \infty)$, $|a|^{p-1} - (a-b)^{p-1} > 0$ if and only if $|a| > a - b$, which happens if and only if $b > 0$ or $b > 2a$. This establishes Proposition 2.3(a). Statement (b) easily follows from (a), while (c) can be easily verified using the result of Proposition 2.2(c). □

We now establish the growth behavior of the proposed families of functions. We first establish the following simple lemma.

Lemma 2.1. *For any $x \in [0,1]$ and any $p > 0$, we have*

$$(1-x)^p \leq \frac{1}{1+px}.$$

Proof. Define $f : [0,1] \to \mathbb{R}$ by $f(x) = (1-x)^p(1+px)$. A simple calculation yields $f'(x) = -p(p+1)x(1-x)^{p-1}$. Then, f monotonically decreases on $[0,1]$ from $f(0) = 1$ to $f(1) = 0$. Consequently, $0 \leq f(x) \leq 1$. This completes the proof. □

Proposition 2.4. *Let $\phi \in \{\widetilde{\phi}^p_{\mathrm{NR}}, \widetilde{\phi}^p_{\mathrm{S-NR}}, \widetilde{\psi}^p_{\mathrm{S-NR}}\}$. Then $|\phi(a^k, b^k)| \to \infty$ for any sequence $\{(a^k, b^k)\}_{k=1}^{\infty}$ in $\mathrm{I\!R}^2$ such that $|a^k| \to \infty$ and $|b^k| \to \infty$.*

Proof. The proposition follows from the preceding lemma and analogous arguments in the proof of [2, Lemma 5.1]. □

3. Stability Analysis

In this section, we consider the neural network given by (9) using the functions $\widetilde{\phi}^p_{\mathrm{NR}}$, $\widetilde{\phi}^p_{\mathrm{S-NR}}$ and $\widetilde{\psi}^p_{\mathrm{S-NR}}$. The corresponding merit functions defined by (8) will be denoted, respectively, by $\widetilde{\Psi}^p_{\mathrm{NR}}$, $\widetilde{\Psi}^p_{\mathrm{S1-NR}}$ and $\widetilde{\Psi}^p_{\mathrm{S2-NR}}$. The case when p is an odd integer greater than 1 is the neural network studied in [2].

We note that since the results presented in Section 2 generalize the results for the discrete families originally formulated in [3, 5, 15], then the discussion presented in [2] can be extended to establish the properties of the induced merit functions $\widetilde{\Psi}^p_{\mathrm{NR}}$, $\widetilde{\Psi}^p_{\mathrm{S1-NR}}$ and $\widetilde{\Psi}^p_{\mathrm{S2-NR}}$ corresponding to the continuous generalization. In the following proposition, we summarize the properties of these merit functions. For conciseness and clarity, we present a shortened proof of the following result pointing out the arguments that needed to be modified in the proofs of the results in [2]. We refer the reader to the monograph [8] for definitions and properties of nonlinear mappings (P_0-functions, monotone functions, etc.) and the book [22] for standard results in the theory of ordinary differential equations.

Proposition 3.1. *Let $p > 1$. Then the following hold:*

(a) *If $(\nabla F - I)$ is a P-matrix, then every stationary point of $\widetilde{\Psi}^p_{\mathrm{NR}}$ is a global minimizer.*

(b) *If $F(x^*) \geq 0$, $(\nabla F(x^*) - I)$ is a P_0-matrix and x^* is a stationary point of $\widetilde{\Psi}^p_{\mathrm{NR}}$, then x^* is a global minimizer of $\widetilde{\Psi}^p_{\mathrm{NR}}$.*

(c) *Suppose that $x^* \in \Omega_{\mathrm{F}}$ and $\nabla F(x^*)$ is a P_0-matrix. If $x^*$is a stationary point of $\widetilde{\Psi}^p_{\mathrm{S1-NR}}$ or $\widetilde{\Psi}^p_{\mathrm{S2-NR}}$, then x^* is a global minimizer.*

Proof. To prove (a) and (b), we define two diagonal matrices $A(x^*)$ and $B(x^*)$ where

$$A_{ii}(x^*) = |x_i^*|^{p-1} \qquad \text{and} \qquad B_{ii}(x^*) = (x_i^* - F_i(x^*))\mathrm{sgn}(x_i^* - F_i(x^*))_+,$$

where x^* is an equilibrium point of (9) with $\Psi_{\rm F} = \widetilde{\Psi}^p_{\rm NR}$. Then, analogous arguments as in the proof of [2, Proposition 4.4 and Remark 4.1] lead to the desired conclusion. To prove (c), we proceed as in the proof of [2, Proposition 4.5]. That is, we verify the following properties:

(P1) $\forall (a,b) \in {\rm IR}^2_+$, we have $\nabla_a \psi(a,b) \cdot \nabla_b \psi(a,b) \geq 0$; and

(P2) $\forall (a,b) \in {\rm IR}^2_+$, we have $\nabla_a \psi(a,b) = 0 \Longleftrightarrow \nabla_b \psi(a,b) = 0 \Longleftrightarrow \phi(a,b) = 0$,

where $\psi := \frac{1}{2}\phi^2$ and $\phi \in \{\widetilde{\phi}^p_{\rm S-NR}, \widetilde{\psi}^p_{\rm S-NR}\}$. Property (P1) can be easily verified. To show (P2), we only need to show that given $a, b \geq 0$, the following holds:

(i) $\nabla_a \widetilde{\phi}^p_{\rm S-NR}(a,b) = 0$ implies $\widetilde{\phi}^p_{\rm S-NR}(a,b) = 0$; and

(ii) $\nabla_a \widetilde{\psi}^p_{\rm S-NR}(a,b) = 0$ implies $\widetilde{\psi}^p_{\rm S-NR}(a,b) = 0$.

We first prove (i). If $\nabla_a \widetilde{\phi}^p_{\rm S-NR}(a,b) = 0$, then we see from Proposition 2.2 (b) that we must have $a > b$ or $a = b = 0$. Otherwise, $\nabla_a \widetilde{\phi}^p_{\rm S-NR}(a,b) = p(b-a)^{p-1}$ would be positive. If $a = b = 0$, then $\widetilde{\phi}^p_{\rm S-NR}(a,b) = 0$ as desired. If $a > b$, then $0 = \frac{1}{p}\nabla_a \widetilde{\phi}^p_{\rm S-NR}(a,b) = a^{p-1} - (a-b)^{p-1}$. Since $t \mapsto t^{p-1}$ is strictly increasing on $[0,\infty)$, then $a = a - b$, i.e. $b = 0$. Then $\widetilde{\phi}^p_{\rm S-NR}(a,b) = 0$ since $a > b = 0$ and $\widetilde{\phi}^p_{\rm S-NR}$ is an NCP function. To prove (ii), assume that $\nabla_a \widetilde{\psi}^p_{\rm S-NR}(a,b) = 0$. From Proposition 2.2 (c), we must have

$$0 = \frac{1}{p}\nabla_a \widetilde{\psi}^p_{\rm S-NR}(a,b) = \begin{cases} a^{p-1}b^p - (a-b)^{p-1}b^p & \text{if} \quad a \geq b, \\ a^{p-1}b^p - (b-a)^p a^{p-1} + (b-a)^{p-1}a^p & \text{if} \quad a < b. \end{cases}$$

If $a \geq b$, then we can proceed as in [2, Prop 4.5]. If $a < b$, then

$$0 = a^{p-1}b^p - (b-a)^p a^{p-1} + (b-a)^{p-1}a^p = a^{p-1}(b^p - (b-a)^p + (b-a)^{p-1}a). \tag{15}$$

From here, we conclude that $a = 0$. Otherwise, we must have $b^p > (b-a)^p$ and so $b^p - (b-a)^p + (b-a)^{p-1}a > (b-a)^{p-1}a > 0$. This contradicts (15). Hence, $a = 0$ and since $b > a = 0$, we obtain that $\widetilde{\psi}^p_{\rm S-NR}(a,b) = 0$ by definition of an NCP function. □

In view of the above proposition and the stability analysis presented in [2], we present herein analogous stability results. The proofs are similar to corresponding propositions for the discrete generalization established in [2], and are thus omitted. In particular, Proposition 3.2 (a) follows from [2, Theorem 5.1], Proposition 3.2 (b) and (c) follow from [2, Theorem 5.2], and Proposition 3.2 (d) is a consequence of [2, Theorem 5.2].

Proposition 3.2. *Let x^* be an equilibrium point of* (9)*.*

- **(a)** *If $\Psi_{\rm F} \in \{\widetilde{\Psi}^p_{\rm NR}, \widetilde{\Psi}^p_{\rm S1-NR}, \widetilde{\Psi}^p_{\rm S2-NR}\}$ and F is a uniformly P-function, then the solution to* (9) *through any $x^0 \in {\rm I\!R}^n$ converges to x^*.*
- **(b)** *If $\Psi_{\rm F} = \widetilde{\Psi}^p_{\rm NR}$, then $x^* \in {\rm SOL}(F)$ provided that $(\nabla F - I)$ is a P-matrix. If x^* is isolated, then it is asymptotically stable.*
- **(c)** *If $x^* \in \Omega_F$ and $\Psi_{\rm F} = \widetilde{\Psi}^p_{\rm S1-NR}$ or $\Psi_{\rm F} = \widetilde{\Psi}^p_{\rm S2-NR}$, then $x^* \in {\rm SOL}(F)$ provided that F is a P_0-function. If x^* is isolated, then it is asymptotically stable.*
- **(d)** *If $\nabla \Phi_{\rm F}(x^*)$ is nonsingular, where $\phi \in \{\widetilde{\phi}^p_{\rm NR}, \widetilde{\phi}^p_{\rm S-NR}, \widetilde{\psi}^p_{\rm S-NR}\}$, and x^* is isolated, then $x^* \in {\rm SOL}(F)$ and x^* is exponentially stable.*

The parameter p has a very significant influence in the rate of convergence of the neural network. For the discrete type families, a few set of test problems was considered in [2], where the numerical experiments revealed that a lower value of $p \in \{3, 5, 7, \dots\}$ often provides faster convergence. However, there is no theoretical evidence yet for this phenomenon.

In fact, as we shall see in Section 4, different convergence behaviors can be observed when we vary the values of p. In particular, a lower value of p does not always lead to faster convergence. There are test instances when a higher value of p offers faster convergence rate. The numerical experiments presented in the next section suggest that there is no simple relation that can be obtained regarding the performance dependence on p of the neural network (9) with $\Psi_{\rm F} \in \{\widetilde{\Psi}^p_{\rm NR}, \widetilde{\Psi}^p_{\rm S1-NR}, \widetilde{\Psi}^p_{\rm S2-NR}\}$. Moreover, the simulations suggest that initial conditions have a significant influence on the performance of the neural network and its dependence on p. To make sense of these phenomenon, we establish the following theorem. The first part of the proof is a derivation of an error bound for the NCP(F) (see equation (18)) where F is a locally Lipschitz uniformly P-function. The technique employed in the derivation is similar to the idea used in [8, Proposition 6.3.1].

Theorem 3.1. *Consider the neural network* (9) *with $\Psi_{\rm F} = \widetilde{\Psi}^p_{\rm S1-NR}$ for a given $p > 1$. Suppose that $x^* \in {\rm SOL}(F)$ is exponentially stable and F is a uniformly P function that is locally Lipschitz continuous. Then there exist positive constants K, ω and δ such that for all $t \geq 0$, we have*

$$\|x(t) - x^*\| \leq K \left(\frac{p+1}{p} \sqrt{2\widetilde{\Psi}^p_{\rm S1-NR}(x^0)} \right)^{\frac{1}{p}} e^{-\omega t} \qquad \forall x^0 \in \Omega_{\rm F} \cap N_\delta(x^*),$$

where $N_\delta(x^*) = \{y : \|y - x^*\| < \delta\}$.

Proof. Suppose F is uniformly P with modulus $\kappa > 0$. Given $x \in \mathbb{R}^n$, let $j \in \{1, \dots, n\}$ such that

$$(x_j - x_j^*)(F_j(x) - F_j(x^*)) \geq (x_i - x_i^*)(F_i(x) - F_i(x^*)) \quad \forall i = 1, \dots, n.$$

Then

$$\kappa \|x - x^*\|^2 \leq (x_j - x_j^*)(F_j(x) - F_j(x^*)) = -x_j F_j(x^*) - (x_j^* - x_j)F_j(x). \tag{16}$$

Meanwhile, note that $(s - t_+)(t_+ - t) \geq 0$ for any $s \geq 0$ and $t \in \mathbb{R}$. Since $\min\{x_j, F_j(x)\} = x_j - (x_j - F_j(x))_+$, then taking $s = x_j^* \geq 0$ and $t = x_j - F_j(x)$, we have

$$(x_j^* - x_j + \min\{x_j, F_j(x)\})(F_j(x) - \min\{x_j, F_j(x)\}) \geq 0$$

which implies that

$$(x_j^* - x_j)F_j(x) \geq (x_j^* - x_j)\min\{x_j, F_j(x)\} - F_j(x)\min\{x_j, F_j(x)\}. \tag{17}$$

Since $x_j \geq \min\{x_j, F_j(x)\}$ and $F_j(x^*) \geq 0$, we have from inequalities (16) and (17) that

$$\begin{aligned}\kappa \|x - x^*\|^2 &\leq [(F_j(x) - F_j(x^*)) - (x_j^* - x_j)]\min\{x_j, F_j(x)\} \\ &\leq (\|F(x) - F(x^*)\| + \|x - x^*\|)|\min\{x_j, F_j(x)\}|\end{aligned}$$

Since F is locally Lipschitz, we conclude that given any $x \in \mathbb{R}^n$ in some neighborhood of x^*, there exists an index $j = j(x)$ and $L > 0$ such that

$$\kappa \|x - x^*\|^2 \leq (1 + L) \cdot |\min\{x_j, F_j(x)\}| \cdot \|x - x^*\|. \tag{18}$$

Now, let $x^0 \in \Omega_{\mathrm{F}}$. We have from part (a) of the proof of [2, Lemma 5.1] and using Lemma 2.1 that $\widetilde{\phi}^p_{\mathrm{S-NR}}(a, b) \geq \frac{p}{p+1}(\min\{a, b\})^p$ for any $a, b \geq 0$. By (18), there exists $j = j(x^0) \in \{1, \dots, n\}$ such that

$$\kappa \|x^0 - x^*\| \leq (1 + L) \cdot \left[\frac{p+1}{p}\widetilde{\phi}^p_{\mathrm{S-NR}}(x_j^0, F_j(x^0))\right]^{\frac{1}{p}}. \tag{19}$$

Since x^* is exponentially stable, there exist positive constants δ, c and ω such that for any $t \geq 0$, 150 $\|x(t) - x^*\| \leq ce^{-\omega t}\|x^0 - x^*\|$ for all $x^0 \in N_\delta(x^*)$. This, together with inequality (19), gives the desired result with $K := \frac{c}{\kappa}(1 + L)$. □

Similarly, we get the following error bound result for the other two merit functions $\widetilde{\Psi}^p_{\mathrm{NR}}$ and $\widetilde{\Psi}^p_{\mathrm{S2-NR}}$.

Theorem 3.2. *Consider the neural network* (9) *for a given* $p > 1$, *and let* $x^* \in \mathrm{SOL}(F)$ *be exponentially stable. Suppose that* F *is a uniformly* P*-function and locally Lipschitz continuous. Then*

(a) *If* $\Psi_{\mathrm{F}} = \widetilde{\Psi}^p_{\mathrm{NR}}$, *there exist positive constants* K, ω *and* δ *such that for all* $t \geq 0$, *we have*

$$\|x(t) - x^*\| \leq K \left(\frac{p+1}{p} \sqrt{2\widetilde{\Psi}^p_{\mathrm{NR}}(x^0)} \right)^{\frac{1}{p}} e^{-\omega t} \qquad \forall x^0 \in \Omega_{\mathrm{F}} \cap N_\delta(x^*).$$

(b) *If* $\Psi_{\mathrm{F}} = \widetilde{\Psi}^p_{\mathrm{S2-NR}}$, *there exist positive constants* K, ω *and* δ *such that for all* $t \geq 0$, *we have*

$$\|x(t) - x^*\| \leq K \left(\frac{p+1}{p} \sqrt{2\widetilde{\Psi}^p_{\mathrm{S2-NR}}(x^0)} \right)^{\frac{1}{2p}} e^{-\omega t} \qquad \forall x^0 \in \Omega_{\mathrm{F}} \cap N_\delta(x^*).$$

Proof. For $a \geq b \geq 0$, then $\widetilde{\phi}^p_{\mathrm{NR}}(a,b) = \widetilde{\phi}^p_{\mathrm{S-NR}}(a,b) \geq \frac{p}{p+1} b^p$ as in part (a) of the proof of [2, Lemma 5.1]. When $0 \leq a < b$, we have $\widetilde{\phi}^p_{\mathrm{NR}}(a,b) = a^p \geq \frac{p}{p+1} a^p$. It follows that $\widetilde{\phi}^p_{\mathrm{NR}}(a,b) \geq \frac{p}{p+1} (\min\{a,b\})^p$. On the other hand, using the identity (14) and the fact that $\widetilde{\phi}^p_{\mathrm{S-NR}}(a,b) \geq \frac{p}{p+1}(\min\{a,b\})^p$ for any $a, b \geq 0$, we derive that $\widetilde{\psi}^p_{\mathrm{S-NR}}(a,b) \geq \frac{p}{p+1}(\min\{a,b\})^{2p}$. Using these identities and the same arguments as in Theorem 3.1, we get the desired inequalities. □

As mentioned in the discussion before Theorem 3.1, there is no simple relation describing the influence of p. To see this clearly, consider the function $\widetilde{\phi}^p_{\mathrm{S-NR}}$. From the proof of Theorem 3.1, there exists an index $j = j(x^0)$ given any $x^0 \in \Omega_{\mathrm{F}}$ close enough to x^* such that

$$\|x(t) - x^*\| \leq \frac{c(1+L)}{\kappa} \left[\frac{p+1}{p} \widetilde{\phi}^p_{\mathrm{S-NR}}(x^0_j, F_j(x^0)) \right]^{\frac{1}{p}} e^{-\omega t,} \qquad \forall t \geq 0. \tag{20}$$

For a fixed $x^0 \in \Omega_{\mathrm{F}} \cap N_\delta(x^*)$, we define the function

$$g_{a,b}(p) := \left[\frac{p+1}{p} \widetilde{\phi}^p_{\mathrm{S-NR}}(a,b) \right]^{\frac{1}{p}},$$

where $a = x^0_j$ and $b = F_j(x^0)$ and $p > 1$. Without loss of generality, by taking into account the symmetry of $\widetilde{\phi}^p_{\mathrm{S-NR}}$, we may suppose that $a \geq b$. Then

$$g_{a,b}(p) = \left[\frac{p+1}{p} (a^p - (a-b)^p) \right]^{\frac{1}{p}}.$$

Note that $M := \lim_{p \to \infty} g_{a,b}(p) = a$. As we shall see in the following example, the function $g_{a,b}$ is not necessarily monotonic, and the values of a and b have a significant effect on the behavior of $g_{a,b}$.

Example 3.1. In Figure 1, we see that $g_{a,b}(p)$ increases for increasing values of p for $(a,b) = (4,0.5)$ on the interval $(1,25]$. In view of the error bound (20), this indicates that lower values of $p \in (1,25]$ will provide faster convergence rate. We shall note that $g_{4,0.5}$ does not continue to increase on $[25,\infty)$. In particular, it is increasing from $p = 1$ to $p \approx 34.4458$, then decreases afterwards (see Figure 2). On the other hand, Figure 1 suggests that for $(a,b) = (4,3)$, higher values of p result to faster convergence rate. Finally, the nonmonotonic graph depicted in Figure 1 for $(a,b) = (4,2)$ indicates different convergence behaviors for values of p on different intervals. However, observe too that the values of $g_{4,2}(p)$ are close to the limit value $M = 4$ when p belongs to some interval $(1, 1+\varepsilon)$, for some small $\varepsilon > 0$.

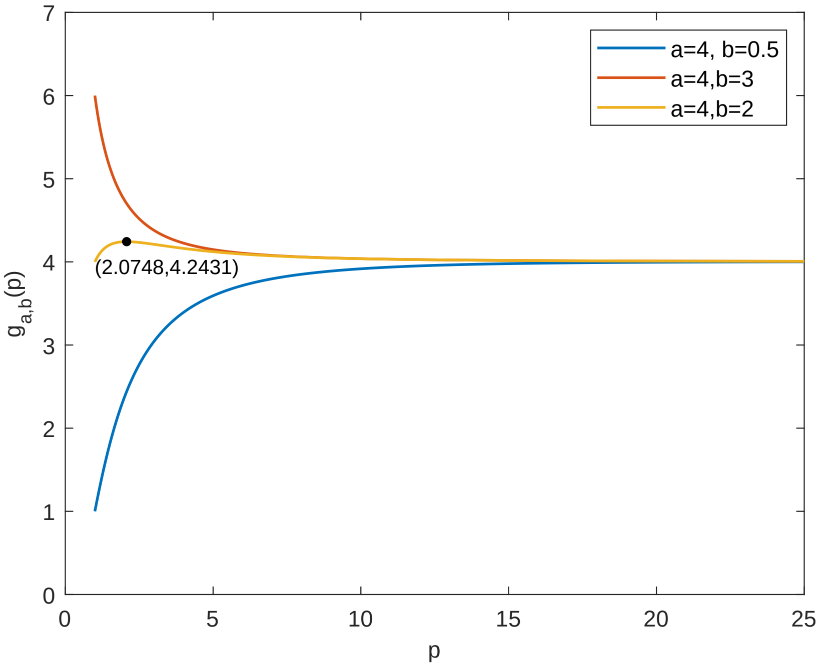


Figure 1: Graph of upper bound for the error term $\|x(t) - x^*\|$ for some values of a and b with $a, b \geq 0$ and $a > b$.

Remark 3.1.

(a) From the preceding example, it is evident that the influence on the upper bound of varying the values of p is heavily dependent on the chosen initial condition for the neural network (9). Despite this, we wish to point out that there is minimal change in the value of $g_{a,b}(p)$ for large

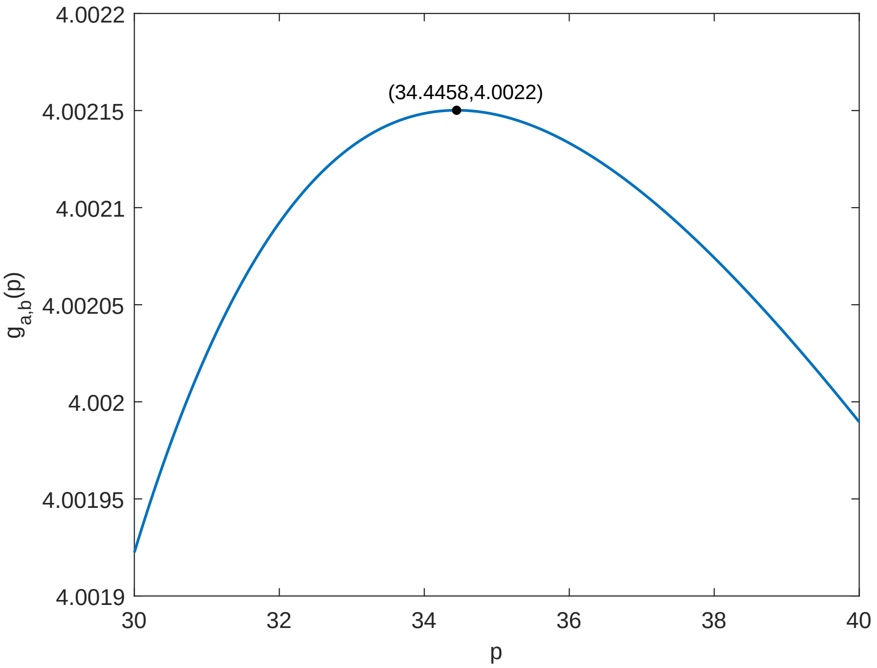


Figure 2: Graph of $g_{4,0.5}(p)$ on the interval $[30, 40]$.

values of p, and thus, we expect that there will be no significant change in the convergence behavior for large values of p.

(b) We remark that these observed behaviors hold under the hypotheses of Theorem 3.1 and Theorem 3.2, which include very strong assumptions on F and on the equilibrium point x^*. Hence, we expect more varying convergence behaviors for other classes of functions F.

(c) Finally, note that for the generalized FB function ϕ^p_{FB} we may define a similar upper bound function $h_{a,b}$ (see [4]) as

$$h_{a,b}(p) = \frac{|\phi^p_{\mathrm{FB}}(a,b)|}{2 - 2^{1/p}}, \quad p > 1.$$

In comparison with the function $g_{a,b}$, the function $h_{a,b}$ defined above can be verified to be always monotonically decreasing. In line with this, it was found in [4] that the neural network approach using ϕ^p_{FB} achieves faster convergence rate when higher values of p are used.

From the above example and remarks, we see the complex dynamics of the role of p, which we will demonstrate via numerical examples in the next section. Hence, it still remains an open question to determine precisely how p affects the convergence behavior of the trajectories of the ODE. Nevertheless, we have provided a theoretical evidence as to why a non-monotonic type of relationship between p and convergence rates are observed in general when using the dynamical systems approach based on $\widetilde{\phi}^p_{\mathrm{NR}}$, $\widetilde{\phi}^p_{\mathrm{S-NR}}$ and $\widetilde{\psi}^p_{\mathrm{S-NR}}$.

4. Numerical Experiments

In this section, we present the results of numerical simulations of the neural networks (9) using the continuous generalization of the NR function and its two symmetrizations proposed in Section 2. We consider several test problems to illustrate the applicability and some advantages of the proposed continuous generalizations. We also present the varying convergence behaviors of trajectories using different values of p for different classes of functions F. Moreover, comparisons of the performance of the three neural networks and the traditional FB and GFB networks will be discussed.

We used the solver *ode23s* in Matlab to simulate the neural network. The simulation is stopped at time t_f if $\|\nabla\Psi_{\mathrm{F}}(x(t_f))\| \leq 10^{-6}$, i.e. when the trajectory is "close" to an equilibrium point. The value of ρ is set to 10^3 for all of the simulations.

The test problems we considered are described in the Appendix, consisting of P_0-functions (NCP1$-$NCP5) and non-P_0-functions (NCP6$-$NCP10). We simulate the three neural networks based on $\widetilde{\phi}^p_{\text{NR}}$, $\widetilde{\phi}^p_{\text{S}-\text{NR}}$ and $\widetilde{\psi}^p_{\text{S}-\text{NR}}$ for different values of p, and the results when $x^0 = (1, 1, \dots 1)^T$ is used as the initial condition are summarized in the Appendix. Using the functions $\widetilde{\phi}^p_{\text{NR}}$, $\widetilde{\phi}^p_{\text{S}-\text{NR}}$ and $\widetilde{\psi}^p_{\text{S}-\text{NR}}$, respectively, to construct the network, **CT1**, **CT2** and **CT3** denote the convergence time of the trajectories, while **Gap1**, **Gap2** and **Gap3** denote the value of $|\langle x(t_f), F(x(t_f))\rangle|$.

We now summarize the findings from the numerical experiments.

Influence of p on convergence time

Using the functions $\widetilde{\phi}^p_{\text{NR}}$ and $\widetilde{\psi}^p_{\text{S}-\text{NR}}$, most of the simulations revealed that lower p value provides faster convergence (see Table 1 to Table 5). For the $\widetilde{\phi}^p_{\text{NR}}-$neural network, this phenomenon can be observed for all NCP test problems except NCP5, while the same conclusion can be made for the $\widetilde{\psi}^p_{\text{S}-\text{NR}}-$neural network when solving all test problems except NCP7. The influence of p when $\widetilde{\phi}^p_{\text{S}-\text{NR}}$ is used for the neural network approach is quite indeterminate. For NCP1-NCP3, higher values of p results to slow convergence of the neural network. In contrast, higher values of p seem desirable for NCP4, NCP5, NCP6 and NCP9 in providing faster convergence rate. For the remaining test problems which involve non-P_0-functions, the pattern is indiscernible.

In summary, we observe that the convergence of the three neural networks has no monotonic type of dependence on the values of p. That is, there is no guarantee whether lower values of p will provide faster convergence or not, even for strongly monotone functions (NCP1-NCP4). For functions F which are not of P_0-type, the observed convergence behaviors are even more varied.

Influence of p on the error term

As mentioned above, the simulations were stopped provided that $\|\nabla\Psi_{\text{F}}(x(t_f))\| \leq 10^{-6}$. Under this stopping criterion, we have observed that for larger values of p, the **Gap** values are not sufficiently close to zero for most of the simulations (see Table 1 to Table 5). In turn, a large error $\|x(t_f) - x^*\|$ is usually obtained for large values of p, and thus, the problem was not successfully solved using the said stopping criterion. For the $\widetilde{\phi}^p_{\text{NR}}-$neural network, this can be observed in all the test problems except for NCP5. For the $\widetilde{\psi}^p_{\text{S}-\text{NR}}$-network, the same phenomenon can be inferred from all test problems except NCP6. Meanwhile, for the neural network based on $\widetilde{\phi}^p_{\text{S}-\text{NR}}$, this also holds true for all test problems except NCP4, NCP5, NCP6 and NCP9 (i.e., the test problems where higher values of p provide faster convergence rate). Nevertheless, the error term for the latter test problems is sufficiently small even when lower values of p is used.

Ill-conditioning

In the numerical experiments, the large values of p we used are 20 and 50. One problem that we usually encountered in simulating the neural networks with large values of p is the ill-conditioning effect which results to failed simulations, which is indicated by "$*$" in Table 1 to Table 5. This problem has been encountered for NCP3, NCP4, NCP5 and NCP8 for neural network based on $\widetilde{\phi}^p_{\mathrm{NR}}$, and NCP3 and NCP5 for the network based on $\widetilde{\psi}^p_{\mathrm{S-NR}}$. On the other hand, the problem is more prominent when we used $\widetilde{\phi}^p_{\mathrm{S-NR}}$ which is evident in the test problems NCP2-NCP6, NCP8 and NCP9. For NCP6 and NCP9, the simulations failed when $p > 2$.

Numerical Comparison of the Neural Networks

Among the three neural networks, we have observed that in all simulations, the one based on $\widetilde{\phi}^p_{\mathrm{S-NR}}$ has the best numerical performance followed by $\widetilde{\phi}^p_{\mathrm{NR}}-$neural network (see Figures 3-12). The simulations reveal that a very slow convergence rate is obtained when using $\widetilde{\psi}^p_{\mathrm{S-NR}}$ compared with the other two NCP functions.

We now compare the $\widetilde{\phi}^p_{\mathrm{S-NR}}-$neural network with the FB and GFB (with $p = 4$) neural networks (see Figures 3-12). In general, we have observed that there is no significant difference in the performance of FB and GFB neural network and the $\widetilde{\phi}^p_{\mathrm{S-NR}}-$neural network when $p = 1.01$ is used, except for NCP8 where the latter has better performance than FB and GFB neural networks. Faster convergence rate, on the other hand, was achieved when $p = 1.1$ was used for the $\widetilde{\phi}^p_{\mathrm{S-NR}}$-neural network, specifically for NCP1, NCP4-NCP7 and NCP10. Finally, note that the FB and GFB networks were significantly outperformed by $\widetilde{\phi}^p_{\mathrm{S-NR}}$-neural network for NCP4, NCP5, NCP6, NCP9 and NCP10 when $p = 7$, $p = 7$ (or 20), $p = 1.5$ (or 2), $p = 1.1$ (or 2) and $p = 2$ were used, respectively, for the latter neural network.

Neural network based on $\widetilde{\phi}^p_{\mathrm{S-NR}}$

With $\widetilde{\phi}^p_{\mathrm{S-NR}}$ as the best NCP function among the functions, we simulate the trajectories of the corresponding neural network using six random initial conditions. From Figures 3-12, we see that the neural network was able to successfully converge to a point in $\mathrm{SOL}(F)$. In particular, note that NCP7 has two solutions, one of which is degenerate. Despite the degeneracy of the solution,

$\widetilde{\phi}^p_{\mathrm{S-NR}}-$neural network was able to converge to this point. Similarly, for the the Matthiesen problem NCP9 which has infinitely many solutions of the form $(k, 0, 0, 0)^T$ where $k \in [0, 3]$, the neural network based on $\widetilde{\phi}^p_{\mathrm{S-NR}}$ was able to converge to a solution.

5. Discussion of Numerical Results

Using the proposed continuous generalization and their symmetrizations, we solved herein several test problems which include P_0- and non-P_0-functions. From the preceding section, we see that the influence of p on the convergence rate of the trajectories is difficult to characterize for all the three neural networks considered. For the neural networks based on $\widetilde{\phi}^p_{\mathrm{NR}}$ and $\widetilde{\psi}^p_{\mathrm{S-NR}}$, most experiments suggest that lower values of p is desirable to obtain faster convergence. However, this cannot be concluded in the general case because there are test instances which involve non-P_0-functions where smaller values of p offer slower convergence rate. The influence of p is even harder to characterize when we use the neural network based on $\widetilde{\phi}^p_{\mathrm{S-NR}}$, where several distinct convergence behaviors are observed when we vary the values of p. Theoretically, we have shown that these varying behaviors are expected because of the non-monotonicity (w.r.t. p) of the error bounds obtained for $\|x(t) - x^*\|$ (see Theorem 3.1, Theorem 3.2, Remark 3.1).

In spite of the non-monotonic dependence on p of the convergence rates of the neural network, we wish to point out that in practical applications, our numerical results suggest that taking $p \in (1, 3)$ is a better choice in general. In most simulations, we have observed that choosing higher values of p result to ill-conditioning problems in the implementations. Moreover, it can be inferred from the numerical experiments that for higher values of p, large values of the error term $\|x(t_f) - x^*\|$ are obtained using the stopping criterion $\|\nabla\Psi(x(t_f))\| \leq \varepsilon = 10^{-6}$. Hence, it is necessary to choose a smaller value of ε in order to obtain more accurate solutions when higher values of p are used. However, the latter might constitute more numerical problems due to higher propensity to encounter ill-conditioning, especially when p becomes larger. In addition, we have also found from numerical experiments that the neural network is often more sensitive to initial conditions when larger values of p are used. We suspect that for larger values of p, an isolated asymptotically stable NCP solution has a smaller region of attraction (i.e. a set S such that the limit $x(t) \to x^*$ (as $t \to \infty$) holds for any initial condition $x(0) = x^0 \in S$).

Finally, the experiments reveal that it is not preferable to use the neural network based on $\widetilde{\psi}^p_{\mathrm{S-NR}}$ because of its unfavorable convergence rate. On the other hand, the neural network based on $\widetilde{\phi}^p_{\rm NR}$ is efficient in solving the test problems, specifically when using values of p close to 1. In particular, its rate of convergence (for small values of p) is almost at par with the neural network based on $\widetilde{\phi}^p_{\rm S-NR}$, which has the best convergence rate among the three families of neural networks considered. Surprisingly, the $\widetilde{\phi}^p_{\rm S-NR}-$neural network can significantly outperform the FB and GFB neural networks for some choices of p.

6. Concluding Remarks

We have successfully constructed a meaningful continuous generalization of the natural residual function, which subsumes the discrete generalization originally proposed in [5]. We also proposed continuous generalization of their symmetrizations, which also subsumes the discrete symmetrization proposed in [3]. The extensions are motivated by the results in our earlier work [2] where preliminary numerical experiments show that lower values of p seem favorable in simulations, which was not theoretically established in [2].

In this paper, we have demonstrated via several test problems that the influence of p on the convergence rate of the neural network is difficult to characterize, even for P_0 or monotone functions. Theoretical evidence for this phenomenon was also provided in this paper by proving some error bound estimates. In any case, as mentioned in the preceding section, lower values of p are more desirable in practical applications due to issues related to ill-conditioning, accuracy of the obtained solution, and sensitivity to initial conditions.

From the numerical experiments, we conclude that the neural network based on $\widetilde{\phi}^p_{\rm NR}$ can be efficiently utilized by choosing values of p close to 1. Its second symmetrization $\psi^p_{\rm S-NR}$, on the other hand, is not recommended to be used for the neural network approach because of its (extremely) slow convergence. Meanwhile, the first symmetrization $\widetilde{\phi}^p_{\rm S-NR}$ offers promising convergence rates despite its complexity. The neural network based on this function is capable of significantly outperforming the well-known FB and GFB neural networks by some suitable choice of p. Hence, it might be worthwhile to revisit some NCP functions-based algorithms to see whether or not this function can be adopted as well, instead of the traditionally used FB or NR functions, to improve numerical performance of the algorithm.

References

[1] B.-H. Ahn, *Iterative methods for linear complementarity problems with upperbounds on primary variables*, Mathematical Programming 26 (1983) 295–315.

[2] J.H. Alcantara, J.-S. Chen, *Neural networks based on three classes of NCP-functions for solving nonlinear complementarity problems*, Neurocomputing 359 (2019) 102–113.

[3] Y.-L. Chang, J.-S. Chen, C.-Y. Yang, *Symmetrization of generalized natural residual function for NCP*, Operations Research Letters 43 (2015) 354–358.

[4] J.-S. Chen, C.-H. Ko, S.-H. Pan, *A neural network based on generalized Fischer-Burmeister function for nonlinear complementarity problems*, Information Sciences 180 (2010) 697–711.

[5] J.-S. Chen, C.-H. Ko, X.-R. Wu, *What is the generalization of natural residual function for NCP?*, Pacific Journal of Optimization 12 (2016) 19–27.

[6] J.-S. Chen, S.-H. Pan, *A family of NCP functions and a descent method for the nonlinear complementarity problem*, Computational Optimization and Applications 40 (2008) 389–404.

[7] C. Dang, Y. Leung, X. Gao, K. Chen, *Neural networks for nonlinear and mixed complementarity problems and their applications*, Neural Networks 17 (2004) 271–283.

[8] F. Facchinei, J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Volumes I and II, Springer-Verlag, New York, 2003.

[9] F. Facchinei, J. Soares, *A new merit function for nonlinear complementarity problems and a related algorithm*, SIAM Journal on Optimization 7 (1997) 225–247.

[10] M. C. Ferris, O. L. Mangasarian, J.-S. Pang, editors, *Complementarity: Applications, Algorithms and Extensions*, Kluwer Academic Publishers, Dordrecht 2001.

[11] M. C. Ferris, J.-S. Pang, *Engineering and economic applications of complementarity problems*, SIAM Review 39 (1997) 669–713.

[12] A. Galantai, *Properties and construction of NCP functions*, Computational Optimization and Applications 52 (2012) 805–824.

[13] B. Huang, C. F. Ma, *The modulus-based Levenberg-Marquardt method for solving linear complementarity problem*, Numerical Mathematics-Theory Methods and Applications 12 (2018) 154–168.

[14] B. Huang, C. F. Ma, *Accelerated modulus-based matrix splitting iteration method for a class of nonlinear complementarity problems*, Computational & Applied Mathematics 37 (2018) 3053–3076.

[15] C.-H. Huang, K.-J. Weng, J.-S. Chen, H.-W. Chu, M.-Y. Li, *On four discrete-type families of NCP Functions*, Journal of Nonlinear and Convex Analysis 20 (2) (2019) 215–228.

[16] C. Kanzow, *Some equation-based methods for the nonlinear complementarity problem*, Optimization Methods and Software 3 (1994) 327–340.

[17] Y. F. Ke, C. F. Ma, *A neural network for the generalized nonlinear complementarity problem over a polyhedral cone*, Journal of the Australian Mathematical Society 99 (2015) 364–379.

[18] Y. F. Ke, C. F. Ma, H. Zhang, *The modulus-based matrix splitting iteration methods for second-order cone linear complementarity problems*, Numerical Algorithms 79 (2018) 1283-1303.

[19] M. Kojima, S. Shindo, *Extensions of Newton and quasi-Newton methods to systems of PC^1 equations*, Journal of Operations Research Society of Japan 29 (1986) 352–374.

[20] L.-Z. Liao, H.-D. Qi, *A neural network for the linear complementarity problem*, Mathematical and Computer Modelling 29 (1999) 9–18.

[21] C. Kanzow, *Some noninterior continuation methods for linear complementarity problems*, SIAM Journal on Matrix Analysis and Applications 17 (1996) 851–868.

[22] R. K. Miller, A. N. Michel, *Ordinary Differential Equations*, Academic Press, 1982.

[23] M. A. G. Ruggiero, J. M. Martinez, S. A. Santos, *Solving nonsmooth equations by means of quasi-Newton methods with globalization*, In: Recent Advances in Nonsmooth Optimization, pp. 121–140. World Scientific, Singapore (1995)

[24] E. Spedicato, Z. Huang, *Numerical experience with Newton-like methods for nonlinear algebraic systems*, Computing 58 (1997) 69–89.

[25] L. T. Watson, *Solving the nonlinear complementarity problem by a homotopy method*, SIAM Journal on Control and Optimization 17 (1979) 36–46.

[26] Y. Xia, H. Leung, J. Wang, *A projection neural network and its application to constrained optimization problems*, IEEE Transactions on Circuits and Systems-I 49 (2002) 447–458.

[27] H. Yu, D. Pu, *Smoothing Levenberg-Marquardt method for general nonlinear complementarity problems under local error bound*, Applied Mathematical Modelling 35 (2011) 1337–1348.

Appendix: Test Problems and Summary of Results

We collect herein some standard test problems for NCP.

Category I: NCP(F) where F is a P_0-function

(NCP1,[1]) Let $F(x) = Ax + b$ where $A = \begin{pmatrix} 4 & 1 & 0 & \cdots & 0 \\ -2 & 4 & 1 & \cdots & 0 \\ 0 & -2 & 4 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 4 \end{pmatrix}$ is a tridiagonal P-matrix and $b = (-1, \cdots, -1)^T$. For the simulations, we take $n = 5$.

(NCP2,[7]) Let $F_i(x) = -x_{i-1} + 2x_i - x_{i+1} + b_i(x) + c_i$ for $i = 1, \ldots, n$, where $x_0 = x_{n+1} = 0$, and let $b_i(x) = \arctan(x_i)$ and $c_i = i - \frac{n}{2}$. For the simulations, we take $n = 5$.

(NCP3,[25]) Let F be given by

$$F(x) = 2\exp\left(\sum_{i=1}^{5}(x_i - i + 2)^2\right)\begin{pmatrix} x_1 + 1 \\ x_2 \\ x_3 - 1 \\ x_4 - 2 \\ x_5 - 3 \end{pmatrix}.$$

(NCP4,[27]) Let $F : \mathbb{R}^3 \to \mathbb{R}^3$ the strictly monotone function

$$F(x) = \begin{pmatrix} x_1 - 2 \\ x_2 - x_3 + x_2^3 + 3 \\ x_2 + x_3 + 2x_3^3 - 3 \end{pmatrix}.$$

The unique solution of NCP(F) is $x^* = (2, 0, 1)^T$.

(NCP5,[20]) We consider the linear complementarity problem with $F(x) = Ax + b$, where $A = \begin{pmatrix} 1 & -4 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$ and $b = (-5, -5, 1, 1)^T$. This problem arises from quadratic programming [20] which has a unique solution $x^* = (1, 1, 8, 4)^T$.

Category II: NCP(F) where F is a non-P_0-function

(NCP6,[19]) Consider the Kojima-Shindo problem where F is defined as

$$F(x) = \begin{pmatrix} 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_1 + x_2^2 + 3x_3 + 2x_4 - 2 \\ 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 3x_4 - 1 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{pmatrix}.$$

The unique (non-degenerate) solution of NCP(F) is $x^* = (\sqrt{6}/2, 0, 0, 1/2)^T$.

(NCP7,[19]) We consider a modification of the Kojima-Shindo problem

$$F(x) = \begin{pmatrix} 3x_1^2 + 2x_1x_2 + 2x_2^2 + x_3 + 3x_4 - 6 \\ 2x_1^2 + x_1 + x_2^2 + 10x_3 + 2x_4 - 2 \\ 3x_1^2 + x_1x_2 + 2x_2^2 + 2x_3 + 9x_4 - 9 \\ x_1^2 + 3x_2^2 + 2x_3 + 3x_4 - 3 \end{pmatrix}.$$

NCP(F) has two solutions: $x^* = (\sqrt{6}/2, 0, 0, 1/2)^T$ and $x^* = (1, 0, 3, 0^T$, which are degenerate and non-degenerate, respectively.

(NCP8,[26]) Let $F : \mathbb{R}^5 \to \mathbb{R}^5$ be given by

$$F(x) = \begin{pmatrix} x_1 + x_2x_3x_4x_5/50 \\ x_2 + x_1x_3x_4x_5/50 - 3 \\ x_3 + x_1x_2x_4x_5/50 - 1 \\ x_4 + x_1x_2x_3x_5/50 + 1/2 \\ x_5 + x_1x_2x_3x_4/50 \end{pmatrix}.$$

The solution of NCP(F) is $x^* = (0, 3, 1, 0, 0)^T$.

(NCP9,[16]) We consider the modified Matthiesen problem, where F is given by

$$F(x) = \begin{pmatrix} -x_2 + x_3 + x_4 \\ x_1 - (4.5x_3 + 2.7x_4)/(x_2 + 1) \\ 5 - x_1 - (0.5x_3 + 0.3x_4)/(x_3 + 1) \\ 3 - x_1 \end{pmatrix},$$

which has infinitely many solutions $x^* = (k, 0, 0, 0)^T$, where $k \in [0, 3]$.

(NCP10,[23, 24]) We follow the construction of F used in [23]. Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a continuously differentiable function, and let $x^* = (0, 1, 0, 1, \dots)^T \in \mathbb{R}^n$. Define $F : \mathbb{R}^n \to \mathbb{R}^n$ by

$$F_i(x) = \begin{cases} f_i(x) - f_i(x^*) + 1 & \text{if } i \text{ is odd} \\ f_i(x) - f_i(x^*) & \text{otherwise} \end{cases}.$$

It is clear that x^* is a nondegenerate solution of NCP(F). For this example, we take f from [24] given by

$$f_i(x) = n - i\sum_{j=1}^{n} \cos(x_j) + i(1 - \cos(x_i)) - \sin(x_i).$$

For the simulations, we take $n = 5$.

Table 1: Numerical results for NCP1 and NCP2 using the neural networks based on $\widetilde{\phi}^p_{\mathrm{NR}}, \widetilde{\phi}^p_{\mathrm{S-NR}}$ and $\widetilde{\psi}^p_{\mathrm{S-NR}}$ for different values of p.

p	NCP1						NCP2					
	CT1	Gap1	CT2	Gap2	CT3	Gap3	CT1	Gap1	CT2	Gap2	CT3	Gap3
1.0	3.0E+0	1.4E-7	3.0E+0	7.6E-8	1.4E+3	6.1E-3	1.9E+1	6.7E-7	1.7E+1	3.4E-7	6.7E+3	3.1E-2
1.1	3.2E+0	1.5E-7	2.7E+0	9.5E-8	1.8E+3	8.4E-3	5.1E+1	1.6E-5	2.0E+1	3.1E-7	7.9E+3	3.7E-2
1.5	5.1E+0	1.9E-7	3.1E+0	1.8E-7	3.9E+3	2.5E-2	1.1E+3	2.4E-3	4.5E+1	6.0E-7	1.4E+4	7.5E-2
1.9	1.0E+1	4.1E-7	6.5E+0	3.6E-7	6.9E+3	5.4E-2	4.7E+3	1.9E-2	1.2E+2	1.9E-6	2.0E+4	1.3E-1
2.0	1.2E+1	4.2E-7	8.0E+0	4.3E-7	7.8E+3	6.4E-2	6.0E+3	2.7E-2	1.6E+2	2.6E-6	2.1E+4	1.5E-1
2.1	1.5E+1	6.1E-7	1.0E+1	5.4E-7	8.7E+3	7.5E-2	7.3E+3	3.6E-2	2.1E+2	3.7E-6	2.2E+4	1.7E-1
2.5	3.5E+1	1.5E-6	2.5E+1	1.5E-6	1.3E+4	1.3E-1	1.3E+4	9.1E-2	6.1E+2	1.5E-5	2.6E+4	2.5E-1
2.9	8.9E+1	4.0E-6	6.7E+1	4.0E-6	1.7E+4	2.0E-1	1.9E+4	1.7E-1	1.6E+3	6.2E-5	2.9E+4	3.4E-1
3.0	1.1E+2	5.2E-6	8.6E+1	5.2E-6	1.7E+4	2.2E-1	2.0E+4	1.9E-1	1.9E+3	8.8E-5	3.0E+4	3.7E-1
3.5	3.9E+2	2.0E-5	3.1E+2	2.0E-5	2.1E+4	3.4E-1	2.7E+4	3.3E-1	4.2E+3	4.2E-4	3.3E+4	5.0E-1
4.0	1.4E+3	8.0E-5	1.1E+3	8.0E-5	2.3E+4	4.8E-1	3.2E+4	4.8E-1	6.1E+3	1.3E-3	3.4E+4	6.3E-1
4.5	5.1E+3	3.4E-4	4.2E+3	3.4E-4	2.3E+4	6.3E-1	3.5E+4	6.5E-1	7.4E+3	2.9E-3	3.5E+4	7.7E-1
5.0	1.8E+4	1.7E-3	1.5E+4	1.7E-3	2.4E+4	7.8E-1	3.8E+4	8.3E-1	8.3E+3	5.1E-3	3.6E+4	9.1E-1
5.5	2.8E+4	2.8E-2	1.7E+4	2.8E-2	2.4E+4	9.2E-1	4.0E+4	1.0E+0	9.0E+3	7.9E-3	3.6E+4	1.0E+0
6.0	2.8E+4	6.6E-2	4.6E+3	5.5E-2	2.3E+4	1.1E+0	4.1E+4	1.2E+0	9.4E+3	1.1E-2	3.6E+4	1.2E+0
6.5	2.8E+4	1.1E-1	1.2E+4	5.7E-2	2.3E+4	1.2E+0	4.2E+4	1.4E+0	9.8E+3	1.5E-2	3.5E+4	1.3E+0
7.0	2.9E+4	1.6E-1	2.9E+4	6.1E-2	2.2E+4	1.3E+0	4.3E+4	1.5E+0	1.0E+4	1.9E-2	3.5E+4	1.4E+0
20.0	3.7E+4	3.3E+0	1.5E+4	1.4E+0	1.2E+4	3.1E+0	3.1E+4	4.4E+0	*	*	1.5E+4	2.9E+0
50.0	1.9E+4	6.9E+0	5.6E+3	2.2E+0	5.5E+3	4.1E+0	1.9E+4	6.3E+0	*	*	6.3E+3	3.5E+0

Table 2: Numerical results for NCP3 and NCP4 using the neural networks based on $\widetilde{\phi}^p_{\rm NR}, \widetilde{\phi}^p_{\rm S-NR}$ and $\widetilde{\psi}^p_{\rm S-NR}$ for different values of p.

p	NCP3						NCP4					
	CT1	Gap1	CT2	Gap2	CT3	Gap3	CT1	Gap1	CT2	Gap2	CT3	Gap3
1.01	1.5E+1	5.2E-6	1.4E+1	1.2E-6	5.1E+3	4.5E-2	1.6E+1	2.3E-6	1.4E+1	3.3E-7	5.1E+3	1.9E-3
1.1	4.3E+1	3.8E-5	2.7E+1	1.9E-10	5.8E+3	4.9E-2	4.3E+1	1.8E-5	1.1E+1	5.2E-8	5.6E+3	2.5E-3
1.5	9.7E+2	3.7E-3	2.4E+2	3.3E-7	8.7E+3	7.6E-2	8.3E+2	1.6E-3	3.5E+0	2.0E-7	7.6E+3	7.1E-3
1.9	4.0E+3	2.7E-2	7.2E+2	9.1E-6	1.1E+4	1.1E-1	3.2E+3	1.2E-2	1.4E+0	1.2E-7	9.3E+3	1.4E-2
2	5.0E+3	3.7E-2	8.6E+2	1.6E-5	1.1E+4	1.2E-1	4.0E+3	1.6E-2	1.1E+0	9.5E-8	9.7E+3	1.7E-2
2.1	6.1E+3	5.0E-2	1.3E+1	6.0E-1	1.1E+4	1.3E-1	4.9E+3	2.1E-2	9.0E-1	7.0E-8	1.0E+4	1.9E-2
2.5	1.1E+4	1.1E-1	8.2E+0	6.9E-1	1.2E+4	1.7E-1	8.5E+3	5.1E-2	4.0E-1	2.5E-8	1.1E+4	1.8E-2
2.9	1.6E+4	1.9E-1	6.9E+0	7.4E-1	1.3E+4	2.1E-1	1.2E+4	9.3E-2	1.9E-1	6.2E-9	1.2E+4	1.7E-2
3	1.7E+4	2.1E-1	6.8E+0	7.5E-1	1.3E+4	2.2E-1	1.3E+4	1.0E-1	1.6E-1	3.6E-9	1.3E+4	2.4E-2
3.5	2.2E+4	3.1E-1	6.6E+0	8.0E-1	1.3E+4	2.7E-1	1.6E+4	1.7E-1	7.0E-2	1.6E-10	1.4E+4	5.3E-2
4	2.6E+4	3.8E-1	7.4E+0	8.2E-1	1.3E+4	3.2E-1	1.9E+4	2.4E-1	4.0E-2	1.1E-10	1.4E+4	8.0E-2
4.5	2.8E+4	4.3E-1	8.9E+0	8.4E-1	1.3E+4	3.6E-1	2.1E+4	3.2E-1	3.0E-2	3.7E-10	1.5E+4	1.1E-1
5	3.0E+4	4.6E-1	1.1E+1	8.6E-1	1.3E+4	4.1E-1	2.3E+4	4.0E-1	3.0E-2	2.5E-13	1.5E+4	1.3E-1
5.5	3.1E+4	4.8E-1	1.4E+1	8.7E-1	1.3E+4	4.5E-1	2.4E+4	4.8E-1	2.0E-2	5.3E-10	1.5E+4	1.6E-1
6	3.1E+4	4.8E-1	1.8E+1	8.8E-1	1.2E+4	4.9E-1	2.5E+4	5.6E-1	2.0E-2	4.5E-13	1.5E+4	1.9E-1
6.5	2.9E+4	4.7E-1	2.4E+1	8.9E-1	1.0E+4	5.4E-1	2.5E+4	6.3E-1	2.0E-2	1.3E-15	1.5E+4	2.1E-1
7	2.6E+4	4.8E-1	3.1E+1	9.0E-1	8.2E+3	6.0E-1	2.5E+4	7.1E-1	2.0E-2	1.5E-15	1.5E+4	2.4E-1
20	*	*	*	*	*	*	4.6E+4	1.9E+0	9.1E+4	4.5E-16	1.1E+4	6.8E-1
50	*	*	*	*	*	*	*	*	*	*	5.2E+3	9.7E-1

Table 3: Numerical results for NCP5 and NCP6 using the neural networks based on $\widetilde{\phi}^p_{\mathrm{NR}}, \widetilde{\phi}^p_{\mathrm{S-NR}}$ and $\widetilde{\psi}^p_{\mathrm{S-NR}}$ for different values of p.

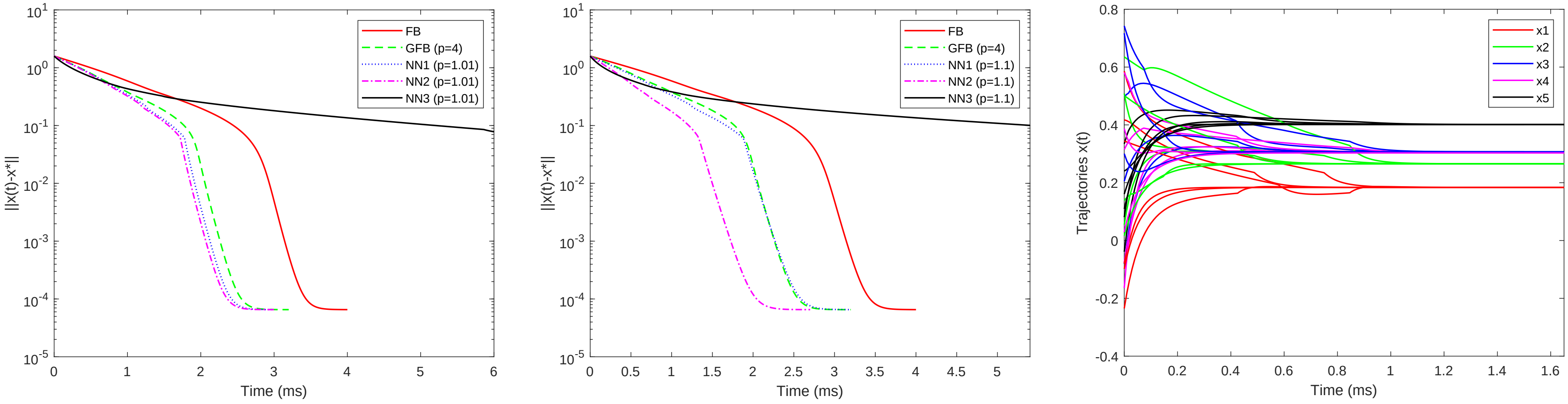
p	NCP5						NCP6					
	CT1	Gap1	CT2	Gap2	CT3	Gap3	CT1	Gap1	CT2	Gap2	CT3	Gap3
1.01	2.4E+2	7.9E-6	2.4E+2	7.9E-6	3.6E+4	1.6E-2	2.8E+1	4.0E-6	2.4E+1	3.0E-6	6.3E+3	5.9E-2
1.1	1.6E+2	5.6E-6	1.6E+2	5.6E-6	3.7E+4	1.6E-2	7.6E+1	2.6E-5	1.4E+1	1.5E-6	6.5E+3	6.9E-2
1.5	3.7E+1	1.1E-6	3.7E+1	1.1E-6	4.1E+4	1.9E-2	1.1E+3	5.6E-3	5.9E+0	4.6E-8	7.5E+3	8.5E-2
1.9	1.1E+1	1.3E-7	1.1E+1	1.3E-7	4.3E+4	3.8E-2	4.4E+3	4.3E-2	4.6E+0	1.6E-7	9.7E+3	7.1E-2
2	8.7E+0	4.7E-8	8.7E+0	4.7E-8	4.4E+4	5.3E-2	5.6E+3	6.0E-2	4.7E+0	1.7E-7	1.0E+4	6.2E-2
2.1	7.0E+0	1.7E-9	7.0E+0	1.7E-9	4.4E+4	6.7E-2	6.8E+3	8.0E-2	*	*	1.1E+4	5.4E-2
2.5	3.4E+0	7.6E-8	3.4E+0	7.6E-8	4.7E+4	1.1E-1	1.2E+4	1.9E-1	*	*	1.2E+4	2.8E-2
2.9	2.1E+0	1.1E-7	2.1E+0	1.1E-7	4.9E+4	1.4E-1	1.8E+4	3.5E-1	*	*	1.4E+4	2.1E-2
3	2.0E+0	5.5E-8	2.0E+0	5.5E-8	4.9E+4	1.5E-1	1.9E+4	4.0E-1	*	*	1.4E+4	2.2E-2
3.5	1.4E+0	3.7E-8	1.4E+0	3.7E-8	5.1E+4	2.0E-1	2.5E+4	6.5E-1	*	*	1.6E+4	3.7E-2
4	1.0E+0	6.0E-8	1.0E+0	6.0E-8	5.2E+4	2.4E-1	3.0E+4	9.2E-1	*	*	1.7E+4	6.2E-2
4.5	8.0E-1	3.1E-8	8.0E-1	3.1E-8	5.2E+4	2.9E-1	3.4E+4	1.2E+0	*	*	1.8E+4	8.8E-2
5	9.0E-1	4.1E-11	9.0E-1	4.1E-11	5.2E+4	3.3E-1	3.8E+4	1.5E+0	*	*	1.8E+4	1.1E-1
5.5	7.0E-1	1.2E-10	7.0E-1	1.2E-10	5.2E+4	3.7E-1	4.0E+4	1.7E+0	*	*	1.9E+4	1.3E-1
6	5.0E+2	8.9E-16	1.0E+3	8.9E-16	5.1E+4	4.0E-1	3.9E+4	2.0E+0	*	*	1.9E+4	1.4E-1
6.5	5.0E-1	8.5E-11	5.0E-1	8.5E-11	5.1E+4	4.3E-1	3.3E+4	2.2E+0	*	*	1.9E+4	1.6E-1
7	4.0E-1	3.3E-10	4.0E-1	3.3E-10	5.0E+4	4.7E-1	4.6E+4	2.4E+0	*	*	1.9E+4	1.7E-1
20	1.0E-1	1.5E-12	1.0E-1	1.5E-12	3.1E+4	8.4E-1	3.6E+4	6.8E+0	*	*	1.4E+4	7.6E-1
50	*	*	*	*	*	*	1.7E+4	1.5E+1	*	*	7.2E+3	1.5E+0

Table 4: Numerical results for NCP7 and NCP8 using the neural networks based on $\widetilde{\phi}^p_{\rm NR}, \widetilde{\phi}^p_{\rm S-NR}$ and $\widetilde{\psi}^p_{\rm S-NR}$ for different values of p.

p	NCP7						NCP8					
	CT1	Gap1	CT2	Gap2	CT3	Gap3	CT1	Gap1	CT2	Gap2	CT3	Gap3
1.01	5.0E+2	5.4E-5	4.4E+2	3.7E-5	5.8E+4	2.7E-1	1.6E+1	8.0E-8	1.6E+1	3.3E-7	6.8E+3	9.3E-3
1.1	1.2E+3	3.7E-4	3.1E+2	1.2E-5	6.0E+4	2.6E-1	4.6E+1	2.9E-6	4.4E+1	9.5E-11	8.6E+3	8.9E-3
1.5	1.4E+4	2.9E-2	4.6E+4	1.4E-1	6.7E+4	2.5E-1	1.1E+3	3.1E-4	9.7E+2	9.4E-7	1.8E+4	5.0E-3
1.9	4.2E+4	1.9E-1	5.4E+2	1.9E-6	1.3E+4	2.8E-2	4.5E+3	2.4E-3	4.0E+3	5.1E-5	2.6E+4	6.6E-3
2	5.0E+4	2.5E-1	6.9E+2	4.1E-6	1.4E+4	2.8E-2	5.7E+3	3.4E-3	5.0E+3	1.0E-4	2.7E+4	1.1E-2
2.1	5.9E+4	3.3E-1	8.6E+2	7.9E-6	1.5E+4	2.8E-2	7.0E+3	4.7E-3	6.1E+3	1.8E-4	2.9E+4	1.6E-2
2.5	9.1E+4	7.4E-1	1.5E+1	3.1E-2	2.1E+4	1.8E-2	1.3E+4	1.2E-2	1.1E+4	1.1E-3	3.3E+4	4.3E-2
2.9	9.1E+4	1.3E+0	6.4E+1	6.7E-2	2.8E+4	3.2E-2	1.8E+4	2.5E-2	1.6E+4	3.5E-3	3.6E+4	7.9E-2
3	9.1E+4	1.4E+0	8.1E+1	6.7E-2	3.0E+4	4.7E-2	2.0E+4	2.9E-2	1.7E+4	4.5E-3	3.7E+4	9.0E-2
3.5	9.1E+4	2.1E+0	9.1E+3	1.4E-1	4.1E+4	1.5E-1	2.6E+4	5.4E-2	2.2E+4	1.2E-2	3.8E+4	1.5E-1
4	9.1E+4	2.6E+0	9.1E+3	1.4E-1	3.9E+4	3.1E-1	3.0E+4	8.6E-2	2.6E+4	2.4E-2	3.9E+4	2.1E-1
4.5	9.1E+4	2.9E+0	9.1E+3	1.4E-1	3.4E+4	2.3E-1	4.6E+4	1.2E-1	2.9E+4	4.0E-2	3.9E+4	2.7E-1
5	9.1E+4	3.0E+0	3.6E+3	8.2E-2	1.7E+4	2.3E-1	4.6E+4	1.6E-1	3.1E+4	6.0E-2	3.8E+4	3.4E-1
5.5	9.1E+4	3.0E+0	5.4E+3	8.9E-2	1.9E+4	2.9E-1	4.6E+4	2.2E-1	1.0E-1	2.0E+1	3.7E+4	4.0E-1
6	6.5E+4	2.9E+0	9.1E+3	1.4E-1	1.9E+4	3.6E-1	4.6E+4	2.7E-1	1.0E-1	2.1E+1	3.6E+4	4.7E-1
6.5	9.1E+4	2.4E+0	9.1E+3	1.4E-1	1.8E+4	4.3E-1	4.5E+4	3.3E-1	1.0E-1	2.2E+1	3.5E+4	5.3E-1
7	9.1E+4	1.2E+0	9.1E+3	1.4E-1	1.8E+4	4.9E-1	4.6E+4	3.9E-1	1.0E-1	2.2E+1	3.4E+4	5.8E-1
20	3.6E+4	7.1E+0	9.1E+3	6.6E-1	9.9E+3	1.3E+0	*	*	*	*	1.8E+4	1.4E+0
50	1.7E+4	1.8E+1	4.3E+3	9.3E-1	4.7E+3	1.8E+0	*	*	*	*	8.1E+3	1.8E+0

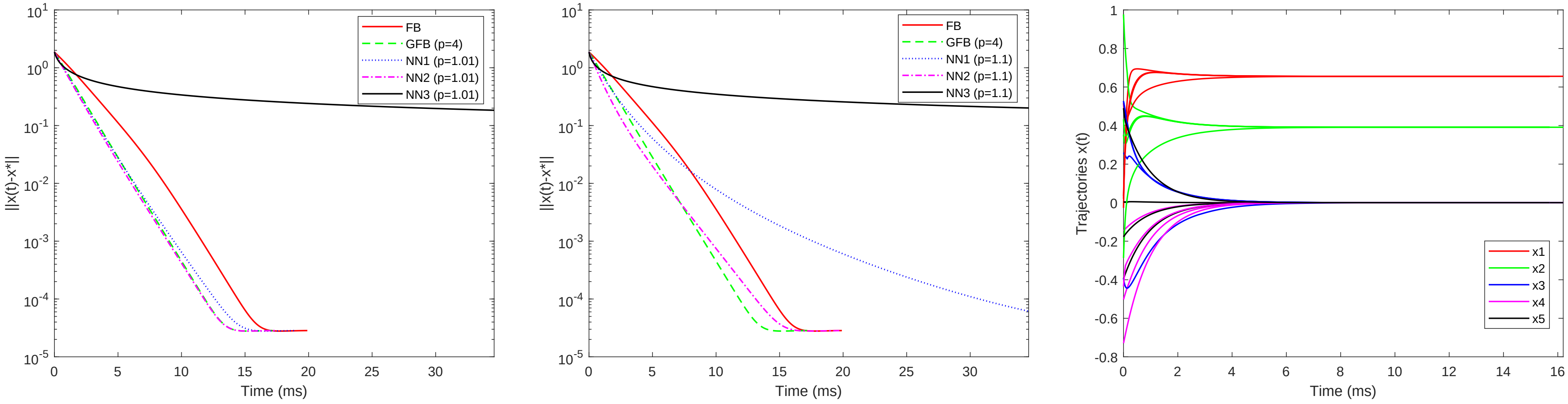
Table 5: Numerical results for NCP9 and NCP10 using the neural networks based on $\widetilde{\phi}^p_{\mathrm{NR}}, \widetilde{\phi}^p_{\mathrm{S-NR}}$ and $\widetilde{\psi}^p_{\mathrm{S-NR}}$ for different values of p.

p	NCP9						NCP10					
	CT1	Gap1	CT2	Gap2	CT3	Gap3	CT1	Gap1	CT2	Gap2	CT3	Gap3
1.01	3.5E+2	4.1E-5	3.0E+2	3.1E-5	4.6E+4	9.7E-2	1.6E+1	2.2E-6	1.4E+1	1.6E-6	6.0E+3	1.9E-2
1.1	9.0E+2	2.3E-4	1.2E+2	2.7E-6	4.8E+4	1.1E-1	4.6E+1	1.7E-5	1.2E+1	1.4E-6	6.8E+3	2.3E-2
1.5	1.2E+4	1.3E-2	6.0E+0	5.4E-7	1.0E+4	9.8E-2	1.1E+3	1.9E-3	9.2E+0	3.7E-7	1.1E+4	5.0E-2
1.9	3.6E+4	7.3E-2	2.9E+0	2.6E-7	1.3E+4	1.5E-1	4.5E+3	1.4E-2	9.0E+0	2.4E-7	1.6E+4	8.6E-2
2	4.2E+4	9.7E-2	2.7E+0	1.6E-7	1.4E+4	1.6E-1	5.7E+3	2.0E-2	9.5E+0	2.1E-7	1.7E+4	9.7E-2
2.1	4.9E+4	1.2E-1	2.8E+0	7.9E-8	1.5E+4	1.8E-1	7.0E+3	2.6E-2	1.0E+1	1.7E-7	1.8E+4	1.1E-1
2.5	7.1E+4	2.6E-1	*	*	1.9E+4	2.4E-1	1.3E+4	6.4E-2	5.2E+1	3.3E-1	2.3E+4	1.5E-1
2.9	8.6E+4	4.2E-1	*	*	2.3E+4	3.0E-1	1.8E+4	1.2E-1	2.2E+1	3.7E-1	2.8E+4	1.6E-1
3	8.9E+4	4.6E-1	*	*	2.4E+4	3.1E-1	2.0E+4	1.3E-1	2.1E+1	3.7E-1	2.9E+4	1.6E-1
3.5	9.1E+4	6.6E-1	*	*	3.0E+4	3.9E-1	2.6E+4	2.1E-1	2.1E+1	3.9E-1	3.6E+4	1.7E-1
4	9.1E+4	8.5E-1	*	*	3.6E+4	4.6E-1	3.1E+4	2.9E-1	4.6E+4	1.8E-1	4.4E+4	2.0E-1
4.5	9.1E+4	1.0E+0	*	*	4.2E+4	5.3E-1	3.4E+4	3.7E-1	4.6E+4	1.8E-1	4.6E+4	2.4E-1
5	9.1E+4	1.2E+0	*	*	4.6E+4	5.9E-1	3.7E+4	4.6E-1	4.8E+2	2.3E-1	4.6E+4	2.8E-1
5.5	9.1E+4	1.3E+0	*	*	5.0E+4	6.4E-1	3.9E+4	5.4E-1	3.9E+2	2.4E-1	4.6E+4	3.3E-1
6	9.1E+4	1.4E+0	*	*	5.2E+4	6.9E-1	4.1E+4	6.2E-1	5.6E+2	2.4E-1	4.6E+4	3.8E-1
6.5	8.2E+4	1.5E+0	*	*	5.3E+4	7.4E-1	4.2E+4	6.9E-1	9.2E+2	2.4E-1	4.6E+4	4.1E-1
7	6.4E+4	1.6E+0	*	*	5.3E+4	7.7E-1	4.3E+4	7.7E-1	1.4E+3	2.4E-1	3.6E+4	4.4E-1
20	2.5E+1	1.6E+0	*	*	2.5E+4	1.2E+0	2.6E+4	2.4E+0	4.3E+4	3.9E-1	1.1E+4	1.8E+0
50	2.5E+4	1.5E+0	*	*	5.1E+3	1.3E+0	1.9E+4	5.5E+0	2.1E+4	6.5E-1	4.6E+3	2.8E+0



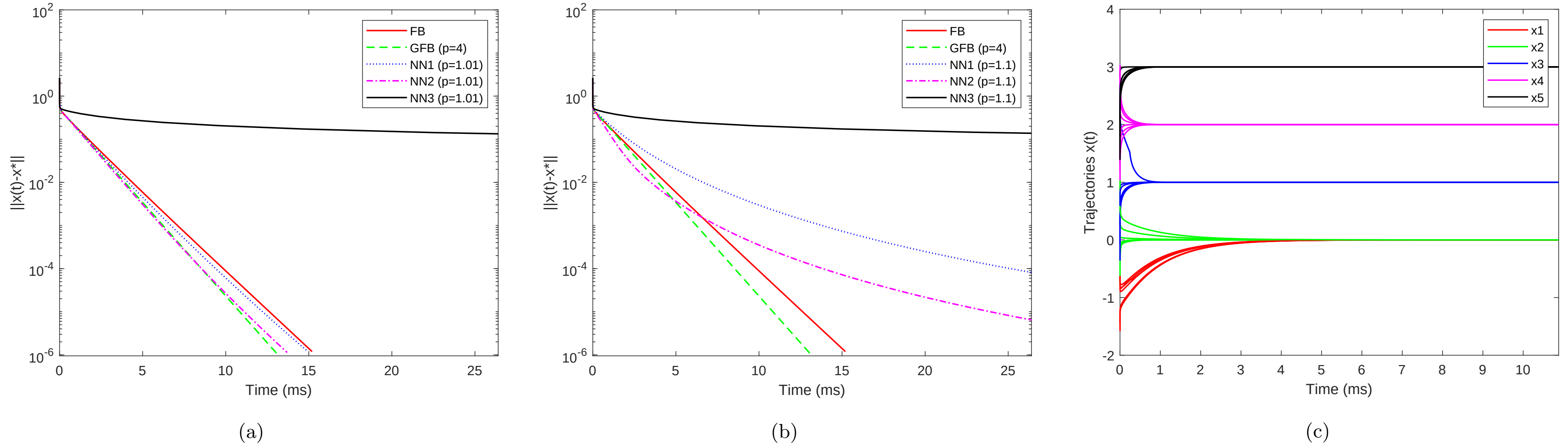
(a) (b) (c)

Figure 3: Comparison of neural networks based on NR and FB functions, and the convergence of the neural network based on $\widetilde{\phi}^{p}_{\mathrm{S-NR}}$ ($p = 1.01$) to the approximate solution $x^* = (0.1837, 0.2652, 0.3068, 0.3030, 0.4015)^T$ for NCP1.



(a) (b) (c)

Figure 4: Comparison of neural networks based on NR and FB functions, and the convergence of the neural network based on $\widetilde{\phi}^{p}_{\mathrm{S-NR}}$ ($p = 1.01$) to the approximate solution $x^* = (0.6555, 0.3913, 0, 0, 0)^T$ for NCP2.

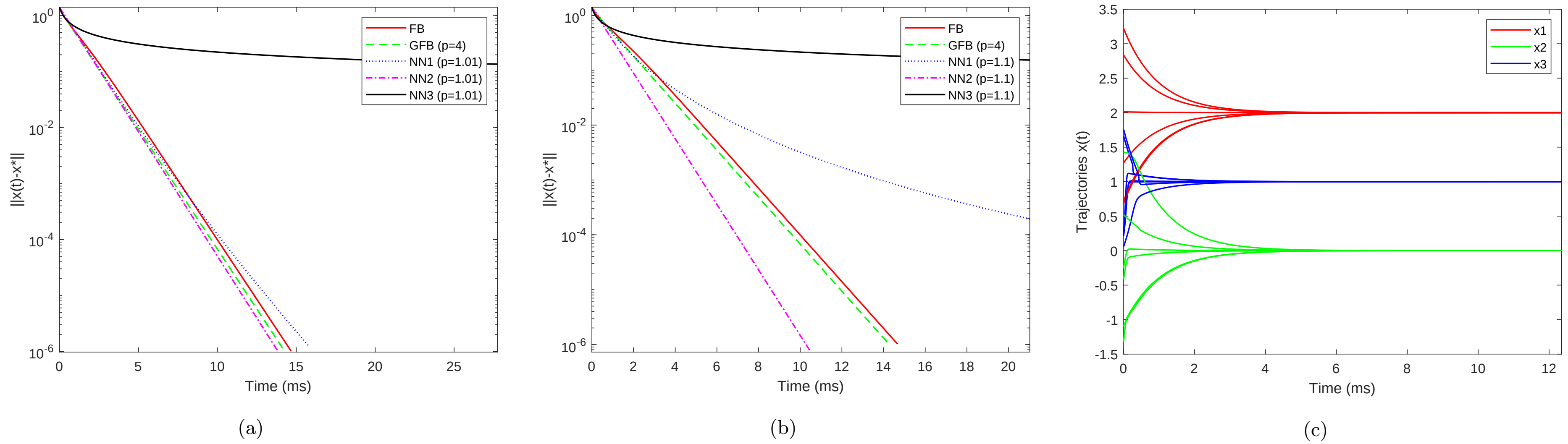


(a)

(b)

(c)

Figure 5: Comparison of neural networks based on NR and FB functions, and the convergence of the neural network based on $\widetilde{\phi}^p_{\rm S-NR}$ $(p = 1.01)$ to the solution $x^* = (0, 0, 1, 2, 3)^T$ for NCP3.

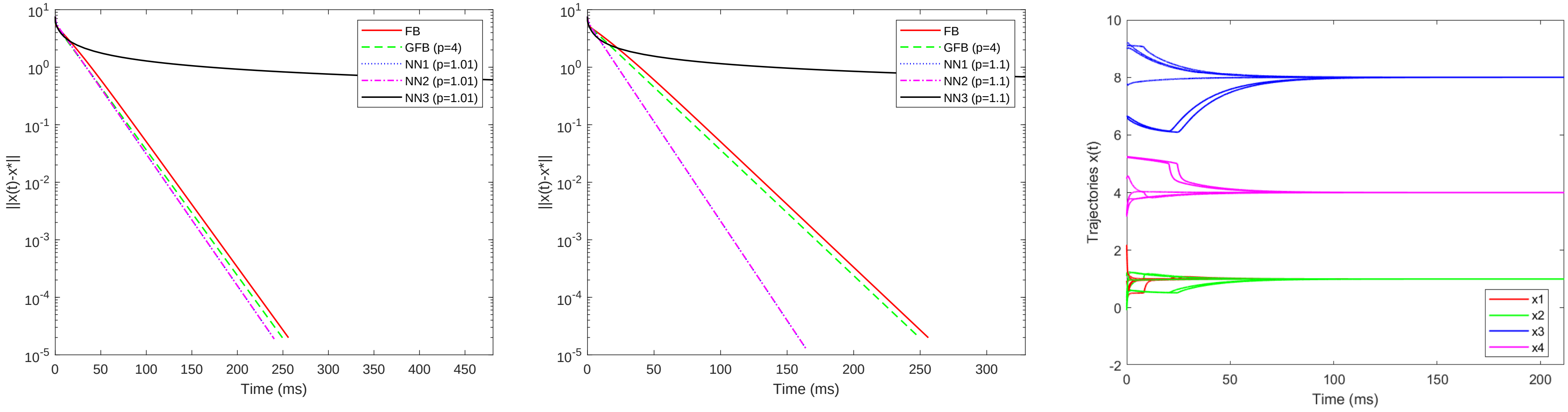


(a)

(b)

(c)

Figure 6: Comparison of neural networks based on NR and FB functions, and the convergence of the neural network based on $\widetilde{\phi}^p_{\rm S-NR}$ $(p = 1.01)$ to the solution $x^* = (2, 0, 1)^T$ for NCP4.

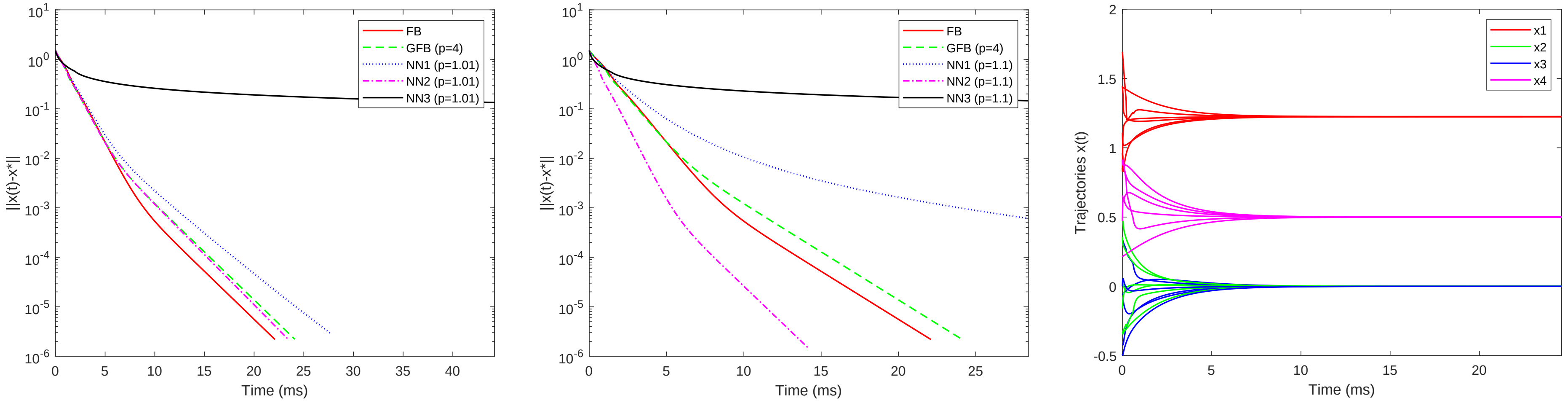


(a)

(b)

(c)

Figure 7: Comparison of neural networks based on NR and FB functions, and the convergence of the neural network based on $\widetilde{\phi}^p_{\mathrm{S-NR}}$ ($p = 1.01$) to the solution $x^* = (1,1,8,4)^T$ for NCP5.

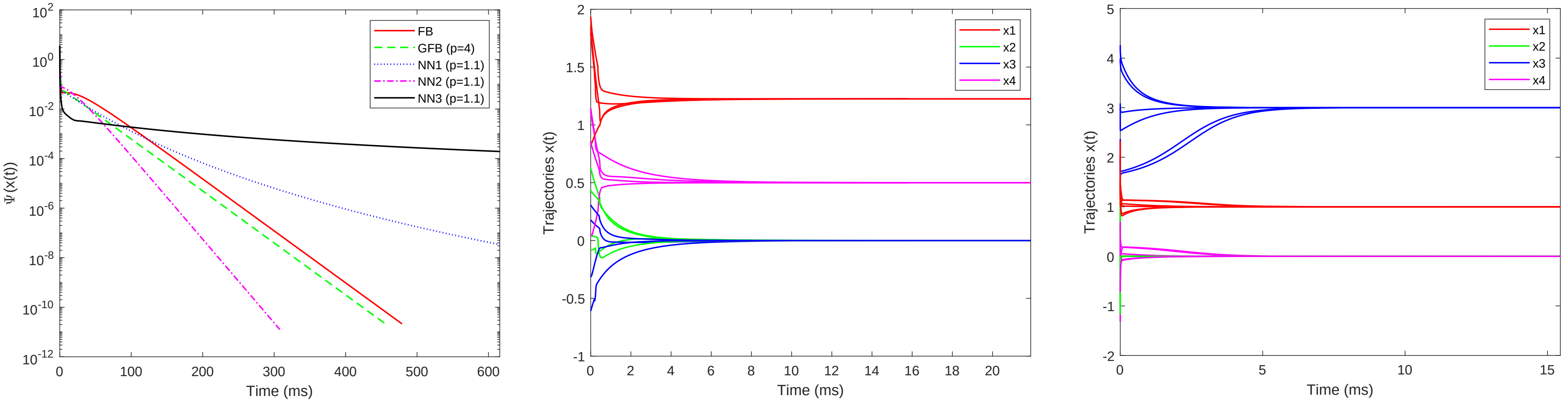


(a)

(b)

(c)

Figure 8: Comparison of neural networks based on NR and FB functions, and the convergence of the neural network based on $\widetilde{\phi}^p_{\mathrm{S-NR}}$ ($p = 1.01$) to the solution $x^* = (\sqrt{6}/2, 0, 0, 0.5)^T$ for NCP6.

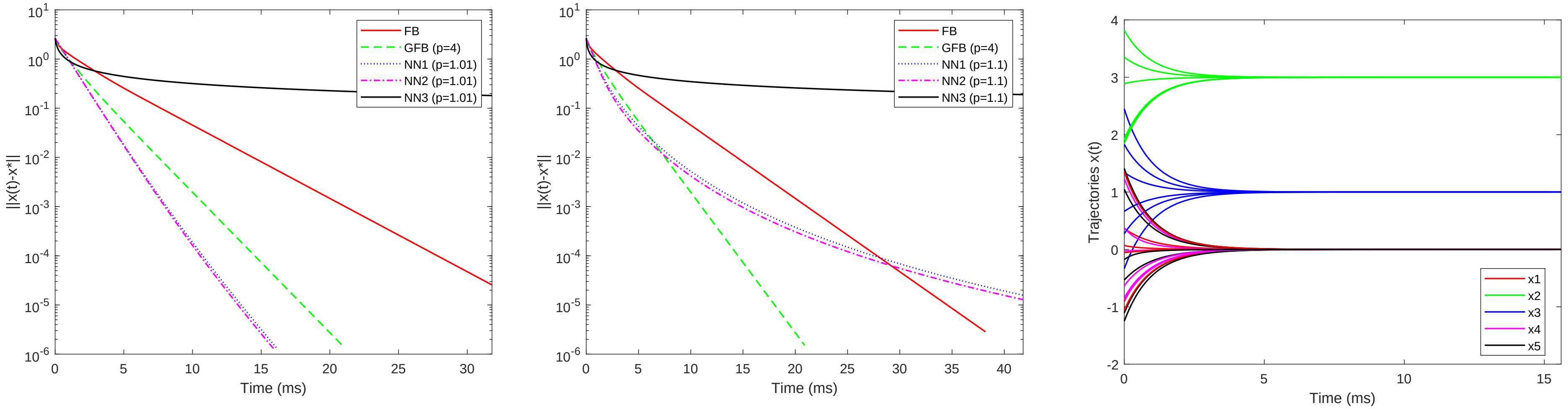


(a)

(b)

(c)

Figure 9: Comparison of neural networks based on NR and FB functions, and the convergence of the neural network based on $\widetilde{\phi}^p_{\mathrm{S-NR}}$ to the degenerate solution $x^* = (\sqrt{6}/2, 0, 0, 0.5)^T$ and non-degenerate solution $x^* = (1, 0, 3, 0)^T$ (using $p = 1.01$ and $p = 2$, respectively) for NCP7.



(a)

(b)

(c)

Figure 10: Comparison of neural networks based on NR and FB functions, and the convergence of the neural network based on $\widetilde{\phi}^p_{\mathrm{S-NR}}$ $(p = 1.01)$ to the solution $x^* = (0, 3, 1, 0, 0)^T$ for NCP8.

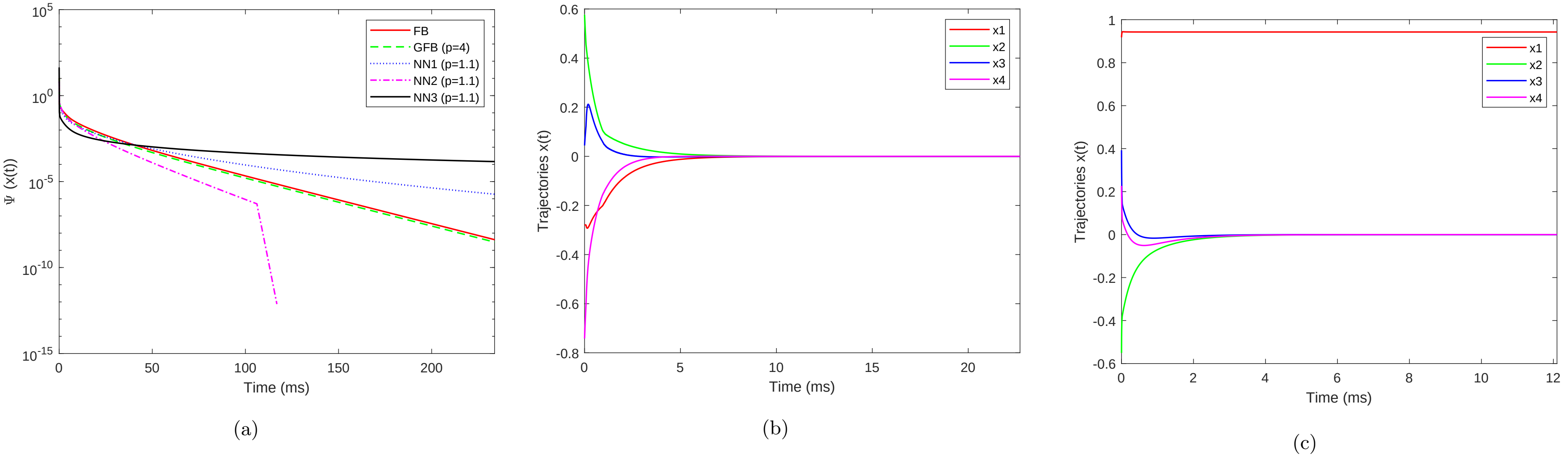


Figure 11: Comparison of neural networks based on NR and FB functions, and the convergence of the neural network based on $\widetilde{\phi}^{p}_{\mathrm{S-NR}}$ $(p = 1.01)$ to a solution $x^* = (k, 0, 0, 0)^T$ (where $k \in [0, 3]$) for NCP9.

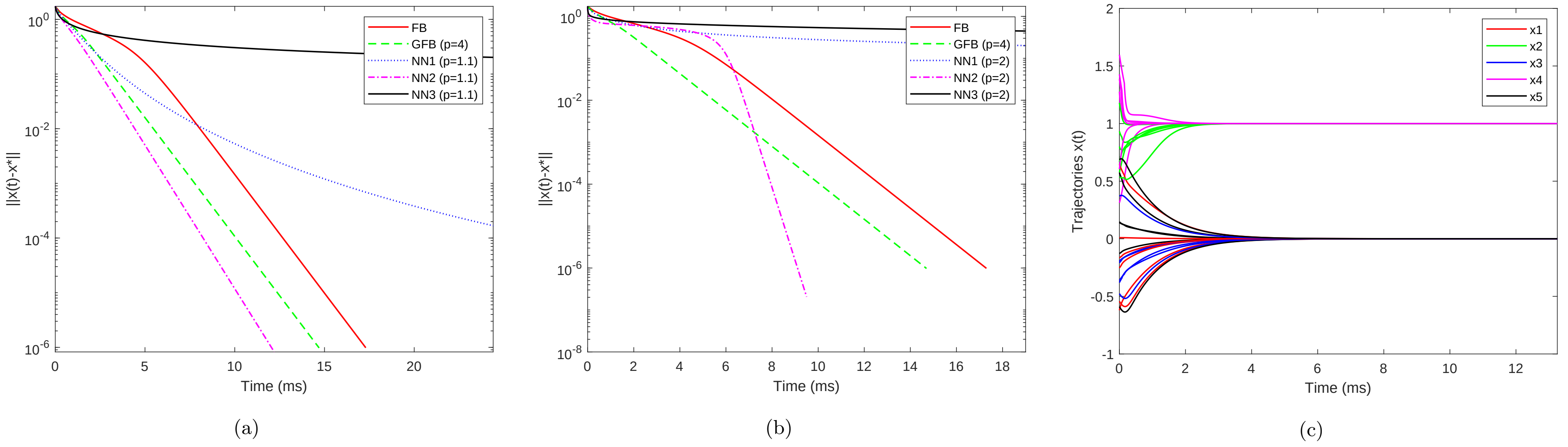


Figure 12: Comparison of neural networks based on NR and FB functions, and the convergence of the neural network based on $\widetilde{\phi}^{p}_{\mathrm{S-NR}}$ to the solution $x^* = (0, 1, 0, 1, 0)^T$ for NCP10.