Fundamental Journal of Mathematics and Applications, 6 (3) (2023) 137-146
Research Article

Fundamental Journal of Mathematics and Applications

Journal Homepage: www.dergipark.org.tr/en/pub/fujma
ISSN: 2645-8845
doi: https://dx.doi.org/10.33401/fujma.1261409

A New Smoothing Algorithm to Solve a System of Nonlinear Inequalities

Nurullah Yılmaz[1*] and Ayşegül Kayacan[1]

[1] *Department of Mathematics, Faculty of Science and Arts, Süleyman Demirel University, Isparta, Türkiye*
[*] *Corresponding author*

Article Info

Keywords: *Iterative methods, Smoothing, System of non-linear inequalities.*
2010 AMS: *65D10, 90C53*
Received: *9 March 2023*
Revised: *5 June 2023*
Accepted: *19 June 2023*
First Online: *24 August 2023*
Published: *30 September 2023*

Abstract

In this study, the system of nonlinear inequalities (SNI) problem is investigated. First, a SNI is reformulated as a system of nonsmooth and nonlinear equations (SNNE). Second, a new smoothing technique for the "max" function is proposed and the smoothing technique is employed for each element of the SNNE. Then, a new smoothing algorithm is developed in order to solve SNNE by combining the smoothing technique with the iterative method. The new algorithm is applied to some numerical examples to show the efficiency of our algorithm.

1. Introduction

In this paper, the following system of non-linear inequalities is considered:

$$H(x) \leq 0, \tag{1.1}$$

where, $H(x) := (h_1(x), h_2(x), \ldots, h_n(x))^T$ with $h_i : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable for any $i \in \{1, \ldots, n\}$ [1,2]. The SNI has been emerged in many real-world applications such as image restoration problems, data analysis, supply chain problems, computer aided design problems, compressive sensing problems and set separation problems [2–4]. In recent years, motion control involving two–joint planar robotic manipulator are modelled as SNI in [5] and it is faced a problem of SNI in process of designing parallel manipulator for aliquoting of biomaterials [6]. Depending on all of these practical applications, SNI has been extensively studied over the years [7–10].
There are many interesting methods have been proposed to solve the problem (1.1) such as modified Newton methods [11], smoothing Newton methods [1,2,8], Broyden-like methods [12], Conjugate Gradient methods [13,14], trust-region method [15], homotopy method [16] and etc. Although there is a parameter that must be tune to construct a smoothing function, among the all methods the smoothing Newton methods come into prominance due to their excellent numerical performance [17]. Smoothing Newton methods developed by modifiying the line search strategies such as exact, inexact, bactracking and etc. types line search techniques [18, 19]. One of the main tool of smoothing Newton method is smoothing functions. The smoothing function is defined as follows:

Definition 1.1. *[20] A function $\tilde{H} : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ is called a smoothing function of a non-smooth function $H : \mathbb{R}^n \to \mathbb{R}^m$ if, for any $\varepsilon > 0$, $\tilde{H}(x, \varepsilon)$ is continuously differentiable and*

$$\lim_{z \to x, \varepsilon \downarrow 0} \tilde{H}(z, \varepsilon) = H(x)$$

for any $x \in R^n$.

Email addresses and ORCID numbers: **nurullahyilmaz@sdu.edu.tr, 0000-0001-6429-7518 (N. Yilmaz), aysegulkayacan98@gmail.com, 0000-0002-6285-5771 (A. Kayacan)**
Cite as: N. Yılmaz, A. Kayacan, *A new smoothing algorithm to solve a system of nonlinear inequalities*, Fundam. J. Math. Appl., 6(3) (2023), 137-146

Smoothing functions have been studied by many scholars [21–24] and they have been applied to solve many interesting non-smooth problems over the years [25–27]. The comprehensive overview on smoothing approaches can be found in [20, 28, 29]. Among the smoothing functions, the smoothing function studied in [30] distinguishes itself from the others due to different structure, formulations and useful properties.
In this study, we propose a new smoothing function inspiring from the smoothing methods given in [30] based on the problem (1.1). By the help of this smoothing approach, we design a family of smooth equations which is surrogate for the original problem (1.1). A practical and user-friendly algorithm is developed to solve the surrogate system. The algorithm is implemented to some test problems in the literature in order to demonstrate the numerical performance of it. The comparison with corner stone studies has been presented in order to show the superiority of our algorithm.
Throughout the paper, $\mathbb{R}^{n+}$ denotes the non-negative part of $\mathbb{R}^n$, I denotes $n\times n$ identity matrix. For any vector $u \in \mathbb{R}^n$, u^T denotes the transpose of u and the Euclidean norm of u is denoted by $\|u\|$. In the following section, we present the new smoothing technique and a new formulation of system of non-linear inequalities. In Section 3, some numerical experiments are illustrated and comparison with the other methods is presented. Some concluding remarks are given in the last section.

2. Main Results

In the first subsection, we re-formulate the SNI (2.9) as a system of nonsmooth and nonlinear equations, then we propose a new smoothing technique to make smooth the each element of reformulated problem. In the next, we propose an algorithm to solve smoothed reformulated problem.

2.1. Smoothing Techniques

Let us define the function for any $y \in \mathbb{R}^n$

$$y_+ := \left(\max\{0, y_1\}, \dots, \max\{0, y_n\}\right)^T. \tag{2.1}$$

Then, problem (1.1) is reformulated as the following system of nonlinear equations:

$$H(x)_+ = 0. \tag{2.2}$$

By considering the smoothing techniques proposed in our another study [30] and adapt it for "max" function. For any $t \in \mathbb{R}$, define the function $\phi(t) = \max\{t, 0\}$ which is also stated as

$$\phi(t) = \begin{cases} 0, & t \le 0, \\ t, & t > 0. \end{cases}$$

The function $\phi(t)$ is re-stated by the help of indicator function as

$$\phi(t) = t\psi(t), \tag{2.3}$$

where,

$$\psi(t) = \begin{cases} 0, & t \le 0, \\ 1, & t > 0. \end{cases} \tag{2.4}$$

The function defined in (2.4) is not smooth. We propose to use the following smoothing function:

$$\psi^{l_j}(t,\varepsilon) = \begin{cases} 0, & t < -\varepsilon, \\ D_j(t,\varepsilon), & -\varepsilon \le t \le \varepsilon, \\ 1, & t > \varepsilon, \end{cases} \tag{2.5}$$

where,

$$D_1(t,\varepsilon) = \frac{-1}{4\varepsilon^3}t^3 + \frac{3}{4\varepsilon}t + \frac{1}{2},$$

and

$$D_2(t,\varepsilon) = \frac{3}{16\varepsilon^5}t^5 - \frac{10}{16\varepsilon^3}t^3 + \frac{15}{16\varepsilon}t + \frac{1}{2},$$

for $j = 1, 2$. Based on the smoothing functions of indicator function, the smoothing function of $\phi(t)$ in (2.3) is defined as

$$\phi^{l_j}(t,\varepsilon) = t\psi^{l_j}(t,\varepsilon), \tag{2.6}$$

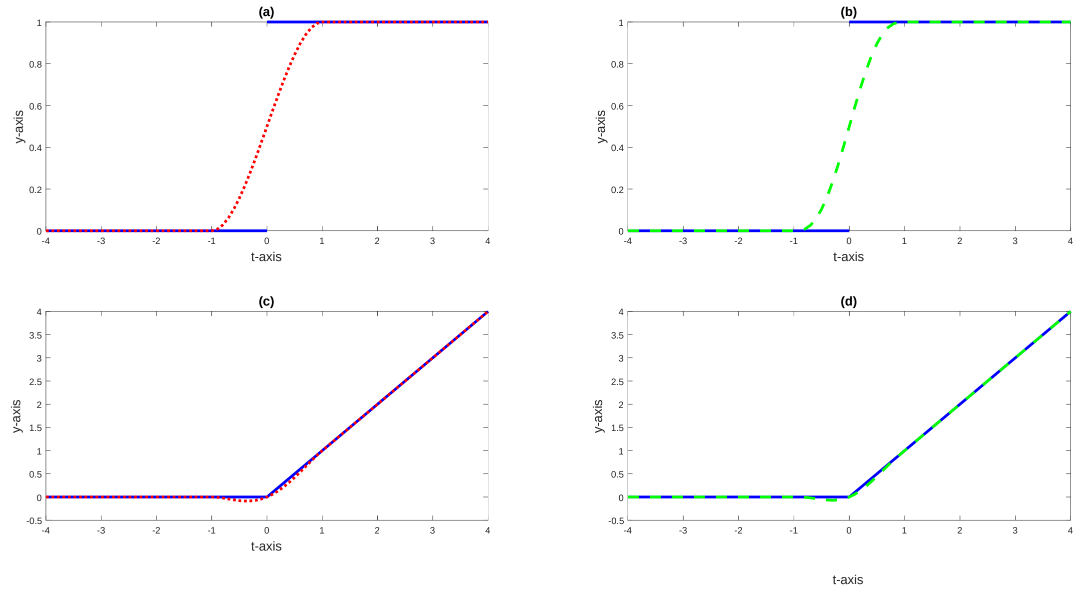

Figure 2.1: The graphics of smoothing functions of indicator functions and smoothing functions of max function.

for $j=1,2$.
We illustrate all the smoothing process by using graphs. The graphs of $\psi(t)$ and $\psi^{l_j}(t,\varepsilon)$ is illustrated in Fig. 2.1 (a) and (b). In Fig. 2.1 (a), the blue and solid graph indicates the function $\psi(t)$ and the red and dotted one indicates $\psi^{l_1}(t,\varepsilon)$ and, Fig. 2.1 (b) the blue and solid graph again indicates the function $\psi(t)$ and the green and dashed one indicates $\psi^{l_2}(t,\varepsilon)$ for $\varepsilon=1$.
The graphs of $\phi(t)$ and $\phi^{l_j}(t,\varepsilon)$ is illustrated in Fig. 2.1 (c) and (d). In Fig. 2.1 (c), the blue and solid graph indicates the function $\phi(t)$ and the red and dotted one indicates $\phi^{l_1}(t,\varepsilon)$ and, Fig. 2.1 (d) the blue and solid graph again indicates the function $\psi(t)$ and the green and dashed one indicates $\phi^{l_2}(t,\varepsilon)$ for $\varepsilon=1$.

Lemma 2.1. *Let $\phi^{l_j}:\mathbb{R}\times\mathbb{R}_+\to\mathbb{R}$ be defined as in (2.6) then,*

i. ϕ^{l_j} is continuously differentiable at $t\in\mathbb{R}$,
ii. $\lim_{\varepsilon\to 0}\phi^{l_j}(t,\varepsilon)=\phi(t)$ for any $t\in\mathbb{R}$,

for any $\varepsilon>0$ and $j=1,2$. Moreover ϕ^{l_2} is second order continuously differentiable.

Proof. i. The derivative of the smoothing functions ϕ^{l_j} are

$$\frac{d}{dt}\left(\phi^{l_j}(t,\varepsilon)\right)=\psi^{l_j}(t,\varepsilon)+t\frac{d}{dt}\left(\psi^{l_j}(t,\varepsilon)\right),$$

where,

$$\frac{d}{dt}\left(\psi^{l_j}(t,\varepsilon)\right)=\begin{cases}0, & t<-\varepsilon,\\ \frac{d}{dt}(D_j(t,\varepsilon)), & -\varepsilon\le t\le\varepsilon,\\ 0, & t>\varepsilon,\end{cases}$$

with

$$\frac{d}{dt}(D_1(t,\varepsilon))=\frac{-3}{4\varepsilon^3}t^2+\frac{3}{4\varepsilon},$$

and

$$\frac{d}{dt}(D_2(t,\varepsilon))=\frac{15}{16\varepsilon^5}t^4-\frac{30}{16\varepsilon^3}t^2+\frac{15}{16\varepsilon},$$

and

$$\frac{d^2}{dt^2}(D_2(t,\varepsilon))=\frac{15}{4\varepsilon^5}t^3-\frac{15}{4\varepsilon^3}t,$$

for $j=1,2$.

ii. For any $\varepsilon > 0$ and $t \notin I = [-\varepsilon, \varepsilon]$, then $\phi_{l_j}(t, \varepsilon) = \phi(t)$ for $j = 1, 2$. Assume that $t \in [-\varepsilon, 0]$, since $\psi(t) = 0$ we have

$$\begin{aligned} 0 \leq \phi(t) - \phi^{l_j}(t,\varepsilon) &= t\psi(t) - t\psi^{l_j}(t,\varepsilon) \\ &\leq -t\psi^{l_j}(t,\varepsilon) \\ &\leq \frac{\varepsilon}{2}. \end{aligned}$$

Now, let $t \in [0, \varepsilon]$ then, we have

$$\begin{aligned} 0 \leq \phi(t) - \phi^{l_j}(t,\varepsilon) &= t\psi(t) - t\psi^{l_j}(t,\varepsilon) \\ &\leq t\left(1 - \psi^{l_j}(t,\varepsilon)\right) \\ &\leq \frac{\varepsilon}{2}. \end{aligned}$$

From the above results the $\phi^{l_j}(t,\varepsilon) \to \phi(t)$ as $\varepsilon \to 0$ for $j = 1, 2$.

The proof is completed. □

By considering the functions $\phi^{l_j}(t,\varepsilon)$ instead of $\phi(t)$, the corresponding smoothing function of $\phi\left(h_i(x)\right)$ is obtained, for $j = 1, 2$ and $i = 1, 2, \ldots, n$. The resulting smoothing approximation of $H(x)_+$ is stated as a system of smooth nonlinear equations by

$$\tilde{H}(x,\varepsilon) = 0, \tag{2.7}$$

where

$$\tilde{H}(x,\varepsilon) = \begin{bmatrix} \phi_1^{l_j}(x,\varepsilon) \\ \phi_2^{l_j}(x,\varepsilon) \\ \vdots \\ \phi_n^{l_j}(x,\varepsilon) \end{bmatrix},$$

and $\phi_1^{l_j}(x,\varepsilon) = \phi^{l_j}(h_1(x),\varepsilon)$, $\phi_2^{l_j}(x,\varepsilon) = \phi^{l_j}(h_2(x),\varepsilon), \ldots, \phi_n^{l_j}(x,\varepsilon) = \phi^{l_j}(h_n(x),\varepsilon)$ for $\varepsilon > 0$.

Theorem 2.2. *Assume the functions $H(x)_+$ and $\tilde{H}(x,\varepsilon)$ be stated as in (1.1) and (2.7), respectively. Then, we obtain*

$$\|H(x)_+ - \tilde{H}(x,\varepsilon)\| \leq \frac{\varepsilon}{2}\sqrt{n}.$$

Proof. For any $\varepsilon > 0$,

$$\begin{aligned} \|H(x)_+ - \tilde{H}(x,\varepsilon)\|^2 &= \sum_{i=1}^{n} \left|\phi_i(x) - \phi_i^{l_j}(x,\varepsilon)\right|^2 \\ &\leq \sum_{i=1}^{n} \left(\frac{\varepsilon}{2}\right)^2 \\ &= n\frac{\varepsilon^2}{4} \end{aligned}$$

for $j = 1, 2$. This completes the proof. □

Theorem 2.3. *The function $\tilde{H}(x,\varepsilon)$ is continuously differentiable and the Jacobian of $\tilde{H}(x,\varepsilon)$ is obtained as*

$$\tilde{H}'(x,\varepsilon) = \begin{bmatrix} \frac{\partial\phi_1^{l_j}(x,\varepsilon)}{\partial x_1} & \frac{\partial\phi_1^{l_j}(x,\varepsilon)}{\partial x_2} & \cdots & \frac{\partial\phi_1^{l_j}(x,\varepsilon)}{\partial x_n} \\ \frac{\partial\phi_2^{l_j}(x,\varepsilon)}{\partial x_1} & \frac{\partial\phi_2^{l_j}(x,\varepsilon)}{\partial x_2} & \cdots & \frac{\partial\phi_2^{l_j}(x,\varepsilon)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\phi_n^{l_j}(x,\varepsilon)}{\partial x_1} & \frac{\partial\phi_n^{l_j}(x,\varepsilon)}{\partial x_2} & \cdots & \frac{\partial\phi_n^{l_j}(x,\varepsilon)}{\partial x_n} \end{bmatrix}, \tag{2.8}$$

for any $\varepsilon > 0$.

2.2. Algorithm

Define the following function

$$G(x,\lambda)=\frac{\lambda}{2}\|H(x)_+\|^2,$$

and its smooth approximation

$$\tilde{G}(x,\lambda,\varepsilon)=\frac{\lambda}{2}\|\tilde{H}(x,\varepsilon)\|^2,$$

where $\lambda,\varepsilon>0$.

Assumption 1. *$\tilde{H}'(x,\varepsilon)$ is invertible for any $x\in\mathbb{R}^n$.*

Theorem 2.4. *Assume that the Assumption 1 is hold. Then, $\nabla\tilde{G}(x,\lambda,\varepsilon)=0$ if and only if $\tilde{H}(x,\varepsilon)=0$.*

Proof. Let $\nabla\tilde{G}(x,\lambda,\varepsilon)=0$ for any $\lambda,\varepsilon>0$. Then, we have that

$$\nabla\tilde{G}(x,\lambda,\varepsilon)=\lambda[\tilde{H}'(x,\varepsilon)]^T\tilde{H}(x,\varepsilon)=0. \tag{2.9}$$

There are two different cases in solving (2.9). At the first one is $\tilde{H}(x,\varepsilon)=0$ in which the proof is directly obtained. At the second one, $\tilde{H}(x,\varepsilon)\neq 0$ and $\tilde{H}'(x,\varepsilon)=0$ are obtained. In this case, $\tilde{H}'(x,\varepsilon)=0$ contradicts the Assumption 1. □

It should be stated that the Assumption 1 is necessary to guarantee the equivalence of problems 2.2 and 2.10.
By considering Theorem 2.4, the following optimization problem

$$\min_{x\in\mathbb{R}^n}\tilde{G}(x,\lambda,\varepsilon) \tag{2.10}$$

can be considered as a surrogate problem for $\tilde{H}(x,\varepsilon)=0$.

Remark 2.5. *It should be stated that the Assumption 1 is necessary to guarantee the equivalence of problems 2.2 and 2.10.*

Theorem 2.6. *Let x^* and $\bar{x}$ are local minimizers of $G(x,\lambda)$ and $\tilde{G}(x,\lambda,\varepsilon)$, respectively. Then,*

$$0\leq G(x^*,\lambda)-\tilde{G}(\bar{x},\lambda,\varepsilon)\leq n\lambda\frac{\varepsilon^2}{8}.$$

Proof. From Theorem 2.2 we obtain

$$\begin{aligned}
0\leq G(x^*,\lambda)-\tilde{G}(x^*,\lambda,\varepsilon) &\leq \frac{\lambda}{2}\left(\|H(x^*)\|^2-\left\|\tilde{H}(\bar{x},\varepsilon)\right\|^2\right)\\
&\leq \frac{\lambda}{2}\left\|H(x^*)-\tilde{H}(\bar{x},\varepsilon)\right\|^2\\
&\leq \frac{\lambda}{2}\left\|H(\bar{x})-\tilde{H}(\bar{x},\varepsilon)\right\|^2\\
&\leq n\lambda\frac{\varepsilon^2}{8}.
\end{aligned}$$

□

It is easy to see that $\tilde{G}(\bar{x},\lambda,\varepsilon)\to G(x^*,\lambda)$ as $\varepsilon\to 0$. We now give the following definition.

Definition 2.7. *A point x is called as $\tau-$approximate solution for (1.1) if the condition*

$$\|H(x)_+\|<\tau$$

is hold.

The following algorithm is proposed in order to solve (2.10), numerically.

Algorithm 1:

---

1 Choose the starting point x^0 and tolerance parameter $\tau=10^{-4}$. Select the parameters $\lambda^0>0$, $\varepsilon^0>0$, $L>1$, $0<\eta<1$ and let $k=0$.
2 Solve the problem (2.10) by using x^k as a starting point. Let x^{k+1} be the optimal solution.
3 If x^{k+1} is $\tau-$approximate solution, then stop. Otherwise, update the parameters $\varepsilon^{k+1}=\eta\varepsilon^k$, $\lambda^{k+1}=L\lambda^k$ and $k=k+1$, then go to Step 2.

It should be noted that Quasi Newton method is used in Step 1. Now, we prove the convergence of the Algorithm I. First, we define the level set as

$$\mathscr{L}(x^0)=\left\{x\in\mathbb{R}^n:\|H(x)_+\|^2\leq\|H(x^0)_+\|^2\right\}$$

for a stating point x^0.

Assumption 2. *For any $\varepsilon > 0$ and for a starting point x^0, the set*

$$\mathscr{L}_\varepsilon(x^0) = \left\{x \in \mathbb{R}^n : \|\tilde{H}(x,\varepsilon)\|^2 \leq \|\tilde{H}(x^0,\varepsilon)\|^2\right\}$$

is bounded.

Theorem 2.8. *Assume that Assumptions 1 and 2 are hold. Then, a sequence $\{x^k\}$ generated by Algorithm I converges to the optimal solution of the problem (2.2).*

Proof. It can be seen that $x^k \in \mathscr{L}_{\varepsilon^k}(x^0)$ for all $k \geq 0$. Since, $\mathscr{L}_{\varepsilon^k}(x^0)$ is bounded then there exists set $K \subset \mathbb{N}$ such that $\{x^k\}$ has a limit point for $k \in K$. Assume that $\bar{x}$ is a limit point of $\{x^k\}$. We have to show that $\bar{x}$ is the optimal solution for (2.2). Thus, it suffices to show that $\bar{x} \in \mathscr{L}(x^0)$ and $\|F(\bar{x})\|^2 \leq \inf_{x\in\mathscr{L}(x^0)} \|H(x)\|^2$.
Let us consider the contrary that $\bar{x} \notin \mathscr{L}(x^0)$, i.e. for sufficiently large $k \in K$, there exist $\beta_0 > \|H(x^0)_+\|^2$ and $i_0 \in \{1,2,\ldots,n\}$ such that

$$h_{i_0}^2(x^k) \geq \beta_0 > 0.$$

Since x^k is the global minimum according $k-$th values of the parameters ε^k and λ^k, for any $x \in \mathscr{L}(x^0)$ we have

$$\tilde{G}(x^0,\lambda^k,\varepsilon^k) \geq \tilde{G}(x^k,\lambda^k,\varepsilon^k) = \frac{\lambda^k}{2}\left(\left(\phi_{i_0}(x^k,\varepsilon)\right)^2 + \sum_{\substack{i=1\\ i\neq i_0}}^{n}\left(\phi_i(x^k,\varepsilon)\right)^2\right) \geq \frac{\lambda^k}{2}\beta_0.$$

If $k \to \infty$ then, $\lambda^k \to \infty$ and $\lim_{k\to\infty} \tilde{G}(x^k,\lambda^k,\varepsilon^k) = \infty$. It contradicts with the boundedness of the $\mathscr{L}_\varepsilon(x^0)$. Moreover, we have

$$\tilde{G}(x^k,\lambda^k,\varepsilon^k) \leq \tilde{G}(x,\lambda^k,\varepsilon^k)$$

for any $x \in \mathscr{L}(x^0)$. When $k \to \infty$, we have $\tilde{G}(\bar{x},\lambda,\varepsilon) \leq \tilde{G}(x,\lambda,\varepsilon)$. $\square$

3. Numerical Results

In this section, we implement the Algorithm I to some test problems in order to evaluate the efficiency of Algorithm I. We compare our numerical results with the methods given in [2,7,8,16]. The numerical experiments have been performed on a PC with Intel Core i5-1035G1 CPU 1.00 GHz and 8GB RAM. The operating system is Windows 10 and the implementations have been done in MATLAB. At the algorithm, the parameters are taken as $\varepsilon_0 = 10^{-1}$ and $\eta = 0.1$. It is accepted that the problem is solved, if the accuracy 10^{-4} with respect to function value is obtained. The “fminunc” function is used as solver. The numerical test problems 1 to 7 are of the form (1.1) and the details are presented as follows:

Problem 1. *[1,2] Consider the function $H : \mathbb{R}^2 \to \mathbb{R}^2$ such that*

$$H(x) = \begin{bmatrix} \sin(x_1) \\ \cos(x_2) \end{bmatrix}.$$

Problem 2. *[1,2] Consider the function $H : \mathbb{R}^3 \to \mathbb{R}^3$ such that*

$$H(x) = \begin{bmatrix} (x_1-0.5)^2 + (x_2-1)^2 - 0.25 \\ -(x_1-0.5)^2 - (x_1-1.1)^2 + x_2^2 - 0.26 \\ x_2 + x_3^2 - 1 \end{bmatrix}.$$

Problem 3. *[1,2] Consider the function $H : \mathbb{R}^6 \to \mathbb{R}^6$ such that*

$$H(x) = \begin{bmatrix} \sin(x_1)+\varepsilon \\ -\cos(x_2)+\varepsilon \\ x_1 - 3\pi + x_3^2 + \varepsilon \\ x_2 - \pi/2 + x_4^2 + \varepsilon \\ -x_1 - \pi + x_5^2 + \varepsilon \\ -x_2 - \pi/2 + x_6^2 + \varepsilon \end{bmatrix},$$

where $\varepsilon = 10^{-5}$.

Problem 4. *[16] Consider $H : \mathbb{R}^2 \to \mathbb{R}^2$ such that*

$$H(x) = \begin{bmatrix} -x_1^2 - x_2^2 - x_1 + 1 + \varepsilon \\ -x_1^2 - x_2^2 + 2x_2 + 2 + \varepsilon \end{bmatrix},$$

where $\varepsilon = 10^{-7}$.

Problem 5. *[16] Consider $H : \mathbb{R}^5 \to \mathbb{R}^5$ such that*

$$H(x) = \begin{bmatrix} \sin(x_1) + \varepsilon \\ -\cos(x_2) + \varepsilon \\ 2x_1 - x_1^2 + x_3^2 + \varepsilon \\ 2x_2 - x_2^2 + x_4^2 + \varepsilon \\ x_1 - x_2 + x_5^2 + \varepsilon \end{bmatrix},$$

where $\varepsilon = 10^{-7}$.

Problem 6. *[16] Consider $H : \mathbb{R}^3 \to \mathbb{R}^3$ such that*

$$H(x) = \begin{bmatrix} x_1(x_1 - 2)(x_1 - 3) + \varepsilon \\ x_2^2 - 3x_2 + 2 + \varepsilon \\ x_1^2 - x_2 + x_3^2 + \varepsilon \end{bmatrix},$$

where $\varepsilon = 10^{-7}$.

Problem 7. *[8] Consider $H : \mathbb{R}^2 \to \mathbb{R}^2$ such that*

$$H(x) = \begin{bmatrix} x_1^2 + x_2^2 - 1 + \varepsilon \\ -x_1^2 - x_2^2 + (0.999)^2 + \varepsilon \end{bmatrix},$$

where $\varepsilon = 10^{-5}$.

We apply our algorithm also to test problems of the following form:

$$H(x) = \begin{cases} h_I(x) \leq 0, \\ h_J(x) = 0, \end{cases} \tag{3.1}$$

where $I = \{1, 2, \ldots, m\}$ and $J = \{m+, m+2, \ldots, n\}$. The function $h_I : \mathbb{R}^n \to \mathbb{R}^m$ is defined as

$$h_I = \begin{bmatrix} h_1(x) \\ h_2(x) \\ \vdots \\ h_m, \end{bmatrix}$$

and $h_J : \mathbb{R}^n \to \mathbb{R}^{n-m}$ is defined as

$$h_J = \begin{bmatrix} h_m(x) \\ h_{m+1}(x) \\ \vdots \\ h_n, \end{bmatrix}$$

where the component functions $f_i, f_j : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable. In other words, the function $H : \mathbb{R}^n \to \mathbb{R}^n$ is defined as

$$H(x) = \begin{bmatrix} h_I(x) \\ \\ h_J(x) \end{bmatrix}.$$

It is easy to see that when the set J is empty, the system (3.1) corresponds to the system (1.1). Now consider the following test problems which are of the form (3.1).

Problem 8. *[8] Consider $H : \mathbb{R}^5 \to \mathbb{R}^5$ such that*

$$h_I(x) = \begin{bmatrix} x_1 + x_3 - 1.6 + \varepsilon \\ \\ 1.333x_2 + x_4 - 3 + \varepsilon \\ \\ -x_3 - x_4 + x_5 + \varepsilon \end{bmatrix} \qquad and \qquad h_J(x) = \begin{bmatrix} x_1^2 + x_3^2 - 1.25, \\ \\ x_2^{1.5} + 1.5x_4 - 3 \end{bmatrix},$$

where $\varepsilon = 10^{-5}$.

Problem 9. *[8] Consider $H : \mathbb{R}^3 \to \mathbb{R}^3$ such that*

$$h_I(x) = \begin{bmatrix} x_1 + x_2 e^{0.8x_3} + e^{1.6} + \varepsilon \\ \\ x_1^2 + x_2^2 + x_3^2 - 5.2675 + \varepsilon \end{bmatrix} \qquad and \qquad h_J(x) = \begin{bmatrix} x_1 + x_2 + x_3 - 0.2605 \end{bmatrix},$$

where $\varepsilon = 10^{-5}$.

Problem 10. *[8] Consider $H : \mathbb{R}^2 \to \mathbb{R}^3$ such that*

$$h_I(x) = \begin{bmatrix} 0.8 - e^{x_1 + x_2} + \varepsilon \end{bmatrix} \qquad and \qquad h_J(x) = \begin{bmatrix} 1.21e^{x_1} + e^{x_2} - 2.2 \\ \\ x_1^2 + x_2^2 + x_2 - 0.1135 \end{bmatrix},$$

where $\varepsilon = 10^{-5}$.

Problem 11. *[8] Consider $H : \mathbb{R}^3 \to \mathbb{R}^3$ such that*

$$h_I(x) = \begin{bmatrix} x_1^2 + x_2^2 + x_3^2 - 10000 + \varepsilon \end{bmatrix} \qquad and \qquad h_J(x) = \begin{bmatrix} x_1 - 0.7\sin(x_1) - 0.2\cos(x_2) \\ \\ x_2 - 0.7\cos(x_1) + 0.2\sin(x_2) \end{bmatrix},$$

where $\varepsilon = 10^{-5}$.

Table 1: The numerical results

PN	SP	FSP	TIN	TFE	SFV	FV	Time
1	$(0,1)$	$(3.2471e-06, 0.3155)$	2	12	$2.9236e-06$	$1.0544e-11$	0.2738
2	$(1,1,1)$	$(0.8759, 0.6721, 0.57163)$	14	72	$6.2936e-07$	$3.3391e-08$	0.3742
3	$(0,1,1,1,1,1)$	$(0.0000, 0.3156, 1.0000, 1.0000, 1.0000, 1.0000)$	2	28	$7.1189e-05$	$9.2784e-08$	0.3744
4	$(1,1)$	$(2,1)$	1	6	0	0	0.3342
5	$(-1,0,1,2,2)$	$(-1.3024, -0.3418, 1, 0.8945, 0.9801)$	11	90	$4.4538e-08$	$1.8001e-11$	0.3986
6	$(1,2,2)$	$(-0.5814, 2.0039, 0.8772)$	2	16	$5.2416e-05$	$1.5545e-05$	0.3118
7	$(0,5)$	$(-2.0903e-09, 0.9994)$	6	36	$7.4149e-05$	0	0.3498
8	$(0.5,2,1,0,0)$	$(0.5018, 2.0535, 0.9991, 0.0382, 0)$	4	48	$7.3160e-09$	$7.275e-09$	0.3065
9	$(-1,1,1)$	$(-0.8353, -0.8601, 1.9564)$	16	84	$5.9864e-06$	$1.5566e-05$	0.3715
10	$(0,0,0)$	$(-0.1015, 0.0992, 0)$	4	32	$4.0599e-05$	$4.0599e-05$	0.3504
11	$(0,1,0)$	$(0.4968, 0.6296, 0)$	6	36	$3.0136e-06$	$3.0136e-06$	0.3166

The results of the numerical experiments are presented in Table 1. In the Table 1, the problem number (PN), the starting point (SP), total iteration number (TIN), total function evaluation (TFE), founded solution point (FSP), the norm value of each smoothed problem $\|\tilde{H}(x,\varepsilon)\|$ in the problem (SFV), the norm value of each problem $\|H(x)\|$ in the problem (FV) and total CPU time (Time) are reported. The satisfactory results are obtained for all test problems. Since, our smoothing functions has the same value with the original function at the same location, the minimum value is obtained with lower number of iteration. This property is the main advantage of our method.

The results are compared with the results obtained from the methods suggested in [2,7,8,16] in terms of "TIN" and "TFE". The results are presented in Table 2 and it is observed that all of the test problems are successfully solved by using our algorithm. Moreover our algorithm presents better results in 82% of all the test problems than other methods in terms of "TIN".

Table 2: The comparison of the numerical results with the competing algorithms

	Algorithm I		Algorithm 3.1 in [2]		Algorithm in [16]		Algorithm 2.1 in [8]		Algorithm 3.1 in [7]	
PN	TIN	TFE	TIN	TFE	TIN	TFE	TIN	TFE	TIN	TFE
1	2	12	3	-	-	-	-	-	3	3
2	14	72	4	-	-	-	-	-	-	-
3	2	28	5	-	-	-	-	-	6	8
4	1	6	-	-	2	-	-	-	-	-
5	11	90	-	-	14	-	-	-	-	-
6	2	16	-	-	18	-	-	-	-	-
7	6	36	-	-	-	-	8	12	8	9
8	4	48	-	-	-	-	5	6	4	4
9	16	84	-	-	-	-	24	39	5	5
10	4	32	-	-	-	-	6	8	4	4
11	6	36	-	-	-	-	10	16	9	14

4. Conclusion

A new algorithm with the new smoothing approach is proposed to solve SNI and the convergence of the algorithm is theoretically presented. The efficiency of our algorithm is illustrated on test problems in the literature. The superiority of our method among the similar algorithms is proved numerically by considering Table 2. According to the comparison of the results with the other methods, it is shown that the Algorithm I has many advantages in terms of computational costs. On the other hand, this study presents a methodology to solve these kinds of problems.

For future works, the proposed smoothing approach can also be applied to other non-smooth problems such as min-max, complementarity, exact penalty, l_1 signal reconstruction and etc. Furthermore, the smoothing function can be used along with the other algorithms such as Newton type and Conjugate gradient algorithms, and related numerical performance can be investigated accordingly.

Article Information

Acknowledgements: The authors would like to express their sincere thanks to the editor and the anonymous reviewers for their helpful comments and suggestions.

Author's Contributions: The authors contributed equally to the writing of this paper.

Conflict of Interest Disclosure: No potential conflict of interest was declared by the author.

Supporting/Supporting Organizations: No grants were received from any public, private or non-profit organizations for this research.

Ethical Approval and Participant Consent: It is declared that during the preparation process of this study, scientific and ethical principles were followed and all the studies benefited from are stated in the bibliography.

Plagiarism Statement: This article was scanned by the plagiarism program. No plagiarism detected.

Availability of Data and Materials: Not applicable.

References

[1] Z. H. Huang, Y. Zhang, W. Wu, *A smoothing-type algorithm for solving system of inequalities*, J. Comput. Appl. Math., **220** (2008), 355–363.
[2] Y. Liuyang, C. Fei, W. Zhongping, L. Weigang, W. Wenbo, *A nonmonotone smoothing Newton method for system of nonlinear inequalities based on a new smoothing function*, Comput Appl. Math., **134** (2019), 38–91.
[3] C. Wu, J. Wang, J. H. Alcantara and J. S. Chen, *Smoothing strategy along with conjugate gradient algorithm for signal seconstruction*, J. Sci. Comput. **87** (2021), 21.
[4] Y. Xiao, Q. Wang, Q. Hu, *Non-smooth equations based method for ℓ_1-norm problems with applications to compressed sensing*, Nonlinear Anal. Theory Methods Appl., **74**(11) (2011), 3570–3577.
[5] A. S. Halilu, A. Majumder, M. Y. Waziri, K. Ahmed, A.M Awwal, *Motion control of the two joint planar robotic manipulators through accelerated Dai–Liao method for solving system of nonlinear equations*, Eng. Comput., **39**(5) (2022), 1802–1840.
[6] D. Malyshev, L. Rybak, G. Carbone, T. Semenenko, A. Nozdracheva, *Optimal design of a parallel manipulator for aliquoting of biomaterials considering workspace and singularity zones*, Appl. Sci., **12**(4) (2022), 2070.
[7] Y. Zhang, Z.-H. Huang, *A nonmonotone smoothing-type algorithm for solving a system of equalities and inequalities*, J. Comput. Appl. Math., **233** (2010) 2312–2321.
[8] J. S. Chen, C. H. Ko, Y. D. Liu, S. P. Wang, *New smoothing functions for solving a system of equalities and inequalities*, Pacific J. Optim, **12**(1) (2016), 185-206.
[9] G. Yuan, T. Li, W. Hu, *A conjugate gradient algorithm for large-scale nonlinear equations and image restoration problems*, Appl. Numer. Math., **147** (2020), 129–141.

[10] R. Li, M. Cao, G. Zhou, *A new adaptive accelerated Levenberg–Marquardt method for solving nonlinear equations and its applications in supply chain problems*, Symmetry, **15**(3) (2023), 588.
[11] D.Q. Mayne, E. Polak, A. J. Heunis, *Solving nonlinear inequalities in a finite number of iterations*, J. Optim. Theory Appl., **33**(2) (1981), 207–221.
[12] H. Che, M. Li, *A smoothing and regularization Broyden-like method for nonlinear inequalities*, J. Appl. Math. Comput., **41** (2013) 209–227.
[13] A. B. Abubakar, P. Kumam, *A descent Dai-Liao conjugate gradient method for nonlinear equations*, Numer. Algorithms, 81 (2019), 197-210.
[14] M. Y. Waziri, K. Ahmed, K. *Two descent Dai-Yuan conjugate gradient methods for systems of monotone nonlinear equations*, J. Sci. Comput., **90** (2022), 1-53.
[15] Z. Saeidian, M. R. Peyghami, M. Habibi, S. Ghasemi, *A new trust-region method for solving systems of equalities and inequalities*, Comp. Appl. Math. **36** (2017), 769 70.
[16] X. Fan, Q. Yan, *Solving system of inequalities via a smoothing homotopy method*, Numer Algorithms **82** (2019), 719–728.
[17] L. Qi, D. Sun, *Smoothing functions and smoothing Newton method for complementarity and variational inequality problems*, J. Optim. Theory Appl. **113** (2002), 121–147.
[18] S. Huang, Z. Wan, X. H. Chen, *A new nonmonotone line search technique for unconstrained optimization*, Numer. Algorithms **68** (4) (2015), 671–689.
[19] H. C. Zhang, W. W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM J. Optim., **14**(4) (2004), 1043–756.
[20] X. Chen, *Smoothing methods for nonsmooth, nonconvex minimization*, Math. Program. Ser. (B), **134** (2012), 71–99.
[21] D. Bertsekas, *Nondifferentiable optimization via approximation*, Math Programmming Study, **3** (1975) 1–25.
[22] I. Zang, *A smooting out technique for min-max optimization*, Math. Programm., **19** (1980), 61–77.
[23] C. Chen, O.L. Mangasarian, *A class of smoothing functions for nonlinear and mixed complementarity problem*, Comput. Optim. Appl., **5** (1996) 97-138.
[24] A. E. Xavier, *Optimal covering of plane domains by circles via hyperbolic smoothing*, J. Global Optim., **31** (2005), 493–504.
[25] A. M. Bagirov, A. Al Nuamiat, N. Sultanova, *Hyperbolic smoothing functions for nonsmooth minimization*, Optimization, **62** (6) (2013), 759-782.
[26] A. Sahiner, G. Kapusuz, N. Yilmaz, *A new smoothing approach to exact penalty functions for inequality constrained optimization problems*, Numer. Algebra Control Optim., **6** (2) (2016) 161–173.
[27] N. Yilmaz, A. Sahiner, *On a new smoothing technique for non-smooth, non-convex optimization*, Numer. Algebra Control Optim., 10 (2020), 317–330.
[28] C. Grossmann, *Smoothing techniques for exact penalty function methods*, *Contemporary Mathematics: A Panaroma of Mathematics Pure and Applied*, American Mathematical Society, Rhode Island, **658** (2016) 249–265.
[29] L. Qi, P. Tseng, *On almost smooth functions and piecewise smooth functions*, Nonlinear Anal. **67** (2007) 773–794.
[30] N. Yilmaz, A. Sahiner, *New smoothing approximations to piecewise smooth functions and applications*, Numer. Func. Anal. Optim., **40** (2019), 513–534.