

Research Article

Computer-Based Fuzzy Numerical Method for Solving Engineering and Real-World Applications

Naila Rafiq ¹, **Naveed Yaqoob** ², **Nasreen Kausar** ³, **Mudassir Shams** ²,
Nazir Ahmad Mir ¹, **Yaé Ulrich Gaba** ^{4,5,6} and **Naveed Khan**²

¹Department of Mathematics, NUML, Islamabad, Pakistan

²Department of Mathematics and Statistics, Riphah International University I-14, Islamabad 44000, Pakistan

³Department of Mathematics, Yildiz Technical University, Faculty of Arts and Science, Esenler, 34210 Istanbul, Turkey

⁴Quantum Leap Africa (QLA), AIMS Rwanda Centre, Remera Sector KN 3, Kigali, Rwanda

⁵Institut de Mathématiques et de Sciences Physiques (IMSP/UAC), Laboratoire de Topologie Fondamentale, Computationnelle et Leurs Applications (Lab-ToFoCApp), BP 613, Porto-Novo, Benin

⁶African Center for Advanced Studies, P.O. Box 4477, Yaounde, Cameroon

Correspondence should be addressed to Yaé Ulrich Gaba; yaeulrich.gaba@gmail.com

Received 27 August 2021; Accepted 12 October 2021; Published 30 October 2021

Academic Editor: Hao Gao

Copyright © 2021 Naila Rafiq et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The nonlinear equation is a fundamentally important area of study in mathematics, and the numerical solutions of the nonlinear equations are also an important part of it. Fuzzy sets introduced by Zadeh are an extension of classical sets, which have several applications in engineering, medicine, economics, finance, artificial intelligence, decision-making, and so on. The most special types of fuzzy sets are fuzzy numbers. The important fuzzy numbers are trapezoidal fuzzy and triangular fuzzy numbers, which have several applications. In this research article, we propose an efficient numerical iterative method for estimating roots of fuzzy nonlinear equations, which are based on the special type of fuzzy number called triangular fuzzy number. Convergence analysis proves that the order of convergence of the numerical method is three. Some real-life applications are considered as numerical test problems from engineering, which contain fuzzy quantities in the parametric form. Engineering models include fractional conversion of nitrogen-hydrogen feed into ammonia and Van der Waal's equation for calculating the volume and pressure of a gas and motion of the object under constant force of gravity. Numerical illustrations are given to show the dominance efficiency of the newly constructed iterative schemes as compared to existing methods in the literature.

1. Introduction

One of the ancient problems of science and engineering in general and in mathematics in particular is to approximate roots of nonlinear equations. The nonlinear equations play a major role in the field of engineering, mathematics, physics, chemistry, economics, medicines, and finance. Many times the particular realization of such type of nonlinear problems involves imprecise and non-probabilistic uncertainties in the parameter, where the approximations are known due to expert knowledge or due to some experimental data. Due to these reasons, several real world applications contain vagueness and

uncertainties. Therefore, in most of real world problems, the parameters involved in the system or variables of the nonlinear equations are presented by a fuzzy number. The concepts of fuzzy numbers and arithmetic operation with fuzzy numbers were first introduced and investigated in [1–10]. Hence, it is necessary to approximate the root of fuzzy nonlinear equation:

$$F(x) = c. \quad (1)$$

The standard analytical technique like Buckley and Qu method [11–14] cannot be suitable for solving the nonlinear equations such as

$ax^6 + bx^4 - cx^3 + dx - e = f$, $x + \cos(x) = g$, $x \ln(x) + e^x - (1/(1+x^2)) + \tan(x) = h$, where a, b, c, d, e, f, g , and h are fuzzy numbers and x is a fuzzy variable.

We therefore look towards numerical iterative schemes, which approximate the roots of fuzzy nonlinear equations. To approximate the roots of fuzzy nonlinear equations, Abbsbandy and Asady [15] used Newton Raphson method, Sulaiman et al. [16] give Levenberg-Marquest method, and Mosleh [17] used Adomian decomposition method; see also [16, 18–20]. Iterative methods presented by them have low convergence order to approximate the roots of fuzzy nonlinear equations. Iterative methods of lower convergence order have high computational time and cost according to Kung–Traub conjecture [21].

Engineers and mathematicians therefore look towards those numerical methods which are more efficient and with high convergence order and low computational cost, or time, to approximate the roots of highly nonlinear fuzzy equations. The main aim of this research article is to propose efficient higher order iterative method as compared to well-known classical methods [15]. Numerical test results, CPU time, and log of residual show the dominance efficiency of our newly constructed method over the existing methods in the literature.

This paper is organized as follows: after introduction in Section 2, we recall some fundamental results of fuzzy numbers. In Section 3, we propose numerical iterative schemes for approximating roots of fuzzy nonlinear equations and their convergence analysis. In Section 4, we illustrate some real world applications as numerical test examples to show the performance and efficiency of the constructed method and conclusions in the last section.

2. Preliminaries

Definition 1. A fuzzy number is a fuzzy set like $x: \mathbb{R} \rightarrow I = [0, 1]$, which satisfies the following [22, 23]:

- (1) x is upper semicontinuous
- (2) $x(a) = 0$ outside some interval $[a_1, a_2]$
- (3) The real numbers are b_1, b_2 such that $a_1 \leq b_1 \leq b_2 \leq a_2$ and
 - $x(a)$ is monotonic increasing on $[a_1, b_1]$
 - $x(a)$ is monotonic decreasing on $[b_2, a_2]$
 - $x(a) = 1$, for $b_1 \leq a \leq b_2$

We denote by E the set of all fuzzy numbers. An equivalent parametric form is also given in [24] as follows.

Definition 2 (see [25]). A fuzzy number x in parametric form is a pair (x^L, x^U) of function $x^L(\tau)$, $x^U(\tau)$, $0 \leq \tau \leq 1$, which satisfies the following requirements:

- (1) $x^L(\tau)$ is a bounded monotonic increasing left continuous function
- (2) $x^U(\tau)$ is a bounded monotonic decreasing left continuous function
- (3) $x^L(\tau) \leq x^U(\tau)$, $0 \leq \tau \leq 1$

A popular fuzzy number is the triangular fuzzy number, which is formed by simply taking $b_1 = b_2$ in Definition 1. Triangular fuzzy number is simply written as $x = (a_1, a_2, a_3)$ and defined in the form of membership function as

$$x(a) = \begin{cases} \frac{x - a_1}{a_2 - a_1}, & \text{if } a_1 < x < a_2, \\ \frac{a_3 - x}{a_3 - a_2}, & \text{if } a_2 < x < a_3, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The parametric form is given as

$$\begin{aligned} x^L(\tau) &= a_1 + \tau(a_2 - a_1), \\ x^U(\tau) &= a_3 + \tau(a_2 - a_3). \end{aligned} \quad (3)$$

Let $\text{TF}(\mathbb{R})$ be the set of all triangular fuzzy numbers. The addition and scalar multiplication of fuzzy numbers are defined by the extension principle and represented as follows.

For arbitrary $x = (x^L, x^U)$, $y = (y^L, y^U)$, and $k > 0$, we define addition $(x + y)$ and multiplication by scalar k as

$$\begin{cases} (x + y)^L(\tau) = x^L(\tau) + y^L(\tau), \\ (x + y)^U(\tau) = x^U(\tau) + y^U(\tau), \\ (kx)^L(\tau) = kx^L(\tau), \\ (kx)^U(\tau) = kx^U(\tau). \end{cases} \quad (4)$$

3. Construction of Iterative Schemes

Now, our aim is to obtain a solution for fuzzy nonlinear equation $F(x) = c$. The parametric form is as follows:

$$\begin{cases} F^L(x^L, x^U, \tau) = c^L(\tau), \\ F^U(x^L, x^U, \tau) = c^U(\tau), \end{cases} \quad \forall \tau \in [0, 1]. \quad (5)$$

Suppose that $x = (\xi^L, \xi^U)$ is the solution to the system; that is,

$$\begin{cases} F^L(\xi^L, \xi^U, \tau) = c^L(\tau), \\ F^U(\xi^L, \xi^U, \tau) = c^U(\tau), \end{cases} \quad \forall \tau \in [0, 1]. \quad (6)$$

Therefore, if $x_0 = (x_0^L, x_0^U)$ is an approximation solution for this system, then $\forall \tau \in [0, 1]$ there exist $h^L(\tau)$, $h^U(\tau)$ such that

$$\begin{cases} \xi^L = x_0^L + h^L(\tau), \\ \xi^U = x_0^U + h^U(\tau). \end{cases} \quad (7)$$

Using Taylor series of F^L, F^U about (x_0^L, x_0^U) , $\forall \tau \in [0, 1]$, we have

$$\begin{cases} F^L(\xi^L, \xi^U, \tau) = F^L(x_0^L, x_0^U, \tau) + h^L F_{x^L}^L(x_0^L, x_0^U, \tau) + h^U F_{x^U}^L(x_0^L, x_0^U, \tau) + \\ O\left((h^L)^2 + h^L h^U + (h^U)^2\right) = c^L, \\ F^U(\xi^L, \xi^U, \tau) = F^U(x_0^L, x_0^U, \tau) + h^L F_{x^L}^U(x_0^L, x_0^U, \tau) + h^U F_{x^U}^U(x_0^L, x_0^U, \tau) + \\ O\left((h^L)^2 + h^L h^U + (h^U)^2\right) = c^U. \end{cases} \quad (8)$$

If x_0^L and x_0^U are near to ξ^L and ξ^U , respectively, then h^L and h^U are small. We assume, of course, that all needed

partial derivatives exist and are bounded. Therefore, for enough small h^L and h^U , we have, $\forall \tau \in [0, 1]$,

$$\begin{cases} F^L(x_0^L, x_0^U, \tau) + h^L F_{x^L}^L(x_0^L, x_0^U, \tau) + h^U F_{x^U}^L(x_0^L, x_0^U, \tau) = c^L(\tau), \\ F^U(x_0^L, x_0^U, \tau) + h^L F_{x^L}^U(x_0^L, x_0^U, \tau) + h^U F_{x^U}^U(x_0^L, x_0^U, \tau) = c^U(\tau), \end{cases} \quad (9)$$

and hence, $h^L(\tau)$ and $h^U(\tau)$ are unknown quantities that can be obtained by solving the following equations, $\forall \tau \in [0, 1]$,

$$J(x_0^L, x_0^U, \tau) \begin{bmatrix} h_0^L(\tau) \\ h_0^U(\tau) \end{bmatrix} = \begin{bmatrix} c^L(\tau) - F^L(x_0^L, x_0^U, \tau) \\ c^U(\tau) - F^U(x_0^L, x_0^U, \tau) \end{bmatrix}, \quad (10)$$

where

$$J(x_0^L, x_0^U, \tau) = \begin{bmatrix} F_{x^L}^L(x_0^L, x_0^U, \tau) & F_{x^U}^L(x_0^L, x_0^U, \tau) \\ F_{x^L}^U(x_0^L, x_0^U, \tau) & F_{x^U}^U(x_0^L, x_0^U, \tau) \end{bmatrix}. \quad (11)$$

Thus, our method in component form becomes

$$\begin{aligned} \begin{bmatrix} x_1^L(\tau) \\ x_1^U(\tau) \end{bmatrix} &= \begin{bmatrix} x_0^L(\tau) \\ x_0^U(\tau) \end{bmatrix} + \begin{pmatrix} ((2-\beta)J(x_0^L, x_0^U, \tau) + \beta K(y_0^L, y_0^U, \tau))^{-1} \\ * ((3-\beta)J(x_0^L, x_0^U, \tau) + (\beta-1)K(y_0^L, y_0^U, \tau)) \end{pmatrix} \\ &\quad \cdot \begin{bmatrix} c^L(\tau) - F_{x^L}^L(x_0^L, x_0^U, \tau) \\ c^U(\tau) - F_{x^L}^U(x_0^L, x_0^U, \tau) \end{bmatrix}, \\ \begin{bmatrix} y_0^L(\tau) \\ y_0^U(\tau) \end{bmatrix} &= \begin{bmatrix} x_0^L(\tau) \\ x_0^U(\tau) \end{bmatrix} + \begin{bmatrix} h_0^L(\tau) \\ h_0^U(\tau) \end{bmatrix}, \\ K(y_0^L, y_0^U, \tau) &= \begin{bmatrix} F_{y^L}^L(y_0^L, y_0^U, \tau) & F_{y^U}^L(y_0^L, y_0^U, \tau) \\ F_{y^L}^U(y_0^L, y_0^U, \tau) & F_{y^U}^U(y_0^L, y_0^U, \tau) \end{bmatrix}. \end{aligned} \quad (12)$$

For approximate solutions of x^L and x^U , we use the following recursive relation:

$$\begin{aligned} \begin{bmatrix} x_{n+1}^L(\tau) \\ x_{n+1}^U(\tau) \end{bmatrix} &= \begin{bmatrix} x_n^L(\tau) \\ x_n^U(\tau) \end{bmatrix} + \begin{pmatrix} ((2-\beta)J(x_n^L, x_n^U, \tau) + \beta K(y_n^L, y_n^U, \tau))^{-1} \\ * ((3-\beta)J(x_n^L, x_n^U, \tau) + (\beta-1)K(y_n^L, y_n^U, \tau)) \end{pmatrix} \\ &\quad \cdot \begin{bmatrix} c^L(\tau) - F_{x^L}^L(x_n^L, x_n^U, \tau) \\ c^U(\tau) - F_{x^L}^U(x_n^L, x_n^U, \tau) \end{bmatrix}, \end{aligned} \quad (13)$$

where

$$\begin{aligned} \begin{bmatrix} y_n^L(\tau) \\ y_n^U(\tau) \end{bmatrix} &= \begin{bmatrix} x_n^L(\tau) \\ x_n^U(\tau) \end{bmatrix} + \begin{bmatrix} h_n^L(\tau) \\ h_n^U(\tau) \end{bmatrix}, \\ J(x_n^L, x_n^U, \tau) \begin{bmatrix} h_n^L(\tau) \\ h_n^U(\tau) \end{bmatrix} &= \begin{bmatrix} F_{x^L}^L(x_n^L, x_n^U, \tau) & F_{x^U}^L(x_n^L, x_n^U, \tau) \\ F_{x^L}^U(x_n^L, x_n^U, \tau) & F_{x^U}^U(x_n^L, x_n^U, \tau) \end{bmatrix}, \\ K(y_n^L, y_n^U, \tau) &= \begin{bmatrix} F_{y^L}^L(y_n^L, y_n^U, \tau) & F_{y^U}^L(y_n^L, y_n^U, \tau) \\ F_{y^L}^U(y_n^L, y_n^U, \tau) & F_{y^U}^U(y_n^L, y_n^U, \tau) \end{bmatrix}, \\ &\forall \tau \in [0, 1]. \end{aligned} \quad (14)$$

For initial guess, one can use the fuzzy number

$$x_0 = (x_0^L(1), x_0^U(1), x_1^L(1) - x_0^L(0), x_0^U(0) - x_0^U(1)), \quad (15)$$

and in the parametric form

$$\begin{aligned} x_0(\tau) &= x_0^L(1) + (x_0^L(1) - x_0^L(0))(\tau - 1), \\ x_0^U(\tau) &= x_0^U(1) + (x_0^U(0) - x_0^U(1))(1 - \tau). \end{aligned} \quad (16)$$

Remark 1. Sequence $\{(x_n^L, x_n^U)\}_{n=0}^\infty$ converges to (ξ^L, ξ^U) iff $\forall \tau \in [0, 1]$, $\lim_{n \rightarrow \infty} x_n^L(\tau) = \xi^L(\tau)$ and $\lim_{n \rightarrow \infty} x_n^U(\tau) = \xi^U(\tau)$.

Lemma 1. Let $F(\xi^L, \xi^U) = (c^L, c^U)$ and if the sequence of $\{(x_n^L, x_n^U)\}_{n=0}^\infty$ converges to (ξ^L, ξ^U) to NM method, then

$$\lim_{n \rightarrow \infty} P_n = 0, \quad (17)$$

where

$$P_n = \sup_{0 \leq \tau \leq 1} \max\{h_n^L(\tau), h_n^U(\tau)\}. \quad (18)$$

Proof. It is obviously because $\forall \tau \in [0, 1]$ in convergent case

$$\lim_{n \rightarrow \infty} h_n^L(\tau) = \lim_{n \rightarrow \infty} h_n^U(\tau) = 0. \quad (19) \quad \square$$

Under certain condition, finally it is shown that NM method is cubic convergence for fuzzy nonlinear equation $F(x) = 0$. Thus, in compact form, we write

$$\begin{cases} y_n = x_n - (F'(x_n))^{-1}F(x_n), \\ x_{n+1} = x_n - Z * (F'(x_n))^{-1}F(x_n), \end{cases} \quad \forall \tau \in [0, 1], \quad (20)$$

where $Z = (((2 - \beta)F'(x_n^L, x_n^U, \tau) + \beta F'(y_n^L, y_n^U, \tau))^{-1} ((3 - \beta)F'(x_n^L, x_n^U, \tau) + (\beta - 1)F'(y_n^L, y_n^U, \tau)))$ and $\beta \in \mathbb{R}$ (set of real numbers).

Theorem 1. Let, $\forall \tau \in [0, 1]$, the functions F^L and F^U be continuously differentiable with respect to $x_n^L(\tau)$ and $x_n^U(\tau)$. Assume that there exist $(\xi^L(\tau), \xi^U(\tau)) \in \mathbb{R}^2$ and $\alpha_1, \alpha_2 > 0$ such that $\|J^{-1}(\xi^L, \xi^U, \tau)\| \leq \alpha_1$, $\|K^{-1}(\xi^L, \xi^U, \tau)^{-1}\| \leq \alpha_2$ and

J, K will be Lipschitz continuous with respect to x_n^L and x_n^U ; then the NM method converges to (ξ^L, ξ^U) and satisfies the following error equation:

$$\mathbf{e}_{n+1} = \left(2(\mathbf{A}_2)^2 + \frac{1}{2}\mathbf{A}_3 - \beta(\mathbf{A}_2)^2\right)(\mathbf{e}_n)^3 + \|O(\mathbf{e}_n)^4\|, \quad (21)$$

where $\mathbf{A}_j = (1/j!) * ((F^{(j)}(x_n, \tau))/F'(x_n, \tau))$, $j = 2, 3, \dots$

Proof. Let $\mathbf{e}_n = x_n - \xi$ and $\mathbf{e}_{n+1} = x_{n+1} - \xi$ be the errors in x_n and x_{n+1} ; then, by Taylor series of $F(x_n, \tau)$ in the neighborhood of ξ , if $J^{-1}(x_n, \tau)$ exist, then

$$\begin{aligned} F(x, \tau) &= F(x_n, \tau) + F'(x_n, \tau)(x - x_n) \\ &\quad + \frac{1}{2!}F''(x_n, \tau)(x - x_n)^2 + \dots, \end{aligned} \quad (22)$$

and $F(\xi, \tau) = 0$,

$$F(x_n, \tau) = F'(x, \xi)(\mathbf{e}_n + \mathbf{A}_2(\mathbf{e}_n)^2 + \mathbf{A}_3(\mathbf{e}_n)^3) + \|O(\mathbf{e}_n)^4\|. \quad (23)$$

This gives

$$\begin{aligned} (F'(x_n, \tau))^{-1}F(x_n, \tau) &= \mathbf{e}_n + \mathbf{A}_2(\mathbf{e}_n)^2 \\ &\quad + (2\mathbf{A}_2 + 2\mathbf{A}_3)(\mathbf{e}_n)^3 + \dots, \\ y_n - \xi &= \mathbf{A}_2(\mathbf{e}_n)^2 + (-2\mathbf{A}_2 \\ &\quad \cdot + 2\mathbf{A}_3)(\mathbf{e}_n)^3 + \dots. \end{aligned} \quad (24)$$

Expanding $F'(y_n, \tau)$ about ξ , we have

$$\begin{aligned} F'(y_n, \tau) &= 1 + 2(\mathbf{A}_2)^2(\mathbf{e}_n)^2 + 2(-2(\mathbf{A}_2)^2 \\ &\quad + 2\mathbf{A}_3)(\mathbf{e}_n)^3 + \dots, \end{aligned}$$

$$Z(F'(x_n, \tau))^{-1}F(x_n, \tau) = (\mathbf{e}_n)^2 + \left(-2(\mathbf{A}_2)^2 - \frac{1}{2}\mathbf{A}_3\right.$$

$$\left. + \beta(\mathbf{A}_2)^2\right)(\mathbf{e}_n)^3 + \dots,$$

$$x_{n+1} - \xi = x_n - \xi - (\mathbf{e}_n)^2 + (-2(\mathbf{A}_2)^2$$

$$- \frac{1}{2}\mathbf{A}_3 + \beta(\mathbf{A}_2)^2)(\mathbf{e}_n)^3 + \dots,$$

$$\begin{aligned} \mathbf{e}_{n+1} &= \left(2(\mathbf{A}_2)^2 + \frac{1}{2}\mathbf{A}_3 - \beta(\mathbf{A}_2)^2\right)(\mathbf{e}_n)^3 \\ &\quad + \|O(\mathbf{e}_n)^4\|. \end{aligned} \quad (25)$$

Hence, the theorem is proved. \square

There are some well-known existing methods in the literature for solving triangular fuzzy nonlinear equations.

Fuzzy version of well-known Newton method [15] (abbreviated as NR) for finding roots of triangular fuzzy equation is as follows:

$$\begin{bmatrix} x_{n+1}^L(\tau) \\ x_{n+1}^U(\tau) \end{bmatrix} = \begin{bmatrix} x_n^L(\tau) \\ x_n^U(\tau) \end{bmatrix} + \begin{bmatrix} h_n^L(\tau) \\ h_n^U(\tau) \end{bmatrix}, \quad (26)$$

where

$$\begin{bmatrix} h_n^L(\tau) \\ h_n^U(\tau) \end{bmatrix} = J^{-1}(x_n^L, x_n^U, \tau) \begin{bmatrix} c^L(\tau) - F^L(x_n^L, x_n^U, \tau) \\ c^U(\tau) - F^U(x_n^L, x_n^U, \tau) \end{bmatrix}. \quad (27)$$

Midpoint iterative schemes [26] (abbreviated as NP) for triangular fuzzy equation are as follows:

$$\begin{bmatrix} x_{n+1}^L(\tau) \\ x_{n+1}^U(\tau) \end{bmatrix} = \begin{bmatrix} y_n^L(\tau) \\ y_n^U(\tau) \end{bmatrix} + J^{-1}\left(\frac{x_n^L + y_n^L}{2}, \frac{y_n^U + x_n^U}{2}, \tau\right) \cdot \begin{bmatrix} c^L(\tau) - F^L(x_n^L, x_n^U, \tau) \\ c^U(\tau) - F^U(x_n^L, x_n^U, \tau) \end{bmatrix}, \quad (28)$$

where

$$\begin{bmatrix} y_n^L(\tau) \\ y_n^U(\tau) \end{bmatrix} = \begin{bmatrix} x_n^L(\tau) \\ x_n^U(\tau) \end{bmatrix} + \begin{bmatrix} h_n^L(\tau) \\ h_n^U(\tau) \end{bmatrix}. \quad (29)$$

4. Numerical Applications

Here, we present examples to illustrating Newton's method for fuzzy nonlinear equations. Examples 1 and 2 are considered from Buckley and Qu [11, 27, 28]. All the

computations are performed using CAS Maple 18 with 64-digit floating point arithmetic with stopping criteria as follows:

$$\begin{aligned} (i) \mathbf{e}_n &= \|F(x, \tau)\| < \epsilon, \\ (ii) \mathbf{e}_n &= \|\mathbf{x}_{n+1}(\tau) - \mathbf{x}_n(\tau)\| < \epsilon, \end{aligned} \quad (30)$$

where \mathbf{e}_n represents the absolute error. We take $\epsilon = 10^{-15}$. In all numerical calculations, we used $\beta = 0.000001$.

Figure 1 shows computational time in seconds of iterative schemes NM, NR, and NP for nonlinear fuzzy equations in Examples 1–3 (Cases 1 and 2) and 4, respectively (Algorithm 1).

4.1. Engineering Applications

Example 1. Fraction conversion of Nitrogen-Hydrogen feed to ammonia is known as fractional conversion. Here, we consider the value of temperature and pressure as triangular fuzzy number, which results in the following fuzzy nonlinear equation:

$$\nabla_1(x(\tau))^4 - \nabla_2(x(\tau))^3 + \nabla_3(x(\tau))^2 + \nabla_4x(\tau) = \nabla_5, \quad (31)$$

where $\nabla_1 = (1, 2.5, 3.7)$, $\nabla_2 = (7.7, 7.9, 8.1)$, $\nabla_3 = (14.7, 14.8, 14.9)$, $\nabla_4 = (2.5, 2.7, 2.9)$, and $\nabla_5 = (1.6, 1.7, 2.0)$ are triangular fuzzy numbers. Without any loss of generality, assume that x is positive; then the parametric form of this equation is as follows:

$$\begin{cases} (1 + 1.5\tau)(x^L(\tau))^4 - (7.7 + 0.2\tau)(x^L(\tau))^3 + (14.7 + 0.1\tau)(x^L(\tau))^2 + (2.5 + 0.2\tau)x^L(\tau) = (1.6 + 0.1\tau), \\ (3.7 - 1.2\tau)(x^U(\tau))^4 - (8.1 - 0.2\tau)(x^U(\tau))^3 + (14.9 - 0.1\tau)x^U(\tau) + (2.9 - 0.2\tau)x^U(\tau) = (2 - 0.3\tau). \end{cases} \quad (32)$$

Figure 2 shows analytical and numerical approximate solution of fuzzy nonlinear equation in Example 1.

Table 1 clearly shows the dominance behavior of NM over NR and NP in terms of absolute error on the same number of iterations $n = 3$ for Example 1.

To obtain initial guess, we use the above system for $\tau = 1$ and $\tau = 0$; therefore,

$$\begin{cases} 2.5(x^L)^4(1) - 7.9(x^L)^2(1) + 14.8(x^L)^2(1) + 2.7(x^L)(1) = 1.7, \\ 2.5(x^U)^4(1) - 7.9(x^U)^2(1) + 14.8(x^U)^2(1) + 2.7(x^U)(1) = 1.7, \\ 1(x^L)^4(0) - 7.7(x^L)^3(0) + 14.7(x^L)^2(0) + 2.5(x^L)(0) = 1.6, \\ 3.7(x^U)^4(0) - 8.1(x^U)^2(0) + 14.9(x^U)(0) + 2.9(x^U)(0) = 2. \end{cases} \quad (33)$$

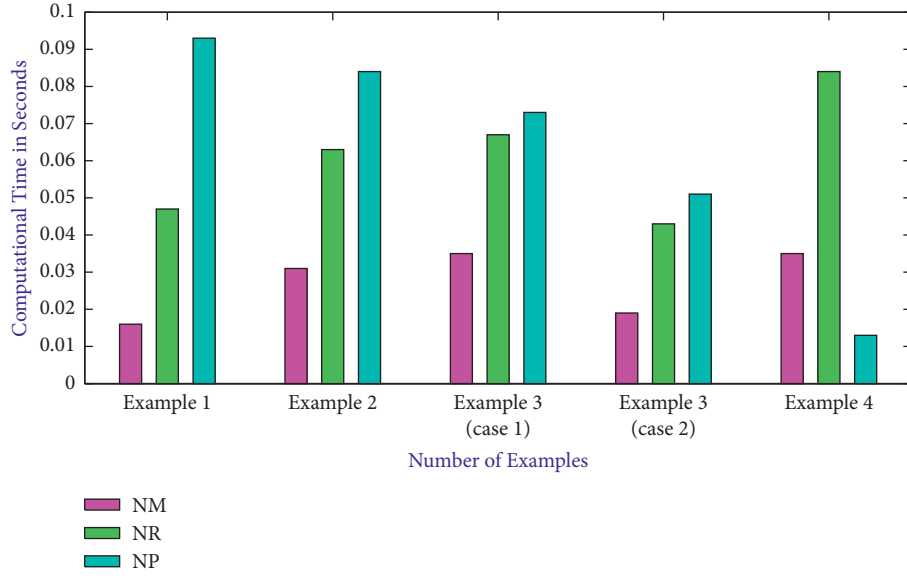


FIGURE 1: Computational time in seconds.

Step 1: transform $F(x, \tau) = c$ into $\begin{cases} F^L(x^L, x^U, \tau) = c^L(\tau), \\ F^U(x^L, x^U, \tau) = c^U(\tau), \end{cases} \forall \tau \in [0, 1].$
 Step 2: solve $\begin{cases} F^L(x^L, x^U, \tau) = c^L(\tau), \\ F^U(x^L, x^U, \tau) = c^U(\tau). \end{cases}$ for $\tau = 0$ and $\tau = 1$ to obtain $x^L(0)$ and $x^U(0)$.
 Step 3: evaluate $F(x, \tau) = c$ at $x^L(0)$ and $x^U(0)$ and compute Jacobian matrix $J(x^L, x^U, \tau)$.
 Step 4: use NM to compute next iterations.
 Step 5: for given $\epsilon > 0$, (i) $\mathbf{e}_n = \|F(x, \tau)\| < \epsilon$ (ii) $\mathbf{e}_n = \|x_{n+1}(\tau) - x_n(\tau)\| < \epsilon$, then stop.
 Step 6: set $k = k + 1$ and go to Step 1.

ALGORITHM 1: NM method.

Consequently $x^L(0) = 0.2$, $x^U(0) = 0.2$, and $x^L(1) = x^U(1) = 0.2737$. Therefore, initial guess is $x_0 = (0.26, 0.2737, 0.29)$. After 3 iterations, we obtain the

solution which the maximum error would be less than 10^{-3} . Now suppose x is negative; we have

$$\begin{cases} (1 + 1.5\tau)(x^L(\tau))^4 - (7.7 + 0.2\tau)(x^L(\tau))^3 + (14.7 + 0.1\tau)(x^L(\tau))^2 + (2.5 + 0.2\tau)x^L(\tau) = (1.6 + 0.1\tau), \\ (3.7 - 1.2\tau)(x^U(\tau))^4 - (8.1 - 0.2\tau)(x^U(\tau))^3 + (14.9 - 0.1\tau)x^U(\tau) + (2.9 - 0.2\tau)x^U(\tau) = (2 - 0.3\tau). \end{cases} \quad (34)$$

For $\tau = 0$, we have $x^U(0) > x^L(0)$, and therefore, negative root does not exist.

Example 2. A chemical engineering problem is finding the volume of van der Waal's equation. Van der Waal's equation interprets real and ideal gas behavior, which results in the following equation:

$$\left(P + \frac{A_1 n^2}{x^2}\right)(x - nA_2) = nRT. \quad (35)$$

By using the specific values of parameter, we have the following fuzzy nonlinear equation:

$$\Delta_1(x(\tau))^3 - \Delta_2(x(\tau))^2 + \Delta_3 x(\tau) = \Delta_4, \quad (36)$$

where $\Delta_1 = (0.3, 0.4, 0.7)$, $\Delta_2 = (1, 2.3, 3.4)$, $\Delta_3 = (6.8, 7, 8.9)$, $\Delta_4 = (4.1, 5.2, 6.7)$ are triangular fuzzy numbers and x represents the volume of the gas under observation, P for pressure, R for general gas constant, n for number of mole, and T for temperature, and A_1, A_2 are used for general parameters. Without any loss of generality,

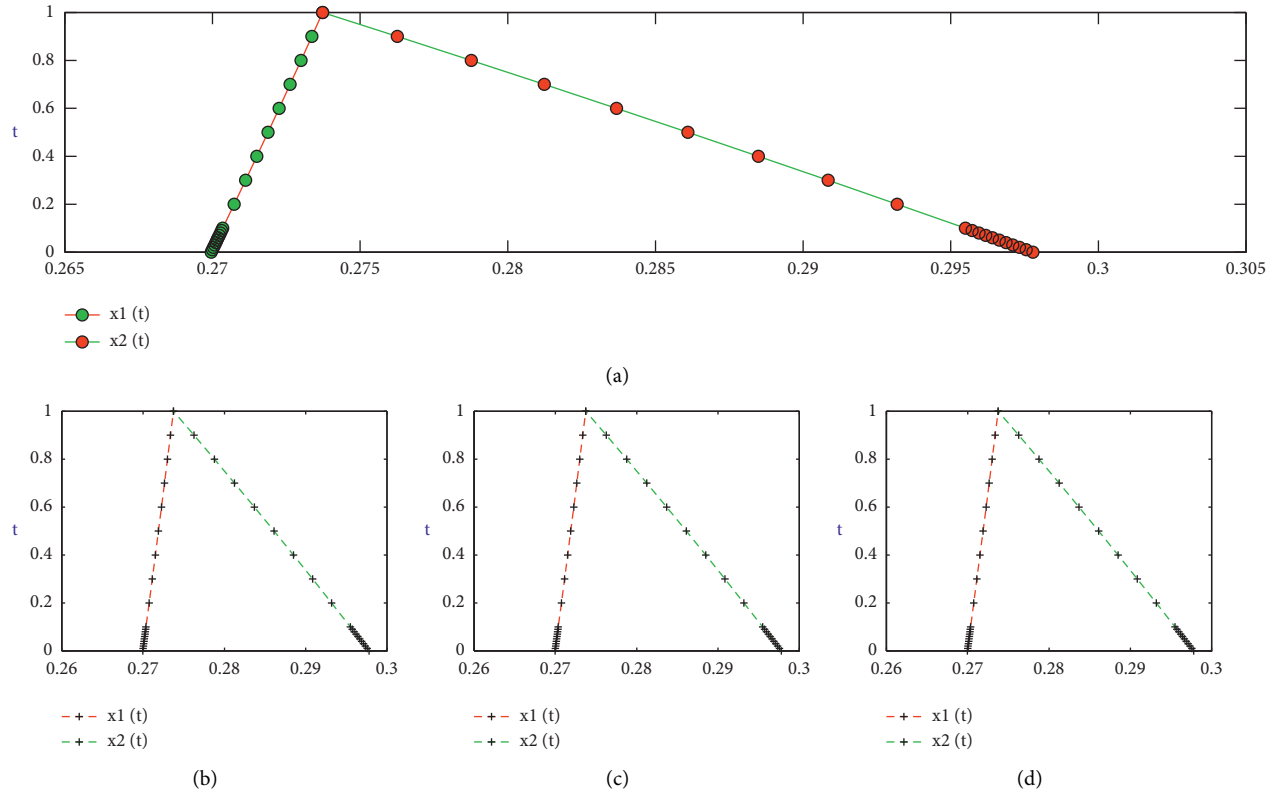


FIGURE 2: Analytical and numerical approximate solutions of Example 1. (a) Analytical solution for Example 1. (b) NM method for Example 1. (c) NR method for Example 1. (d) NP method for Example 1.

TABLE 1: Comparison of iterative schemes NM, NR, and NP on the same number of iterations $n = 3$.

τ	x^L	x^U	NM		NR		NP	
			$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $	$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $	$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $
0.0	0.2699	0.2977	$3.9e-19$	$6.9e-20$	$0.3e-5$	$2.7e-4$	$6.3e-8$	$7.3e-9$
0.1	0.2703	0.2954	$1.3e-19$	$1.3e-18$	$6.2e-5$	$5.2e-6$	$6.2e-9$	$5.9e-8$
0.2	0.2707	0.2931	$2.1e-17$	$7.1e-18$	$5.2e-5$	$7.4e-5$	$1.6e-9$	$6.4e-7$
0.3	0.2711	0.2908	$9.7e-15$	$1.8e-19$	$6.4e-6$	$4.6e-4$	$8.8e-9$	$4.3e-8$
0.4	0.2715	0.2884	$9.5e-16$	$8.5e-16$	$6.1e-5$	$5.3e-6$	$7.8e-9$	$1.4e-9$
0.5	0.2718	0.2860	$1.4e-15$	$1.9e-15$	$5.7e-7$	$8.3e-5$	$7.7e-9$	$0.5e-8$
0.6	0.2722	0.2836	$4.0e-16$	$6.9e-19$	$7.5e-5$	$3.4e-4$	$5.5e-8$	$3.8e-7$
0.7	0.2726	0.2812	$7.6e-17$	$3.9e-18$	$8.1e-6$	$4.8e-3$	$8.7e-9$	$1.2e-7$
0.8	0.2729	0.2787	$0.4e-18$	$8.2e-19$	$5.8e-7$	$0.4e-6$	$8.9e-8$	$2.6e-9$
0.9	0.2733	0.2762	$2.6e-17$	$6.1e-18$	$3.8e-5$	$1.3e-8$	$4.8e-9$	$2.3e-9$
1	0.2737	0.2737	$9.6e-19$	$1.2e-17$	$7.9e-5$	$7.8e-7$	$9.8e-8$	$1.8e-8$

assume that $x(\tau)$ is positive; then the parametric form of this equation is as follows:

$$\begin{cases} (0.3 + 0.1\tau)(x^L(\tau))^3 - (1 + 1.3\tau)(x^L(\tau))^2 + (6.8 + 0.2\tau)x^L(\tau) = (4.1 + 1.1\tau), \\ (0.7 - 0.3\tau)(x^U(\tau))^3 - (3.4 - 1.1\tau)(x^U(\tau))^2 + (8.9 - 1.9\tau)x^U(\tau) = (6.7 - 1.5\tau). \end{cases} \quad (37)$$

Figure 3 shows analytical and numerical approximate solution of fuzzy nonlinear equation in Example 2.

Figure 3 shows analytical and numerical approximate solution of fuzzy nonlinear equation in Example 2.

Table 2 clearly shows the dominance behavior of NM over NR and NP in terms of absolute error on the same number of iterations $n = 3$ for Example 2.

To obtain initial guess, we use the above system for $\tau = 1$ and $\tau = 0$; therefore

$$\begin{cases} 0.4(x^L)^3(1) - 2.3(x^L)^2(1) + 7.0(x^L)(1) = 5.2, \\ 0.4(x^U)^3(1) - 2.3(x^U)^2(1) + 7.0(x^U)(1) = 5.2, \\ 0.3(x^L)^3(0) - 1.0(x^L)^2(0) + 6.8(x^L)(0) = 4.1, \\ 0.7(x^U)^3(0) - 3.4(x^U)^2(0) + 8.9(x^U)(0) = 6.7. \end{cases} \quad (38)$$

$$\begin{cases} (0.3 + 0.1\tau)(x^L(\tau))^3 - (1 + 1.3\tau)(x^L(\tau))^2 + (6.8 + 0.2\tau)x^L(\tau) = (4.1 + 1.1\tau), \\ (0.7 - 0.3\tau)(x^U(\tau))^3 - (3.4 - 1.1\tau)(x^U(\tau))^2 + (8.9 - 1.9\tau)x^L(\tau) = (6.7 - 1.5\tau). \end{cases} \quad (39)$$

For $\tau = 0$, we have $x^U(0) > x^L(0)$, and therefore, negative root does not exist.

Example 3. In engineering, the problem concerns the motion of object under fuzzy environment, resulting in two cases.

Case 1. Vertical motion of object.

Here, we are concerned with the vertical motion of ball by neglecting air resistance and assume constant acceleration 32 ft/sec^2 . Positive direction of the object y is upward from earth. Then, the resulting fuzzy nonlinear equation is written as

$$A_1 * x(\tau)^2 + V_1 * x(\tau) = Y_1, \quad (40)$$

where $A_1 = (0.8, 1.0, 1.2)$, free parameter, $V_1 = (1.3, 1.75, 2.2)$, velocity of the moving object under constant force of gravity, and $Y_1 = (0.1, 0.15, 0.20)$, initial position of the object. By substituting the value of A_1, V_1 , and Y_1 in the above equation of motion, we get

$$(0.8, 1.0, 1.2)(x(\tau)^2) + (1.3, 1.75, 2.2)x(\tau) = (0.1, 0.15, 0.20). \quad (41)$$

Then, we find the time $x(\tau)$ of the ball to hit the ground which depends on fuzzy parameter $\tau \in [0, 1]$. Without any loss of generality, assume that $x(\tau)$ is positive; then the parametric form of this equation is as follows:

$$\begin{cases} (0.8 + 0.2\tau)(x^L(\tau))^2 + (1.3 + 0.45\tau)x^L(\tau) = (0.1 + 0.05\tau), \\ (1.2 - 0.2\tau)(x^U(\tau))^2 + (2.2 - 0.45\tau)x^U(\tau) = (0.2 - 0.05\tau). \end{cases} \quad (42)$$

Figure 4 shows analytical and numerical approximate solution of fuzzy nonlinear equation in Example 3 (Case 1).

Figure 4 shows analytical and numerical approximate solution of fuzzy nonlinear equation in Example 3 (Case 1).

Consequently $x^L(0) = 0.6$, $x^U(0) = 1.1$, and $x^L(1) = x^U(1) = 1.028015150$. Therefore, initial guess is $x_0 = (0.6, 1.028015150, 1.1)$. After 3 iterations, we obtain the solution which the maximum error would be less than 10^{-3} . Now suppose $x(\tau)$ is negative; we have

Table 3 clearly shows the dominance behavior of NM over NR and NP in terms of absolute error on the same number of iterations $n = 3$ for Example 3 (Case 1).

To obtain initial guess, we use the above system for $\tau = 0$ and $\tau = 1$; therefore

$$\begin{cases} 1(x^L)^2(1) + 1.75(x^L)(1) = 0.6, \\ 1(x^U)^2(1) + 1.75(x^U)(1) = 0.15, \\ 0.8(x^L)^2(0) + 1.3(x^L)(0) = 0.1, \\ 1.2(x^U)^2(0) + 2.2(x^U)(0) = 0.2. \end{cases} \quad (43)$$

Consequently $x^L(0) = 0.0735$, $x^U(0) = 0.0867$, and $x^L(1) = x^U(1) = 0.0818$. Therefore, initial guess is $x_0 = (0.0735, 0.0818, 0.0867)$. After 3 iterations, we obtain the solution which the maximum error would be less than 10^{-3} . Now suppose x is negative; we have

$$\begin{cases} (0.8 + 0.2\tau)(x^L(\tau))^2 + (1.3 + 0.45\tau)x^L(\tau) = (0.1 + 0.05\tau), \\ (1.2 - 0.2\tau)(x^U(\tau))^2 + (2.2 - 0.45\tau)x^U(\tau) = (0.2 - 0.05\tau). \end{cases} \quad (44)$$

For $\tau = 0$, we have $x^U(0) > x^L(0)$, and therefore, negative root does not exist.

Case 2. Downward motion of object.

Here, we discuss the downward motion of object which results in the following fuzzy nonlinear equation:

$$(3, 4, 5)(x(\tau))^2 + (1, 2, 3)x(\tau) = (1, 2, 3). \quad (45)$$

Without any loss of generality, assume that x is positive; then the parametric form of this equation is as follows:

$$\begin{cases} (3 + \tau)(x^L(\tau))^2 + (1 + \tau)x^L(\tau) = (1 + \tau), \\ (5 - \tau)(x^U(\tau))^2 + (3 - \tau)x^U(\tau) = (3 - \tau). \end{cases} \quad (46)$$

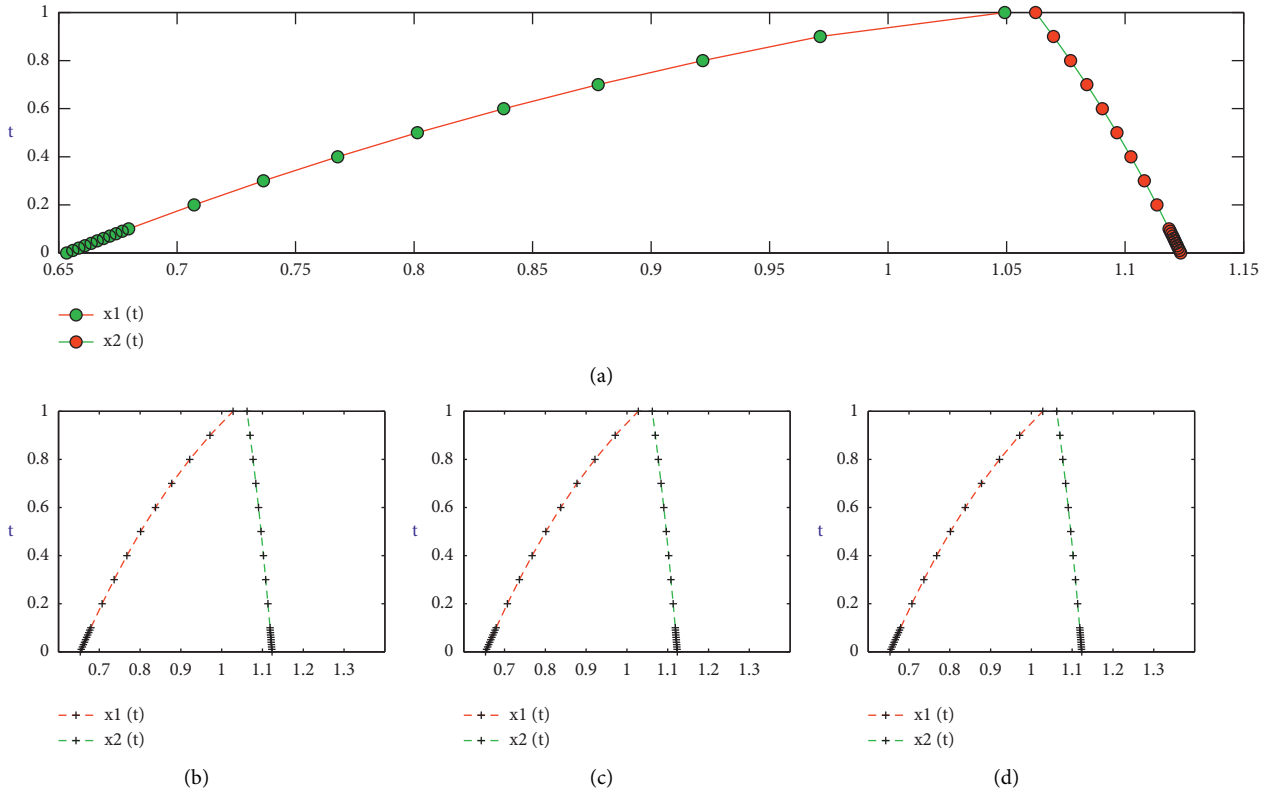


FIGURE 3: Analytical and numerical approximate solution of Example 2. (a) Analytical solution for Example 2. (b) NM method for Example 2. (c) NR method for Example 2. (d) NP method for Example 2.

TABLE 2: Comparison of iterative schemes NM, NR, and NP on the same number of iterations $n = 3$.

τ	x^L	x^U	NM		NR		NP	
			$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $	$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $	$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $
0.0	0.6534	1.1234	$3.9e-17$	$2.9e-16$	$7.3e-5$	$3.7e-3$	$5.3e-9$	$6.3e-8$
0.1	0.6795	1.1185	$1.3e-15$	$1.3e-16$	$1.2e-3$	$8.2e-6$	$4.2e-9$	$6.9e-8$
0.2	0.7071	1.1134	$3.1e-16$	$4.1e-16$	$1.2e-5$	$7.6e-6$	$5.6e-9$	$8.4e-9$
0.3	0.7364	1.1080	$2.7e-15$	$1.3e-17$	$4.4e-3$	$1.6e-4$	$8.8e-8$	$3.3e-8$
0.4	0.7677	1.1024	$3.5e-15$	$8.3e-13$	$6.0e-4$	$8.3e-4$	$7.2e-9$	$5.4e-9$
0.5	0.8013	1.0965	$1.4e-14$	$1.9e-15$	$6.7e-6$	$0.3e-5$	$7.7e-7$	$8.5e-9$
0.6	0.8378	1.0903	$4.0e-16$	$7.9e-17$	$7.5e-5$	$1.4e-6$	$5.2e-7$	$9.8e-8$
0.7	0.8776	1.0838	$7.6e-16$	$3.9e-18$	$8.6e-6$	$0.8e-6$	$8.7e-9$	$2.2e-8$
0.8	0.9218	1.0770	$8.4e-18$	$8.2e-16$	$6.8e-6$	$4.4e-6$	$3.9e-7$	$0.6e-9$
0.9	0.9713	1.0689	$9.6e-17$	$2.1e-17$	$4.8e-5$	$5.3e-7$	$4.1e-9$	$1.3e-9$
1	1.0280	1.0280	$9.0e-18$	$1.2e-17$	$7.9e-5$	$7.8e-7$	$9.6e-7$	$1.0e-7$

Figure 5 shows analytical and numerical approximate solution of fuzzy nonlinear equation in Example 3 (Case 2).

Figure 5 shows analytical and numerical approximate solution of fuzzy nonlinear equation in Example 3 (Case 2).

Table 4 clearly shows the dominance behavior of NM over NR and NP in terms of absolute error on the same number of iterations $n = 3$ for Example 3 (Case 2).

To obtain initial guess, we use the above system for $\tau = 0$ and $\tau = 1$; therefore,

$$\begin{cases} 4(x^L)^2(1) + 2x^L(1) = 2, \\ 4(x^U)^2(1) + 2x^U(1) = 2, \\ 3(x^L)^2(0) + x^L(0) = 1, \\ 5(x^U)^2(0) + 3x^U(0) = 3. \end{cases} \tag{47}$$

Consequently $x^L(0) = 0.4343$, $x^U(0) = 0.5307$, and $x^L(1) = x^U(1) = 0.5$. Therefore, initial guess is

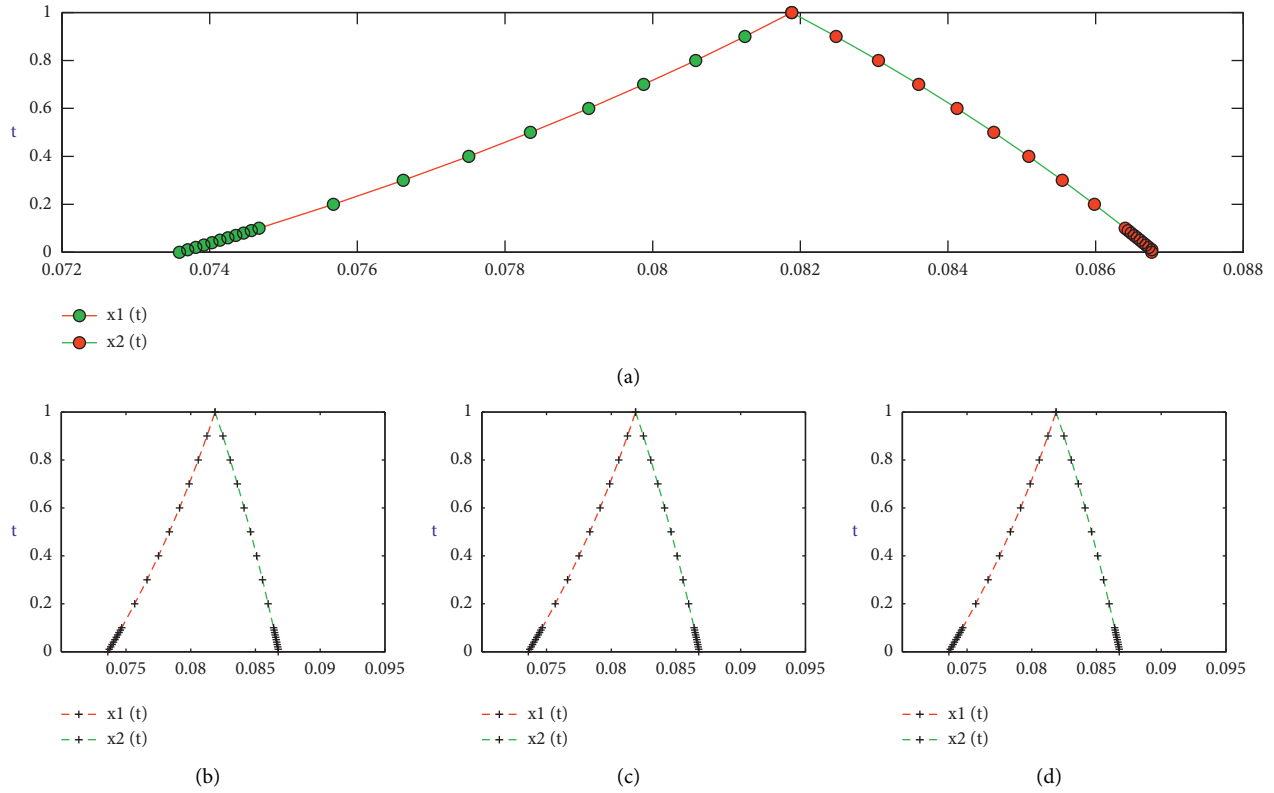


FIGURE 4: Analytical and numerical approximate solution of Example 3 (Case 1). (a) Analytical solution for Example 3 (Case 1). (b) NM method for Example 3 (Case 1). (c) NR method for Example 3 (Case 1). (d) NP method for Example 3 (Case 1).

TABLE 3: Comparison of iterative schemes NM, NR, and NP on the same number of iterations $n = 3$.

τ	x^L	x^U	NM		NR		NP	
			$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $	$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $	$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $
0.0	0.0735	0.0867	$4.9e-16$	$2.9e-16$	$7.3e-6$	$3.7e-6$	$1.4e-8$	$1.3e-8$
0.1	0.0746	0.0863	$2.3e-15$	$1.3e-15$	$1.2e-5$	$1.2e-5$	$3.2e-8$	$6.6e-8$
0.2	0.0756	0.0859	$3.1e-15$	$0.1e-16$	$1.2e-5$	$1.6e-5$	$5.6e-8$	$5.4e-8$
0.3	0.0766	0.0855	$1.7e-14$	$1.2e-15$	$2.4e-5$	$1.3e-5$	$8.8e-8$	$4.3e-8$
0.4	0.0775	0.0850	$3.5e-14$	$5.3e-14$	$3.0e-4$	$1.0e-4$	$1.2e-7$	$3.4e-8$
0.5	0.0783	0.0846	$6.4e-15$	$1.3e-15$	$3.7e-5$	$0.3e-5$	$1.7e-7$	$2.5e-8$
0.6	0.0791	0.0841	$1.0e-16$	$7.0e-16$	$4.5e-5$	$5.4e-5$	$2.2e-7$	$1.8e-8$
0.7	0.0798	0.0836	$1.6e-16$	$3.9e-16$	$5.6e-6$	$0.2e-6$	$2.7e-7$	$1.2e-8$
0.8	0.0805	0.0830	$2.4e-13$	$1.2e-16$	$4.8e-7$	$1.4e-7$	$3.3e-7$	$7.6e-9$
0.9	0.0812	0.0824	$3.6e-17$	$3.1e-17$	$3.8e-6$	$2.3e-6$	$4.0e-7$	$1.4e-9$
1	0.0818	0.0818	$7.0e-18$	$1.0e-18$	$7.0e-6$	$1.8e-6$	$4.6e-7$	$1.8e-9$

$x_0 = (0.4343, 0.5, 0.5307)$. After 3 iterations, we obtain the solution which the maximum error would be less than 10^{-3} . Now suppose x is negative; we have

$$\begin{cases} (3 + \tau)(x^L(\tau))^2 + (1 + \tau)x^L(\tau) = (1 + \tau), \\ (5 - \tau)(x^U(\tau))^2 + (3 - \tau)x^U(\tau) = (3 - \tau). \end{cases} \quad (48)$$

For $\tau = 0$, we have $x^U(0) > x^L(0)$, and therefore, negative root does not exist.

Example 4. Consider the fuzzy nonlinear equation:

$$(3, 4, 5)(x(\tau))^2 + (1, 2, 3)\sin(x(\tau)) = (1, 2, 3). \quad (49)$$

Without any loss of generality, assume that x is positive; then the parametric form of this equation is as follows:

$$\begin{cases} (3 + \tau)(x^L)^2 + (1 + \tau)\sin(x^L(\tau)) = (1 + \tau), \\ (5 - \tau)(x^U)^2 + (3 - \tau)\sin(x^U(\tau)) = (3 - \tau). \end{cases} \quad (50)$$

Figure 6 shows analytical and numerical approximate solution of fuzzy nonlinear equation in Example 4.

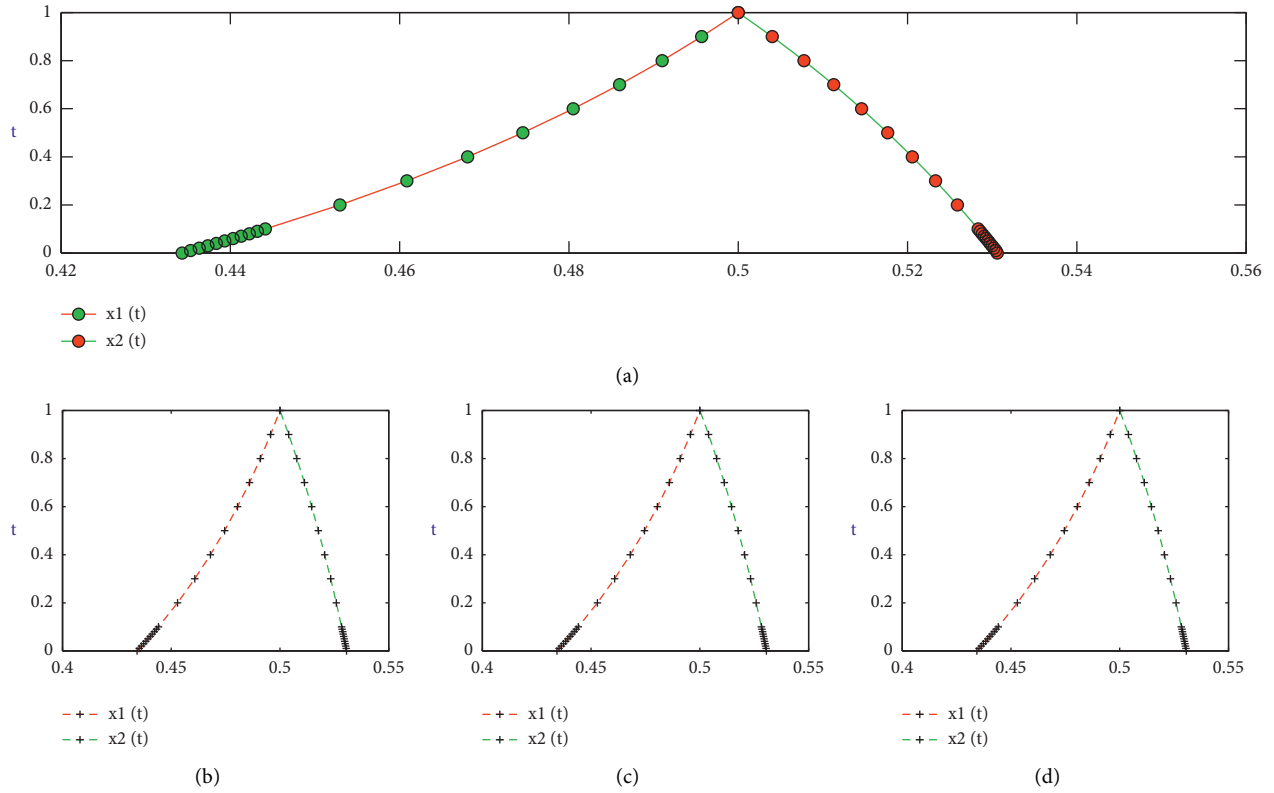


FIGURE 5: Analytical and numerical approximate solution of Example 3 (Case 2). (a) Analytical solution for Example 3 (Case 2). (b) NM method for Example 3 (Case 2). (c) NR method for Example 3 (Case 2). (d) NP method for Example 3 (Case 2).

TABLE 4: Comparison of iterative schemes NM, NR, and NP on the same number of iterations $n = 3$.

τ	x^L	x^U	NM		NR		NP	
			$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $	$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $	$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $
0.0	0.4343	0.5306	$1.4e-15$	$1.1e-29$	$3.0e-4$	$2.0e-7$	$4.4e-9$	$5.3e-16$
0.1	0.4441	0.5258	$6.9e-24$	$2.1e-29$	$2.0e-6$	$1.5e-7$	$1.5e-13$	$2.1e-16$
0.2	0.4529	0.5233	$1.3e-22$	$4.8e-30$	$4.4e-6$	$1.0e-7$	$6.5e-13$	$2.2e-16$
0.3	0.4608	0.5205	$1.1e-21$	$7.2e-31$	$7.2e-5$	$7.3e-8$	$1.9e-12$	$0.1e-20$
0.4	0.4680	0.5176	$6.1e-21$	$0.1e-32$	$1.0e-5$	$4.5e-8$	$4.5e-12$	$6.0e-17$
0.5	0.4745	0.5145	$2.6e-20$	$8.1e-35$	$1.4e-5$	$2.5e-8$	$9.0e-12$	$3.1e-17$
0.6	0.4805	0.5144	$8.3e-20$	$8.7e-33$	$1.9e-5$	$1.1e-8$	$1.5e-11$	$1.0e-16$
0.7	0.4859	0.5112	$2.1e-19$	$9.4e-33$	$2.3e-5$	$4.4e-9$	$2.5e-11$	$6.6e-17$
0.8	0.4910	0.5077	$4.9e-19$	$3.0e-33$	$2.8e-5$	$1.0e-9$	$3.8e-11$	$1.1e-15$
0.9	0.4956	0.5040	$9.9e-19$	$0.1e-36$	$3.3e-5$	$7.5e-11$	$5.3e-11$	$2.1e-16$
1	0.5	0.5	$1.8e-18$	$2.0e-36$	$3.9e-5$	$0.1e-15$	$7.2e-11$	$0.1e-18$

Figure 6 shows analytical and numerical approximate solution of fuzzy nonlinear equation in Example 4.

Table 5 clearly shows the dominance behavior of NM over NR and NP in terms of absolute error on the same number of iterations $n = 3$ for Example 4.

To obtain initial guess, we use the above system for $\tau = 0$ and $\tau = 1$; therefore,

$$\begin{cases} 4(x^L)^2(1) + 2 \sin(x^L(1)) = 2, \\ 4(x^U)^2(1) + 2 \sin(x^U(1)) = 2, \\ 3(x^L)^2(0) + \sin(x^L(0)) = 1, \\ 5(x^U)^2(0) + 3 \sin(x^U(0)) = 3. \end{cases} \quad (51)$$

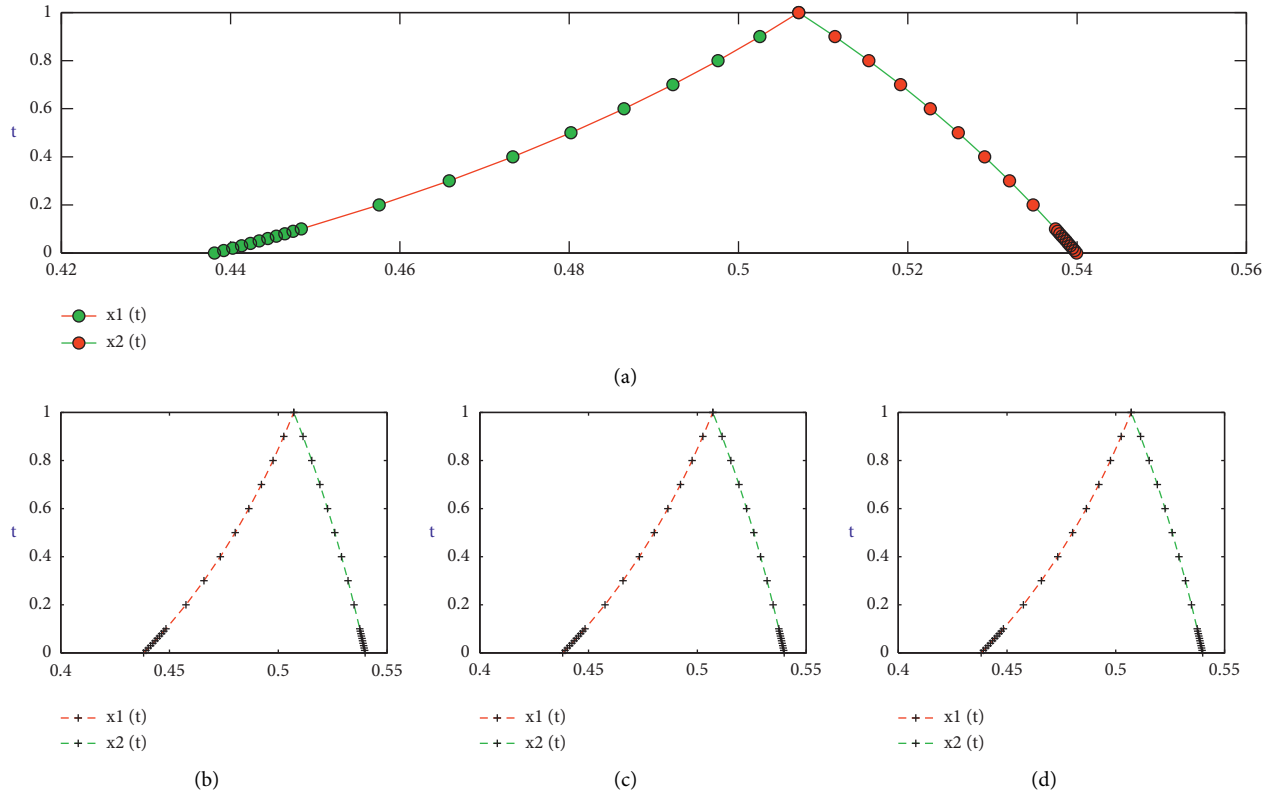


FIGURE 6: Analytical and numerical approximate solution of Example 4. (a) Analytical solution for Example 4. (b) NM method for Example 4. (c) NR method for Example 4. (d) NP method for Example 4.

TABLE 5: Comparison of iterative schemes NM, NR, and NP on the same number of iterations $n = 3$.

τ	x^L	x^U	NM		NR		NP	
			$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $	$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $	$\ x_{n+1} - x_n\ $	$\ f(x_n)\ $
0.0	0.4380	0.5399	$2.7e-25$	$4.7e-28$	$3.6e-6$	$2.4e-7$	$1.1e-14$	$4.2e-15$
0.1	0.4483	0.5377	$1.2e-23$	$1.7e-28$	$2.4e-6$	$1.4e-7$	$2.8e-13$	$3.3e-15$
0.2	0.4575	0.5374	$1.8e-22$	$5.3e-29$	$9.5e-6$	$6.4e-7$	$5.2e-13$	$2.5e-16$
0.3	0.4658	0.5348	$1.4e-21$	$1.3e-29$	$2.6e-6$	$3.6e-7$	$8.2e-12$	$1.8e-16$
0.4	0.4733	0.5320	$6.8e-21$	$3.0e-30$	$5.8e-5$	$1.8e-7$	$1.1e-12$	$1.3e-16$
0.5	0.4801	0.5290	$2.4e-20$	$4.6e-31$	$1.1e-5$	$0.1e-8$	$1.7e-11$	$4.2e-17$
0.6	0.4864	0.5259	$6.8e-20$	$3.5e-32$	$4.5e-5$	$2.1e-8$	$2.0e-17$	$5.1e-18$
0.7	0.4922	0.5226	$1.6e-20$	$9.4e-33$	$2.8e-5$	$1.2e-8$	$2.5e-11$	$2.7e-18$
0.8	0.4975	0.5191	$3.3e-19$	$3.0e-34$	$4.1e-5$	$3.4e-8$	$2.4e-11$	$1.1e-19$
0.9	0.5025	0.5113	$6.2e-19$	$2.1e-35$	$5.7e-5$	$7.3e-9$	$3.4e-11$	$3.6e-19$
1	0.5071	0.5071	$1.0e-19$	$3.0e-36$	$7.6e-5$	$2.0e-10$	$3.9e-11$	$5.8e-19$

Consequently $x^L(0) = 0.4380$, $x^U(0) = 0.5399$, and $x^L(1) = x^U(1) = 0.5071$. Therefore, initial guess is $x_0 = (0.4380, 0.5071, 0.5399)$. After 3 iterations, we obtain the solution which the maximum error would be less than 10^{-3} . Now suppose x is negative; we have

$$\begin{cases} (3 + \tau)(x^L(\tau))^2 + (1 + \tau)x^L(\tau) = (1 + \tau), \\ (5 - \tau)(x^U(\tau))^2 + (3 - \tau)x^U(\tau) = (3 - \tau). \end{cases} \quad (52)$$

For $\tau = 0$, we have $x^U(0) > x^L(0)$, and therefore, negative root does not exist.

5. Conclusion

In this research paper, we construct highly efficient family of two-step numerical iterative method to approximate roots of triangular fuzzy nonlinear equations. A set of real life applications are considered as numerical test problems show the practical performance and dominance efficiency of NM over NP and NR. From Tables 1–5 and Figures 1–6, we observe that numerical results of test examples, CPU time, and residual errors corroborate theoretical analysis and illustrate the effectiveness and rapid convergence of our proposed family of iterative method NM as compared to the methods NP and NR.

Data Availability

No data were used to support this study.

Disclosure

The statements made and views expressed are solely the responsibility of the authors.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Authors' Contributions

All authors contributed equally to the preparation of this manuscript.

Acknowledgments

The author (YUG) would like to acknowledge that this publication was made possible by a grant from Carnegie Corporation of New York.

References

- [1] M. Akram, G. Muhammad, and T. Allahviranloo, "Bipolar fuzzy linear system of equations," *Computational and Applied Mathematics*, vol. 38, no. 69, pp. 1–29, 2019.
- [2] M. Akram, G. Muhammad, A. N. A. Koam, and N. Hussain, "Iterative methods for solving a system of linear equation in bipolar fuzzy environment," *Mathematics*, vol. 7, no. 728, pp. 1–25, 2019.
- [3] S. S. L. Chang and L. A. Zadeh, "On fuzzy mapping and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-2, no. 1, pp. 30–34, 1972.
- [4] D. Dubois and H. Prade, "Operations on fuzzy numbers," *International Journal of Systems Science*, vol. 9, no. 6, pp. 613–626, 1978.
- [5] A. N. A. Koam, M. Akram, G. Muhammad, and N. Hussain, "LU decomposition scheme for solving m-polar fuzzy system of linear equations," *Mathematical Problems in Engineering*, vol. 2020, Article ID 8384593, 19 pages, 2020.
- [6] M. Mizumoto, "Some properties of fuzzy numbers," in *Advances in Fuzzy Sets Theory and Applications*, M. M. Gupta, R. K. Ragarde, and R. R. Yager, Eds., pp. 156–164, North-Holland, Amsterdam, Netherlands, 1979.
- [7] M. Mizumoto and K. Tanaka, "The four operations of arithmetic on fuzzy numbers," *System—Computers—Controls*, vol. 7, no. 5, pp. 73–81, 1976.
- [8] S. Nahmias, "Fuzzy variables," *Fuzzy Sets and Systems*, vol. 1, no. 2, pp. 97–110, 1978.
- [9] M. Saqib, M. Akram, and B. Shahida, "Certain efficient iterative methods for bipolar fuzzy system of linear equations," *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 3, pp. 3971–3985, 2020.
- [10] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning-I," *Information Sciences*, vol. 8, no. 3, pp. 199–249, 1975.
- [11] J. J. Buckley and Y. Qu, "Solving linear and quadratic fuzzy equations," *Fuzzy Sets and Systems*, vol. 38, no. 1, pp. 43–59, 1990.
- [12] J. J. Buckley and Y. Qu, "On using α -cuts to evaluate fuzzy equations," *Fuzzy Sets and Systems*, vol. 38, no. 3, pp. 309–312, 1990.
- [13] J. J. Buckley and Y. Qu, "Solving fuzzy equations: a new solution concept," *Fuzzy Sets and Systems*, vol. 39, no. 3, pp. 291–301, 1991.
- [14] J. J. Buckley and Y. Qu, "Solving systems of linear fuzzy equations," *Fuzzy Sets and Systems*, vol. 43, no. 1, pp. 33–43, 1991.
- [15] S. Abbasbandy and B. Asady, "Newton's method for solving fuzzy nonlinear equations," *Applied Mathematics and Computation*, vol. 159, no. 2, pp. 349–356, 2004.
- [16] I. M. Sulaiman, M. Mamat, M. Y. Waziri, M. A. Mohamed, and F. S. Mohamad, "Solving fuzzy nonlinear equation via Levenberg-Marquardt method," *Far East Journal of Mathematical Sciences*, vol. 103, no. 10, pp. 1547–1558, 2018.
- [17] M. Mosleh, "Solution of dual fuzzy polynomial equations by modified Adomian decomposition method," *Fuzzy Information and Engineering*, vol. 5, no. 1, pp. 45–56, 2013.
- [18] Y. J. Cho, N. J. Huang, and S. M. Kang, "Nonlinear equations for fuzzy mappings in probabilistic normed spaces," *Fuzzy Sets and Systems*, vol. 110, no. 1, pp. 115–122, 2000.
- [19] J.-X. Fang, "On nonlinear equations for fuzzy mappings in probabilistic normed spaces," *Fuzzy Sets and Systems*, vol. 131, no. 3, pp. 357–364, 2002.
- [20] J. Ma and G. Feng, "An approach to H_∞ control of fuzzy dynamic systems," *Fuzzy Sets and Systems*, vol. 137, no. 3, pp. 367–386, 2003.
- [21] H. T. Kung and J. F. Traub, "Optimal order of one-point and multipoint iteration," *Journal of the ACM*, vol. 21, no. 4, pp. 643–651, 1974.
- [22] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Application*, Academic Press, New York, NY, USA, 1980.
- [23] H. J. Zimmermann, *Fuzzy Sets Theory and its Application*, Kluwer Academic Press, Dordrecht, Netherlands, 1991.
- [24] R. Goetschel and W. Voxman, "Elementary fuzzy calculus," *Fuzzy Sets and Systems*, vol. 18, no. 1, pp. 31–43, 1986.
- [25] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [26] M. Keyanpour and T. Akbarian, "Solving intuitionistic fuzzy nonlinear equations," *Journal of Fuzzy Set Valued Analysis*, vol. 2014, pp. 1–6, 2014.
- [27] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Hoboken, NJ, USA, 1983.
- [28] M. A. Rehman, A. Naseem, and T. Abdeljawad, "Some novel sixth-order iteration schemes for computing zeros of non-linear scalar equations and their applications in engineering," *Journal of Function Spaces*, vol. 2021, Article ID 5566379, 11 pages, 2021.