



# Numerical solution of interval-valued trapezoidal fuzzy non-linear system of equations and its applications

Mudassir Shams<sup>a,b,c</sup>, Nasreen Kausar<sup>a,d</sup>, Harish Garg<sup>e,\*</sup>, Shams Forruque Ahmed<sup>f,g</sup>

<sup>a</sup> Department of Mathematics, Faculty of Arts and Science, Balikesir University, Balikesir, Turkey

<sup>b</sup> Faculty of Engineering, Free University of Bozen-Bolzano (BZ), Bolzano, Italy

<sup>c</sup> Department of Mathematics and Statistics, Riphah International University I-14, Islamabad, 44000, Pakistan

<sup>d</sup> Department of Mathematics, Faculty of Arts and Sciences, Yildiz Technical University, Istanbul, Esenler, 34220, Turkey

<sup>e</sup> Department of Mathematics, Thapar Institute of Engineering and Technology (Deemed University), Patiala, 147004, Punjab, India

<sup>f</sup> School of Mathematical Sciences, Sunway University, Bandar Sunway, Petaling Jaya, 47500, Selangor Darul Ehsan, Malaysia

<sup>g</sup> Mijan Research Institute, International University of Business Agriculture and Technology, Dhaka, 1230, Bangladesh

## ARTICLE INFO

### Keywords:

Numerical schemes

Interval-valued trapezoidal fuzzy number

Fuzzy set

Analytical solutions

## ABSTRACT

In many areas of applied mathematics, physical and chemical sciences, and engineering sciences—particularly bio engineering—the topic of fuzzy systems of nonlinear equations with an interval-valued trapezoidal fuzzy (IVTF) number or crisp real-valued coefficient matrix and with a vector of interval-valued trapezoidal fuzzy numbers or crisp real value numbers on the right-hand side arises. However, insufficient investigation has been made into the development and assessment of numerical schemes for resolving interval value trapezoidal fuzzy nonlinear systems of equations (IVTFNSEs). In this paper, we develop and analyze efficient higher order IVTF-numerical schemes for solving IVTFNSEs. Supply chain networks with unpredictable needs and IVTF data are taken into consideration for some problems in econometrics. Numerical examples using interval-valued trapezoidal fuzzy quantities in the parametric form are explored. The outcomes of numerical methods are tested using various stopping criteria, then compared and analyzed with other existing methods on the basis of computational CPU time in seconds, local errors, maximum errors, residual errors and consistency. The numerical outcomes frequently demonstrate that our newly developed method performs better than existing classical techniques, obtaining higher accuracy, faster convergence, and lower computing cost. The graphical findings are analyzed in order to demonstrate and support the theoretical significance, illustrating the better performance of the newly modified numerical techniques over the existing numerical scheme. The suggested approach for solving trapezoidal fuzzy nonlinear systems is a useful tool for real-world applications involving uncertainty and complex systems, since these results confirm its robustness, efficiency, and applicability.

## 1. Introduction

Nonlinear systems of equations are utilized in many field of science and engineering to simulate and comprehend the behavior of complex systems. The importance of these types of models lies in their ability to capture the nonlinear relationships in nature

\* Corresponding author.

E-mail addresses: [shams6262@gmail.com](mailto:shams6262@gmail.com) (M. Shams), [kausar.nasreen57@gmail.com](mailto:kausar.nasreen57@gmail.com) (N. Kausar), [harishg58iitr@gmail.com](mailto:harishg58iitr@gmail.com), [harish.garg@thapar.edu](mailto:harish.garg@thapar.edu) (H. Garg), [shams.f.ahmed@gmail.com](mailto:shams.f.ahmed@gmail.com) (S.F. Ahmed).

<https://doi.org/10.1016/j.cam.2026.117468>

Received 3 October 2024; Received in revised form 23 December 2024

Available online 3 March 2026

0377-0427/© 2026 Elsevier B.V. All rights reserved, including those for text and data mining, AI training, and similar technologies.

and engineering, which allows for more accurate predictions, analyses, and designs to be made [1,2]. Nonlinear system parameters and measurements are most often represented with fuzzy numbers rather than crisp numbers. Nonlinear systems of equations with fuzzy numbers hold great significance in various fields due to their aptitude to model and analyze intricate systems with imprecise or uncertain information [3,4]. The significance of such systems lies in their ability to model and analyze complex systems where imprecision and uncertainty are inherent. These nonlinear system of equations have a broad range of applications in science and engineering, such as in supply chain management, resource allocation, decision-making, control systems, robotics, image processing, etc. [5,6].

The concept of fuzzy numbers and arithmetic operations on fuzzy numbers was first introduced and investigated by authors [7–11]. Intuitionistic fuzzy sets, proposed by Atanassov [12], are one of the most important types, as they contain a membership function and a non-membership function and can better meet the needs of real-world applications. Based on intuitionistic fuzzy sets, Atanassov and Gargov [13] introduced the concept of interval-valued intuitionistic fuzzy sets, whose membership and non-membership functions are all intervals and have more practical properties. Many researchers have devoted themselves to studying the theory and applications of interval-valued intuitionistic sets [14–18]. Bipolar fuzzy set [19] is an improvement of classical fuzzy set that use three-dimensional membership degrees to provide a more precise depiction of uncertainty. In a variety of fields, including engineering, finance, and artificial intelligence, they are used in risk analysis, multi-criteria decision-making, and complex system modeling [20]. Bipolar triangular [21], bipolar intuitionistic [22], and bipolar trapezoidal fuzzy numbers [21] are extended versions of traditional fuzzy ideas that may handle both positive and negative preferences or evaluations. This approach captures a considerably wider range of uncertainties and ambiguities. In artificial intelligence, engineering, and economics, they are employed bipolar fuzzy number in decision-making, multi-criteria optimization, and conflict scenario modeling [23,24]. Pentagonal fuzzy number [25] and hexagonal fuzzy number [26] are other expansions of fuzzy numbers that can be defined by five and six parameters, respectively. These are more flexible representations of uncertainty and vagueness. The importance of such representations is in the accurate modeling of complicated systems in decision-making processes, optimization, and risk assessment under imprecise conditions [27]. The solution of nonlinear systems of equations using distinct fuzzy numbers is intended to incorporate inaccurate input parameters into the problem in order to handle uncertainty and enable for more realistic and flexible modeling of complex systems in engineering, finance, and biology. Its significance stems from improving making choices, robustness, and adaptation to real-world settings in which precise data is not accessible.

The subject of fuzzy systems of nonlinear equations with interval-valued trapezoidal fuzzy number [28] or crisp real-valued coefficient matrix and with a vector of interval-valued trapezoidal number or crisp real value number on the right hand side arise in many branches of applied mathematics, physical and chemical sciences. Frequently, the specific understanding of such nonlinear problems comprises ambiguous and non-probabilistic reservations in the parameter, where the computations are determined based on expert information or untested data. To address the issue of unpredictable demand in supply chain networks, we aim to minimize the overall system cost by determining the level of uncertainty. This often leads to the creation of a system of interval-valued trapezoidal fuzzy polynomial equations, known as nonlinear IVTF equations. Additionally, we explore solutions for economic problems such as the Supply function equilibrium, which involves IVTF values in parametric form and provides a suitable framework for modeling competitive interactions among suppliers in electricity markets. Consequently, solving interval value trapezoidal fuzzy nonlinear systems of equations (IVTFNSEs) is important. The classical semi-analytical and analytical schemes similar to Buckley and Qu [29,30] are not appropriate for solving the interval-valued trapezoidal fuzzy system of non-linear equations

$$\begin{cases} F(x) = C, \\ G(x) = D, \end{cases} \quad (1)$$

where in Eq. (1),  $x$  is an IVTF variable and  $C, D$  are IVTF numbers. Incorporating interval analysis and fuzzy arithmetic into computation can help reduce uncertainties in a trapezoidal fuzzy nonlinear system. This guarantees that uncertainty spreads throughout the system. Furthermore, sensitivity analysis aids in measuring how input uncertainties affect the outcome. We therefore look towards numerical schemes which solve IVTFNSEs and analyze the uncertainty involve in the system.

An analytical framework for dynamic systems with intrinsic uncertainties is provided by simulating fuzzy differential equations using nonlinear systems of equations. Because it captures imprecision in initial conditions, parameters, or external influences, this approach is especially helpful in scientific and technical domains. It is crucial to solve and analyze these nonlinear fuzzy systems of equations in order to comprehend how different physical processes behave. Analytical techniques are highly computationally extensive and almost always inappropriate when working with fuzzy data and nonlinear equations. Therefore, these types of complex fuzzy nonlinear equations are solved numerically. In the literature, there are numerous numerical techniques for solving fuzzy nonlinear systems of equations. Abasbandy et al. [31] used Newton's method to solve a fuzzy nonlinear system of equations with triangular fuzzy numbers. In order to solve fuzzy nonlinear equations. Keyanpour al. [32] used a numerical iterative approach. Sulaiman and Omesa [33,34] created the Levenberg-Marquardt approach, which uses triangular fuzzy numbers to solve fuzzy nonlinear equations and their system. Cho et al. [35] approximates fuzzy mapping in probabilistic normed spaces by solving nonlinear equations. Dual fuzzy nonlinear polynomial equations were solved by Mosleh et al. [36] using semi-numerical techniques. Allahviranloo et al. [37] solved fuzzy polynomials using the Newton-Raphson method. Jafari and Yu [38] discussed fuzzy modelling and control of uncertain nonlinear systems. Ma and Feng [39] discussed a unified method to H1 control of fuzzy dynamic systems.

The literature review reveals a scarcity of studies focusing on the development and analysis of efficient and stable numerical iterative schemes of higher order for solving IVTFNSEs. Motivated by the lack of research in this area and previous research works, it is critical to identify the core concepts and results needed for the correct development of this topic by constructing higher order schemes and performing convergence analysis. Therefore, the present study aims to construct and analyze efficient higher order

IVTF-numerical schemes that solve IVTFNSEs. Using numerical examples, the consistency and stability of the proposed numerical scheme for solving IVTFNSEs are demonstrated. The research study makes the following main contributions:

1. Define TFN and its mathematical operations.
2. To modify an existing numerical approach for solving more generalized fuzzy nonlinear systems of equations and compare it to a well-known classical method.
3. Compute the theoretical convergence order of the proposed method by using local convergence theorems.
4. Numerical test examples are provided to demonstrate the precision and accuracy of the proposed approach as well as the graphical representations of the problem’s solution.
5. The outcomes of our proposed method are compared to those of existing methods.
6. The results of numerical benchmark problems, as well as CPU time, errors, residual errors, and computational order of convergence, show that our modified scheme outperforms the current classical technique.
7. Stability and consistency analysis are discussed in detail.
8. The rate of convergence is accurately analyzed by calculating the percentage convergence, divergence, and the total number of function evaluations per iteration.

The rest of the contents of this study are organized as follows. Section 2 give basic concepts and definition. Section 3 propose and investigates a numerical iterative method for solving an interval-valued trapezoidal fuzzy nonlinear system of equations. We demonstrate various numerical test examples in Section 4. Finally, concrete conclusion is drawn in Section 5.

## 2. Preliminaries

Here, we discuss some preliminary results that are essential to understanding the proposed methodology because they provide fundamental validation of its practicality and effectiveness. They contribute to an understanding of the method’s behavior, convergence qualities, and practical application. By evaluating these preliminary results, we may identify the methodology’s strengths, potential limitations, and areas for improvement, ensuring that it is powerful and well-suited to solve the target problem domain.

**Definition 2.1.** A fuzzy number [40] is a fuzzy set like  $x : \mathbb{R} \rightarrow I = [0, 1]$  which satisfies

- 1)  $x$  is an upper semi-continuous,
  - 2)  $x(\ddot{e}) = 0$  outside some interval  $[\ddot{e}_1, \ddot{e}_2]$ ,
  - 3) there are real numbers  $\check{a}_1, \check{a}_2$  such that  $\ddot{e}_1 \leq \check{a}_1 \leq \check{a}_2 \leq \ddot{e}_2$  and
- $x(\ddot{e})$  is monotonically increasing on  $[\ddot{e}_1, \check{a}_1]$ ,
  - $x(\ddot{e})$  is monotonically decreasing on  $[\check{a}_2, \ddot{e}_2]$ ,
  - $x(\ddot{e}) = 1$ , for  $\check{a}_1 \leq \ddot{e} \leq \check{a}_2$ .

We denote by  $E$ , the set of all fuzzy numbers.

**Definition 2.2.** [41] 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. We define  $x^L(\tau)$  is a bounded monotonic increasing left continuous function over the close interval 0 and 1.
2. We define  $x^U(\tau)$  is a bounded monotonic decreasing left continuous function over the closed interval 0 and 1.
3.  $x^L(\tau) \leq x^U(\tau), 0 \leq \tau \leq 1$ .

A popular fuzzy number is the generalized interval-valued trapezoidal fuzzy numbers [41]  $A$ , denoted by  $A = (\ddot{e}_1, \ddot{e}_2, \check{e}_3, \check{e}_4; \omega)$ ,  $0 < \omega < 1$  is a fuzzy number with membership function as:

$$A(x) = \begin{cases} \omega \frac{x - \ddot{e}_1}{\ddot{e}_2 - \ddot{e}_1} & \text{if } \ddot{e}_1 < x < \ddot{e}_2, \\ \omega & \text{if } \ddot{e}_2 \leq x \leq \check{e}_3, \\ \omega \frac{\check{e}_4 - x}{\check{e}_4 - \check{e}_3} & \text{if } \check{e}_3 < x < \check{e}_4, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Assume  $F_{TN}(\omega)$  be the family of all  $\omega$ -trapezoidal fuzzy numbers i.e.,

$$F_{TN}(\omega) = \left\{ A = (\ddot{e}_1, \ddot{e}_2, \check{e}_3, \check{e}_4; \omega), \ddot{e}_1 \leq \ddot{e}_2 \leq \check{e}_3 \leq \check{e}_4; \right\} \tag{3}$$

**Definition 2.3.** [41] Let  $A^L \in F_{TN}(\omega^L)$  and  $A^U \in F_{TN}(\omega^U)$  then a level  $(\omega^L, \omega^U)$ - of IVTF number  $A$ , denoted by

$$A = [A^L, A^U] = \langle (\ddot{e}_1^L, \ddot{e}_2^L, \check{e}_3^L, \check{e}_4^L; \omega^L), (\ddot{e}_1^U, \ddot{e}_2^U, \check{e}_3^U, \check{e}_4^U; \omega^U) \rangle \tag{4}$$

is interval-valued fuzzy number on set  $\mathbb{R}$  with

$$A^L(x) = \begin{cases} \omega^L \frac{x - \ddot{e}_1^L}{\ddot{e}_2^L - \ddot{e}_1^L} & \text{if } \ddot{e}_1^L < x < \ddot{e}_2^L, \\ \omega^L & \text{if } \ddot{e}_2^L \leq x \leq \check{e}_3^L, \\ \omega^L \frac{\check{e}_4^L - x}{\check{e}_4^L - \check{e}_3^L} & \text{if } \check{e}_3^L < x < \check{e}_4^L, \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

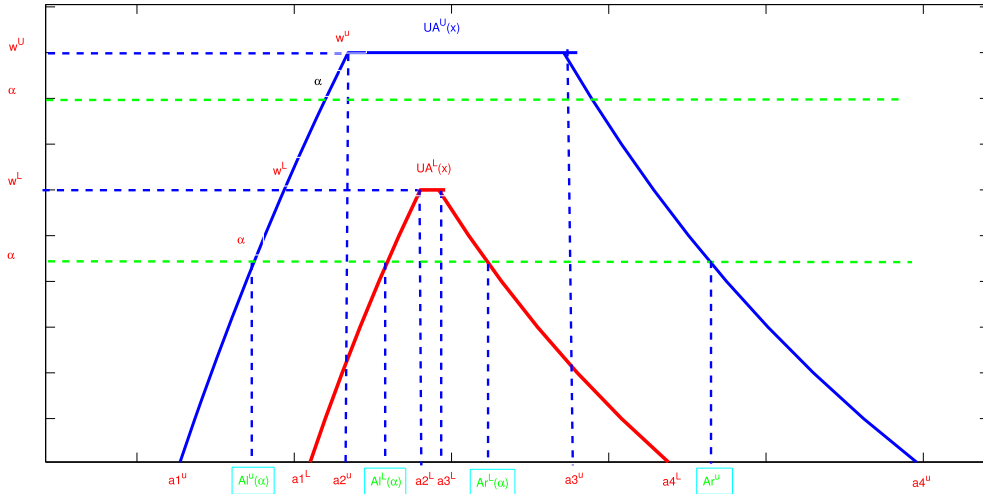


Fig. 1. Alpha-cut level of interval-valued trapezoidal fuzzy number.

and

$$A^U(x) = \begin{cases} \omega^U \frac{x - \check{e}_1^U}{\check{e}_2^U - \check{e}_1^U} & \text{if } \check{e}_1^U < x < \check{e}_2^U, \\ \omega^U & \text{if } \check{e}_2^U \leq x \leq \check{e}_3^U, \\ \omega^U \frac{\check{e}_4^U - x}{\check{e}_4^U - \check{e}_3^U} & \text{if } \check{e}_3^U < x < \check{e}_4^U, \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where  $\check{e}_1^L \leq \check{e}_2^L \leq \check{e}_3^L \leq \check{e}_4^L$ ,  $\check{e}_1^U \leq \check{e}_2^U \leq \check{e}_3^U \leq \check{e}_4^U$ ,  $0 \leq \omega^L \leq \omega^U \leq 1$ ,  $\check{e}_1^U \leq \check{e}_1^L$  and  $\check{e}_4^L \leq \check{e}_4^U$ . This IVTF is shown in Fig. 1 [42]. Moreover,  $A^L(x) \leq A^U(x)$ , which means the grade of membership  $x \in \mathbf{A} = [A^L(x), A^U(x)]$ , the latest and greatest grade of membership at  $x$  are  $A^L(x)$ , and  $A^U(x)$  respectively. We therefore, denote the family of all IVTF numbers by  $F(\omega^L, \omega^U) = \mathbf{A} = [A^L(x), A^U(x)]$  i.e.,

$$F(\omega^L, \omega^U) = \mathbf{A} = [A^L(x), A^U(x)] = \{ \langle (\check{e}_1^L, \check{e}_2^L, \check{e}_3^L, \check{e}_4^L; \omega^L), (\check{e}_1^U, \check{e}_2^U, \check{e}_3^U, \check{e}_4^U; \omega^U) \rangle : \check{e}_1^U \leq \check{e}_1^L; \check{e}_4^L \leq \check{e}_4^U \\ A^L(x) \in F_{TN}(\omega^L), A^U(x) \in F_{TN}(\omega^U), 0 \leq \omega^L \leq \omega^U \leq 1. \}$$

**Definition 2.4.** [41] A  $(\omega^L, \omega^U)$  is said to be non-negative-F  $(\omega^L, \omega^U)$  iff  $\check{e}_1^U \geq 0$  and denoted by  $F^+(\omega^L, \omega^U)$ .

**Definition 2.5.** [41] Extend addition, scalar multiplication and extend multiplication in  $(\omega^L, \omega^U)$  interval-valued trapezoidal fuzzy number is defined as if  $A = \langle (\check{e}_1^L, \check{e}_2^L, \check{e}_3^L, \check{e}_4^L; \omega^L), (\check{e}_1^U, \check{e}_2^U, \check{e}_3^U, \check{e}_4^U; \omega^U) \rangle$  and

$B = \langle (\check{a}_1^L, \check{a}_2^L, \check{a}_3^L, \check{a}_4^L; \omega^L), (\check{a}_1^U, \check{a}_2^U, \check{a}_3^U, \check{a}_4^U; \omega^U) \rangle \in F(\omega^L, \omega^U)$  and  $k \in \mathbb{R}$ , then

$$A \oplus B = \left\langle \left( \check{e}_1^L + \check{a}_1^L, \check{e}_2^L + \check{a}_2^L, \check{e}_3^L + \check{a}_3^L, \check{e}_4^L + \check{a}_4^L; \omega^L \right), \left( \check{e}_1^U + \check{a}_1^U, \check{e}_2^U + \check{a}_2^U, \check{e}_3^U + \check{a}_3^U, \check{e}_4^U + \check{a}_4^U; \omega^U \right) \right\rangle,$$

$$A \otimes B = \left\langle \left( \check{e}_1^L * \check{a}_1^L, \check{e}_2^L * \check{a}_2^L, \check{e}_3^L * \check{a}_3^L, \check{e}_4^L * \check{a}_4^L; \omega^L \right), \left( \check{e}_1^U * \check{a}_1^U, \check{e}_2^U * \check{a}_2^U, \check{e}_3^U * \check{a}_3^U, \check{e}_4^U * \check{a}_4^U; \omega^U \right) \right\rangle, \check{e}_1^U, \check{a}_1^U \geq 0 \text{ and}$$

$$kA = \begin{cases} \left\langle \left( k\check{a}_1^L, k\check{a}_2^L, k\check{a}_3^L, k\check{a}_4^L; \omega^L \right), \left( k\check{a}_1^U, k\check{a}_2^U, k\check{a}_3^U, k\check{a}_4^U; \omega^U \right) \right\rangle, k > 0 \\ \left\langle \left( k\check{a}_4^L, k\check{a}_3^L, k\check{a}_2^L, k\check{a}_1^L; \omega^L \right), \left( k\check{a}_4^U, k\check{a}_3^U, k\check{a}_2^U, k\check{a}_1^U; \omega^U \right) \right\rangle, k < 0 \\ \left\langle (0, 0, 0, 0; \omega^L), (0, 0, 0, 0; \omega^U) \right\rangle, k = 0, \end{cases}$$

**Definition 2.6.** [41] Let  $A \in F(\omega^L, \omega^U)$ , then the alpha-cut set of  $\mathbf{A}$  can be and defined by

$$\mathbf{A}(\alpha) = [A^L(\alpha), A^U(\alpha)] = \left\{ \begin{aligned} & [A^U(\alpha), A^L(\alpha)] \cup [A^L(\alpha), A^U(\alpha)]; 0 \leq \alpha \leq \omega^L \\ & [A^U(\alpha), A^U(\alpha)]; \omega^L \leq \alpha \leq \omega^U \end{aligned} \right.$$

where

$$\begin{cases} A^L(\alpha) = \check{e}_1^L + (\check{e}_2^L - \check{e}_1^L) \frac{\alpha}{\omega^L}, \\ A^L(\alpha) = \check{e}_4^L + (\check{e}_4^L - \check{e}_3^L) \frac{\alpha}{\omega^L}, \\ A^U(\alpha) = \check{e}_1^U + (\check{e}_2^U - \check{e}_1^U) \frac{\alpha}{\omega^U}, \\ A^U(\alpha) = \check{e}_4^U + (\check{e}_4^U - \check{e}_3^U) \frac{\alpha}{\omega^U}. \end{cases}$$

### 3. Construction of iterative scheme(SFQM1)

Iterative approaches can be crucial in solving nonlinear systems of equations, particularly in situations when calculating analytical solutions is either impossible or impractical. Analytical methods are always precise, but they can only be used to very basic or idealized issues that fail to account for high dimensional or fuzzy uncertainty in reality [43–45]. Iterative approaches, on the other hand, are extremely adaptable and scalable, making them ideal for dealing with complicated systems. Through a series of iterations, they approximate the solutions; they achieve the required precision with less computational work than obtaining explicit formulas. Iterative approaches have significant implications for fuzzy trapezoidal nonlinear systems. Because of the ambiguity inherent in the system’s coefficients and variables, nonanalytical solutions are extremely difficult to get. Once more, iterative approaches offer a concept—of course, a framework-like solution—whereby the technique fits into the plan without making an effort to avoid system fuzziness. Iterative approaches are well-suited to cope with the interplay between fuzziness and nonlinearity, such that it converges toward a solution that respects the trapezoidal fuzzy framework. Further, iterative methods enable the optimization of performance parameters such as residual error, computational order of convergence, and CPU time, resulting in systems that are not only practically solvable but also computationally efficient. Their flexibility allows them to be easily modified or extended to handle various types of fuzzy systems or fields of application, ensuring their place in the field.

In order to approximate the roots of interval-valued trapezoidal fuzzy non-linear equation  $F(x) = \begin{cases} F(x) = C \\ G(x) = D \end{cases}$ , we stated the two-step iterative scheme as:

$$\begin{cases} F_l^L(x_l^L, x_l^L, x_l^U, x_l^U, \tau) = C_l^L(\tau), \\ F_r^L(x_l^L, x_l^L, x_l^U, x_l^U, \tau) = C_l^L(\tau), \\ F_l^U(x_l^L, x_l^L, x_l^U, x_l^U, \tau) = C_l^U(\tau), \\ F_r^U(x_l^L, x_l^L, x_l^U, x_l^U, \tau) = C_l^U(\tau), \\ G_l^L(y_l^L, y_l^L, y_l^U, y_l^U, \tau) = C_l^L(\tau), \\ G_r^L(y_l^L, y_l^L, y_l^U, y_l^U, \tau) = C_l^L(\tau), \\ G_l^U(y_l^L, y_l^L, y_l^U, y_l^U, \tau) = C_l^U(\tau), \\ G_r^U(y_l^L, y_l^L, y_l^U, y_l^U, \tau) = C_l^U(\tau). \end{cases} \quad \forall \tau \in [0, 1]. \tag{7}$$

Suppose that  $x = (\alpha_l^L, \alpha_r^L, \alpha_l^U, \alpha_r^U), y = (\beta_l^L, \beta_r^L, \beta_l^U, \beta_r^U)$  is the solution of the above system Eq. (7) and  $x_0 = (x_{l0}^L, x_{r0}^L, x_{l0}^U, x_{r0}^U)$  and  $y_0 = (y_{l0}^L, y_{r0}^L, y_{l0}^U, y_{r0}^U)$  are approximate solutions of the system then

$$\begin{cases} \alpha_l^L(\tau) = x_{l0}^L(\tau) + h_l^L(\tau), \\ \alpha_r^L(\tau) = x_{r0}^L(\tau) + h_r^L(\tau), \\ \alpha_l^U(\tau) = x_{l0}^U(\tau) + h_l^U(\tau), \\ \alpha_r^U(\tau) = x_{r0}^U(\tau) + h_r^U(\tau), \\ \beta_l^L(\tau) = y_{l0}^L(\tau) + k_l^L(\tau), \\ \beta_r^L(\tau) = y_{r0}^L(\tau) + k_r^L(\tau), \\ \beta_l^U(\tau) = y_{l0}^U(\tau) + k_l^U(\tau), \\ \beta_r^U(\tau) = y_{r0}^U(\tau) + k_r^U(\tau). \end{cases} \tag{8}$$

By using Taylor’s series of  $F_l^L, F_r^L, F_l^U, F_r^U$  about  $(x_{l0}^L(\tau), x_{r0}^L(\tau), x_{l0}^U(\tau), x_{r0}^U(\tau))$ , in Eq. (8), then, we have

$$\begin{cases} F_l^L(\alpha_l^L, \alpha_r^L, \alpha_l^U, \alpha_r^U, \beta_l^L, \beta_r^L, \beta_l^U, \beta_r^U, \tau) = F_l^L(\Phi) + h_l^L F_l^L{}_{x_{l0}^L}(\Phi) + h_r^L F_l^L{}_{x_{r0}^L}(\Phi) + h_l^U F_l^L{}_{x_{l0}^U}(\Phi) + h_r^U F_l^L{}_{x_{r0}^U}(\Phi) + k_l^L F_l^L{}_{y_{l0}^L}(\Phi) + k_r^L F_l^L{}_{y_{r0}^L}(\Phi) + k_l^U F_l^L{}_{y_{l0}^U}(\Phi) + k_r^U F_l^L{}_{y_{r0}^U}(\Phi) + O(\Phi_1), \\ F_r^L(\alpha_l^L, \alpha_r^L, \alpha_l^U, \alpha_r^U, \beta_l^L, \beta_r^L, \beta_l^U, \beta_r^U, \tau) = F_r^L(\Phi) + h_l^L F_r^L{}_{x_{l0}^L}(\Phi) + h_r^L F_r^L{}_{x_{r0}^L}(\Phi) + h_l^U F_r^L{}_{x_{l0}^U}(\Phi) + h_r^U F_r^L{}_{x_{r0}^U}(\Phi) + k_l^L F_r^L{}_{y_{l0}^L}(\Phi) + k_r^L F_r^L{}_{y_{r0}^L}(\Phi) + k_l^U F_r^L{}_{y_{l0}^U}(\Phi) + k_r^U F_r^L{}_{y_{r0}^U}(\Phi) + O(\Phi_1), \\ F_l^U(\alpha_l^L, \alpha_r^L, \alpha_l^U, \alpha_r^U, \beta_l^L, \beta_r^L, \beta_l^U, \beta_r^U, \tau) = F_l^U(\Phi) + h_l^L F_l^U{}_{x_{l0}^L}(\Phi) + h_r^L F_l^U{}_{x_{r0}^L}(\Phi) + h_l^U F_l^U{}_{x_{l0}^U}(\Phi) + h_r^U F_l^U{}_{x_{r0}^U}(\Phi) + k_l^L F_l^U{}_{y_{l0}^L}(\Phi) + k_r^L F_l^U{}_{y_{r0}^L}(\Phi) + k_l^U F_l^U{}_{y_{l0}^U}(\Phi) + k_r^U F_l^U{}_{y_{r0}^U}(\Phi) + O(\Phi_1), \\ F_r^U(\alpha_l^L, \alpha_r^L, \alpha_l^U, \alpha_r^U, \beta_l^L, \beta_r^L, \beta_l^U, \beta_r^U, \tau) = F_r^U(\Phi) + h_l^L F_r^U{}_{x_{l0}^L}(\Phi) + h_r^L F_r^U{}_{x_{r0}^L}(\Phi) + h_l^U F_r^U{}_{x_{l0}^U}(\Phi) + h_r^U F_r^U{}_{x_{r0}^U}(\Phi) + k_l^L F_r^U{}_{y_{l0}^L}(\Phi) + k_r^L F_r^U{}_{y_{r0}^L}(\Phi) + k_l^U F_r^U{}_{y_{l0}^U}(\Phi) + k_r^U F_r^U{}_{y_{r0}^U}(\Phi) + O(\Phi_1), \\ G_l^L(\alpha_l^L, \alpha_r^L, \alpha_l^U, \alpha_r^U, \beta_l^L, \beta_r^L, \beta_l^U, \beta_r^U, \tau) = G_l^L(\Phi) + h_l^L G_l^L{}_{x_{l0}^L}(\Phi) + h_r^L G_l^L{}_{x_{r0}^L}(\Phi) + h_l^U G_l^L{}_{x_{l0}^U}(\Phi) + h_r^U G_l^L{}_{x_{r0}^U}(\Phi) + k_l^L G_l^L{}_{y_{l0}^L}(\Phi) + k_r^L G_l^L{}_{y_{r0}^L}(\Phi) + k_l^U G_l^L{}_{y_{l0}^U}(\Phi) + k_r^U G_l^L{}_{y_{r0}^U}(\Phi) + O(\Phi_1), \\ G_r^L(\alpha_l^L, \alpha_r^L, \alpha_l^U, \alpha_r^U, \beta_l^L, \beta_r^L, \beta_l^U, \beta_r^U, \tau) = G_r^L(\Phi) + h_l^L G_r^L{}_{x_{l0}^L}(\Phi) + h_r^L G_r^L{}_{x_{r0}^L}(\Phi) + h_l^U G_r^L{}_{x_{l0}^U}(\Phi) + h_r^U G_r^L{}_{x_{r0}^U}(\Phi) + k_l^L G_r^L{}_{y_{l0}^L}(\Phi) + k_r^L G_r^L{}_{y_{r0}^L}(\Phi) + k_l^U G_r^L{}_{y_{l0}^U}(\Phi) + k_r^U G_r^L{}_{y_{r0}^U}(\Phi) + O(\Phi_1), \\ G_l^U(\alpha_l^L, \alpha_r^L, \alpha_l^U, \alpha_r^U, \beta_l^L, \beta_r^L, \beta_l^U, \beta_r^U, \tau) = G_l^U(\Phi) + h_l^L G_l^U{}_{x_{l0}^L}(\Phi) + h_r^L G_l^U{}_{x_{r0}^L}(\Phi) + h_l^U G_l^U{}_{x_{l0}^U}(\Phi) + h_r^U G_l^U{}_{x_{r0}^U}(\Phi) + k_l^L G_l^U{}_{y_{l0}^L}(\Phi) + k_r^L G_l^U{}_{y_{r0}^L}(\Phi) + k_l^U G_l^U{}_{y_{l0}^U}(\Phi) + k_r^U G_l^U{}_{y_{r0}^U}(\Phi) + O(\Phi_1), \\ F_l^U(\alpha_l^L, \alpha_r^L, \alpha_l^U, \alpha_r^U, \beta_l^L, \beta_r^L, \beta_l^U, \beta_r^U, \tau) = G_l^U(\Phi) + h_l^L G_l^U{}_{x_{l0}^L}(\Phi) + h_r^L G_l^U{}_{x_{r0}^L}(\Phi) + h_l^U G_l^U{}_{x_{l0}^U}(\Phi) + h_r^U G_l^U{}_{x_{r0}^U}(\Phi) + k_l^L G_l^U{}_{y_{l0}^L}(\Phi) + k_r^L G_l^U{}_{y_{r0}^L}(\Phi) + k_l^U G_l^U{}_{y_{l0}^U}(\Phi) + k_r^U G_l^U{}_{y_{r0}^U}(\Phi) + O(\Phi_1). \end{cases}$$



and

$$\begin{bmatrix} x_{l11}^L(\tau) \\ x_{t11}^L(\tau) \\ x_{l011}^U(\tau) \\ x_{t11}^U(\tau) \\ y_{l11}^L(\tau) \\ y_{t11}^L(\tau) \\ y_{l011}^U(\tau) \\ y_{t11}^U(\tau) \end{bmatrix} = \begin{bmatrix} x_{l01}^L(\tau) \\ x_{t01}^L(\tau) \\ x_{l01}^U(\tau) \\ x_{t01}^U(\tau) \\ y_{l01}^L(\tau) \\ y_{t01}^L(\tau) \\ y_{l01}^U(\tau) \\ y_{t01}^U(\tau) \end{bmatrix} + Z * \begin{bmatrix} h_l^L(\tau) \\ h_t^L(\tau) \\ h_l^U(\tau) \\ h_t^U(\tau) \\ k_l^U(\tau) \\ k_t^U(\tau) \\ k_l^U(\tau) \\ k_t^U(\tau) \end{bmatrix},$$

$$Z = (4J_{**}(\Phi_*) - 2J_*(\Phi))^{-1} * (J_*(\Phi) - J_{**}(\Phi_*)),$$

$$J_{**}(\Phi_*) = \begin{bmatrix} F_l^{L, x_{l01}^L}(\Phi_*) & F_l^{L, x_{t01}^L}(\Phi_*) & F_l^{L, x_{l01}^U}(\Phi_*) & F_l^{L, x_{t01}^U}(\Phi_*) & F_l^{L, y_{l01}^L}(\Phi_*) & F_l^{L, y_{t01}^L}(\Phi_*) & F_l^{L, y_{l01}^U}(\Phi_*) & F_l^{L, y_{t01}^U}(\Phi_*) \\ F_t^{L, x_{l01}^L}(\Phi_*) & F_t^{L, x_{t01}^L}(\Phi_*) & F_t^{L, x_{l01}^U}(\Phi_*) & F_t^{L, x_{t01}^U}(\Phi_*) & F_t^{L, y_{l01}^L}(\Phi_*) & F_t^{L, y_{t01}^L}(\Phi_*) & F_t^{L, y_{l01}^U}(\Phi_*) & F_t^{L, y_{t01}^U}(\Phi_*) \\ F_l^{U, x_{l01}^L}(\Phi_*) & F_l^{U, x_{t01}^L}(\Phi_*) & F_l^{U, x_{l01}^U}(\Phi_*) & F_l^{U, x_{t01}^U}(\Phi_*) & F_l^{U, y_{l01}^L}(\Phi_*) & F_l^{U, y_{t01}^L}(\Phi_*) & F_l^{U, y_{l01}^U}(\Phi_*) & F_l^{U, y_{t01}^U}(\Phi_*) \\ F_t^{U, x_{l01}^L}(\Phi_*) & F_t^{U, x_{t01}^L}(\Phi_*) & F_t^{U, x_{l01}^U}(\Phi_*) & F_t^{U, x_{t01}^U}(\Phi_*) & F_t^{U, y_{l01}^L}(\Phi_*) & F_t^{U, y_{t01}^L}(\Phi_*) & F_t^{U, y_{l01}^U}(\Phi_*) & F_t^{U, y_{t01}^U}(\Phi_*) \\ G_l^{L, x_{l01}^L}(\Phi_*) & G_l^{L, x_{t01}^L}(\Phi_*) & G_l^{L, x_{l01}^U}(\Phi_*) & G_l^{L, x_{t01}^U}(\Phi_*) & G_l^{L, y_{l01}^L}(\Phi_*) & G_l^{L, y_{t01}^L}(\Phi_*) & G_l^{L, y_{l01}^U}(\Phi_*) & G_l^{L, y_{t01}^U}(\Phi_*) \\ G_t^{L, x_{l01}^L}(\Phi_*) & G_t^{L, x_{t01}^L}(\Phi_*) & G_t^{L, x_{l01}^U}(\Phi_*) & G_t^{L, x_{t01}^U}(\Phi_*) & G_t^{L, y_{l01}^L}(\Phi_*) & G_t^{L, y_{t01}^L}(\Phi_*) & G_t^{L, y_{l01}^U}(\Phi_*) & G_t^{L, y_{t01}^U}(\Phi_*) \\ G_l^{U, x_{l01}^L}(\Phi_*) & G_l^{U, x_{t01}^L}(\Phi_*) & G_l^{U, x_{l01}^U}(\Phi_*) & G_l^{U, x_{t01}^U}(\Phi_*) & G_l^{U, y_{l01}^L}(\Phi_*) & G_l^{U, y_{t01}^L}(\Phi_*) & G_l^{U, y_{l01}^U}(\Phi_*) & G_l^{U, y_{t01}^U}(\Phi_*) \\ G_t^{U, x_{l01}^L}(\Phi_*) & G_t^{U, x_{t01}^L}(\Phi_*) & G_t^{U, x_{l01}^U}(\Phi_*) & G_t^{U, x_{t01}^U}(\Phi_*) & G_t^{U, y_{l01}^L}(\Phi_*) & G_t^{U, y_{t01}^L}(\Phi_*) & G_t^{U, y_{l01}^U}(\Phi_*) & G_t^{U, y_{t01}^U}(\Phi_*) \end{bmatrix},$$

where  $\Phi_* = (x_{l01}^L, x_{t01}^L, x_{l01}^U, x_{t01}^U, y_{l01}^L, y_{t01}^L, y_{l01}^U, y_{t01}^U, \tau)$ ,  $J_* = J_*(\Phi)$ ,  $J_{**} = J_{**}(\Phi_*)$ , the next approximation for  $(x_l^L(\tau), x_t^L(\tau), x_l^U(\tau), x_t^U(\tau), x_l^L(\tau), x_t^L(\tau), x_l^U(\tau), x_t^U(\tau), \tau)$  are found by using a recursive scheme as follows:

$$\begin{bmatrix} x_{l(n+1)}^L(\tau) \\ x_{t(n+1)}^L(\tau) \\ x_{l(n+1)}^U(\tau) \\ x_{t(n+1)}^U(\tau) \\ y_{l(n+1)}^L(\tau) \\ y_{t(n+1)}^L(\tau) \\ y_{l(n+1)}^U(\tau) \\ y_{t(n+1)}^U(\tau) \end{bmatrix} = \begin{bmatrix} x_{l(n)}^L(\tau) \\ x_{t(n)}^L(\tau) \\ x_{l(n)}^U(\tau) \\ x_{t(n)}^U(\tau) \\ y_{l(n)}^L(\tau) \\ y_{t(n)}^L(\tau) \\ y_{l(n)}^U(\tau) \\ y_{t(n)}^U(\tau) \end{bmatrix} + Z * \begin{bmatrix} h_{ln}^L(\tau) \\ h_{tn}^L(\tau) \\ h_{ln}^U(\tau) \\ h_{tn}^U(\tau) \\ k_{ln}^U(\tau) \\ k_{tn}^U(\tau) \\ k_{ln}^U(\tau) \\ k_{tn}^U(\tau) \end{bmatrix}, \tag{10}$$

where

$$\begin{bmatrix} x_{l(n)}^L(\tau) \\ x_{t(n)}^L(\tau) \\ x_{l(n)}^U(\tau) \\ x_{t(n)}^U(\tau) \\ y_{l(n)}^L(\tau) \\ y_{t(n)}^L(\tau) \\ y_{l(n)}^U(\tau) \\ y_{t(n)}^U(\tau) \end{bmatrix} = \begin{bmatrix} x_{l(n)}^L(\tau) \\ x_{t(n)}^L(\tau) \\ x_{l(n)}^U(\tau) \\ x_{t(n)}^U(\tau) \\ y_{l(n)}^L(\tau) \\ y_{t(n)}^L(\tau) \\ y_{l(n)}^U(\tau) \\ y_{t(n)}^U(\tau) \end{bmatrix} + \begin{bmatrix} h_{ln}^L(\tau) \\ h_{tn}^L(\tau) \\ h_{ln}^U(\tau) \\ h_{tn}^U(\tau) \\ k_{ln}^U(\tau) \\ k_{tn}^U(\tau) \\ k_{ln}^U(\tau) \\ k_{tn}^U(\tau) \end{bmatrix},$$

$$Z = (4J_{**}(\Phi_{*n}) - 2J_*(\Phi_n))^{-1} * (J_*(\Phi_n) - J_{**}(\Phi_{*n})),$$

$$J_*(\Phi_n) = \begin{bmatrix} h_{ln}^L(\tau) \\ h_{ln}^L(\tau) \\ h_{ln}^U(\tau) \\ h_{ln}^U(\tau) \\ k_{ln}^U(\tau) \\ k_{ln}^U(\tau) \\ k_{ln}^U(\tau) \\ k_{ln}^U(\tau) \end{bmatrix} = \begin{bmatrix} C_l^L(\tau) - F_l^L(x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U, \tau) \\ C_l^L(\tau) - F_l^L(x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U, \tau) \\ C_l^U(\tau) - F_l^U(x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U, \tau) \\ C_l^U(\tau) - F_l^U(x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U, \tau) \\ D_l^L(\tau) - G_l^L(x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U, \tau) \\ D_l^L(\tau) - G_l^L(x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U, \tau) \\ D_l^U(\tau) - G_l^U(x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U, \tau) \\ D_l^U(\tau) - G_l^U(x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U, \tau) \end{bmatrix},$$

$$J_* = \begin{bmatrix} F_l^L x_{ln}^L(\Phi_n) & F_l^L x_{ln}^L(\Phi_n) & F_l^L x_{ln}^U(\Phi_n) & F_l^L x_{ln}^U(\Phi_n) & F_l^L x_{ln}^L(\Phi_n) & F_l^L y_{ln}^L(\Phi_n) & F_l^L y_{ln}^U(\Phi_n) & F_l^L y_{ln}^U(\Phi_n) \\ F_l^L x_{ln}^L(\Phi_n) & F_l^L x_{ln}^L(\Phi_n) & F_l^L x_{ln}^U(\Phi_n) & F_l^L x_{ln}^U(\Phi_n) & F_l^L x_{ln}^L(\Phi_n) & F_l^L y_{ln}^L(\Phi_n) & F_l^L y_{ln}^U(\Phi_n) & F_l^L y_{ln}^U(\Phi_n) \\ F_l^U x_{ln}^L(\Phi_n) & F_l^U x_{ln}^L(\Phi_n) & F_l^U x_{ln}^U(\Phi_n) & F_l^U x_{ln}^U(\Phi_n) & F_l^U x_{ln}^L(\Phi_n) & F_l^U y_{ln}^L(\Phi_n) & F_l^U y_{ln}^U(\Phi_n) & F_l^U y_{ln}^U(\Phi_n) \\ F_l^U x_{ln}^L(\Phi_n) & F_l^U x_{ln}^L(\Phi_n) & F_l^U x_{ln}^U(\Phi_n) & F_l^U x_{ln}^U(\Phi_n) & F_l^U x_{ln}^L(\Phi_n) & F_l^U y_{ln}^L(\Phi_n) & F_l^U y_{ln}^U(\Phi_n) & F_l^U y_{ln}^U(\Phi_n) \\ G_l^L x_{ln}^L(\Phi_n) & G_l^L x_{ln}^L(\Phi_n) & G_l^L x_{ln}^U(\Phi_n) & G_l^L x_{ln}^U(\Phi_n) & G_l^L x_{ln}^L(\Phi_n) & G_l^L y_{ln}^L(\Phi_n) & G_l^L y_{ln}^U(\Phi_n) & G_l^L y_{ln}^U(\Phi_n) \\ G_l^L x_{ln}^L(\Phi_n) & G_l^L x_{ln}^L(\Phi_n) & F_l^L x_{ln}^U(\Phi_n) & G_l^L x_{ln}^U(\Phi_n) & G_l^L x_{ln}^L(\Phi_n) & G_l^L y_{ln}^L(\Phi_n) & F_l^L y_{ln}^U(\Phi_n) & G_l^L y_{ln}^U(\Phi_n) \\ G_l^U x_{ln}^L(\Phi_n) & G_l^U x_{ln}^L(\Phi_n) & F_l^U x_{ln}^U(\Phi_n) & G_l^U x_{ln}^U(\Phi_n) & G_l^U x_{ln}^L(\Phi_n) & G_l^U y_{ln}^L(\Phi_n) & F_l^U y_{ln}^U(\Phi_n) & G_l^U y_{ln}^U(\Phi_n) \\ G_l^U x_{ln}^L(\Phi_n) & G_l^U x_{ln}^L(\Phi_n) & F_l^U x_{ln}^U(\Phi_n) & G_l^U x_{ln}^U(\Phi_n) & G_l^U x_{ln}^L(\Phi_n) & G_l^U y_{ln}^L(\Phi_n) & F_l^U y_{ln}^U(\Phi_n) & G_l^U y_{ln}^U(\Phi_n) \end{bmatrix}$$

and

$$J_{**} = \begin{bmatrix} F_l^L x_{ln}^L(\Phi_n^*) & F_l^L x_{ln}^L(\Phi_n^*) & F_l^L x_{ln}^U(\Phi_n^*) & F_l^L x_{ln}^U(\Phi_n^*) & F_l^L x_{ln}^L(\Phi_n^*) & F_l^L y_{ln}^L(\Phi_n^*) & F_l^L y_{ln}^U(\Phi_n^*) & F_l^L y_{ln}^U(\Phi_n^*) \\ F_l^L x_{ln}^L(\Phi_n^*) & F_l^L x_{ln}^L(\Phi_n^*) & F_l^L x_{ln}^U(\Phi_n^*) & F_l^L x_{ln}^U(\Phi_n^*) & F_l^L x_{ln}^L(\Phi_n^*) & F_l^L y_{ln}^L(\Phi_n^*) & F_l^L y_{ln}^U(\Phi_n^*) & F_l^L y_{ln}^U(\Phi_n^*) \\ F_l^U x_{ln}^L(\Phi_n^*) & F_l^U x_{ln}^L(\Phi_n^*) & F_l^U x_{ln}^U(\Phi_n^*) & F_l^U x_{ln}^U(\Phi_n^*) & F_l^U x_{ln}^L(\Phi_n^*) & F_l^U y_{ln}^L(\Phi_n^*) & F_l^U y_{ln}^U(\Phi_n^*) & F_l^U y_{ln}^U(\Phi_n^*) \\ F_l^U x_{ln}^L(\Phi_n^*) & F_l^U x_{ln}^L(\Phi_n^*) & F_l^U x_{ln}^U(\Phi_n^*) & F_l^U x_{ln}^U(\Phi_n^*) & F_l^U x_{ln}^L(\Phi_n^*) & F_l^U y_{ln}^L(\Phi_n^*) & F_l^U y_{ln}^U(\Phi_n^*) & F_l^U y_{ln}^U(\Phi_n^*) \\ G_l^L x_{ln}^L(\Phi_n^*) & G_l^L x_{ln}^L(\Phi_n^*) & G_l^L x_{ln}^U(\Phi_n^*) & G_l^L x_{ln}^U(\Phi_n^*) & G_l^L x_{ln}^L(\Phi_n^*) & G_l^L y_{ln}^L(\Phi_n^*) & G_l^L y_{ln}^U(\Phi_n^*) & G_l^L y_{ln}^U(\Phi_n^*) \\ G_l^L x_{ln}^L(\Phi_n^*) & G_l^L x_{ln}^L(\Phi_n^*) & F_l^L x_{ln}^U(\Phi_n^*) & G_l^L x_{ln}^U(\Phi_n^*) & G_l^L x_{ln}^L(\Phi_n^*) & G_l^L y_{ln}^L(\Phi_n^*) & F_l^L y_{ln}^U(\Phi_n^*) & G_l^L y_{ln}^U(\Phi_n^*) \\ G_l^U x_{ln}^L(\Phi_n^*) & G_l^U x_{ln}^L(\Phi_n^*) & F_l^U x_{ln}^U(\Phi_n^*) & G_l^U x_{ln}^U(\Phi_n^*) & G_l^U x_{ln}^L(\Phi_n^*) & G_l^U y_{ln}^L(\Phi_n^*) & F_l^U y_{ln}^U(\Phi_n^*) & G_l^U y_{ln}^U(\Phi_n^*) \\ G_l^U x_{ln}^L(\Phi_n^*) & G_l^U x_{ln}^L(\Phi_n^*) & F_l^U x_{ln}^U(\Phi_n^*) & G_l^U x_{ln}^U(\Phi_n^*) & G_l^U x_{ln}^L(\Phi_n^*) & G_l^U y_{ln}^L(\Phi_n^*) & F_l^U y_{ln}^U(\Phi_n^*) & G_l^U y_{ln}^U(\Phi_n^*) \end{bmatrix}$$

where  $\Phi_n = (x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U, \tau)$ ,  $\Phi_n^* = (x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U, \tau)$  and  $\forall \tau \in [0, 1]$ .

For initial guess, one can use the fuzzy number

$$x_0 = (x_{l0}^U(0), x_{l0}^L(0), x_{l0}^L(1), x_{l0}^L(1), x_{l0}^U(1), x_{l0}^U(1), x_{l0}^L(0)x_{l0}^U(0), (y_{l0}^U(0), y_{l0}^L(0), y_{l0}^L(1), y_{l0}^L(1), y_{l0}^U(1), y_{l0}^U(1), y_{l0}^L(0)y_{l0}^U(0)). \tag{11}$$

**Remark 1.** Sequence  $\{ (x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U) \}_{n=0}^\infty$  converges to  $(\alpha_l^L, \alpha_l^L, \alpha_l^U, \alpha_l^U, \beta_l^L, \beta_l^L, \beta_l^U, \beta_l^U)$  iff  $\forall \tau \in [0, 1]$ ,  $\lim_{n \rightarrow \infty} x_{ln}^L(\tau) = \alpha_l^L(\tau)$ ,  $\lim_{n \rightarrow \infty} x_{ln}^L(\tau) = \alpha_l^L(\tau)$ ,  $\lim_{n \rightarrow \infty} x_{ln}^U(\tau) = \alpha_l^U(\tau)$ ,  $\lim_{n \rightarrow \infty} y_{ln}^L(\tau) = \beta_l^L(\tau)$ ,  $\lim_{n \rightarrow \infty} y_{ln}^L(\tau) = \beta_l^L(\tau)$ ,  $\lim_{n \rightarrow \infty} y_{ln}^U(\tau) = \beta_l^U(\tau)$  and  $\lim_{n \rightarrow \infty} y_{ln}^U(\tau) = \beta_l^U(\tau)$

**Lemma 3.1.** Let  $F(\alpha_l^L, \alpha_l^L, \alpha_l^U, \alpha_l^U, \beta_l^L, \beta_l^L, \beta_l^U, \beta_l^U) = (C_l^L, C_l^L, C_l^U, C_l^U, D_l^L, D_l^L, D_l^U, D_l^U)$  and if the sequence of  $\{ (x_{ln}^L, x_{ln}^L, x_{ln}^U, x_{ln}^U, y_{ln}^L, y_{ln}^L, y_{ln}^U, y_{ln}^U) \}_{n=0}^\infty$  converges to  $(\alpha_l^L, \alpha_l^L, \alpha_l^U, \alpha_l^U, \beta_l^L, \beta_l^L, \beta_l^U, \beta_l^U)$  according to  $M$ , then

$$\lim_{n \rightarrow \infty} P_n = 0, \tag{12}$$

where

$$P_n = \sup_{0 \leq \tau \leq 1} \max \{ \Psi \} \tag{13}$$

and  $\Psi = h_{ln}^L(\tau) + h_{ln}^L(\tau) + h_{ln}^L(\tau) + h_{ln}^L(\tau) + k_{ln}^U(\tau) + k_{ln}^U(\tau) + k_{ln}^U(\tau) + k_{ln}^U(\tau)$ .

**Proof.** It is obvious, because for all  $\forall \tau \in [0, 1]$  in convergent case,

$$\lim_{n \rightarrow \infty} h_{ln}^L(\tau) = \lim_{n \rightarrow \infty} h_{ln}^L(\tau) = \lim_{n \rightarrow \infty} h_{ln}^L(\tau) = \lim_{n \rightarrow \infty} h_{ln}^L(\tau) = 0, \\ \lim_{n \rightarrow \infty} k_{ln}^U(\tau) = \lim_{n \rightarrow \infty} k_{ln}^U(\tau) = \lim_{n \rightarrow \infty} k_{ln}^U(\tau) = \lim_{n \rightarrow \infty} k_{ln}^U(\tau) = 0,$$

Hence the result.  $\square$

Finally, it is shown that under certain conditions SFQM1 method (14) for fuzzy equation  $F(x) = 0$  is cubic convergent. In compact form for  $F(x) = 0$  SFQM1 method can be written as:

$$\begin{cases} y_n(\tau) = x_n(\tau) - (F'(x_n(\tau)))^{-1} F(x_n(\tau)), \\ x_{n+1}(\tau) = y_n(\tau) - Z * (F'(x_n(\tau)))^{-1} F(x_n(\tau)), \end{cases} \quad \forall \tau \in [0, 1] \tag{14}$$

where

$$Z = (4J_{**} - 2J_*)^{-1} * (J_* - J_{**}).$$

**Theorem 3.1.** Let  $F : H \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ , be  $u$ -times Fréchet differential function on a convex set  $H$  containing the root  $\alpha$  of  $F(x) = 0$ , then the MM method has cubic convergence and satisfies the following error equation.

$$e_{n+1} = 2 * \left( (\hat{C}_2)^2 - \frac{1}{2} \hat{C}_3 - 4(\hat{C}_2)^2 \right) (e_n)^3 + \|O(e_n)^4\|, \tag{15}$$

where  $\hat{C}_m = \frac{1}{m!} * \frac{F^m(x_n, \tau)}{F'(x_n, \tau)}$ ,  $m = 2, 3, \dots$

**Proof.** Let  $e_n = r_n - \alpha$  and  $e_{n+1} = r_{n+1} - \alpha$  then by Taylor series of  $F(r_n, \tau)$  in the neighborhood of  $\alpha$  if  $J_*$  and  $J_{**}$  exist, then

$$F(r_n, \tau) = F(r_n, \tau) + F'(r_n, \tau)(r - r_n) + \frac{1}{2!} F''(r_n, \tau)(r - r_n)^2 + \dots \tag{16}$$

and  $F(r, \alpha) = 0$ ,

$$F(r_n, \tau) = F'(r, \xi)(e_n + \hat{C}_2(e_n)^2 + \hat{C}_3(e_n)^3) + \|O(e_n)^4\|. \tag{17}$$

Dividing (16) by  $F'(r_n, \tau)$ , gives

$$(F'(r_n, \tau))^{-1} F(r_n, \tau) = e_n + \hat{C}_2(e_n)^2 + (2\hat{C}_2 + 2\hat{C}_3)(e_n)^3 + \dots \tag{18}$$

$$y_n - \alpha = \hat{C}_2(e_n)^2 + (-2\hat{C}_2 + 2\hat{C}_3)(e_n)^3 + \dots \tag{19}$$

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

$$F'(y_n, \tau) = 1 + 2(\hat{C}_2)^2(e_n)^2 + 2(-2(\hat{C}_2)^2 + 2\hat{C}_3)(e_n)^3 + \dots \tag{20}$$

and

$$Z * (F'(r_n, \tau))^{-1} F(r_n, \tau) = -\hat{C}_2(e_n)^2 + \left( 4(\hat{C}_2)^2 - \frac{3}{2} \hat{C}_3 - 4(\hat{C}_2)^2 \right) (e_n)^3 + \dots \tag{21}$$

$$r_{n+1} - \alpha = y_n - \alpha - \hat{C}_2(e_n)^2 + \left( 4(\hat{C}_2)^2 - \frac{3}{2} \hat{C}_3 - 4(\hat{C}_2)^2 \right) (e_n)^3 + \dots \tag{22}$$

$$e_{n+1} = 2 * \left( (\hat{C}_2)^2 - \frac{1}{2} \hat{C}_3 - 4(\hat{C}_2)^2 \right) (e_n)^3 + \|O(e_n)^4\|. \tag{23}$$

Hence the theorem is proved.  $\square$

We modify classical Newton Raphson’s method (SFQN1) [44] for solving IVTFNSEs as follows:

$$\begin{bmatrix} x_l^L(n+1)(\tau) \\ x_l^L(n+1)(\tau) \\ x_l^U(n+1)(\tau) \\ x_l^U(n+1)(\tau) \\ y_l^L(n+1)(\tau) \\ y_l^L(n+1)(\tau) \\ y_l^U(n+1)(\tau) \\ y_l^U(n+1)(\tau) \end{bmatrix} = \begin{bmatrix} x_l^L(n)(\tau) \\ x_l^L(n)(\tau) \\ x_l^U(n)(\tau) \\ x_l^U(n)(\tau) \\ y_l^L(n)(\tau) \\ y_l^L(n)(\tau) \\ y_l^U(n)(\tau) \\ y_l^U(n)(\tau) \end{bmatrix} + \begin{bmatrix} h_l^L(\tau) \\ h_l^L(\tau) \\ h_l^U(\tau) \\ h_l^U(\tau) \\ k_l^U(\tau) \\ k_l^U(\tau) \\ k_l^U(\tau) \\ k_l^U(\tau) \end{bmatrix}, \tag{24}$$

where

$$J_{**}(\Phi_n) = \begin{bmatrix} h_l^L(\tau) \\ h_l^L(\tau) \\ h_l^U(\tau) \\ h_l^U(\tau) \\ k_l^U(\tau) \\ k_l^U(\tau) \\ k_l^U(\tau) \\ k_l^U(\tau) \end{bmatrix} = \begin{bmatrix} C_l^L(\tau) - F_l^L(\Phi_n) \\ C_l^L(\tau) - F_l^L(\Phi_n) \\ C_l^U(\tau) - F_l^U(\Phi_n) \\ C_l^U(\tau) - F_l^U(\Phi_n) \\ D_l^L(\tau) - G_l^L(\Phi_n) \\ D_l^L(\tau) - G_l^L(\Phi_n) \\ D_l^U(\tau) - G_l^U(\Phi_n) \\ D_l^U(\tau) - G_l^U(\Phi_n) \end{bmatrix}.$$

The numerical scheme (24) has local quadratic convergence.

The steps followed for solving the nonlinear system using iterative scheme SFQM1 Method are stated as below. The flow chart of the proposed SFQM1 scheme is shown in Fig. 2.

### 4. Numerical applications

In this section, we considered some numerical examples [44]-[45] to illustrate the performance and efficiency of proposed SFQM1, and SFQN1 techniques for solving IVTFNSEs. Algorithm 1, and flow chart represents Fig. 2, shows the implementation of SFQM1 iterative method on CAS-Maple 18.

- (i)  $e_n = \|F(x, \tau)\| < \epsilon = 10^{-15}$  (ii)  $e_n = \|x_{n+1}(\tau) - x_n(\tau)\| < \epsilon = 10^{-15}$
- (iii)  $e_n = \frac{\|x_{n+1}(\tau) - x_n(\tau)\|}{\|x_n(\tau)\|} \times 100 < \epsilon = 10^{-15}$ .

**Algorithm 1** Proposed iterative: **SFQM1 method.**

Step 1: Transform

$$F(x) = \begin{cases} F(x) = C \\ G(x) = D \end{cases}$$

into

$$\begin{cases} F_l^L(x_l^L, x_l^L, x_l^U, x_l^U, y_l^L, y_l^L, y_l^U, y_l^U, \tau) = C_l^L(\tau), \\ F_t^L(x_t^L, x_t^L, x_t^U, x_t^U, y_t^L, y_t^L, y_t^U, y_t^U, \tau) = C_t^L(\tau), \\ F_l^U(x_l^L, x_l^L, x_l^U, x_l^U, y_l^L, y_l^L, y_l^U, y_l^U, \tau) = C_l^U(\tau), \\ F_t^U(x_t^L, x_t^L, x_t^U, x_t^U, y_t^L, y_t^L, y_t^U, y_t^U, \tau) = C_t^U(\tau), \\ G_l^L(x_l^L, x_l^L, x_l^U, x_l^U, y_l^L, y_l^L, y_l^U, y_l^U, \tau) = D_l^L(\tau), \\ G_t^L(x_t^L, x_t^L, x_t^U, x_t^U, y_t^L, y_t^L, y_t^U, y_t^U, \tau) = D_t^L(\tau), \\ G_l^U(x_l^L, x_l^L, x_l^U, x_l^U, y_l^L, y_l^L, y_l^U, y_l^U, \tau) = D_l^U(\tau), \\ G_t^U(x_t^L, x_t^L, x_t^U, x_t^U, y_t^L, y_t^L, y_t^U, y_t^U, \tau) = D_t^U(\tau). \end{cases}$$

$$\forall \tau \in [0, 1].$$

Step 2: Solve

$$\begin{cases} F_l^L(x_l^L, x_l^L, x_l^U, x_l^U, y_l^L, y_l^L, y_l^U, y_l^U, \tau) = C_l^L(\tau), \\ F_t^L(x_t^L, x_t^L, x_t^U, x_t^U, y_t^L, y_t^L, y_t^U, y_t^U, \tau) = C_t^L(\tau), \\ F_l^U(x_l^L, x_l^L, x_l^U, x_l^U, y_l^L, y_l^L, y_l^U, y_l^U, \tau) = C_l^U(\tau), \\ F_t^U(x_t^L, x_t^L, x_t^U, x_t^U, y_t^L, y_t^L, y_t^U, y_t^U, \tau) = C_t^U(\tau), \\ G_l^L(x_l^L, x_l^L, x_l^U, x_l^U, y_l^L, y_l^L, y_l^U, y_l^U, \tau) = D_l^L(\tau), \\ G_t^L(x_t^L, x_t^L, x_t^U, x_t^U, y_t^L, y_t^L, y_t^U, y_t^U, \tau) = D_t^L(\tau), \\ G_l^U(x_l^L, x_l^L, x_l^U, x_l^U, y_l^L, y_l^L, y_l^U, y_l^U, \tau) = D_l^U(\tau), \\ G_t^U(x_t^L, x_t^L, x_t^U, x_t^U, y_t^L, y_t^L, y_t^U, y_t^U, \tau) = D_t^U(\tau). \end{cases}$$

for  $\tau = 0$  and  $\tau = 1$  to obtain the initial guess value.

Step 3: Evaluate

$$F(x, \tau) = c,$$

at initial starting values and approximate the Jacobin matrix  $\mathbf{J}_*, \mathbf{J}_{**}$ .

Step 4: Use SFQM1 to compute the next iteration

$$\begin{cases} \mathbf{y}_n(\tau) = \mathbf{x}_n(\tau) - (\mathbf{F}'(\mathbf{x}_n(\tau)))^{-1} \mathbf{F}(\mathbf{x}_n(\tau)), \\ \mathbf{x}_{n+1}(\tau) = \mathbf{y}_n(\tau) - \mathbf{Z} * (\mathbf{F}'(\mathbf{x}_n(\tau)))^{-1} \mathbf{F}(\mathbf{x}_n(\tau)), \end{cases} \quad \forall \tau \in [0, 1]$$

where

$$\mathbf{Z} = (4\mathbf{J}_{**} - 2\mathbf{J}_*)^{-1} * (\mathbf{J}_* - \mathbf{J}_{**}).$$

Step 5: For given  $\epsilon > 0$ , if

$$(i) \quad \epsilon_n = \|\mathbf{F}(\mathbf{x}, \tau)\| < \epsilon \quad \text{or} \quad (ii) \quad \epsilon_n = \|\mathbf{x}_{n+1}(\tau) - \mathbf{x}_n(\tau)\| < \epsilon$$

then stop.

Step 6: Set  $k = k + 1$  and go to step 1.

**4.1. Applications in economics: supply chain problems**

The supply function equilibrium (SFE) framework is appealing for modeling strategic interactions among electricity suppliers. The submission of supply functions prior to the fulfillment of uncertain demand is common in structured electricity markets. If the SFE model is a strong predictor of oligopoly behavior, it must be thoroughly understood for market outcomes and performance monitoring. Incorporating uncertainty and ambiguity, however, theoretical research and computational analysis have demonstrated considerable complexity. Therefore, we obtained the following system autonomous system of equations: the solution of this system resulted in the following system of nonlinear IVTFNSEs:

$$\begin{cases} X(t) = A_{11}e^{\eta_1 t}v_1 + A_{12}e^{\eta_2 t}v_2, \\ Y(t) = a + A_{21}e^{\eta_1 t} + A_{22}e^{\eta_2 t}, \end{cases} \tag{25}$$

where  $A_{11}, A_{12}, A_{21}, A_{22}, v_1, v_2, \eta_1, \eta_2$  are IVTF numbers. In general, we get the following IVTFNSEs as:

$$\begin{cases} G = A_{11}e^{\eta_1 x}v_1 + A_{12}e^{\eta_2 y}, \\ C = A_{21}e^{\eta_1 x} + A_{22}e^{\eta_2 y}. \end{cases} \tag{26}$$

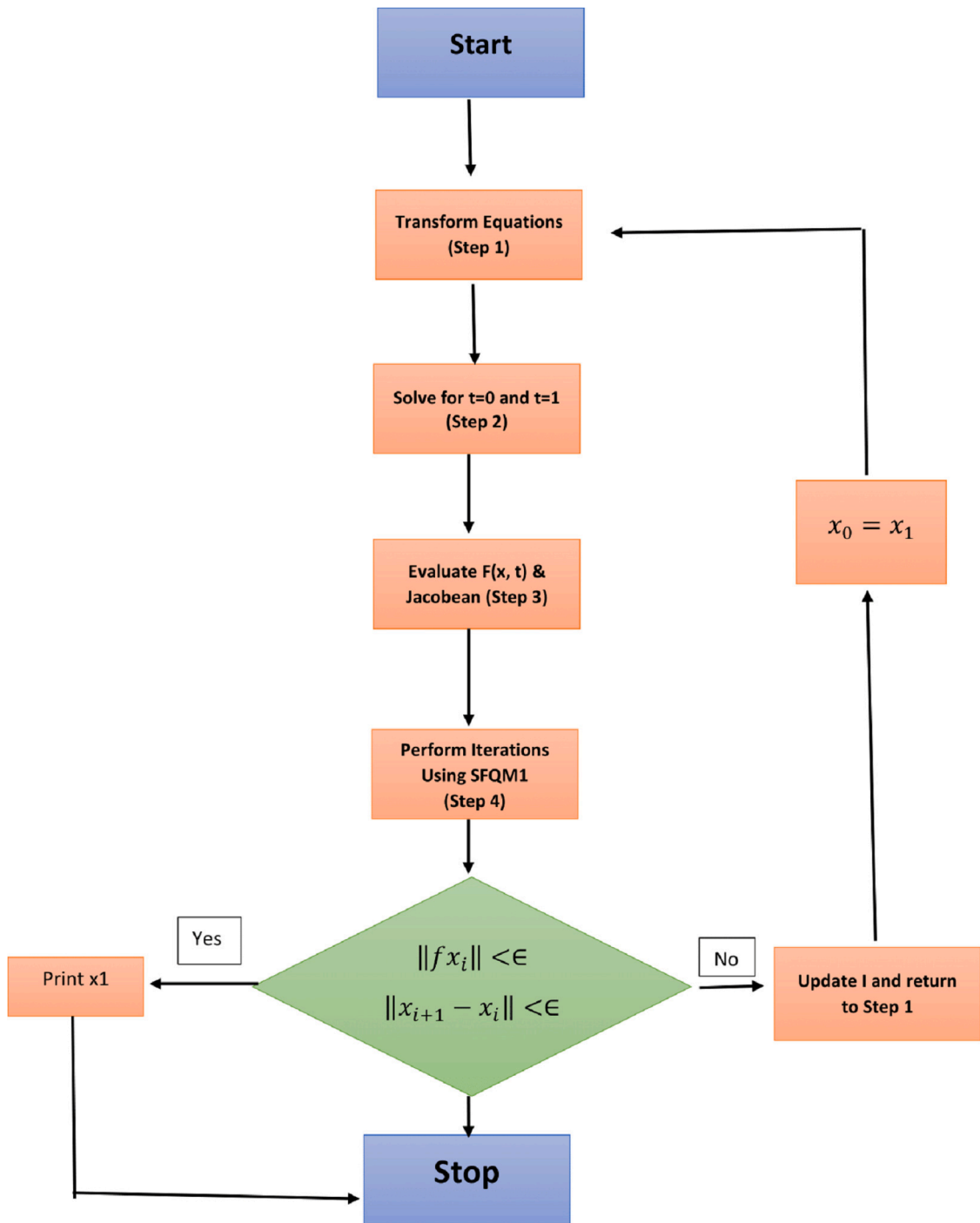


Fig. 2. Flow chart of the proposed SFQM1 scheme for solving IVTFNSEs.

Similarly, we discuss various problems in economics involving a system of nonlinear algebraic or transcendental IVTF numbers, such as demonstrating the uniqueness of equilibrium in some models of trade with increasing returns.

$$\begin{cases} \frac{AX_1}{AX_1^2 + X_2^2} + \frac{x_1}{x_1^2 + AX_2^2} = G_1, \\ \frac{X_2}{AX_1^2 + X_2^2} + \frac{BX_2}{x_1^2 + AX_2^2} = C_1, \end{cases} \tag{27}$$

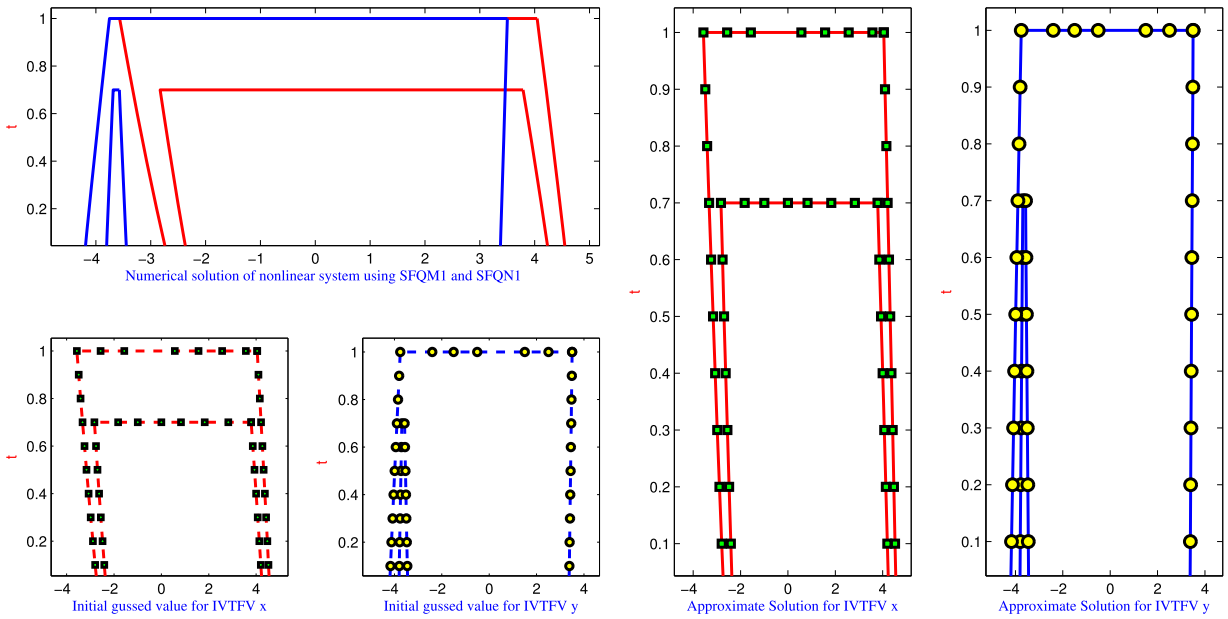


Fig. 3. Approximate solution of IVTFNEs used in Example 1.

or

$$\begin{cases} A_{11}x^2 + B_{11}y^2 = G, \\ A_{22}x^2 + B_{22}y^2 = G, \end{cases} \tag{28}$$

where  $A_{11} = AX_1, B_{11} = X_1, A_{22} = X_2, B_{22} = AX_2, x^2 = \frac{1}{AX_1^2 + X_2^2}$  and  $y^2 = \frac{1}{X_1^2 + AX_2^2}$ .

Here, we consider different cases of the above IVTSFs to find the approximate solution using numerical techniques SFQM1 and SFQN1 respectively.

**Example 1:** Consider the nonlinear system of IVTFNEs in which the coefficient matrix is real valued crisp number and the right side column vector contains IVTF number as

$$\begin{cases} 3x^2 - y^2 = \langle (10, 20, 30, 40; \frac{2}{3}), (5, 15, 35, 45; 1) \rangle, \\ 2x^2 + 3y^2 = \langle (50, 60, 70, 80; \frac{2}{3}), (45, 55, 75, 95; 1) \rangle. \end{cases}$$

Analytical methods are not capable of handling this type of highly nonlinear IVTFNEs. We thus use SFQM1 and SFQN1. Without any loss of generality, assume that  $x, y$  are positive and so the parametric form of this equation is as follows:

$$\begin{cases} 3x^2 - y^2 = \langle (10 + 15\tau, 40 - 15\tau), (5 + 10\tau, 45 - 10\tau) \rangle, \\ 2x^2 + 3y^2 = \langle (50 + 15\tau, 80 - 15\tau), (45 + 10\tau, 95 - 20\tau) \rangle, \end{cases}$$

or

$$\begin{cases} 3(\cdot x_t^L(\tau))^2 - (\cdot y_t^L(\tau))^2 = 10 + 15\tau, \\ 3(\cdot x_t^U(\tau))^2 - (\cdot y_t^U(\tau))^2 = 40 - 15\tau, \\ 2(\cdot\cdot x_t^L(\tau))^2 + 3(\cdot\cdot y_t^L(\tau))^2 = 50 + 15\tau, \\ 2(\cdot\cdot x_t^U(\tau))^2 + 3(\cdot\cdot y_t^U(\tau))^2 = 80 - 15\tau, \\ 2(\cdot\cdot\cdot x_t^L(\tau))^2 + 3(\cdot\cdot\cdot y_t^L(\tau))^2 = 45 + 10\tau, \\ 2(\cdot\cdot\cdot x_t^U(\tau))^2 + 3(\cdot\cdot\cdot y_t^U(\tau))^2 = 95 - 20\tau. \end{cases}$$

Fig. 3 depicts the numerical solution of IVTFNEs, with the initial starting values and solution of the IVTF variables  $x$  and  $y$  utilized in Example 1. Also, this figure clearly shows numerical approximations obtained with SFQM1 that are matched with analytical solutions up to 36 decimal places.

**Table 1**  
Approximate solution of Example 1 using SFQM1 and SFQN1.

$\tau$	$\dot{x}_t^L(\tau)$	$\ddot{x}_t^L(\tau)$	$\dot{x}_t^U(\tau)$	$\ddot{x}_t^U(\tau)$	$\dot{y}_t^L(\tau)$	$\ddot{y}_t^L(\tau)$	$\dot{y}_t^U(\tau)$	$\ddot{y}_t^U(\tau)$
0.0	-2.6967	4.2640	-2.3354	4.5726	-4.2103	-3.8138	-3.4377	3.3709
0.1	-2.7961	4.1995	-2.4120	4.5226	-4.1669	-3.7959	-3.4575	3.3844
0.2	-2.8919	4.1341	-2.4863	4.4721	-4.1231	-3.7779	-3.4771	3.3978
0.3	-2.9848	4.0676	-2.5584	4.4210	-2.0787	-3.7598	-3.4967	3.4112
0.4	-3.0748	4.000	-2.6285	4.3693	-4.0339	-3.7416	-3.5161	3.4245
0.5	-3.1622	3.9312	-2.6967	4.3169	-3.9886	-3.7233	-3.5355	3.4377
0.6	-3.2473	3.8612	-2.7633	4.2640	-3.9427	-3.7050	-3.5547	3.4509
0.7	-3.3303	3.7899	-2.8284	4.2103	-3.8963	-3.7665	-3.5738	3.4641
0.8	-3.112	3.7899	1.8284	4.1560	-3.0494	-3.6665	-3.5735	3.4771
0.9	-3.4920	3.7899	2.1425	4.1009	-3.8019	-3.5241	-3.5732	3.4902
1.0	-3.4902	3.7899	3.7899	4.0451	-3.7537	-3.5731	-3.5731	3.5032

**Table 2**  
Error comparison of the numerical scheme SFQM1 and SFQN1.

Scheme	SFQM1				SFQN1			
	C1		C2		C1		C2	
Criteria	x	y	x	y	x	y	x	y
$\tau$								
0.0	2.1e-11	1.2e-16	1.5e-26	0.4e-20	0.9e-11	3.8e-12	4.6e-26	2.1e-26
0.1	2.0e-21	1.2e-15	2.3e-22	3.9e-21	0.3e-12	9.6e-15	1.2e-25	1.2e-25
0.2	3.2e-10	6.3e-15	3.1e-21	1.5e-22	0.8e-14	3.6e-11	3.1e-21	3.5e-23
0.3	5.6e-0	4.5e-14	8.9e-25	6.3e-25	0.9e-17	2.0e-14	1.2e-22	3.9e-25
0.4	4.5e-09	9.6e-14	0.9e-24	1.0e-21	0.3e-15	1.8e-15	6.3e-24	3.4e-22
0.5	6.9e-10	4.5e-11	3.6e-27	1.3e-23	1.5e-14	1.6e-15	0.3e-28	3.5e-24
0.6	0.3e-11	6.3e-10	3.6e-28	3.2e-22	2.6e-18	1.5e-16	1.0e-29	3.6e-23
0.7	5.6e-12	4.5e-14	3.6e-25	1.2e-21	1.6e-16	1.2e-10	0.3e-25	1.2e-20
0.8	1.0e-12	6.3e-15	5.6e-24	3.6e-22	3.0e-15	3.5e-11	0.9e-24	1.5e-24
0.9	5.9e-14	3.0e-11	0.3e-29	3.2e-21	1.6e-14	3.0e-12	9.0e-26	0.5e-21
1.0	4.1e-18	1.8e-10	1.0e-28	6.3e-24	2.0e-15	3.6e-10	0.8e-25	0.9e-28

For the initial guess we solve the system for  $\tau = 0$  and  $\tau = 1$ , therefore

$$\begin{cases} 3(\dot{x}_t^L(0))^2 - (\dot{y}_t^L(0))^2 = 10 \\ 3(\dot{x}_t^L(1))^2 - (\dot{y}_t^L(1))^2 = 25 \\ 3(\dot{x}_t^U(0))^2 - (\dot{y}_t^U(0))^2 = 5 \\ 3(\dot{x}_t^U(1))^2 - (\dot{y}_t^U(1))^2 = 15 \\ 2(\ddot{x}_t^L(0))^2 + 3(\ddot{y}_t^L(0))^2 = 50 \\ 2(\ddot{x}_t^L(1))^2 + 3(\ddot{y}_t^L(1))^2 = 65 \\ 2(\ddot{x}_t^U(0))^2 + 3(\ddot{y}_t^U(0))^2 = 45 \\ 2(\ddot{x}_t^U(1))^2 + 3(\ddot{y}_t^U(1))^2 = 55 \end{cases} \quad \text{and} \quad \begin{cases} 3(\dot{x}_t^L(0))^2 - (\dot{y}_t^L(0))^2 = 40 \\ 3(\dot{x}_t^L(1))^2 - (\dot{y}_t^L(1))^2 = 25 \\ 3(\dot{x}_t^U(0))^2 - (\dot{y}_t^U(0))^2 = 5 \\ 3(\dot{x}_t^U(1))^2 - (\dot{y}_t^U(1))^2 = 15 \\ 2(\ddot{x}_t^L(0))^2 + 3(\ddot{y}_t^L(0))^2 = 80 \\ 2(\ddot{x}_t^L(1))^2 + 3(\ddot{y}_t^L(1))^2 = 65 \\ 2(\ddot{x}_t^U(0))^2 + 3(\ddot{y}_t^U(0))^2 = 45 \\ 2(\ddot{x}_t^U(1))^2 + 3(\ddot{y}_t^U(1))^2 = 55 \end{cases}$$

Consequently  $x_t^U(0) = 0.5, x_t^L(0) = 0.5, x_t^L(1) = 0.5, x_t^U(1) = 0.5$  and  $x_t^U(0) = x_t^L(0) = x_t^L(1) = x_t^U(1) = \frac{1}{2}$ . After 4 iterations, we obtain the solution with a maximum error less than  $10^{-30}$ . Now suppose  $x, y$  are negative, we have

$$\begin{cases} 3(\dot{x}_t^L(\tau))^2 - (\dot{y}_t^L(\tau))^2 = 10 + 15\tau, \\ 3(\dot{x}_t^L(\tau))^2 - (\dot{y}_t^L(\tau))^2 = 40 - 15\tau, \\ 3(\dot{x}_t^U(\tau))^2 - (\dot{y}_t^U(\tau))^2 = 5 + 10\tau, \\ 3(\dot{x}_t^U(\tau))^2 - (\dot{y}_t^U(\tau))^2 = 45 - 10\tau, \\ 2(\ddot{x}_t^L(\tau))^2 + 3(\ddot{y}_t^L(\tau))^2 = 50 + 15\tau, \\ 2(\ddot{x}_t^L(\tau))^2 + 3(\ddot{y}_t^L(\tau))^2 = 80 - 15\tau, \\ 2(\ddot{x}_t^U(\tau))^2 + 3(\ddot{y}_t^U(\tau))^2 = 45 + 10\tau, \\ 2(\ddot{x}_t^U(\tau))^2 + 3(\ddot{y}_t^U(\tau))^2 = 95 - 20\tau. \end{cases}$$

For  $\tau = 0$ , we have,  $y_t^L(0) > y_t^L(0)$ , therefore negative solutions of IVTFNSEs does not exist.

Table 1 shows the numerical approximate solution of the interval-valued trapezoidal system of nonlinear equations used in Example 1 by utilizing numerical schemes SFQM1 and SFQN1 up to 30 decimal places with 64-digit floating point arithmetic. Table displays the approximated solution to four decimal places. The numerical approximation solution presented in Table 1 agrees with the analytical solution illustrated in Fig. 3.

In Table 2, C1, C2 and C3 represents the stopping criteria 1, 2 and 3 i.e.,  $\|F(x, \tau)\|, \|x_{n+1}(\tau) - x_n(\tau)\|$ , and  $\frac{\|x_{n+1}(\tau) - x_n(\tau)\|}{\|x_n(\tau)\|} \times 100$  for different values of  $\tau$  respectively. The approximate IVTFNE solutions presented in this table clearly show that our newly developed family is more efficient than the classical method used in the literature.

**Table 3**  
Stability-consistency analysis.

Method	SFQM1	SFQN1
M-Error <sub>C1</sub>	3.1e-47	4.1e-10
M-Error <sub>C2</sub>	1.2e-36	1.6e-8
M-CPU-Time <sub>C1</sub>	0.0114	0.0427
M-CPU-Time <sub>C2</sub>	0.0121	0.0327
COC	2.912	1.901

**Table 4**  
Percentage convergence and total function evaluations for SFQM1, SFQN1.

Method	Num- F-J	Percentage-Con	M-CPU-Time <sub>C3</sub>	M-Error <sub>C3</sub>	COC
SFQM1	7	86.6575756	1.34635643	0.543443e-16	2.0344
SFQN1	13	65.8665456	3.56464674	4.214345e-09	1.5674

In Table 3, M-Error<sub>C1</sub>, M-Error<sub>C2</sub> show the error of numerical schemes SFQM1 and SFQN1 using stopping criteria I and II, respectively, M-CPU-Time<sub>C1</sub>, M-CPU-Time<sub>C2</sub> present the computational time in seconds of the numerical schemes for solving the solution of a fuzzy nonlinear system used in Example 1, employing stopping criteria I and II, and COC shows the computational order of convergence of the SFQM1 and SFQN1 methods. Thus, Tables 1–3 demonstrates that our newly developed method, SFQM1, is better than SFQN1 in terms of residual error, CPU time, and computational order of convergence. A numerical method is considered stable if it prevents numerical solution errors from being amplified. When the solution produced by a technique is unaffected by small changes in the inputs and parameters, and when it is expected that changes in the parameters will have an impact on equations and conditions, the method is said to be consistent if the overall error of the numerical scheme is under control. Further, Table 3 shows that the SFQM1 method is more stable and consistent than the SFQN1 method.

In Table 4, Num- F-J represents the total number of functions and Jacobine evaluations each iteration, M-Error<sub>C3</sub>, shows the maximum error, Percentage-Con represents the percentage convergence, and M-CPU-Time<sub>C3</sub> represents the computational time required to compute the percentage convergence and total functions evaluations using stopping criteria C3 i.e.,  $\frac{\|x_{n+1}(\tau) - x_n(\tau)\|}{\|x_n(\tau)\|} \times 100$ . Table 4 also demonstrates that in terms of percentage convergence, total functions, and Jacob’s evaluations, our technique SFQM1 outperforms the existing method SFQN1 and provides an alternative way to solve complex IVTFNE for greater comprehension and analysis.

**Example 2:** Consider the nonlinear system of IVTFNEs in which the coefficient matrix and right side column vector both contains IVTF number as

$$\left\{ \begin{aligned} & \left\langle (0.65, 0.73, 0.87, 0.95; \frac{2}{3}), (0.6, 0.7, 0.9, 1.1; 1) \right\rangle x^3 - \left\langle (0.25, 0.33, 0.47, 0.55; \frac{2}{3}), (0.2, 0.3, 0.5, 0.6; 1) \right\rangle y^2 \\ & = \left\langle (0.45, 0.53, 0.67, 0.75; \frac{2}{3}), (0.4, 0.5, 0.7, 0.8; 1) \right\rangle, \\ & \left\langle (0.45, 0.53, 0.67, 0.75; \frac{2}{3}), (0.4, 0.5, 0.7, 0.8; 1) \right\rangle x^4 + \left\langle (0.65, 0.73, 0.87, 0.95; \frac{2}{3}), (0.6, 0.7, 0.9, 1; 1) \right\rangle y^3 \\ & = \left\langle (0.25, 0.37, 0.47, 0.55; \frac{2}{3}), (0.2, 0.3, 0.5, 0.6; 1) \right\rangle. \end{aligned} \right.$$

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

$$\left\{ \begin{aligned} & \langle (0.65 + 0.12\tau, 0.95 - 0.12\tau), (0.6 + 0.1\tau, 1 - 0.1\tau) \rangle x^3 - \langle (0.25 + 0.12\tau, 0.55 - 0.12\tau), (0.2 + 0.1\tau, 0.6 - 0.1\tau) \rangle y^2 \\ & = \langle (0.45 + 0.12\tau, 0.75 - 0.12\tau), (0.4 + 0.1\tau, 0.8 - 0.1\tau) \rangle, \\ & \langle (0.45 + 0.12\tau, 0.75 - 0.12\tau), (0.4 + 0.1\tau, 0.8 - 0.1\tau) \rangle x^4 + \langle (0.65 + 0.12\tau, 0.95 - 0.12\tau), (0.6 + 0.1\tau, 1 - 0.1\tau) \rangle y^3 \\ & = \langle (0.25 + 0.12\tau, 0.55 - 0.12\tau), (0.2 + 0.1\tau, 0.6 - 0.1\tau) \rangle, \end{aligned} \right.$$

or

$$\left\{ \begin{aligned} & (0.65 + 0.12\tau)(\cdot x_I^L(\tau))^3 - (0.25 + 0.12\tau)(\cdot y_I^L(\tau))^2 = 0.45 + 0.12\tau, \\ & (0.95 - 0.12\tau)(\cdot x_I^L(\tau))^3 - (0.55 - 0.12\tau)(\cdot y_I^L(\tau))^2 = 0.75 - 0.12\tau, \\ & (0.6 + 0.1\tau)(\cdot x_I^U(\tau))^3 - (0.2 + 0.1\tau)(\cdot y_I^U(\tau))^2 = 0.4 + 0.1\tau, \\ & (1 - 0.1\tau)(\cdot x_I^U(\tau))^3 - (0.6 - 0.1\tau)(\cdot y_I^U(\tau))^2 = 0.8 - 0.1\tau, \\ & (0.45 + 0.12\tau)(\cdot x_I^L(\tau))^4 + (0.65 + 0.12\tau)(\cdot y_I^L(\tau))^3 = 0.25 + 0.12\tau, \\ & (0.75 - 0.12\tau)(\cdot x_I^L(\tau))^4 + (0.95 - 0.12\tau)(\cdot y_I^L(\tau))^3 = 0.55 - 0.12\tau, \\ & (0.4 + 0.1\tau)(\cdot x_I^U(\tau))^4 + (0.6 + 0.1\tau)(\cdot y_I^U(\tau))^3 = 0.2 + 0.1\tau, \\ & (0.8 - 0.1\tau)(\cdot x_I^U(\tau))^4 + (1 - 0.1\tau)(\cdot y_I^U(\tau))^3 = 0.6 - 0.1\tau. \end{aligned} \right.$$

Fig. 4 depicts the numerical solution of IVTFNEs, as well as the initial starting values and solution of the IVTF variables  $x$  and  $y$  used in Example 2. This figure shows numerical approximations obtained with SFQM1 that are matched with analytical solutions up to 33 decimal places.

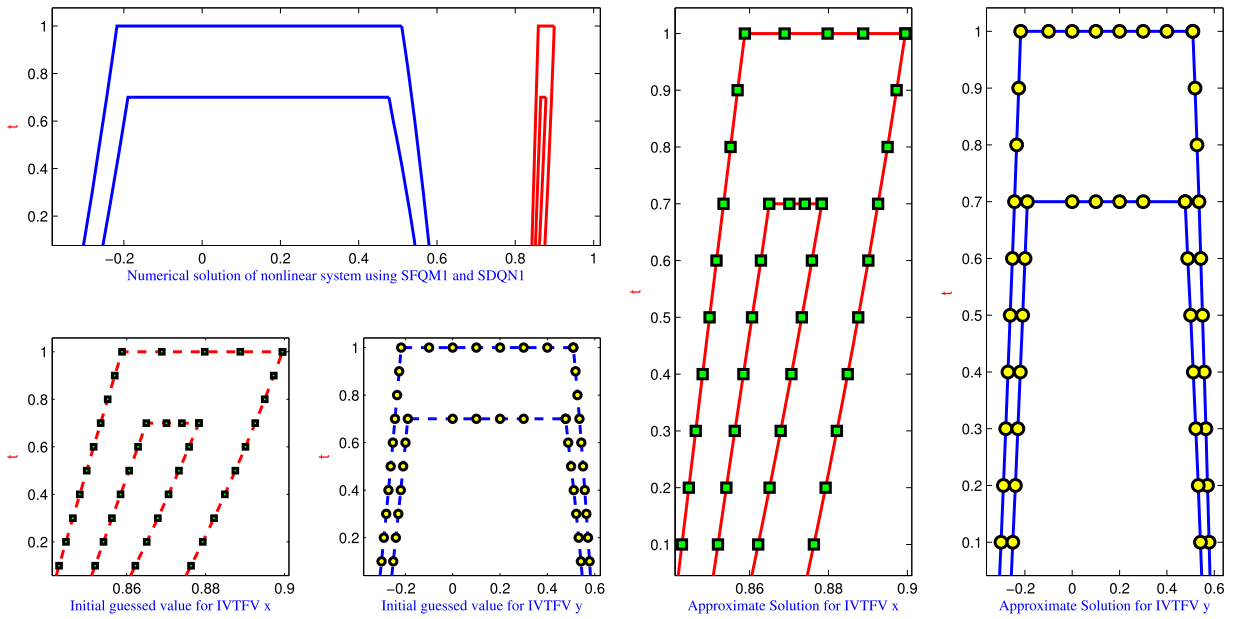


Fig. 4. Approximate solution of IVTFNEs used in Example 2.

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

$$\begin{cases} (0.65)(\dot{x}_t^L(0))^3 - (0.25)(\dot{y}_t^L(0))^2 = 0.45 \\ (0.95)(\dot{x}_t^L(0))^3 - (0.55)(\dot{y}_t^L(0))^2 = 0.75 \\ (0.6)(\dot{x}_t^U(0))^3 - (0.2)(\dot{y}_t^U(0))^2 = 0.4 \\ (1)(\dot{x}_t^U(0))^3 - (0.6)(\dot{y}_t^U(0))^2 = 0.8 \\ (0.45)(\ddot{x}_t^L(0))^4 + (0.65)(\ddot{y}_t^L(0))^3 = 0.25 \\ (0.75)(\ddot{x}_t^L(0))^4 + (0.95)(\ddot{y}_t^L(0))^3 = 0.55 \\ (0.4\tau)(\ddot{x}_t^U(0))^4 + (0.6)(\ddot{y}_t^U(0))^3 = 0.2 \\ (0.8)(\ddot{x}_t^L(0))^4 + (1)(\ddot{y}_t^L(0))^3 = 0.6 \end{cases} \quad \text{and} \quad \begin{cases} (0.77)(\dot{x}_t^L(1))^3 - (0.37)(\dot{y}_t^L(1))^2 = 0.62 \\ (0.83)(\dot{x}_t^L(1))^3 - (0.43)(\dot{y}_t^L(1))^2 = 0.63 \\ (0.7)(\dot{x}_t^U(1))^3 - (0.3)(\dot{y}_t^U(1))^2 = 0.5 \\ (0.9)(\dot{x}_t^U(1))^3 - (0.5)(\dot{y}_t^U(1))^2 = 0.7 \\ (0.57)(\ddot{x}_t^L(1))^4 + (0.77)(\ddot{y}_t^L(1))^3 = 0.37 \\ (0.63)(\ddot{x}_t^L(1))^4 + (0.83)(\ddot{y}_t^L(1))^3 = 0.43 \\ (0.5)(\ddot{x}_t^U(1))^4 + (0.7)(\ddot{y}_t^U(1))^3 = 0.3 \\ (0.7)(\ddot{x}_t^L(1))^4 + (0.9)(\ddot{y}_t^L(1))^3 = 0.5 \end{cases}$$

Consequently  $x_t^U(0) = 0.5, x_t^L(0) = 0.5, x_t^L(0) = 0.5, x_t^U(0) = 0.5$  and  $x_t^U(0) = x_t^L(0) = x_t^L(0) = x_t^U(0) = \frac{1}{2}$ . After 4 iterations, we obtain the solution with a maximum error less than  $10^{-15}$ . Now assume  $x, y$  are negative, we have

$$\begin{cases} (0.65 + 0.12\tau)(\dot{x}_t^L(\tau))^3 - (0.25 + 0.12\tau)(\dot{y}_t^L(\tau))^2 = 0.45 + 0.12\tau, \\ (0.95 - 0.12\tau)(\dot{x}_t^L(\tau))^3 - (0.55 - 0.12\tau)(\dot{y}_t^L(\tau))^2 = 0.75 - 0.12\tau, \\ (0.6 + 0.1\tau)(\dot{x}_t^U(\tau))^3 - (0.2 + 0.1\tau)(\dot{y}_t^U(\tau))^2 = 0.4 + 0.1\tau, \\ (1 - 0.1\tau)(\dot{x}_t^U(\tau))^3 - (0.6 - 0.1\tau)(\dot{y}_t^U(\tau))^2 = 0.8 - 0.1\tau, \\ (0.45 + 0.12\tau)(\ddot{x}_t^L(\tau))^4 + (0.65 + 0.12\tau)(\ddot{y}_t^L(\tau))^3 = 0.25 + 0.12\tau, \\ (0.75 - 0.12\tau)(\ddot{x}_t^L(\tau))^4 + (0.95 - 0.12\tau)(\ddot{y}_t^L(\tau))^3 = 0.55 - 0.12\tau, \\ (0.4 + 0.1\tau)(\ddot{x}_t^U(\tau))^4 + (0.6 + 0.1\tau)(\ddot{y}_t^U(\tau))^3 = 0.2 + 0.1\tau, \\ (0.8 - 0.1\tau)(\ddot{x}_t^L(\tau))^4 + (1 - 0.1\tau)(\ddot{y}_t^L(\tau))^3 = 0.6 - 0.1\tau. \end{cases}$$

For  $\tau = 0$ , we have,  $y_t^L(0) > y_t^L(0)$ , therefore negative solutions of IVTFNSEs does not exist.

Table 5, shows the numerical approximate solution of the interval-valued trapezoidal system of nonlinear equations used in Example 2 by utilizing numerical schemes SFQM1 and SFQN1 up to 30 decimal places with 64-digit floating point arithmetic. It displays the approximated solution to four decimal places. The numerical approximations solution presented in Table 5 agrees with the analytical solution illustrated in Fig. 4.

In Table 6, C1 and C2 represents the stopping criteria 1 and 2 i.e.,  $\|F(x, \tau)\|, \|x_{n+1}(\tau) - x_n(\tau)\|$  for different values of  $\tau$  respectively. The approximate IVTFNE solutions tabulated in Table 6 clearly show that our newly developed family is more efficient than the classical method used in the literature.

In Table 7, M-Error<sub>C1</sub>, M-Error<sub>C2</sub> show the error of numerical schemes SFQM1 and SFQN1 using stopping criteria I and II, respectively, M-CPU-Time<sub>C1</sub>, M-CPU-Time<sub>C2</sub> represent the computational time in seconds of the numerical schemes for solving the solution of a fuzzy nonlinear system used in Example 2, employing stopping criteria I and II, and COC shows the computational order of convergence of the SFQM1 and SFQN1 methods. Tables 5–7, demonstrates that our newly developed method, SFQM1, is better than

**Table 5**  
Approximate solution of Example 2 using SFQM1 and SFQN1.

$\tau$	$x_1^L(\tau)$	$x_1^U(\tau)$	$x_2^L(\tau)$	$x_2^U(\tau)$	$y_1^L(\tau)$	$y_1^U(\tau)$	$y_2^L(\tau)$	$y_2^U(\tau)$
0.0	0.5508	0.5722	0.7478	0.7900	-0.8778	-0.8440	-0.7032	-0.6865
0.1	0.5550	0.5775	0.7385	0.7811	-0.8707	-0.8364	-0.7074	-0.6892
0.2	0.5592	0.5830	0.7294	0.7725	-0.8637	-0.8291	-0.7117	-0.6930
0.3	0.5634	0.5886	0.7206	0.7641	-0.8570	-0.8220	-0.7161	-0.6964
0.4	0.5678	0.5942	0.7120	0.7558	-0.8504	-0.8151	-0.7206	-0.6998
0.5	0.5722	0.6000	0.7037	0.7478	-0.8440	-0.8083	-0.7252	-0.7032
0.6	0.5766	0.6059	0.6955	0.7460	-0.8377	-0.8018	-0.7298	-0.7067
0.7	0.5812	0.6119	0.6876	0.7324	-0.8315	-0.7954	-0.7346	-0.7103
0.8	0.5858	0.6119	0.6875	0.7250	-0.8255	-0.7954	-0.7345	-0.7139
0.9	0.5904	0.6119	0.6128	0.7170	-0.8197	-0.7954	-0.7868	-0.7176
1.0	0.5952	0.6119	0.6119	0.7106	-0.8139	-0.7954	-0.7954	-0.7213

**Table 6**  
Error comparison of the numerical schemes SFQM1 and SFQN1.

Scheme	SFQM1				SFQN1			
	C1		C2		C1		C2	
$\tau$	x	y	x	y	x	y	x	y
0.0	1.2e-30	2.1e-32	0.6e-43	1.3e-42	3.2e-15	1.1e-17	2.2e-25	1.7e-26
0.1	2.8e-35	2.2e-36	0.5e-44	2.6e-45	4.2e-16	1.4e-18	2.4e-26	1.6e-27
0.2	3.2e-31	2.3e-33	1.1e-41	3.7e-42	4.5e-17	2.2e-19	2.6e-28	5.5e-29
0.3	3.1e-35	2.6e-36	2.3e-42	3.3e-43	3.3e-18	2.6e-20	2.4e-29	5.6e-30
0.4	9.8e-36	1.9e-37	3.4e-43	3.6e-44	4.6e-19	2.8e-21	2.1e-30	6.6e-32
0.5	0.3e-33	1.7e-35	1.2e-44	2.8e-45	5.1e-21	3.7e-24	2.7e-31	1.3e-33
0.6	1.1e-34	2.5e-36	1.8e-43	2.7e-46	5.2e-22	3.8e-23	3.6e-32	2.2e-34
0.7	0.9e-38	3.1e-39	2.9e-42	2.2e-47	5.2e-23	4.5e-24	3.7e-30	2.3e-31
0.8	1.7e-37	4.1e-38	3.8e-43	2.5e-46	6.1e-20	4.7e-22	3.5e-31	5.6e-34
0.9	1.5e-39	2.7e-41	3.7e-44	2.1e-45	6.6e-23	4.8e-24	4.4e-27	3.5e-28
1.0	1.4e-40	2.8e-42	3.5e-40	4.6e-41	6.7e-24	5.1e-25	4.7e-32	3.8e-33

**Table 7**  
Stability-consistency analysis.

Method	SFQM1	SFQN1
M-Error <sub>C1</sub>	6.1e-43	3.1e-10
M-Error <sub>C2</sub>	1.2e-32	1.8e-8
M-CPU-Time <sub>C1</sub>	0.0346	1.1231
M-CPU-Time <sub>C2</sub>	0.1221	1.0327
COC	3.012	2.013

**Table 8**  
Percentage convergence and total function evaluations for SFQM1, SFQN1.

Method	Num- F-J	Percentage-Con	M-CPU-Time <sub>C3</sub>	M-Error <sub>C3</sub>	COC
SFQM1	6	76.006576	2.00056435	6.0051e-11	2.123245
SFQN1	9	55.6537868	2.6742274	4.26505e-07	1.76654

SFQN1 in terms of residual error, CPU time, and computational order of convergence. Table 7 also shows that the SFQM1 method is more stable and consistent than the SFQN1 method.

Table 8 demonstrates that in terms of percentage convergence, total functions, and Jacob’s evaluations, our technique SFQM1 outperforms the existing method SFQN1 and provides an alternative way to solve complex IVTFNE for greater comprehension and analysis.

**Example 3:** Consider IVTFNEs in which the coefficient matrix is IVTF number and the right side column vector contains real valued crisp numbers as

$$\begin{aligned} &\left\langle (0.65, 0.73, 0.87, 0.95; \frac{2}{3}), (0.6, 0.7, 0.9, 1.1; 1) \right\rangle x^3 \\ &\quad - \left\langle (0.25, 0.33, 0.47, 0.55; \frac{2}{3}), (0.2, 0.3, 0.5, 0.6; 1) \right\rangle y^2 = 0.45, \\ &\left\langle (0.45, 0.53, 0.67, 0.75; \frac{2}{3}), (0.4, 0.5, 0.7, 0.8; 1) \right\rangle x^2 \\ &\quad + \left\langle (0.65, 0.73, 0.87, 0.95; \frac{2}{3}), (0.6, 0.7, 0.9, 1; 1) \right\rangle y^2 = -0.25. \end{aligned}$$

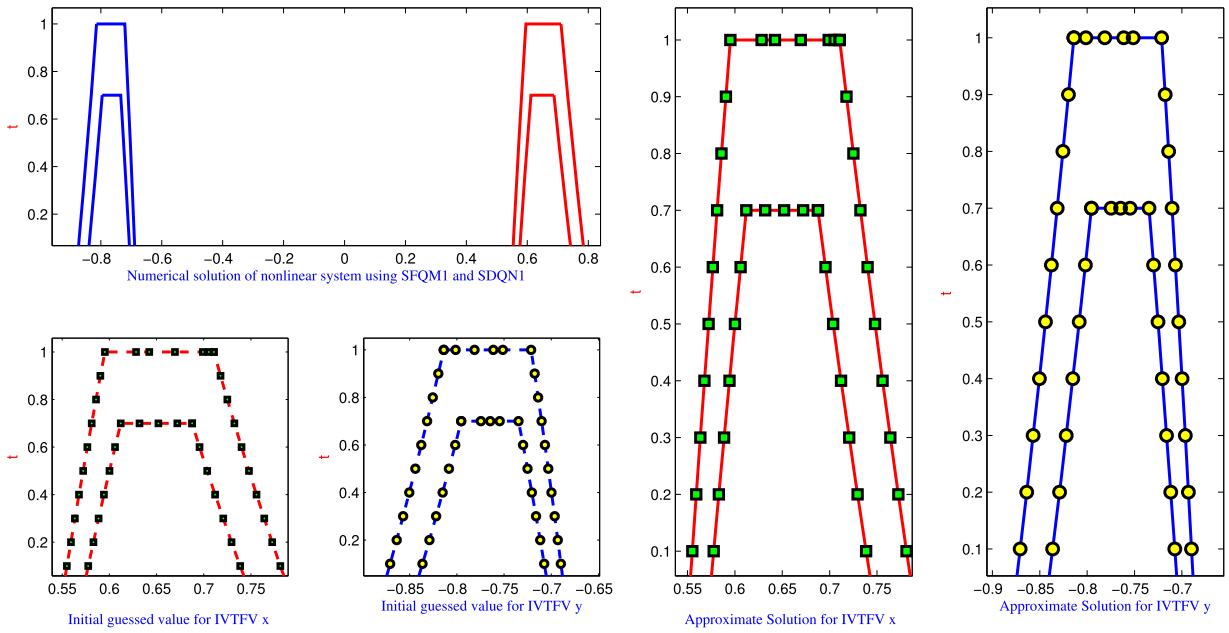


Fig. 5. Approximate solution of IVTFNEs used in Example 3.

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

$$\begin{aligned} &\langle(0.65 + 0.12\tau, 0.95 - 0.12\tau), (0.6 + 0.1\tau, 1 - 0.1\tau)\rangle x^3 \\ &- \langle(0.25 + 0.12\tau, 0.55 - 0.12\tau), (0.2 + 0.1\tau, 0.6 - 0.1\tau)\rangle y^2 = 0.45, \\ &\langle(0.45 + 0.12\tau, 0.75 - 0.12\tau), (0.4 + 0.1\tau, 0.8 - 0.1\tau)\rangle x^4 \\ &+ \langle(0.65 + 0.12\tau, 0.95 - 0.12\tau), (0.6 + 0.1\tau, 1 - 0.1\tau)\rangle y^3 = -0.25. \end{aligned}$$

or

$$\begin{cases} (0.65 + 0.12\tau)(\dot{x}_t^L(\tau))^3 - (0.25 + 0.12\tau)(\dot{y}_t^L(\tau))^2 = 0.45, \\ (0.95 - 0.12\tau)(\dot{x}_t^L(\tau))^3 - (0.55 - 0.12\tau)(\dot{y}_t^L(\tau))^2 = 0.45, \\ (0.6 + 0.1\tau)(\dot{x}_t^U(\tau))^3 - (0.2 + 0.1\tau)(\dot{y}_t^U(\tau))^2 = 0.45, \\ (1 - 0.1\tau)(\dot{x}_t^U(\tau))^3 - (0.6 - 0.1\tau)(\dot{y}_t^U(\tau))^2 = 0.45, \\ (0.45 + 0.12\tau)(\ddot{x}_t^L(\tau))^4 + (0.65 + 0.12\tau)(\ddot{y}_t^L(\tau))^3 = -0.25, \\ (0.75 - 0.12\tau)(\ddot{x}_t^L(\tau))^4 + (0.95 - 0.12\tau)(\ddot{y}_t^L(\tau))^3 = -0.25, \\ (0.4 + 0.1\tau)(\ddot{x}_t^U(\tau))^4 + (0.6 + 0.1\tau)(\ddot{y}_t^U(\tau))^3 = -0.25, \\ (0.8 - 0.1\tau)(\ddot{x}_t^U(\tau))^4 + (1 - 0.1\tau)(\ddot{y}_t^U(\tau))^3 = -0.25. \end{cases}$$

Fig. 5, depicts the numerical solution of IVTFNEs, as well as the initial starting values and solution of the IVTF variables  $x$  and  $y$  utilized in Example 3. This figure clearly shows numerical approximations obtained with SFQMI that are matched with analytical solutions up to 34 decimal places.

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

$$\begin{cases} (0.65)(\dot{x}_t^L(0))^3 - (0.25)(\dot{y}_t^L(0))^2 = 0.45 \\ (0.95)(\dot{x}_t^L(0))^3 - (0.55)(\dot{y}_t^L(0))^2 = 0.45 \\ (0.6)(\dot{x}_t^U(0))^3 - (0.2)(\dot{y}_t^U(0))^2 = 0.45 \\ (1)(\dot{x}_t^U(0))^3 - (0.6)(\dot{y}_t^U(0))^2 = 0.45 \\ (0.45)(\ddot{x}_t^L(0))^4 + (0.65)(\ddot{y}_t^L(0))^3 = -0.25 \\ (0.75)(\ddot{x}_t^L(0))^4 + (0.95)(\ddot{y}_t^L(0))^3 = -0.25 \\ (0.4\tau)(\ddot{x}_t^U(0))^4 + (0.6)(\ddot{y}_t^U(0))^3 = -0.25 \\ (0.8)(\ddot{x}_t^U(0))^4 + (1)(\ddot{y}_t^U(0))^3 = -0.25 \end{cases} \quad \text{and} \quad \begin{cases} (0.77)(\dot{x}_t^L(1))^3 - (0.37)(\dot{y}_t^L(1))^2 = 0.45 \\ (0.83)(\dot{x}_t^L(1))^3 - (0.43)(\dot{y}_t^L(1))^2 = 0.45 \\ (0.7)(\dot{x}_t^U(1))^3 - (0.3)(\dot{y}_t^U(1))^2 = 0.45 \\ (0.9)(\dot{x}_t^U(1))^3 - (0.5)(\dot{y}_t^U(1))^2 = 0.45 \\ (0.57)(\ddot{x}_t^L(1))^4 + (0.77)(\ddot{y}_t^L(1))^3 = -0.25 \\ (0.63)(\ddot{x}_t^L(1))^4 + (0.83)(\ddot{y}_t^L(1))^3 = -0.25 \\ (0.5)(\ddot{x}_t^U(1))^4 + (0.7)(\ddot{y}_t^U(1))^3 = -0.25 \\ (0.7)(\ddot{x}_t^U(1))^4 + (0.9)(\ddot{y}_t^U(1))^3 = -0.25 \end{cases}$$

**Table 9**  
Approximate solution of Example 3 using SFQM1 and SFQN1.

$\tau$	$\dot{x}_t^L(\tau)$	$\dot{x}_t^U(\tau)$	$\dot{x}_t^L(\tau)$	$\dot{x}_t^U(\tau)$	$\dot{y}_t^L(\tau)$	$\dot{y}_t^U(\tau)$	$\dot{y}_t^L(\tau)$	$\dot{y}_t^U(\tau)$
0.0	0.5508	0.5722	0.7478	0.7900	-0.8778	-0.8440	-0.7032	-0.6865
0.1	0.5550	0.5775	0.7385	0.7811	-0.8707	-0.8364	-0.7074	-0.6892
0.2	0.5592	0.5830	0.7294	0.7725	-0.8637	-0.8291	-0.7117	-0.6930
0.3	0.5634	0.5886	0.7206	0.7641	-0.8570	-0.8220	-0.7161	-0.6964
0.4	0.5678	0.5942	0.7120	0.7558	-0.8504	-0.8151	-0.7206	-0.6998
0.5	0.5722	0.6000	0.7037	0.7478	-0.8440	-0.8083	-0.7252	-0.7032
0.6	0.5766	0.6059	0.6955	0.7460	-0.8377	-0.8018	-0.7298	-0.7067
0.7	0.5812	0.6119	0.6876	0.7324	-0.8315	-0.7954	-0.7346	-0.7103
0.8	0.5858	0.6252	0.6876	0.7250	-0.8255	-0.7954	-0.7415	-0.7139
0.9	0.5904	0.6412	0.6876	0.7170	-0.8197	-0.7954	-0.7623	-0.7176
1.0	0.5952	0.6876	0.6876	0.7106	-0.8139	-0.7954	-0.7954	-0.7213

**Table 10**  
Error comparison of the numerical schemes SFQM1 and SFQN1.

$\tau$	SFQM1				SFQN1			
	C1		C2		C1		C2	
	x	y	x	y	x	y	x	y
0.0	2.1e-30	3.3e-32	2.2e-42	3.4e-45	3.8e-15	3.9e-16	4.4e-25	4.6e-31
0.1	2.6e-31	3.5e-33	2.3e-44	1.2e-45	3.5e-16	0.7e-18	1.5e-27	0.1e-29
0.2	3.4e-33	3.7e-34	2.6e-46	1.4e-47	3.8e-18	1.4e-20	2.3e-28	2.4e-30
0.3	3.6e-35	3.8e-36	2.8e-42	2.2e-46	1.7e-19	3.4e-21	3.3e-31	1.1e-32
0.4	4.3e-36	1.6e-37	3.3e-42	2.4e-43	0.5e-20	1.3e-22	3.6e-29	1.4e-34
0.5	4.4e-32	2.7e-34	3.7e-43	3.5e-44	3.4e-23	2.5e-24	4.4e-33	0.4e-35
0.6	4.5e-34	5.5e-35	4.1e-47	3.6e-48	4.9e-24	2.6e-25	4.8e-30	1.5e-32
0.7	4.7e-37	4.8e-38	4.2e-40	3.9e-41	0.5e-17	3.4e-22	1.9e-27	1.3e-28
0.8	1.1e-38	7.8e-39	5.2e-41	4.4e-42	5.2e-21	3.7e-23	1.0e-28	2.3e-29
0.9	1.6e-31	9.0e-32	5.6e-42	4.8e-43	2.9e-18	3.9e-24	2.2e-30	2.7e-32
1.0	1.8e-32	1.1e-33	6.6e-43	5.5e-44	1.3e-22	1.1e-23	2.5e-32	2.8e-33

**Table 11**  
Stability-consistency analysis.

Method	SFQM1	SFQN1
M-Error <sub>C1</sub>	2.1e-30	3.8e-15
M-Error <sub>C2</sub>	3.9e-41	4.4e-25
M-CPU-Time <sub>C1</sub>	0.0013	0.1031
M-CPU-Time <sub>C2</sub>	0.0217	0.0527
COC	3.1528	2.0013

Consequently  $x_t^U(0) = 0.5, x_t^L(0) = 0.5, x_t^L(0) = 0.5, x_t^U(0) = 0.5$  and  $x_t^U(0) = x_t^L(0) = x_t^L(0) = x_t^U(0) = \frac{1}{2}$ . After 4 iterations, we obtain the solution with a maximum error less than  $10^{-30}$ . Suppose  $x, y$  are negative, we have

$$\begin{cases} (0.65 + 0.12\tau)(\dot{x}_t^L(\tau))^3 - (0.25 + 0.12\tau)(\dot{y}_t^L(\tau))^2 = 0.45, \\ (0.95 - 0.12\tau)(\dot{x}_t^L(\tau))^3 - (0.55 - 0.12\tau)(\dot{y}_t^L(\tau))^2 = 0.45, \\ (0.6 + 0.1\tau)(\dot{x}_t^U(\tau))^3 - (0.2 + 0.1\tau)(\dot{y}_t^U(\tau))^2 = 0.45, \\ (1 - 0.1\tau)(\dot{x}_t^U(\tau))^3 - (0.6 - 0.1\tau)(\dot{y}_t^U(\tau))^2 = 0.45, \\ (0.45 + 0.12)(\ddot{x}_t^L(\tau))^4 + (0.65 + 0.12\tau)(\ddot{y}_t^L(\tau))^3 = -0.25, \\ (0.75 - 0.12\tau)(\ddot{x}_t^L(\tau))^4 + (0.95 - 0.12\tau)(\ddot{y}_t^L(\tau))^3 = -0.25, \\ (0.4 + 0.1\tau)(\ddot{x}_t^U(\tau))^4 + (0.6 + 0.1\tau)(\ddot{y}_t^U(\tau))^3 = -0.25, \\ (0.8 - 0.1\tau)(\ddot{x}_t^U(\tau))^4 + (1 - 0.1\tau)(\ddot{y}_t^U(\tau))^3 = -0.25. \end{cases}$$

For  $\tau = 0$ , we have,  $y_t^L(0) > y_t^U(0)$ , therefore negative solutions of IVTFNSEs does not exist.

Table 9, shows the numerical approximate solution of the interval-valued trapezoidal system of nonlinear equations used in Example 3 by utilizing numerical schemes SFQM1 and SFQN1 up to 30 decimal places with 64-digit floating point arithmetic. This table displays the approximated solution to four decimal places. The numerical approximate solution presented in Table 9 agrees with the analytical solution shown in Fig. 5.

In Table 10, C1 and C2 represents the stopping criteria 1 and 2 i.e.,  $\|F(x, \tau)\|, \|x_{n+1}(\tau) - x_n(\tau)\|$  for different values of  $\tau$  respectively. The approximate IVTFNE solutions in Table 10 clearly show that our newly developed family is more efficient than the classical method used in the literature.

**Table 12**  
Percentage convergence and total function evaluations for SFQM1, SFQN1.

Method	Num- F-J	Percentage-Con	M-CPU-Time <sub>C3</sub>	M-Error <sub>C3</sub>	COC
SFQM1	7	64.133254	3.0098722	9.500970e-18	1.9997
SFQN1	13	41.954552	4.995007	4.0098733e-11	1.76543

In Table 11, M-Error<sub>C1</sub>, M-Error<sub>C2</sub> show the error of numerical schemes SFQM1 and SFQN1 using stopping criteria I and II, respectively, M-CPU-Time<sub>C1</sub>, M-CPU-Time<sub>C2</sub> present the computational time in seconds of the numerical schemes for solving the solution of a fuzzy nonlinear system used in Example 3, employing stopping criteria I and II, and COC shows the computational order of convergence of the SFQM1 and SFQN1 methods. Tables 9–11, demonstrates that our newly developed method, SFQM1, is better than SFQN1 in terms of residual error, CPU time, and computational order of convergence. Table 11 also shows that the SFQM1 method is more stable and consistent than the SFQN1 method.

Table 12, demonstrates that in terms of percentage convergence, total functions, and Jacob’s evaluations, our technique SFQM1 outperforms the existing method SFQN1 and provides an alternative way to solve complex IVTFNE for greater comprehension and analysis.

**Example 4:** Consider the nonlinear system of IVTFNEs in which the coefficient matrix is IVTF number and the right side column vector contains real valued crisp number along with trigonometric functions as:

$$\begin{cases} \langle (0.25 + 0.12\tau, 0.55 - 0.12\tau), (0.2 + 0.1\tau, 0.6 - 0.1\tau) \rangle \tan(x) + \\ \langle (0.65 + 0.12\tau, 0.95 - 0.12\tau), (0.6 + 0.1\tau, 1 - 0.1\tau) \rangle y = -0.3, \\ \langle (0.65 + 0.12\tau, 0.95 - 0.12\tau), (0.6 + 0.1\tau, 1 - 0.1\tau) \rangle x^3 + \\ \langle (0.45 + 0.12\tau, 0.75 - 0.12\tau), (0.4 + 0.1\tau, 0.8 - 0.1\tau) \rangle y = -0.2. \end{cases}$$

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

$$\begin{cases} \langle (0.25 + 0.12\tau, 0.55 - 0.12\tau), (0.2 + 0.1\tau, 0.6 - 0.1\tau) \rangle \tan(x) + \\ \langle (0.65 + 0.12\tau, 0.95 - 0.12\tau), (0.6 + 0.1\tau, 1 - 0.1\tau) \rangle y = -0.3, \\ \langle (0.65 + 0.12\tau, 0.95 - 0.12\tau), (0.6 + 0.1\tau, 1 - 0.1\tau) \rangle x^3 + \\ \langle (0.45 + 0.12\tau, 0.75 - 0.12\tau), (0.4 + 0.1\tau, 0.8 - 0.1\tau) \rangle y = -0.2, \end{cases}$$

or

$$\begin{cases} (0.25 + 0.12\tau) \tan(\dot{x}_\tau^L) + (0.65 + 0.12\tau)(\dot{y}_\tau^L) = -0.3, \\ (0.55 - 0.12\tau) \tan(\dot{x}_\tau^U) + (0.95 - 0.12\tau)(\dot{y}_\tau^U) = -0.3, \\ (0.2 + 0.1\tau) \tan(\dot{x}_\tau^U) + (0.6 + 0.1\tau)(\dot{y}_\tau^U) = -0.3, \\ (0.6 - 0.1\tau) \tan(\dot{x}_\tau^L) + (1 - 0.1\tau)(\dot{y}_\tau^L) = -0.3, \\ (0.65 + 0.12\tau)(\ddot{x}_\tau^L)^3 + (0.45 + 0.12\tau)(\ddot{y}_\tau^L) = -0.2, \\ (0.95 - 0.12\tau)(\ddot{x}_\tau^U)^3 + (0.75 - 0.12\tau)(\ddot{y}_\tau^U) = -0.2, \\ (0.6 + 0.1\tau)(\ddot{x}_\tau^U)^3 + (0.4 + 0.1\tau)(\ddot{y}_\tau^U) = -0.2, \\ (1 - 0.1\tau)(\ddot{x}_\tau^L)^3 + (0.8 - 0.1\tau)(\ddot{y}_\tau^L) = -0.2. \end{cases}$$

Fig. 6, depicts the numerical solution of IVTFNEs, as well as the initial starting values and solution of the IVTF variables  $x$  and  $y$  utilized in Example 4. This figure shows numerical approximations obtained with SFQMI that are matched with analytical solutions up to 34 decimal places.

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

$$\begin{cases} (0.250) \tan(\dot{x}_0^L) + (0.650)(\dot{y}_0^L) = -0.3 \\ (0.550) \tan(\dot{x}_0^U) + (0.950)(\dot{y}_0^U) = -0.3 \\ (0.20) \tan(\dot{x}_0^U) + (0.60)(\dot{y}_0^U) = -0.3 \\ (0.60) \tan(\dot{x}_0^L) + (1.0)(\dot{y}_0^L) = -0.3 \\ (0.650)(\ddot{x}_0^L)^3 + (0.450)(\ddot{y}_0^L) = -0.2 \\ (0.950)(\ddot{x}_0^U)^3 + (0.750)(\ddot{y}_0^U) = -0.2 \\ (0.60)(\ddot{x}_0^U)^3 + (0.40)(\ddot{y}_0^U) = -0.2 \\ (1.0)(\ddot{x}_0^L)^3 + (0.80)(\ddot{y}_0^L) = -0.2 \end{cases} \quad \text{and} \quad \begin{cases} (0.25 + 0.12\tau) \tan(\dot{x}_1^L) + (0.65 + 0.12\tau)(\dot{y}_1^L) = -0.3 \\ (0.55 - 0.12\tau) \tan(\dot{x}_1^U) + (0.95 - 0.12\tau)(\dot{y}_1^U) = -0.3 \\ (0.2 + 0.1\tau) \tan(\dot{x}_1^U) + (0.6 + 0.1\tau)(\dot{y}_1^U) = -0.3 \\ (0.6 - 0.1\tau) \tan(\dot{x}_1^L) + (1 - 0.1\tau)(\dot{y}_1^L) = -0.3 \\ (0.65 + 0.12\tau)(\ddot{x}_1^L)^3 + (0.45 + 0.12\tau)(\ddot{y}_1^L) = -0.2 \\ (0.95 - 0.12\tau)(\ddot{x}_1^U)^3 + (0.75 - 0.12\tau)(\ddot{y}_1^U) = -0.2 \\ (0.6 + 0.1\tau)(\ddot{x}_1^U)^3 + (0.4 + 0.1\tau)(\ddot{y}_1^U) = -0.2 \\ (1 - 0.1\tau)(\ddot{x}_1^L)^3 + (0.8 - 0.1\tau)(\ddot{y}_1^L) = -0.2 \end{cases}$$

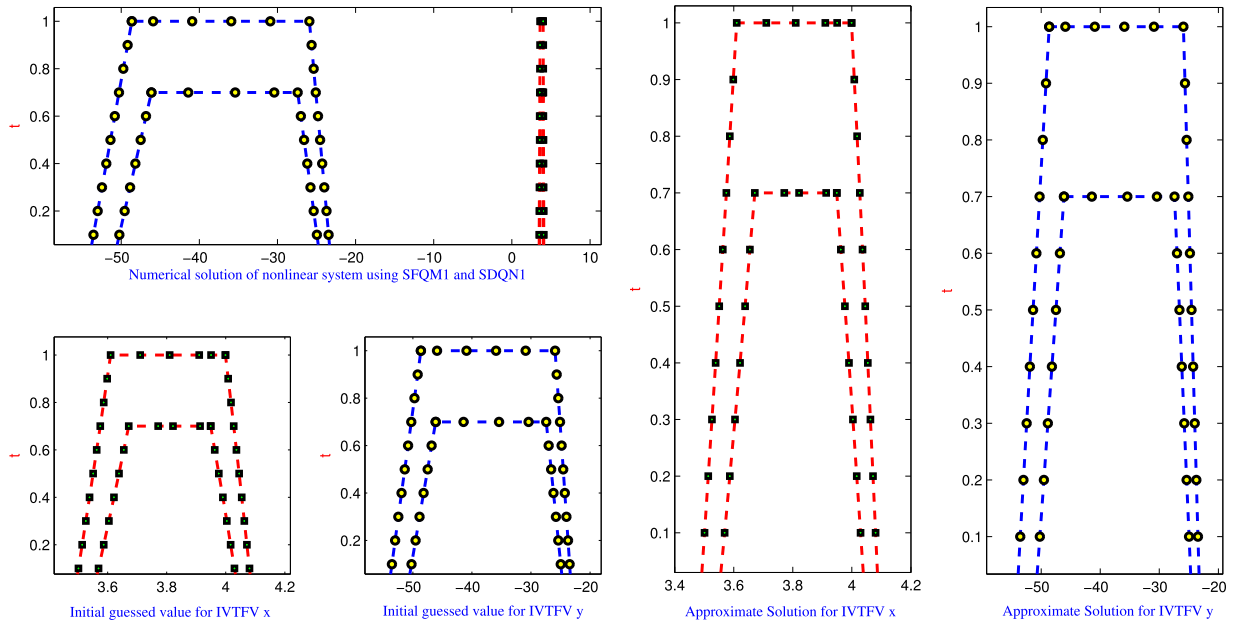


Fig. 6. Approximate solution of IVTFNEs used in Example 4.

**Table 13**  
Approximate solution of Example 4 using SFQM1 and SFQN1.

$\tau$	$\dot{x}_t^L(\tau)$	$x_t^L(\tau)$	$\dot{x}_t^U(\tau)$	$x_t^U(\tau)$	$\dot{y}_t^L(\tau)$	$y_t^L(\tau)$	$\dot{y}_t^U(\tau)$	$y_t^U(\tau)$
0.0	4.0892	3.5510	4.0425	3.4882	-54.0909	-50.9090	-24.5454	-23.1818
0.1	4.0806	3.5691	4.0296	3.5010	-53.5454	-50.2272	-24.9545	-23.4545
0.2	4.0718	3.5868	4.0166	3.5138	-53.0000	-49.5454	-25.3636	-23.7272
0.3	4.0600	3.6043	4.0034	3.5263	-52.4545	-48.8636	-25.7727	-24.0060
0.4	4.0541	3.6215	3.9901	3.5387	-51.9090	-48.1818	-26.1818	-24.2727
0.5	4.4051	3.6383	3.9765	3.5510	-51.3636	-47.5000	-26.5909	-24.5454
0.6	4.0361	3.6549	3.9628	3.5631	-50.8181	-46.8181	-27.0000	-24.8181
0.7	4.0269	3.6713	3.9489	3.5750	-49.2727	-47.1363	-27.4090	-25.0909
0.8	4.0177	3.6713	3.8417	3.5868	-49.1818	-47.1363	-37.5841	-25.3636
0.9	4.0084	3.6713	3.7412	3.5985	-48.6363	-47.1363	-43.5412	-25.6363
1.0	3.9990	3.6713	3.6713	3.6100	-48.2701	-47.1363	-47.1363	-25.9090

Consequently  $x_t^U(0) = 0.5, x_t^L(0) = 0.5, x_t^L(0) = 0.5, x_t^U(0) = 0.5$  and  $x_t^U(0) = x_t^L(0) = x_t^L(0) = x_t^U(0) = \frac{1}{2}$ . After 4 iterations, we obtain the solution with maximum error less than  $10^{-30}$ . Suppose  $x, y$  are negative, we have

$$\begin{cases} (0.25 + 0.12\tau) \tan(\dot{x}_t^L(\tau)) + (0.65 + 0.12\tau)(\dot{y}_t^L(\tau)) = -0.3, \\ (0.55 - 0.12\tau) \tan(\dot{x}_t^L(\tau)) + (0.95 - 0.12\tau)(\dot{y}_t^L(\tau)) = -0.3, \\ (0.2 + 0.1\tau) \tan(\dot{x}_t^U(\tau)) + (0.6 + 0.1\tau)(\dot{y}_t^U(\tau)) = -0.3, \\ (0.6 - 0.1\tau) \tan(\dot{x}_t^U(\tau)) + (1 - 0.1\tau)(\dot{y}_t^U(\tau)) = -0.3, \\ (0.65 + 0.12\tau)(\ddot{x}_t^L(\tau))^3 + (0.45 + 0.12\tau)(\ddot{y}_t^L(\tau)) = -0.2, \\ (0.95 - 0.12\tau)(\ddot{x}_t^L(\tau))^3 + (0.75 - 0.12\tau)(\ddot{y}_t^L(\tau)) = -0.2, \\ (0.6 + 0.1\tau)(\ddot{x}_t^U(\tau))^3 + (0.4 + 0.1\tau)(\ddot{y}_t^U(\tau)) = -0.2, \\ (1 - 0.1\tau)(\ddot{x}_t^U(\tau))^3 + (0.8 - 0.1\tau)(\ddot{y}_t^U(\tau)) = -0.2. \end{cases}$$

For  $\tau = 0$ , we have,  $y_t^L(0) > y_t^L(0)$ , therefore negative solutions of IVTFNSEs does not exist.

Table 13 shows the numerical approximate solution of the interval-valued trapezoidal system of nonlinear equations used in Example 4 by utilizing numerical schemes SFQM1 and SFQN1 up to 30 decimal places using 64-digit floating point arithmetic. This table displays the approximated solution to four decimal places.

In Table 14, C1 and C2 represents the stopping criteria 1 and 2 i.e.,  $\|F(x, \tau)\|, \|x_{n+1}(\tau) - x_n(\tau)\|$  for different values of  $\tau$  respectively. The approximate IVTFNE solutions presented in Table 14 clearly show that our newly developed family is more efficient than the classical method used in the literature.

In Table 15, M-Error<sub>C1</sub>, M-Error<sub>C2</sub> show the error of numerical schemes SFQM1 and SFQN1 using stopping criteria I and II, respectively, M-CPU-Time<sub>C1</sub>, M-CPU-Time<sub>C2</sub> present the computational time in seconds of the numerical schemes for solving the

**Table 14**  
Error comparison of the numerical schemes SFQM1 and SFQN1.

Scheme	SFQM1				SFQN1			
	C1		C2		C1		C2	
$\tau$	x	y	x	y	x	y	x	y
0.0	1.4e-30	2.2e-33	4.5e-40	0.6e-41	3.3e-15	1.1e-17	1.4e-26	2.5e-28
0.1	0.5e-33	1.5e-34	0.7e-43	4.4e-46	2.5e-16	0.5e-19	2.2e-27	2.8e-29
0.2	0.7e-32	2.2e-33	0.5e-42	1.4e-45	1.5e-17	0.7e-20	0.5e-28	3.7e-30
0.3	1.5e-35	2.5e-37	3.3e-43	2.3e-44	1.6e-19	1.5e-21	1.8e-29	3.9e-31
0.4	1.6e-31	3.3e-32	3.2e-45	3.3e-46	4.4e-20	3.5e-22	0.7e-30	1.1e-32
0.5	2.5e-36	3.6e-38	3.8e-46	3.8e-47	4.3e-23	5.5e-24	2.3e-33	0.5e-34
0.6	3.5e-38	3.7e-39	4.6e-47	0.6e-48	2.3e-20	7.2e-23	2.8e-32	0.6e-33
0.7	1.8e-36	0.5e-37	4.9e-43	1.5e-44	1.9e-17	1.5e-20	2.9e-31	1.7e-35
0.8	2.7e-31	2.5e-32	5.5e-42	2.5e-45	0.4e-18	2.5e-19	3.7e-25	1.5e-26
0.9	3.4e-33	3.7e-34	5.6e-41	3.5e-42	4.5e-21	2.8e-20	3.7e-28	2.7e-29
1.0	3.2e-35	1.7e-36	5.9e-40	1.5e-47	4.7e-22	2.9e-21	3.9e-29	4.4e-30

**Table 15**  
Stability-consistency analysis.

Method	SFQM1	SFQN1
M-Error <sub>C1</sub>	4.3e-46	7.7e-10
M-Error <sub>C2</sub>	3.2e-34	1.9e-8
M-CPU-Time <sub>C1</sub>	0.1146	2.1131
M-CPU-Time <sub>C2</sub>	0.1027	1.1220
COC	3.0021	2.0013

**Table 16**  
Percentage convergence and total function evaluations for SFQM1, SFQN1.

Method	Number- F-J	Percentage-Con	M-CPU-Time <sub>C3</sub>	M-Error <sub>C3</sub>	COC
SFQM1	5	71.6977009	1.2243334	0.012252e-21	2.7434
SFQN1	9	65.1103124	3.9788656	1.877567e-11	2.0074

solution of a fuzzy nonlinear system used in Example 4, employing stopping criteria I and II, and COC shows the computational order of convergence of the SFQM1 and SFQN1 methods. Tables 13–15 demonstrates that our newly developed method, SFQM1, is better than SFQN1 in terms of residual error, CPU time, and computational order of convergence. Table 15 also shows that the SFQM1 method is more stable and consistent than the SFQN1 method.

Table 16 demonstrates that in terms of percentage convergence, total functions, and Jacob’s evaluations, our technique SFQM1 outperforms the existing method SFQN1 and provides an alternative way to solve complex IVTFNE for greater comprehension and analysis.

**Example 5:** Consider the nonlinear system of IVTFNEs in which the coefficient matrix is the crisp real valued number and the right side column vector contains IVTF along with exponential and logarithmic fuzzy functions as:

$$\begin{cases} 3e^x + 7 \sin y = \langle (80, 90, 110, 120; \frac{2}{3}), (75, 85, 115, 125; 1) \rangle. \\ 12e^x + 15 \sin y = \langle (50, 60, 70, 80; \frac{2}{3}), (45, 55, 75, 95; 1) \rangle. \end{cases}$$

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

$$\begin{cases} 3e^x + y = \langle (80 + 15\tau, 120 - 15\tau), (75 + 10\tau, 125 - 10\tau) \rangle, \\ 12e^x + 15y = \langle (50 + 15\tau, 80 - 15\tau), (45 + 10\tau, 95 - 20\tau) \rangle. \end{cases}$$

or

$$\begin{cases} 3e^{x_L(\tau)} + ( \cdot y_L^L(\tau) ) = 80 + 15\tau, \\ 3e^{x_U(\tau)} + ( \cdot y_U^L(\tau) ) = 120 - 15\tau, \\ 3e^{x_L(\tau)} + ( \cdot y_L^U(\tau) ) = 75 + 10\tau, \\ 3e^{x_U(\tau)} + ( \cdot y_U^U(\tau) ) = 125 - 10\tau, \\ 12e^{x_L(\tau)} + 15( \cdot y_L^L(\tau) ) = 50 + 15\tau, \\ 12e^{x_U(\tau)} + 15( \cdot y_U^L(\tau) ) = 80 - 15\tau, \\ 12e^{x_L(\tau)} + 15( \cdot y_L^U(\tau) ) = 45 + 10\tau, \\ 12e^{x_U(\tau)} + 15( \cdot y_U^U(\tau) ) = 95 - 20\tau. \end{cases}$$

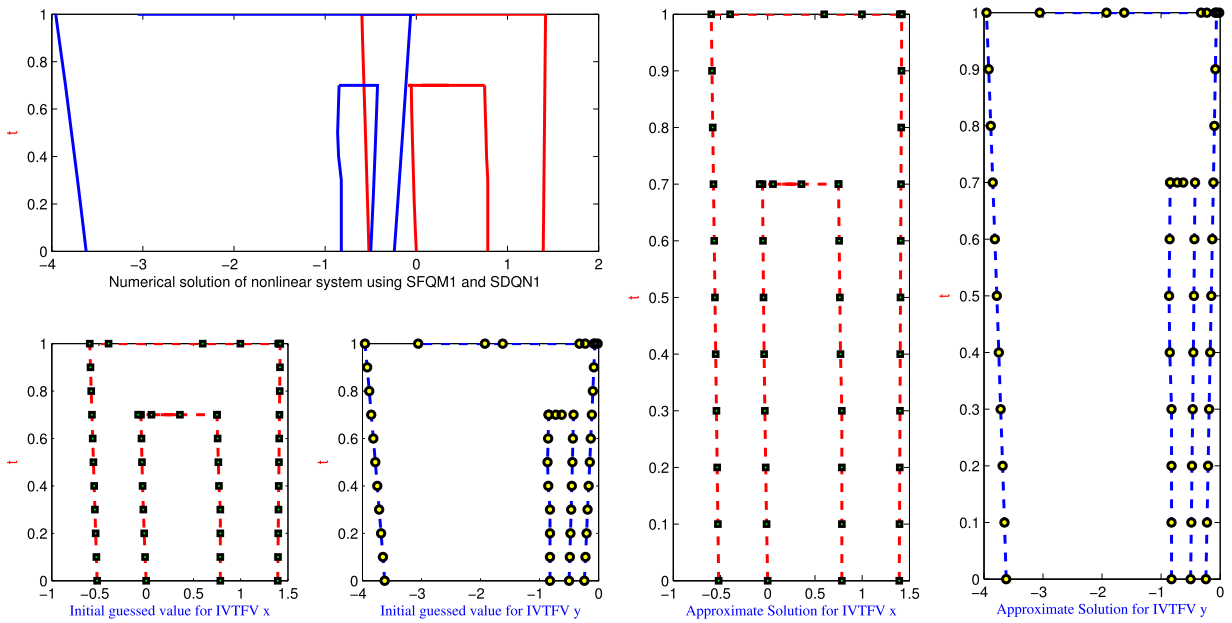


Fig. 7. Approximate solution of IVTFNEs used in Example 5.

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

$$\begin{cases} 3e^{\cdot x_t^L(0)} + (\cdot y_t^L(0)) = 80 \\ 3e^{\cdot x_t^L(0)} + (\cdot y_t^L(0)) = 120 \\ 3e^{\cdot x_t^U(0)} + (\cdot y_t^U(0)) = 75 \\ 3e^{\cdot x_t^U(0)} + (\cdot y_t^U(0)) = 125 \\ 12e^{\cdot\cdot x_t^L(0)} + 15(\cdot\cdot y_t^L(0)) = 50 \\ 12e^{\cdot\cdot x_t^L(0)} + 15(\cdot\cdot y_t^L(0)) = 80 \\ 12e^{\cdot\cdot x_t^U(0)} + 15(\cdot\cdot y_t^U(0)) = 45 \\ 12e^{\cdot\cdot x_t^U(0)} + 15(\cdot\cdot y_t^U(0)) = 95 \end{cases} \text{ and } \begin{cases} 3e^{\cdot x_t^L(1)} + (\cdot y_t^L(1)) = 96\tau \\ 3e^{\cdot x_t^L(1)} + (\cdot y_t^L(1)) = 105 \\ 3e^{\cdot x_t^U(1)} + (\cdot y_t^U(1)) = 85 \\ 3e^{\cdot x_t^U(1)} + (\cdot y_t^U(1)) = 115 \\ 12e^{\cdot\cdot x_t^L(1)} + 15(\cdot\cdot y_t^L(1)) = 65 \\ 12e^{\cdot\cdot x_t^L(1)} + 15(\cdot\cdot y_t^L(1)) = 65 \\ 12e^{\cdot\cdot x_t^U(1)} + 15(\cdot\cdot y_t^U(1)) = 55 \\ 12e^{\cdot\cdot x_t^U(1)} + 15(\cdot\cdot y_t^U(1)) = 75 \end{cases}$$

Consequently  $x_t^U(0) = 0.5, x_t^L(0) = 0.5, x_t^L(0) = 0.5, x_t^U(0) = 0.5$  and  $x_t^U(0) = x_t^L(0) = x_t^L(0) = x_t^U(0) = \frac{1}{2}$ . After 4 iterations, we obtain the solution with a maximum error less than  $10^{-30}$ . Now suppose  $x, y$  are negative, we have

$$\begin{cases} 3e^{\cdot x_t^L(\tau)} + (\cdot y_t^L(\tau)) = 80 + 15\tau, \\ 3e^{\cdot x_t^L(\tau)} + (\cdot y_t^L(\tau)) = 120 - 15\tau, \\ 3e^{\cdot x_t^U(\tau)} + (\cdot y_t^U(\tau)) = 75 + 10\tau, \\ 3e^{\cdot x_t^U(\tau)} + (\cdot y_t^U(\tau)) = 125 - 10\tau, \\ 12e^{\cdot\cdot x_t^L(\tau)} + 15(\cdot\cdot y_t^L(\tau)) = 50 + 15\tau, \\ 12e^{\cdot\cdot x_t^L(\tau)} + 15(\cdot\cdot y_t^L(\tau)) = 80 - 15\tau, \\ 12e^{\cdot\cdot x_t^U(\tau)} + 15(\cdot\cdot y_t^U(\tau)) = 45 + 10\tau, \\ 12e^{\cdot\cdot x_t^U(\tau)} + 15(\cdot\cdot y_t^U(\tau)) = 95 - 20\tau. \end{cases}$$

For  $\tau = 0$ , we have,  $y_t^L(0) > y_t^L(0)$ , therefore negative solutions of IVTFNSEs does not exist.

Fig. 7, depicts the numerical solution of IVTFNEs, as well as the initial starting values and solution of the IVTF variables  $x, y$  and  $z$  utilized in Example 5. It also shows numerical approximations obtained with SFQM1 that are matched with analytical solutions up to 32 decimal places.

Table 17 shows the numerical approximate solution of the interval-valued trapezoidal system of nonlinear equations used in Example 5 by utilizing numerical schemes SFQM1 and SFQN1 up to 30 decimal places with 64-digit floating point arithmetic. Table 17 displays the approximated solution to four decimal places. The numerical approximate solution in Table 17 agrees with the analytical solution in Fig. 7.

In Table 18, C1 and C2 represents the stopping criteria 1 and 2 i.e.,  $\|F(x, \tau)\|, \|x_{n+1}(\tau) - x_n(\tau)\|$  for different values of  $\tau$  respectively. The approximate IVTFNE solutions in Table 18 clearly show that our newly developed family is more efficient than the classical method used in the literature.

**Table 17**  
Approximate solution of Example 5 using SFQM1 and SFQN1.

$\tau$	$x_1^L(\tau)$	$x_1^U(\tau)$	$x_2^L(\tau)$	$x_2^U(\tau)$	$y_1^L(\tau)$	$y_1^U(\tau)$	$y_2^L(\tau)$	$y_2^U(\tau)$
0.0	-0.5200	0.0	0.7842	1.3921	-3.6225	-0.8224	-0.5	-0.2413
0.1	-0.5274	-0.1162e-1	0.7842	1.3947	-3.6533	-0.8224	-0.4878	-0.2226
0.2	-0.5350	-0.2166e-1	0.7842	1.3973	-3.6846	-0.8224	-0.4761	-0.2041
0.3	-0.5427	-0.3043e-1	0.7842	1.4000	-3.7165	-0.8442	-0.4650	-0.1858
0.4	-0.5505	-0.3807e-1	0.7706	1.4026	-3.7490	-0.8224	-0.4544	-0.1677
0.5	-0.5583	-0.4474e-1	0.7635	1.4052	-3.7820	-0.8641	-0.4447	-0.1497
0.6	-0.5662	-0.5059e-1	0.7563	1.4079	-3.8156	-0.8554	-0.4343	-0.1319
0.7	-0.5741	-0.5571e-1	0.7490	1.4106	-3.8496	-0.8467	-0.4252	-0.1143
0.8	-0.5820	-0.5671e-1	0.7261	1.4136	-3.8647	-0.8467	-0.4215	-0.9694e-1
0.9	-0.5898	-0.5671e-1	0.7015	1.4154	-3.9203	-0.8467	-0.4315	-0.7966e-1
1.0	-0.5976	-0.5912e-1	0.7012	1.4186	-3.9566	-0.8467	-0.8467	-0.6254e-1

**Table 18**  
Error comparison of the numerical schemes SFQM1 and SFQN1.

Scheme	SFQM1				SFQN1			
	C1		C2		C1		C2	
$\tau$	x	y	x	y	x	y	x	y
0.0	0.3e-30	1.1e-32	1.1e-40	2.4e-41	3.4e-15	1.1e-17	0.6e-25	3.8e-26
0.1	4.4e-32	0.8e-33	2.2e-42	3.3e-44	1.9e-16	1.9e-18	1.7e-26	4.7e-27
0.2	2.2e-33	2.4e-35	2.7e-44	3.7e-45	2.2e-17	8.1e-19	2.6e-27	5.3e-28
0.3	2.7e-35	3.3e-36	2.9e-46	1.1e-47	3.2e-18	4.4e-20	3.3e-29	2.2e-30
0.4	3.8e-32	4.4e-34	3.5e-47	1.8e-48	3.8e-19	7.9e-21	3.7e-28	2.6e-29
0.5	3.0e-34	1.7e-36	3.7e-41	2.6e-42	1.9e-21	0.7e-22	4.2e-30	3.7e-31
0.6	4.8e-35	2.8e-37	4.1e-43	5.1e-44	0.8e-22	0.3e-23	4.7e-31	4.4e-32
0.7	1.8e-36	1.6e-38	4.2e-46	2.4e-47	3.8e-23	2.2e-24	2.6e-32	4.6e-33
0.8	2.7e-37	3.3e-39	5.1e-45	2.5e-46	4.4e-24	2.8e-25	3.3e-33	0.6e-34
0.9	4.7e-33	0.6e-35	1.1e-44	2.9e-45	4.8e-20	3.8e-22	4.8e-34	1.4e-35
1.0	4.6e-30	1.1e-32	2.2e-43	3.1e-44	4.9e-21	0.6e-24	5.7e-33	1.7e-34

**Table 19**  
Stability-consistency analysis.

Method	SFQM1	SFQN1
M-Error <sub>C1</sub>	0.3e-41	0.5e-16
M-Error <sub>C2</sub>	8.2e-32	1.0e-9
M-CPU-Time <sub>C1</sub>	0.11461	0.41311
M-CPU-Time <sub>C2</sub>	0.21074	0.52202
COC	3.2021	2.1013

**Table 20**  
Percentage convergence and total function evaluations for SFQM1, SFQN1.

Method	Number- F-J	Percentage-Con	M-CPU-Time <sub>C3</sub>	M-Error <sub>C3</sub>	COC
SFQM1	7	51.87756	2.5545	0.6454e-19	2.7645
SFQN1	13	21.99786	4.5434	4.7656e-07	1.8767

In Table 19, M-Error<sub>C1</sub>, M-Error<sub>C2</sub> show the error of numerical schemes SFQM1 and SFQN1 using stopping criteria I and II, respectively, M-CPU-Time<sub>C1</sub>, M-CPU-Time<sub>C2</sub> present the computational time in seconds of the numerical schemes for solving the solution of a fuzzy nonlinear system used in Example 5, employing stopping criteria I and II, and COC shows the computational order of convergence of the SFQM1 and SFQN1 methods. Tables 18–19 demonstrates that our newly developed method, SFQM1, is better than SFQN1 in terms of residual error, CPU time, and computational order of convergence. Table 19 also shows that the SFQM1 method is more stable and consistent than the SFQN1 method.

Table 20 demonstrates that in terms of percentage convergence, total functions, and Jacob’s evaluations, our technique SFQM1 outperforms the existing method SFQN1 and provides an alternative way to solve complex IVTFNE for greater comprehension and analysis.

**Table 21**  
Approximate solution of Example 6 using SFQM1 and SFQN1.

$\tau$	$\dot{x}_1^L(\tau)$	$\dot{x}_1^U(\tau)$	$\dot{x}_2^L(\tau)$	$\dot{x}_2^U(\tau)$	$\dot{y}_1^L(\tau)$	$\dot{y}_1^U(\tau)$	$\dot{y}_2^L(\tau)$	$\dot{y}_2^U(\tau)$
0.0	-0.4664	-0.4476	-0.1560	0.4142	-0.4801	-0.3403	0.73553	0.4238
0.1	-0.4630	-0.4515	-0.1584	0.4119	-0.4877	-0.3426	0.3589	0.4219
0.2	-0.4596	-0.4554	-0.1608	0.4096	-0.4747	-0.3449	0.3626	0.4200
0.3	-0.4528	-0.4592	-0.1631	0.4072	-0.4719	-0.3495	0.3697	0.4160
0.4	-0.4528	-0.4630	-0.1654	0.4049	-0.4692	-0.3496	0.3732	0.4161
0.5	-0.4593	-0.4668	-0.1676	0.4025	-0.4664	-0.3518	0.33767	0.4142
0.6	-0.4458	-0.4705	-0.1699	0.4001	-0.4636	-0.3540	0.3802	0.4123
0.7	-0.4423	-0.4742	-0.1721	0.3977	-0.4608	-0.3563	0.3836	0.4104
0.8	-0.4388	-0.4742	-0.2743	0.3953	-0.4579	0.0585	0.3870	0.4084
0.9	-0.4352	-0.4742	-0.3764	0.3929	-0.4551	0.1607	0.3903	0.4064
1.0	-0.4316	-0.4742	-0.4742	0.3904	-0.4522	0.3905	0.3905	0.4045

**Example 6:** Consider the nonlinear homogeneous system of IVTFNEs of degree 2 in which the coefficient matrix is the crisp real valued number and the right side column vector contains IVTF along with exponential and logarithmic fuzzy functions as:

$$\begin{cases} 3x^2 + 2y^2 = \left\langle (0.65, 0.73, 0.87, 0.95; \frac{2}{3}), (0.6, 0.7, 0.9, 1.1; 1) \right\rangle, \\ 4xy + 7y^2 = \left\langle (0.45, 0.53, 0.67, 0.75; \frac{2}{3}), (0.4, 0.5, 0.7, 0.8; 1) \right\rangle. \end{cases}$$

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

$$\begin{cases} 3x^2 + 2y^2 = \langle (0.65 + 0.12\tau, 0.95 - 0.12\tau), (0.6 + 0.1\tau, 1 - 0.1\tau) \rangle, \\ 4xy + 7y^2 = \langle (0.45 + 0.12\tau, 0.75 - 0.12\tau), (0.4 + 0.1\tau, 0.8 - 0.1\tau) \rangle. \end{cases}$$

or

$$\begin{cases} 3(\dot{x}_1^L(\tau))^2 + 2(\dot{y}_1^L(\tau))^2 = 0.65 + 0.12\tau, \\ 3(\dot{x}_1^U(\tau))^2 + 2(\dot{y}_1^U(\tau))^2 = 0.95 - 0.12\tau, \\ 3(\dot{x}_2^L(\tau))^2 + 2(\dot{y}_2^L(\tau))^2 = 0.6 + 0.1\tau, \\ 3(\dot{x}_2^U(\tau))^2 + 2(\dot{y}_2^U(\tau))^2 = 1 - 0.1\tau, \\ 4\dot{x}_1^L(\tau) * \dot{y}_1^L(\tau) + 7(\dot{y}_1^L(\tau))^2 = 0.45 + 0.12\tau, \\ 4\dot{x}_1^U(\tau) * \dot{y}_1^U(\tau) + 7(\dot{y}_1^U(\tau))^2 = 0.75 - 0.12\tau, \\ 4\dot{x}_2^L(\tau) * \dot{y}_2^L(\tau) + 7(\dot{y}_2^L(\tau))^2 = 0.4 + 0.1\tau, \\ 4\dot{x}_2^U(\tau) * \dot{y}_2^U(\tau) + 7(\dot{y}_2^U(\tau))^2 = 0.8 - 0.1\tau. \end{cases}$$

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

$$\begin{cases} 3(\dot{x}_1^L(0))^2 + 2(\dot{y}_1^L(0))^2 = 0.650 \\ 3(\dot{x}_1^U(0))^2 + 2(\dot{y}_1^U(0))^2 = 0.950 \\ 3(\dot{x}_2^L(0))^2 + 2(\dot{y}_2^L(0))^2 = 0.60 \\ 3(\dot{x}_2^U(0))^2 + 2(\dot{y}_2^U(0))^2 = 1.0 \\ 4\dot{x}_1^L(0) * \dot{y}_1^L(0) + 7(\dot{y}_1^L(0))^2 = 0.450 \\ 4\dot{x}_1^U(0) * \dot{y}_1^U(0) + 7(\dot{y}_1^U(0))^2 = 0.750 \\ 4\dot{x}_2^L(0) * \dot{y}_2^L(0) + 7(\dot{y}_2^L(0))^2 = 0.40 \\ 4\dot{x}_2^U(0) * \dot{y}_2^U(0) + 7(\dot{y}_2^U(0))^2 = 0.80 \end{cases} \quad \text{and} \quad \begin{cases} 3(\dot{x}_1^L(1))^2 + 2(\dot{y}_1^L(1))^2 = 0.77 \\ 3(\dot{x}_1^U(1))^2 + 2(\dot{y}_1^U(1))^2 = 0.83 \\ 3(\dot{x}_2^L(1))^2 + 2(\dot{y}_2^L(1))^2 = 0.7 \\ 3(\dot{x}_2^U(1))^2 + 2(\dot{y}_2^U(1))^2 = 0.9 \\ 4\dot{x}_1^L(1) * \dot{y}_1^L(1) + 7(\dot{y}_1^L(1))^2 = 0.57 \\ 4\dot{x}_1^U(1) * \dot{y}_1^U(1) + 7(\dot{y}_1^U(1))^2 = 0.63 \\ 4\dot{x}_2^L(1) * \dot{y}_2^L(1) + 7(\dot{y}_2^L(1))^2 = 0.5 \\ 4\dot{x}_2^U(1) * \dot{y}_2^U(1) + 7(\dot{y}_2^U(1))^2 = 0.7 \end{cases}$$

Consequently  $\dot{x}_1^U(0) = 0.5, \dot{x}_1^L(0) = 0.5, \dot{x}_2^L(0) = \dot{x}_2^U(0) = 0.75$  and  $\dot{x}_1^U(1) = \dot{x}_1^L(1) = \dot{x}_2^L(1) = \dot{x}_2^U(1) = 0.5$ . After 4 iterations, we obtain the solution with a maximum error less than  $10^{-30}$ . Now suppose  $x, y$  are negative, we have

$$\begin{cases} 3(\dot{x}_1^L(\tau))^2 + 2(\dot{y}_1^L(\tau))^2 = 0.65 + 0.12\tau, \\ 3(\dot{x}_1^U(\tau))^2 + 2(\dot{y}_1^U(\tau))^2 = 0.95 - 0.12\tau, \\ 3(\dot{x}_2^L(\tau))^2 + 2(\dot{y}_2^L(\tau))^2 = 0.6 + 0.1\tau, \\ 3(\dot{x}_2^U(\tau))^2 + 2(\dot{y}_2^U(\tau))^2 = 1 - 0.1\tau, \\ 4\dot{x}_1^L(\tau) * \dot{y}_1^L(\tau) + 7(\dot{y}_1^L(\tau))^2 = 0.45 + 0.12\tau, \\ 4\dot{x}_1^U(\tau) * \dot{y}_1^U(\tau) + 7(\dot{y}_1^U(\tau))^2 = 0.75 - 0.12\tau, \\ 4\dot{x}_2^L(\tau) * \dot{y}_2^L(\tau) + 7(\dot{y}_2^L(\tau))^2 = 0.4 + 0.1\tau, \\ 4\dot{x}_2^U(\tau) * \dot{y}_2^U(\tau) + 7(\dot{y}_2^U(\tau))^2 = 0.8 - 0.1\tau. \end{cases}$$

For  $\tau = 0$ , we have,  $\dot{y}_1^L(0) > \dot{y}_1^U(0)$ , therefore negative solutions of IVTFNSEs does not exist.

**Fig. 8,** depicts the numerical solution of IVTFNEs, as well as the initial starting values and solution of the IVTF variables  $x, y$  and  $z$  utilized in Example 6. This figure shows numerical approximations obtained with SFQM1 that are matched with analytical solutions up to 32 decimal places.

**Table 21** shows the numerical approximate solution of the interval-valued trapezoidal system of nonlinear equations used in Example 6 by utilizing numerical schemes SFQM1 and SFQN1 up to 30 decimal places with 64-digit floating point arithmetic. It

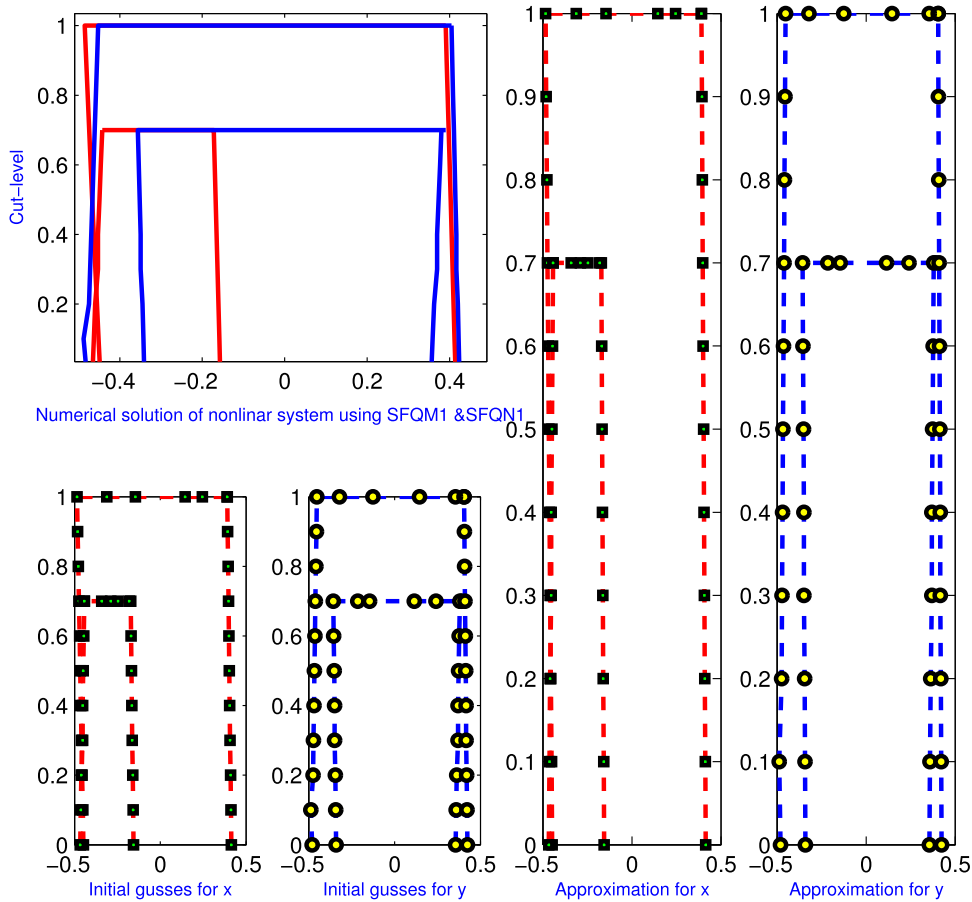


Fig. 8. Approximate solution of IVTFNEs used in Example 6.

Table 22

Error comparison of the numerical schemes SFQM1 and SFQN1.

Scheme	SFQM1				SFQN1			
	C1		C2		C1		C2	
$\tau$	x	y	x	y	x	y	x	y
0.0	8.3e-21	0.1e-37	3.1e-40	2.3e-42	8.4e-13	0.3e-15	0.6e-22	3.8e-26
0.1	4.4e-22	8.8e-33	2.4e-40	4.3e-44	1.9e-10	4.9e-14	7.7e-21	7.7e-21
0.2	0.2e-28	9.4e-33	4.7e-44	3.7e-41	2.0e-15	8.4e-18	7.6e-20	5.0e-22
0.3	0.7e-27	9.3e-31	3.9e-46	5.1e-37	1.2e-14	0.4e-24	7.3e-29	2.3e-25
0.4	1.8e-29	4.9e-30	4.5e-45	3.3e-48	6.8e-14	6.9e-22	7.7e-28	2.6e-27
0.5	1.0e-25	1.9e-36	5.7e-43	5.5e-42	3.9e-25	6.7e-21	7.2e-30	3.7e-21
0.6	2.8e-25	3.8e-34	5.1e-43	5.1e-47	7.8e-20	9.3e-21	0.7e-30	9.4e-22
0.7	3.8e-26	4.6e-32	6.2e-48	2.4e-44	3.0e-24	2.0e-23	1.6e-31	3.9e-33
0.8	3.3e-27	7.3e-33	5.6e-45	6.5e-40	0.4e-21	9.8e-26	1.3e-33	0.6e-31
0.9	3.7e-33	6.6e-30	1.1e-40	7.9e-41	9.8e-21	3.8e-21	4.1e-32	1.0e-32
1.0	3.8e-30	0.1e-32	9.8e-41	7.1e-42	8.9e-20	7.6e-20	5.7e-30	8.7e-31

displays the approximated solution to four decimal places. The numerical approximate solution in Table 21 agrees with the analytical solution in Fig. 8.

In Table 22, C1 and C2 represents the stopping criteria 1 and 2 i.e.,  $\|F(x, \tau)\|$ ,  $\|x_{n+1}(\tau) - x_n(\tau)\|$  for different values of  $\tau$  respectively. The approximate IVTFNE solutions in Table 22 clearly show that our newly developed family is more efficient than the classical method used in the literature.

In Table 23, M-Error<sub>C1</sub>, M-Error<sub>C2</sub> show the error of numerical schemes SFQM1 and SFQN1 using stopping criteria I and II, respectively, M-CPU-Time<sub>C1</sub>, M-CPU-Time<sub>C2</sub> present the computational time in seconds of the numerical schemes for solving the solution of a fuzzy nonlinear system used in Example 6, employing stopping criteria I and II, and COC shows the computational order

**Table 23**  
Stability-consistency analysis.

Method	SFQM1	SFQN1
M-Error <sub>C1</sub>	8.3e-21	1.9e-10
M-Error <sub>C2</sub>	5.1e-37	7.6e-20
M-CPU-Time <sub>C1</sub>	0.11461	0.41311
M-CPU-Time <sub>C2</sub>	0.21074	0.52202
COC	3.2021	2.1013

**Table 24**  
Percentage convergence and total function evaluations for SFQM1, SFQN1.

Method	Number- F-J	Percentage-Con	M-CPU-Time <sub>C3</sub>	M-Error <sub>C3</sub>	COC
SFQM1	7	67.95434	0.3875743	9.945e-17	2.03001
SFQN1	6	66.87655	0.9654522	4.945e-15	2.00434

of convergence of the SFQM1 and SFQN1 methods. Tables 21–23, demonstrates that our newly developed method, SFQM1, is better than SFQN1 in terms of residual error, CPU time, and computational order of convergence. Table 23 also shows that the SFQM1 method is more stable and consistent than the SFQN1 method.

Table 24, demonstrates that in terms of percentage convergence, total functions, and Jacob’s evaluations, our technique SFQM1 outperforms the existing method SFQN1 and provides an alternative way to solve complex IVTFNE for greater comprehension and analysis.

**Example 7:** Consider the nonlinear system of IVTFNEs in which the coefficient matrix is the crisp real valued number and the right side column vector contains IVTF along with trigonometric fuzzy functions as:

$$\begin{cases} 3x + 7y + 8 \sin z = \left\langle (10, 20, 30, 40; \frac{2}{3}), (5, 15, 35, 45; 1) \right\rangle, \\ x + 3y + 7z = \left\langle (50, 60, 70, 80; \frac{2}{3}), (45, 55, 75, 95; 1) \right\rangle, \\ 2x + y + z = \left\langle (10, 20, 30, 40; \frac{2}{3}), (5, 15, 35, 45; 1) \right\rangle. \end{cases}$$

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

$$\begin{cases} 3x + 7y + 8 \sin z = \langle (10 + 15\tau, 40 - 15\tau), (5 + 10\tau, 45 - 10\tau) \rangle, \\ x + 3y + 7z = \langle (50 + 15\tau, 80 - 15\tau), (45 + 10\tau, 95 - 20\tau) \rangle, \\ 2x + y + z = \langle (10 + 15\tau, 40 - 15\tau), (5 + 10\tau, 45 - 10\tau) \rangle. \end{cases}$$

or

$$\begin{cases} 3 \cdot x^L(\tau) + 7(\cdot y^L(\tau)) + \sin(\cdot z^L(\tau)) = 10 + 15\tau, \\ 3 \cdot x^L(\tau) + 7(\cdot y^L(\tau)) + \sin(\cdot z^L(\tau)) = 40 - 15\tau, \\ 3 \cdot x^U(\tau) + 7(\cdot y^U(\tau)) + \sin(\cdot z^U(\tau)) = 5 + 10\tau, \\ 3 \cdot x^U(\tau) + 7(\cdot y^U(\tau)) + \sin(\cdot z^U(\tau)) = 45 - 10\tau, \\ \cdot x^L(\tau) + 3(\cdot y^L(\tau)) + 7(\cdot z^L(\tau)) = 50 + 15\tau, \\ \cdot x^L(\tau) + 3(\cdot y^L(\tau)) + 7(\cdot z^L(\tau)) = 80 - 15\tau, \\ \cdot x^U(\tau) + 3(\cdot y^U(\tau)) + 7(\cdot z^U(\tau)) = 45 + 10\tau, \\ \cdot x^U(\tau) + 3(\cdot y^U(\tau)) + 7(\cdot z^U(\tau)) = 95 - 20\tau, \\ 2 \cdot x^L(\tau) + (\cdot y^L(\tau)) + (\cdot z^L(\tau)) = 10 + 15\tau, \\ 2 \cdot x^L(\tau) + (\cdot y^L(\tau)) + (\cdot z^L(\tau)) = 40 - 15\tau, \\ 2 \cdot x^U(\tau) + (\cdot y^U(\tau)) + (\cdot z^U(\tau)) = 5 + 10\tau, \\ 2 \cdot x^U(\tau) + (\cdot y^U(\tau)) + (\cdot z^U(\tau)) = 45 - 10\tau. \end{cases}$$

**Table 25**  
Approximate solution of Example 7 using SFQM1 and SFQN1.

$\tau$	$\dot{x}_t^L(\tau)$	$\dot{x}_t^U(\tau)$	$\dot{x}_t^L(\tau)$	$\dot{x}_t^U(\tau)$	$\dot{y}_t^L(\tau)$	$\dot{y}_t^U(\tau)$	$\dot{y}_t^L(\tau)$	$\dot{y}_t^U(\tau)$	$\dot{z}_t^L(\tau)$	$\dot{z}_t^U(\tau)$	$\dot{z}_t^L(\tau)$	$\dot{z}_t^U(\tau)$
0.0	1.4857	0.1711	6.8572	15.6437	-0.8421	9.5546	-1.300	1.7251	5.8749	16.8101	0.3668	11.0127
0.1	2.2840	-0.1730	7.1050	14.9629	-0.8796	9.4537	0.7438	1.4175	6.0701	16.4267	0.3629	10.7834
0.2	3.0580	-0.4387	7.3225	14.2822	-0.9173	9.3528	-0.1810	1.0884	6.2736	16.0605	0.3033	10.5756
0.3	3.8124	-0.6405	7.5155	13.6013	-0.9545	9.2517	0.3820	0.7584	6.4775	15.7065	0.2037	10.3831
0.4	4.5512	-0.7923	7.6893	12.9200	-0.9900	9.1500	0.9390	0.4481	6.6738	15.3615	0.0750	10.2019
0.5	5.2784	-0.9050	7.8480	12.3378	-1.0231	9.0473	1.4857	0.1711	6.8572	15.0230	-0.0750	10.0288
0.6	5.995	-0.9869	7.9949	11.5546	-1.0526	8.9433	2.0208	-0.0676	7.0260	14.6893	-0.2404	9.8616
0.7	6.7059	-1.0442	8.1324	10.8700	-1.0775	8.8375	2.5444	-0.2696	7.1806	14.3588	-0.4162	9.6985
0.8	7.4097	-1.0817	8.2621	10.1835	-1.0965	8.7294	3.0580	0.4387	7.3225	14.0302	-0.5983	9.5378
0.9	8.1086	-1.1030	8.3857	9.4948	-1.1082	8.6185	3.6290	-0.5794	7.4536	13.7024	-0.7829	9.3780
1.0	8.5754	-1.1110	8.5042	8.8034	-1.1110	8.5042	4.0603	-0.6960	7.5754	13.3743	-0.9665	9.2117

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

$$\left\{ \begin{array}{l} 3 \dot{x}_t^L(0) + 7(\dot{y}_t^L(0)) + \sin(\dot{z}_t^L(0)) = 10 \\ 3 \dot{x}_t^U(0) + 7(\dot{y}_t^U(0)) + \sin(\dot{z}_t^U(0)) = 40 \\ 3 \dot{x}_t^L(0) + 7(\dot{y}_t^L(0)) + \sin(\dot{z}_t^U(0)) = 5 \\ 3 \dot{x}_t^U(0) + 7(\dot{y}_t^U(0)) + \sin(\dot{z}_t^L(0)) = 45 \\ \dot{x}_t^L(0) + 3(\dot{y}_t^L(0)) + 7(\dot{z}_t^L(0)) = 50 \\ \dot{x}_t^U(0) + 3(\dot{y}_t^U(0)) + 7(\dot{z}_t^U(0)) = 80 \\ \dot{x}_t^L(0) + 3(\dot{y}_t^U(0)) + 7(\dot{z}_t^L(0)) = 45 \\ \dot{x}_t^U(0) + 3(\dot{y}_t^L(0)) + 7(\dot{z}_t^U(0)) = 95 \\ 2 \dot{x}_t^L(0) + (\dot{y}_t^L(0)) + (\dot{z}_t^L(0)) = 10 \\ 2 \dot{x}_t^U(0) + (\dot{y}_t^U(0)) + (\dot{z}_t^U(0)) = 40 \\ 2 \dot{x}_t^L(0) + (\dot{y}_t^U(0)) + (\dot{z}_t^L(0)) = 5 \\ 2 \dot{x}_t^U(0) + (\dot{y}_t^L(0)) + (\dot{z}_t^U(0)) = 45 \end{array} \right. \text{ and } \left\{ \begin{array}{l} 3 \dot{x}_t^L(1) + 7(\dot{y}_t^L(1)) + \sin(\dot{z}_t^L(1)) = 25 \\ 3 \dot{x}_t^U(1) + 7(\dot{y}_t^U(1)) + \sin(\dot{z}_t^U(1)) = 25 \\ 3 \dot{x}_t^L(1) + 7(\dot{y}_t^L(1)) + \sin(\dot{z}_t^U(1)) = 25 \\ 3 \dot{x}_t^U(1) + 7(\dot{y}_t^U(1)) + \sin(\dot{z}_t^L(1)) = 35 \\ \dot{x}_t^L(1) + 3(\dot{y}_t^L(1)) + 7(\dot{z}_t^L(1)) = 65 \\ \dot{x}_t^U(1) + 3(\dot{y}_t^U(1)) + 7(\dot{z}_t^U(1)) = 65 \\ \dot{x}_t^L(1) + 3(\dot{y}_t^U(1)) + 7(\dot{z}_t^L(1)) = 55 \\ \dot{x}_t^U(1) + 3(\dot{y}_t^L(1)) + 7(\dot{z}_t^U(1)) = 75 \\ 2 \dot{x}_t^L(1) + (\dot{y}_t^L(1)) + (\dot{z}_t^L(1)) = 25 \\ 2 \dot{x}_t^U(1) + (\dot{y}_t^U(1)) + (\dot{z}_t^U(1)) = 25 \\ 2 \dot{x}_t^L(1) + (\dot{y}_t^U(1)) + (\dot{z}_t^L(1)) = 15 \\ 2 \dot{x}_t^U(1) + (\dot{y}_t^L(1)) + (\dot{z}_t^U(1)) = 35 \end{array} \right.$$

Consequently  $x_t^U(0) = 0.5, x_t^L(0) = 0.5, x_t^L(0) = 0.5, x_t^U(0) = 0.5$  and  $x_t^U(0) = x_t^L(0) = x_t^L(0) = x_t^U(0) = \frac{1}{2}$ . After 4 iterations, we obtain the solution with the maximum error less than  $10^{-30}$ . Suppose  $x, y, z$  are negative, we have

$$\left\{ \begin{array}{l} 3 \dot{x}_t^L(\tau) + 7(\dot{y}_t^L(\tau)) + \sin(\dot{z}_t^L(\tau)) = 10 + 15\tau, \\ 3 \dot{x}_t^U(\tau) + 7(\dot{y}_t^U(\tau)) + \sin(\dot{z}_t^U(\tau)) = 40 - 15\tau \\ 3 \dot{x}_t^L(\tau) + 7(\dot{y}_t^L(\tau)) + \sin(\dot{z}_t^U(\tau)) = 5 + 10\tau, \\ 3 \dot{x}_t^U(\tau) + 7(\dot{y}_t^U(\tau)) + \sin(\dot{z}_t^L(\tau)) = 45 - 10\tau, \\ \dot{x}_t^L(\tau) + 3(\dot{y}_t^L(\tau)) + 7(\dot{z}_t^L(\tau)) = 50 + 15\tau, \\ \dot{x}_t^U(\tau) + 3(\dot{y}_t^U(\tau)) + 7(\dot{z}_t^U(\tau)) = 80 - 15\tau, \\ \dot{x}_t^L(\tau) + 3(\dot{y}_t^U(\tau)) + 7(\dot{z}_t^L(\tau)) = 45 + 10\tau, \\ \dot{x}_t^U(\tau) + 3(\dot{y}_t^L(\tau)) + 7(\dot{z}_t^U(\tau)) = 95 - 20\tau, \\ 2 \dot{x}_t^L(\tau) + (\dot{y}_t^L(\tau)) + (\dot{z}_t^L(\tau)) = 10 + 15\tau, \\ 2 \dot{x}_t^U(\tau) + (\dot{y}_t^U(\tau)) + (\dot{z}_t^U(\tau)) = 40 - 15\tau, \\ 2 \dot{x}_t^L(\tau) + (\dot{y}_t^U(\tau)) + (\dot{z}_t^L(\tau)) = 5 + 10\tau, \\ 2 \dot{x}_t^U(\tau) + (\dot{y}_t^L(\tau)) + (\dot{z}_t^U(\tau)) = 45 - 10\tau. \end{array} \right.$$

For  $\tau = 0$ , we have,  $y_t^L(0) > y_t^U(0)$ , therefore negative solutions of IVTFNSES does not exist.

Fig. 9 depicts the numerical solution of IVTFNSES, as well as the initial starting values and solution of the IVTF variables  $x, y$  and  $z$  utilized in Example 7. This figure also shows numerical approximations obtained with SFQM1 that are matched with analytical solutions up to 30 decimal places. Table 25, shows the numerical approximate solution of the interval-valued trapezoidal system of nonlinear equations used in Example 7 by utilizing numerical schemes SFQM1 and SFQN1 up to 30 decimal places with 64-digit floating point arithmetic. It display the approximated solution to four decimal places. The numerical approximate solution in Table 25 agrees with the analytical solution shown in Fig. 9.

In Table 26, C1 and C2 represents the stopping criteria 1 and 2 i.e.,  $\|F(x, \tau)\|, \|x_{n+1}(\tau) - x_n(\tau)\|$  respectively for different values of  $\tau$  respectively. The approximate IVTFNSES in Table 26 clearly show that our newly developed family is more efficient than the classical method used in the literature.

In Table 27, M-Error<sub>C1</sub>, M-Error<sub>C2</sub> show the error of numerical schemes SFQM1 and SFQN1 using stopping criteria I and II, respectively, M-CPU-Time<sub>C1</sub>, M-CPU-Time<sub>C2</sub> present the computational time in seconds of the numerical schemes for solving the solution of a fuzzy nonlinear system used in Example 7, employing stopping criteria I and II, and COC shows the computational order of convergence of the SFQM1 and SFQN1 methods. Tables 25–27, demonstrates that our newly developed method, SFQM1, is better than SFQN1 in terms of residual error, CPU time, and computational order of convergence. Table 27 also shows that the SFQM1 method is more stable and consistent than the SFQN1 method.

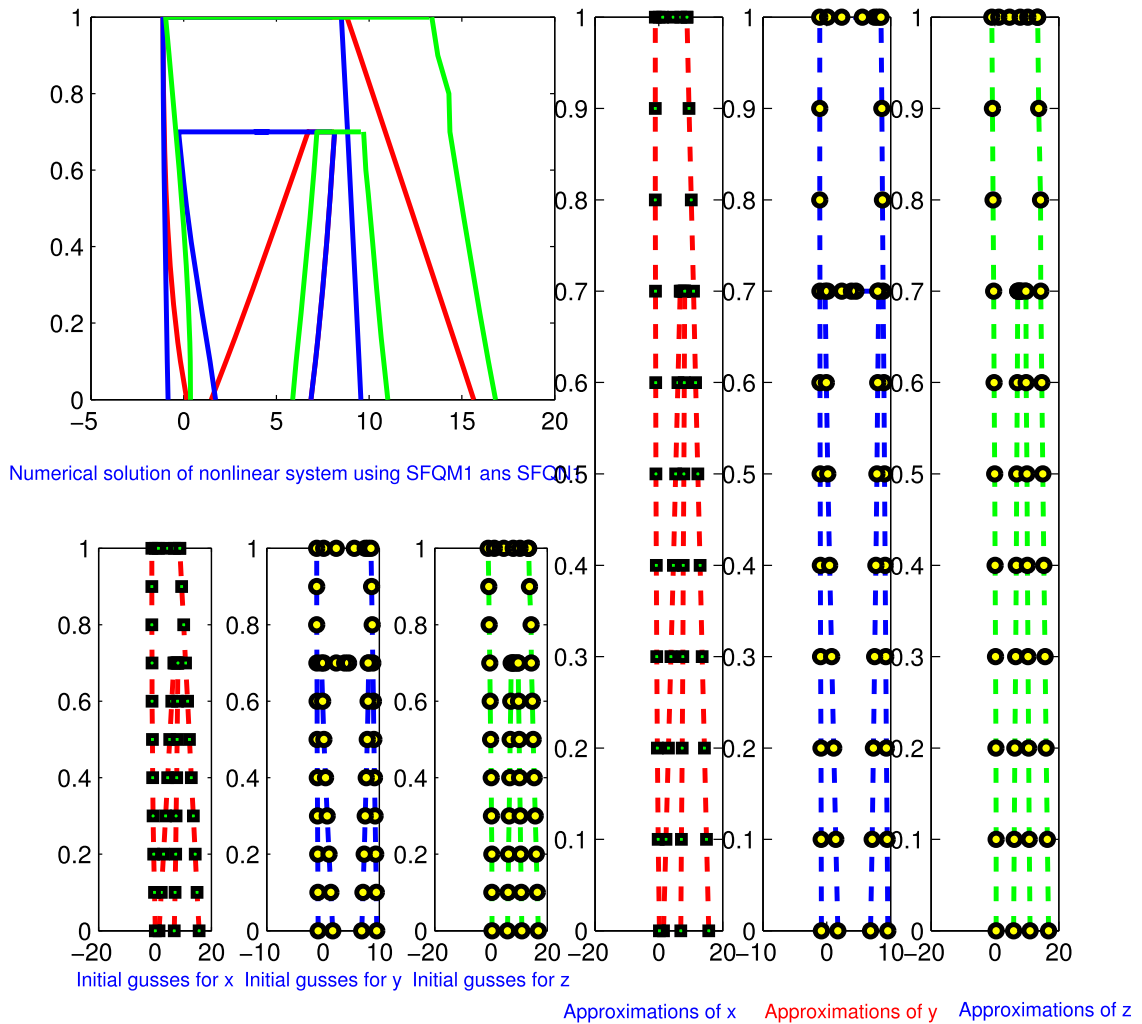


Fig. 9. Approximate solution of IVTFNEs used in Example 7.

Table 28 demonstrates that in terms of percentage convergence, total functions, and Jacob’s evaluations, our technique SFQM1 outperforms the existing method SFQ1 and provides an alternative way to solve complex IVTFNE for greater comprehension and analysis.

**Example 8:** Consider the nonlinear system of IVTFNEs in which the coefficient matrix is crisp real valued number as well as IVTF and the right side column vector contains IVTF along with trigonometric fuzzy functions as:

$$\begin{cases} x - 3y + z = \langle (50, 60, 70, 80; \frac{2}{3}), (45, 55, 75, 95; 1) \rangle, \\ \sin(x) + y = \langle (10, 20, 30, 40; \frac{2}{3}), (5, 15, 35, 45; 1) \rangle, \\ \left( \langle (10, 20, 30, 40; \frac{2}{3}), (5, 15, 35, 45; 1) \rangle \right) x - y - z = \langle (50, 60, 70, 80; \frac{2}{3}), (45, 55, 75, 95; 1) \rangle. \end{cases}$$

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

$$\begin{cases} x - 3y + z = \langle (50 + 15\tau, 80 - 15\tau), (45 + 10\tau, 95 - 20\tau) \rangle, \\ \sin(x) + y = \langle (10 + 15\tau, 40 - 15\tau), (5 + 10\tau, 45 - 10\tau) \rangle, \\ \left( \langle (10 + 15\tau, 40 - 15\tau), (5 + 10\tau, 45 - 10\tau) \rangle \right) x - y - z = \langle (50 + 15\tau, 80 - 15\tau), (45 + 10\tau, 95 - 20\tau) \rangle. \end{cases}$$

**Table 26**  
Error comparison of the numerical schemes SFQM1 and SFQN1.

Scheme		SFQM1						SFQN1					
Criteria	C1			C2			C1			C2			
$\tau$	x	y	z	x	y	z	x	y	z	x	y	z	
0.0	0.3e-30	1.1e-32	1.1e-32	1.1e-40	2.4e-41	0.4e-48	3.4e-15	1.1e-17	1.1e-17	0.9e-25	0.6e-25	3.0e-26	
0.1	8.4e-32	0.8e-33	0.8e-33	2.2e-42	3.3e-44	3.0e-49	8.9e-16	1.9e-18	1.9e-18	1.9e-26	1.7e-26	4.2e-27	
0.2	8.2e-33	2.4e-35	2.4e-35	2.7e-44	3.0e-45	4.7e-55	8.2e-17	8.1e-19	8.1e-19	2.8e-27	2.0e-27	5.0e-28	
0.3	2.7e-35	1.3e-36	3.3e-36	0.9e-46	0.1e-47	2.1e-47	3.2e-18	4.4e-20	4.4e-20	3.4e-29	3.0e-29	2.2e-30	
0.4	3.8e-32	0.4e-34	4.4e-34	0.5e-47	0.8e-48	0.8e-48	3.3e-19	7.9e-18	7.9e-21	3.2e-28	3.0e-28	2.0e-29	
0.5	3.0e-34	0.7e-36	1.7e-36	0.7e-41	0.6e-42	2.0e-50	1.0e-21	0.7e-18	0.7e-18	4.0e-30	4.0e-30	3.0e-31	
0.6	4.0e-35	0.8e-37	2.8e-37	4.0e-43	5.1e-44	5.1e-49	0.8e-22	0.3e-23	0.3e-19	4.0e-29	4.0e-31	4.4e-32	
0.7	1.8e-36	1.0e-38	1.6e-38	5.2e-46	2.0e-47	7.4e-55	3.0e-22	2.2e-20	2.2e-20	2.3e-29	2.1e-32	4.0e-33	
0.8	2.0e-37	0.3e-39	3.3e-39	5.0e-45	2.5e-46	2.7e-48	4.4e-22	2.8e-20	2.8e-20	3.2e-29	3.1e-33	0.6e-34	
0.9	4.7e-33	0.6e-35	0.6e-35	1.6e-44	2.0e-45	2.7e-49	4.8e-20	3.8e-19	3.8e-21	4.0e-28	4.2e-34	0.4e-35	
1.0	4.0e-30	1.0e-32	1.1e-32	2.6e-43	3.0e-44	3.19e-48	4.9e-18	0.6e-20	0.6e-20	5.7e-28	1.0e-33	0.7e-34	

**Table 27**  
Stability-consistency analysis.

Method	SFQM1	SFQN1
M-Error <sub>C1</sub>	0.6e-30	0.5e-15
M-Error <sub>C2</sub>	7.2e-40	1.0e-28
M-CPU-Time <sub>C1</sub>	0.02421	0.11051
M-CPU-Time <sub>C2</sub>	0.01313	0.32272
COC	3.0021	2.00131

**Table 28**  
Percentage convergence and total function evaluations for SFQM1, SFQN1.

Method	Number- F-J	Percentage-Con	M-CPU-Time <sub>C3</sub>	M-Error <sub>C3</sub>	COC
SFQM1	9	96.007676	3.5664665	0.509e-21	2.1132
SFQN1	14	75.80881	4.0067577	4.287e-13	1.8876

or

$$\begin{cases} \dot{x}_t^L(\tau) - 3(\dot{y}_t^L(\tau)) + (\dot{z}_t^L(\tau)) = 50 + 15\tau, \\ \dot{x}_t^L(\tau) - 3(\dot{y}_t^L(\tau)) + (\dot{z}_t^L(\tau)) = 80 - 15\tau, \\ \dot{x}_t^U(\tau) - 3(\dot{y}_t^U(\tau)) + (\dot{z}_t^U(\tau)) = 45 + 10\tau, \\ \dot{x}_t^U(\tau) - 3(\dot{y}_t^U(\tau)) + (\dot{z}_t^U(\tau)) = 95 - 20\tau, \\ \sin(\dot{x}_t^L(\tau)) + (\dot{y}_t^L(\tau)) = 10 + 15\tau, \\ \sin(\dot{x}_t^L(\tau)) + (\dot{y}_t^L(\tau)) = 40 - 15\tau, \\ \sin(\dot{x}_t^U(\tau)) + (\dot{y}_t^U(\tau)) = 5 + 10\tau, \\ \sin(\dot{x}_t^U(\tau)) + (\dot{y}_t^U(\tau)) = 45 - 10\tau, \\ (10 + 15\tau)\dot{x}_t^L(\tau) - (\dot{y}_t^L(\tau)) - (\dot{z}_t^L(\tau)) = 50 + 15\tau, \\ (40 - 15\tau)\dot{x}_t^L(\tau) - (\dot{y}_t^L(\tau)) - (\dot{z}_t^L(\tau)) = 80 - 15\tau, \\ (5 + 10\tau)\dot{x}_t^U(\tau) - (\dot{y}_t^U(\tau)) - (\dot{z}_t^U(\tau)) = 45 + 10\tau, \\ (45 - 10\tau)\dot{x}_t^U(\tau) - (\dot{y}_t^U(\tau)) - (\dot{z}_t^U(\tau)) = 95 - 20\tau. \end{cases}$$

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

$$\begin{cases} \dot{x}_t^L(0) - 3(\dot{y}_t^L(0)) + (\dot{z}_t^L(0)) = 50 \\ \dot{x}_t^L(0) - 3(\dot{y}_t^L(0)) + (\dot{z}_t^L(0)) = 80 \\ \dot{x}_t^U(0) - 3(\dot{y}_t^U(0)) + (\dot{z}_t^U(0)) = 45 \\ \dot{x}_t^U(0) - 3(\dot{y}_t^U(0)) + (\dot{z}_t^U(0)) = 95 \\ \sin(\dot{x}_t^L(0)) + (\dot{y}_t^L(0)) = 10 \\ \sin(\dot{x}_t^L(0)) + (\dot{y}_t^L(0)) = 40 \\ \sin(\dot{x}_t^U(0)) + (\dot{y}_t^U(0)) = 5 \\ \sin(\dot{x}_t^U(0)) + (\dot{y}_t^U(0)) = 45 \\ (10)\dot{x}_t^L(0) - (\dot{y}_t^L(0)) - (\dot{z}_t^L(0)) = 50 \\ (40)\dot{x}_t^L(0) - (\dot{y}_t^L(0)) - (\dot{z}_t^L(0)) = 80 \\ (5)\dot{x}_t^U(0) - (\dot{y}_t^U(0)) - (\dot{z}_t^U(0)) = 45 \\ (45)\dot{x}_t^U(0) - (\dot{y}_t^U(0)) - (\dot{z}_t^U(0)) = 95 \end{cases} \quad \text{and} \quad \begin{cases} \dot{x}_t^L(1) - 3(\dot{y}_t^L(1)) + (\dot{z}_t^L(1)) = 65 \\ \dot{x}_t^L(1) - 3(\dot{y}_t^L(1)) + (\dot{z}_t^L(1)) = 65 \\ \dot{x}_t^U(1) - 3(\dot{y}_t^U(1)) + (\dot{z}_t^U(1)) = 55 \\ \dot{x}_t^U(1) - 3(\dot{y}_t^U(1)) + (\dot{z}_t^U(1)) = 75 \\ \sin(\dot{x}_t^L(1)) + (\dot{y}_t^L(1)) = 25 \\ \sin(\dot{x}_t^L(1)) + (\dot{y}_t^L(1)) = 25 \\ \sin(\dot{x}_t^U(1)) + (\dot{y}_t^U(1)) = 25 \\ \sin(\dot{x}_t^U(1)) + (\dot{y}_t^U(1)) = 25 \\ (10 + 15\tau)\dot{x}_t^L(1) - (\dot{y}_t^L(1)) - (\dot{z}_t^L(\tau)) = 65 \\ (40 - 15\tau)\dot{x}_t^L(1) - (\dot{y}_t^L(1)) - (\dot{z}_t^L(\tau)) = 65 \\ (5 + 10\tau)\dot{x}_t^U(1) - (\dot{y}_t^U(1)) - (\dot{z}_t^U(\tau)) = 55 \\ (45 - 10\tau)\dot{x}_t^U(1) - (\dot{y}_t^U(1)) - (\dot{z}_t^U(\tau)) = 75 \end{cases}$$

**Table 29**  
Approximate solution of Example 8 using SFQM1 and SFQN1.

$\tau$	$\dot{x}_I^L(\tau)$	$\dot{x}_I^U(\tau)$	$\dot{x}_I^L(\tau)$	$\dot{x}_I^U(\tau)$	$\dot{y}_I^L(\tau)$	$\dot{y}_I^U(\tau)$	$\dot{y}_I^L(\tau)$	$\dot{y}_I^U(\tau)$	$\dot{z}_I^L(\tau)$	$\dot{z}_I^U(\tau)$	$\dot{z}_I^L(\tau)$	$\dot{z}_I^U(\tau)$
0.0	7.7083	9.8822	12.684	66.96	5.3067	18.5378	39.010	189.32	7.9569	44.005	42.382	219.058
0.1	7.7724	11.974	12.071	75.35	6.9891	17.1371	37.503	183.23	7.9560	43.005	49.832	214.059
0.2	7.8421	13.859	11.531	83.04	6.2652	14.8826	36.000	177.15	7.9550	42.005	50.913	209.060
0.3	7.9180	15.499	11.032	89.96	7.0554	13.8024	34.502	171.08	7.9539	41.004	55.364	204.061
0.4	8.0012	16.909	10.566	96.16	8.4305	13.1722	33.010	165.03	7.9528	40.004	61.119	199.061
0.5	8.0929	18.157	10.142	101.83	9.8822	12.6844	31.528	158.99	7.9516	39.004	66.962	194.062
0.6	8.1946	19.335	9.7671	107.24	11.2963	12.26546	30.057	152.97	7.9504	38.004	72.623	189.063
0.7	8.3083	20.520	9.445	112.61	12.6291	11.8858	28.601	146.99	7.9491	37.004	78.001	184.064
0.8	8.4365	20.520	13.174	118.08	13.8598	18.5313	28.601	141.05	7.9478	47.004	78.001	179.065
0.9	8.5823	20.520	18.945	123.67	14.9802	20.1942	28.601	135.17	7.9464	67.004	78.001	174.066
1.0	8.7500	20.520	20.520	129.37	15.9925	28.6017	28.601	129.37	7.9449	78.001	78.001	169.067

**Table 30**  
Error comparison of the numerical schemes SFQM1 and SFQN1.

Scheme	SFQM1						SFQN1					
	C1			C2			C1			C2		
$\tau$	x	y	z	x	y	z	x	y	z	x	y	z
0.0	1.3e-45	1.1e-42	1.1e-42	1.1e-54	0.4e-46	9.4e-46	3.4e-17	1.1e-19	1.1e-18	8.9e-35	7.6e-28	3.0e-36
0.1	1.4e-45	0.8e-33	7.8e-43	2.2e-52	0.3e-66	3.0e-69	8.9e-18	1.9e-28	1.9e-28	1.9e-36	7.7e-27	4.2e-37
0.2	2.2e-43	2.4e-45	7.4e-45	2.7e-54	3.0e-55	4.7e-65	8.2e-19	8.1e-29	8.1e-19	2.8e-37	2.7e-28	0.1e-38
0.3	2.7e-45	1.3e-46	3.7e-46	0.9e-56	9.1e-57	2.1e-57	0.2e-28	4.4e-22	4.4e-20	3.4e-39	3.8e-29	2.2e-34
0.4	3.2e-42	3.4e-44	7.4e-44	0.5e-57	9.8e-58	0.8e-68	7.3e-29	3.9e-18	7.9e-25	8.2e-28	3.8e-38	2.4e-25
0.5	3.0e-44	3.7e-46	7.7e-34	0.7e-51	9.6e-45	2.0e-55	1.6e-22	7.7e-28	9.7e-15	7.0e-33	4.4e-33	3.7e-35
0.6	4.4e-45	0.3e-47	9.8e-34	4.0e-53	5.0e-45	5.6e-59	5.8e-22	9.3e-22	9.3e-29	4.5e-22	4.4e-34	4.4e-37
0.7	1.5e-46	1.3e-34	9.6e-48	5.2e-56	2.5e-67	7.6e-55	3.0e-32	7.2e-21	2.5e-20	1.3e-29	2.1e-38	4.0e-33
0.8	2.5e-47	3.3e-34	9.3e-49	5.0e-55	5.5e-46	2.3e-58	1.4e-32	6.8e-21	4.8e-25	1.2e-22	3.1e-39	3.6e-35
0.9	1.7e-43	3.6e-45	9.6e-45	1.6e-54	2.3e-65	2.2e-69	0.8e-24	4.8e-19	4.8e-28	4.1e-23	4.3e-38	2.4e-35
1.0	4.0e-40	1.4e-42	0.1e-42	2.6e-63	3.3e-54	0.19e-68	9.9e-28	5.6e-20	4.6e-27	0.7e-38	1.5e-37	1.7e-34

Consequently  $x_I^U(0) = x_I^L(0) = x_I^L(0) = x_I^U(0) = 0.35$  and  $x_I^U(0) = x_I^L(0) = x_I^L(0) = x_I^U(0) = 0.5$ . After 4 iterations, we obtain the solution with the maximum error less than  $10^{-30}$ . Suppose  $x, y, z$  are negative, we have

$$\begin{cases} \dot{x}_I^L(\tau) - 3(\dot{y}_I^L(\tau)) + (\dot{z}_I^L(\tau)) = 50 + 15\tau, \\ \dot{x}_I^L(\tau) - 3(\dot{y}_I^U(\tau)) + (\dot{z}_I^L(\tau)) = 80 - 15\tau, \\ \dot{x}_I^U(\tau) - 3(\dot{y}_I^U(\tau)) + (\dot{z}_I^U(\tau)) = 45 + 10\tau, \\ \dot{x}_I^U(\tau) - 3(\dot{y}_I^L(\tau)) + (\dot{z}_I^U(\tau)) = 95 - 20\tau, \\ \sin(\dot{x}_I^L(\tau)) + (\dot{y}_I^L(\tau)) = 10 + 15\tau, \\ \sin(\dot{x}_I^L(\tau)) + (\dot{y}_I^U(\tau)) = 40 - 15\tau, \\ \sin(\dot{x}_I^U(\tau)) + (\dot{y}_I^U(\tau)) = 5 + 10\tau, \\ \sin(\dot{x}_I^U(\tau)) + (\dot{y}_I^L(\tau)) = 45 - 10\tau, \\ (10 + 15\tau)\dot{x}_I^L(\tau) - (\dot{y}_I^L(\tau)) - (\dot{z}_I^L(\tau)) = 50 + 15\tau, \\ (40 - 15\tau)\dot{x}_I^L(\tau) - (\dot{y}_I^U(\tau)) - (\dot{z}_I^L(\tau)) = 80 - 15\tau, \\ (5 + 10\tau)\dot{x}_I^U(\tau) - (\dot{y}_I^U(\tau)) - (\dot{z}_I^U(\tau)) = 45 + 10\tau, \\ (45 - 10\tau)\dot{x}_I^U(\tau) - (\dot{y}_I^L(\tau)) - (\dot{z}_I^U(\tau)) = 95 - 20\tau. \end{cases}$$

For  $\tau = 0$ , we have,  $y_I^L(0) > y_I^U(0)$ , therefore negative solutions of IVTFNSEs does not exist.

Fig. 10, depicts the numerical solution of IVTFNes, as well as the initial starting values and solution of the IVTF variables  $x, y$  and  $z$  utilized in Example 8. This figure also shows numerical approximations obtained with SFQM1 that are matched with analytical solutions up to 30 decimal places. Table 29 shows the numerical approximate solution of the interval-valued trapezoidal system of nonlinear equations used in Example 8 by utilizing numerical schemes SFQM1 and SFQN1 up to 30 decimal places with 64-digit floating point arithmetic. It display the approximated solution to four decimal places. The numerical approximate solution in Table 29 agrees with the analytical solution shown in Fig. 10.

In Table 30, C1 and C2 represents the stopping criteria 1 and 2 i.e.,  $\|F(x, \tau)\|, \|x_{n+1}(\tau) - x_n(\tau)\|$  respectively for different values of  $\tau$  respectively. The approximate IVTFNes solutions in Table 30 clearly show that our newly developed family is more efficient than the classical method used in the literature.

In Table 31, M-Error<sub>C1</sub>, M-Error<sub>C2</sub> show the error of numerical schemes SFQM1 and SFQN1 using stopping criteria I and II, respectively, M-CPU-Time<sub>C1</sub>, M-CPU-Time<sub>C2</sub> present the computational time in seconds of the numerical schemes for solving the solution of a fuzzy nonlinear system used in Example 8, employing stopping criteria I and II, and COC shows the computational order of convergence of the SFQM1 and SFQN1 methods. Tables 29–31, demonstrates that our newly developed method, SFQM1, is better

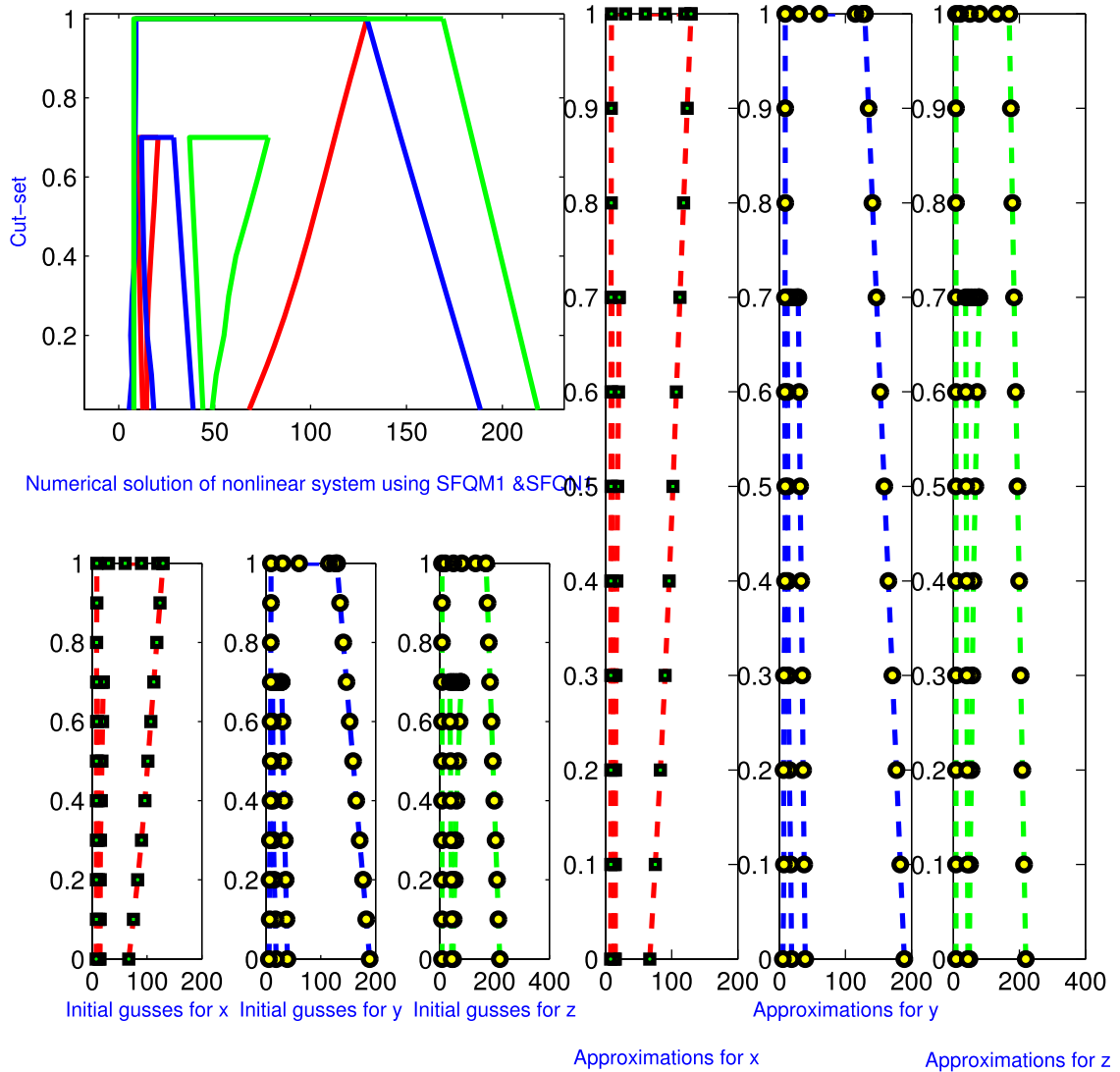


Fig. 10. Approximate solution of IVTFNEs used in Example 8.

**Table 31**  
Stability-consistency analysis.

Method	SFQM1	SFQN1
M-Error <sub>C1</sub>	0.6e-30	0.5e-15
M-Error <sub>C2</sub>	7.2e-40	1.0e-28
M-CPU-Time <sub>C1</sub>	0.02421	0.11051
M-CPU-Time <sub>C2</sub>	0.01313	0.32272
COC	3.0021	2.00131

than SFQN1 in terms of residual error, CPU time, and computational order of convergence. Table 31 also shows that the SFQM1 method is more stable and consistent than the SFQN1 method.

Table 32, demonstrates that in terms of percentage convergence, total functions, and Jacob’s evaluations, our technique SFQM1 outperforms the existing method SFQN1 and provides an alternative way to solve complex IVTFNE for greater comprehension and analysis.

**Table 32**  
Percentage convergence and total function evaluations for SFQM1, SFQN1.

Method	Number- F-J	Percentage-Con	M-CPU-Time <sub>C3</sub>	M-Error <sub>C3</sub>	COC
SFQM1	7	77.900613	2.17764511	9.985e-19	2.00776
SFQN1	13	35.761132	3.21344355	7.876e-17	1.76546

#### 4.2. Advantages of the proposed numerical scheme SFQM1

We discuss how the computing efficiency, stability, and residual error robustness of the proposed numerical scheme SFQM1 outperforms those of the classical existing technique.

- Tables 1 to 4 demonstrate that the SFQM1 is more reliable than the SFQN1 method. While solving interval-valued trapezoidal fuzzy nonlinear system of equations, SFQM1 converges more quickly and accurately than SFQN1.
- The approximate solutions for various parameter values are presented in Tables 1, 5, 9, 13, 17, 21, 25 and 29, all of which agree completely with the analytical solutions.
- In terms of iterations, residual error, and CPU time, Tables 2, 6, 10, 14, 18, 22, 26 and 30 clearly show that SFQM1 outperforms SFQN1.
- Tables 3, 7, 11, 15, 19, 23, 27 and 31 show that SFQM1 is more stable and consistent than SFQN1 in terms of maximum and overall error.
- The maximum error utilizing C3 of SFQM1 is significantly superior than SFQN1 in terms of percentage convergence, total number of functions, and its Jacobine evaluations, as indicated in Table 4, 8, 12, 16, 20, 24, 28 and 22 respectively.
- Figs. 3-10 compare the numerical simulation of our recently developed fuzzy numerical scheme SFQM1 to an analytical solution of the IVTFNSEs used in Example 1–8 respectively.
- The numerical results presented in Tables 1–4 and Figs. 3-10 clearly demonstrate that the exact and approximate solutions are matched up to 30 decimal places with SFQM1 and 7 decimal places with SFQN1 using stopping criteria C1, C2 and C3, respectively. Compared to classical methods in the literature, our method is clearly superior, as shown by the numerical simulation of our methods.

Here are some remarks on the advantages of the proposed numerical schemes:

- 1) The stability and consistency of SFQM1 relative to SFQN1 are demonstrated in Tables 3, 7, 11, 15, 19, 23, 27 and 31, with an emphasis on maximum and total error. This presents an in-depth analysis of the proposed methodology.
- 2) Based on the numerical scheme proposed, the comparison between SFQM1 and SFQN1 is effectively communicated in terms of computing efficiency, stability, residual error and robustness.
- 3) The proposed numerical technique compares the two methods using several performance parameters such as iterations, residual error, and CPU time. The entire summary in Tables 2, 6, 10, 14, 18, 22, 26 and 30 provides an in-depth review of their relative performance.
- 4) The numerical findings of the proposed numerical scheme show the precision of SFQM1 by claiming that precise and approximate solutions match 30 decimal places, whereas SFQN1 matches 7 decimal places. A standard for the numerical stability of SFQM1 is established by its outstanding precision.

#### 4.3. Research work limitations

Despite its benefits, the proposed approach has limits that must be investigated further. The approach is specifically made for trapezoidal fuzzy systems, and handling other kinds of fuzzy numbers—such as triangular or generalized fuzzy numbers—which are frequently encountered in various real-world applications, may require considerable adjustment. However, the method's performance has not yet been tested for dynamic or highly complex systems where uncertainty is either large or fluctuates over time. This may limit the method's applicability in these instances. Thirdly, even if this method is efficient in the situations examined, the computing cost may go up for larger variables or system limits. This may limit scalability for large problems. Addressing these constraints will improve the method's versatility, robustness, and applicability in future research.

## 5. Conclusion

In this paper, a new numerical iterative scheme for computing the approximate solution of the interval-valued trapezoidal fuzzy system of nonlinear equations. The practical performance and dominance efficiency of the SFQM1 technique over the SFQN1 method on the same number of iterations were demonstrated by numerical test examples. It has been tested on a variety of cases, yielding results in terms of residual error, CPU time, computational order of convergence, percentage convergence, and divergence. The results clearly show that our method outperforms other methods in terms of accuracy, convergence speed, and computing efficiency. Its key advantages include reduced CPU time, large reductions in residual errors, and good convergence features, as demonstrated by the computational order of convergence and percentage convergence results. The numerical results of the SFQM1 approach outperform those of the SFQN1 approach in terms of absolute error and CPU time, as shown in Tables 1-32 and Figs. 1-10. In terms of maximum

and overall error, the SFQM1 numerical scheme is superior to the classical method for solving IVTFNSEs, and stability and consistency analyses validate this. Furthermore, we have demonstrated the superiority of our method by comparing the approximation results with an analytical solution. The method's effective application to realistic cases demonstrates its applicability in real-world scenarios.

There are several limitations, however, such as its current emphasis on trapezoidal fuzzy systems, which may need to be adapted for other types of fuzzy numbers, and its untested functionality in particularly dynamic or complex systems. Higher processing costs may also be the consequence of issues with a much greater number of variables or restrictions.

Further research can address these possible limits by improving the accuracy and efficiency of this approach towards larger order convergence, generalizing it to address new kinds of fuzzy systems, or adapting it to dynamic and time-dependent situations. To improve its applicability, adaptive iterative approaches that reduce computational effort and uncertainty quantification frameworks might be applied. Comprehensive comparison studies using state-of-the-art techniques on a range of problem domains with research for parallel or distributed computing implementations offer an additional avenue to valuable information, in addition to the possibility of larger, more complex systems.

Consequently, future studies will center on applying SFQM1 in a more generalized fuzzy setting to solve higher-dimensional interval-valued trapezoidal fuzzy systems of nonlinear equations and related boundary value problems. We also create higher-order numerical techniques for solving IVTFNSE problems that may be used in a wide range of scientific and engineering fields.

## Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## Data availability

No data was used for the research described in the article.

## Declaration of competing interest

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

## References

- [1] I.M. Batiha, M.W. Alomari, N. Anakira, S. Meqdad, I.H. Jebri, S. Momani, Numerical advancements: a duel between Euler-Maclaurin and Runge-Kutta for initial value problem, *Int. J. Neutrosophic Sci.* 25 (3) (2025) 76–91.
- [2] M.W. Alomari, I.M. Batiha, N. Anakira, A. Amourah, I.H. Jebri, S. Momani, Revolutionizing numerical approximations: a novel higher-order implicit method vs. Runge-Kutta for initial value problems, *J. Rob. Contr.* 6 (2) (2025) 624–637.
- [3] M. Abualhomos, Numerical solution of complex fuzzy differential equations by Euler and Taylor methods, *Int. J. Membr. Sci. Technol.* 10 (3) (2023) 340–353. <https://doi.org/10.15379/ijmst.v10i3.1539>
- [4] A. Bouchenak, I.M. Batiha, M. Aljazzazi, N. Anakira, M. Odeh, R.I. Hajaj, Numerical and analytical investigations of fractional self-adjoint equations and fractional Sturm-Liouville problems via modified conformable operator, *J. Rob. Contr.* 6 (3) (2025) 1410–1424.
- [5] Z.T. Yassin, W. Al-Hayani, A.F. Jameel, A. Amourah, N. Anakira, Solving fuzzy system of Fredholm integro-differential equations of the second kind by using homotopy analysis method, *AIMS Math.* 10 (1) (2025) 1704–1740.
- [6] M. Tariq, M. Suleman, A.S. Khan, P. Kuma, A.A. Shaikh, Delay differential-algebraic equations (DDAEs), *Babylonian J. Math.* 2023, pp. 40–44. <https://doi.org/10.58496/BJM/2023/008>
- [7] L.A. Zadeh, Fuzzy sets, *Inf. Contr.* 8 (3) (1965) 338–353.
- [8] L.A. Zadeh, Toward extended fuzzy logic a first step, *Fuzzy Sets Syst.* 160 (21) (2009) 3175–3181.
- [9] S.S. Chang, L.A. Zadeh, On fuzzy mapping and control, *IEEE Trans. Syst. Man Cybern.* 1, 1972, pp. 30–34.
- [10] D. Dubois, H. Prade, Operations on fuzzy numbers, *Int. J. Syst. Sci.* 9 (6) (1978) 613–626.
- [11] M. Mizumoto, K. Tanaka, Some properties of fuzzy sets of type 2, *Inf. Contr.* 31 (4) (1976) 312–340.
- [12] K.T. Atanassov, Intuitionistic fuzzy sets, *Fuzzy Sets Syst.* 20 (1) (1986) 87–96.
- [13] K.T. Atanassov, G. Gargov, Interval valued intuitionistic fuzzy sets, *Fuzzy Sets Syst.* 31 (3) (1989) 343–349.
- [14] H. Bustince, P. Burillo, Correlation of interval-valued intuitionistic fuzzy sets, *Fuzzy Sets Syst.* 74 (2) (1995) 237–244.
- [15] Q.S. Zhang, S. Jiang, B. Jia, S. Luo, Some information measures for interval-valued intuitionistic fuzzy sets, *Inf. Sci.* 180 (24) (2010) 5130–5145.
- [16] Y. Liu, W. Jiang, A new distance measure of interval-valued intuitionistic fuzzy sets and its application in decision making, *Soft Comput.*, 24, 2020. 6987–7003.
- [17] T.K. Mondal, S.K. Samanta, Topology of interval-valued intuitionistic fuzzy sets, *Fuzzy Sets Syst.*, 119, 2001. 483–494.
- [18] H.M. Talib, A.S. Albahri, T.O.C. Edoh, Fuzzy decision-making framework for sensitively prioritizing autism patients with moderate emergency level, *Appl. Data Sci. Anal.* 2023, pp. 16–41. <https://doi.org/10.58496/ADSA/2023/002>
- [19] I. Bloch, Mathematical morphology on bipolar fuzzy sets: general algebraic framework, *Int. J. Approximate Reasoning* 53 (7) (2012) 1031–1060.
- [20] N. Deva, A. Felix, Bipolar intuitionistic fuzzy matrices and its determinant, *TWMS J. Appl. Eng. Math.* 14 (3) (2024) 1085–1098.
- [21] C. Jain, et al. Fully fuzzy distribution planning problem by using interval-valued bipolar trapezoidal fuzzy number, *HyperSoft Set Methods Eng.* 1 (2024) 95–108.
- [22] A. Farooq, M. Nabeel, G. Ali, New MCDM applications using cubic bipolar fuzzy model in medicine and engineering, *Soft Comput.* 2023, pp. 1–26.
- [23] A. Güldürdek, G.Y. Tütüncü, Arithmetic operations on generalized pentagonal fuzzy numbers, *Int. J. Neutrosophic Sci.* 24 (2) (2024).
- [24] K. Kalaiarasi, S. Swathi, Arithmetic operations of trigonal fuzzy numbers using alpha cuts: a flexible approach to handling uncertainty, *Indian J. Sci. Technol.* 16 (47) (2023) 4585–4593.
- [25] A. Chakraborty, S.P. Mondal, S. Alam, A. Ahmadian, N. Senu, D. De, S. Salahshour, The pentagonal fuzzy number: its different representations, properties, ranking, defuzzification and application in game problems, *Symmetry* 11 (2) (2019) 248.
- [26] R.M. Zulqarnain, M. Saeed, B. Ali, N. Ahmad, L. Ali, S. Abdal, Application of interval valued fuzzy soft max-min decision making method, *Int. J. Math. Res.* 9 (1) (2020) 11–19.
- [27] P. Dutta, G. Borah, S. Borkotokey, Neo arithmetic and ranking techniques for trapezoidal generalized interval valued fuzzy numbers: their applications in decision making for medical investigation, *Neural Process. Lett.* 54 (4) (2022) 3045–3096.
- [28] N. Jan, J. Gwak, S. Hussain, A. Nasir, Mathematical investigation of communication and network securities under interval-valued complex spherical fuzzy information, *Int. J. Fuzzy Syst.* 26 (1) (2024) 87–104.
- [29] J.J. Buckley, Y. Qu, Solving linear and quadratic fuzzy equations, *Fuzzy Sets Syst.* 38 (1) (1990) 43–59.

- [30] J.J. Buckley, Y. Qu, Solving fuzzy equations: a new solution concept, *Fuzzy Sets Syst.* 39 (3) (1991) 291–301.
- [31] S. Abbasbandy, T. Hajjari, A new approach for ranking of trapezoidal fuzzy numbers, *Comput. Math. Appl.* 57 (3) (2009) 413–419.
- [32] M. Keyanpour, T. Akbarian, Solving intuitionistic fuzzy nonlinear equations, *J. Fuzzy Set Valued Anal.* 2014 2014, pp. Article ID jfsva-00142, 6 Pages.
- [33] I.M. Sulaiman, M. Mamat, M.Y. Waziri, M.A. Mohamed, F.S. Mohamad, Solving fuzzy nonlinear equation via Levenberg-Marquardt method, *Far East J. Math. Sci.* 103 (10) (2018) 1547–1558.
- [34] A.U. Omesa, I.M. Sulaiman, M. Mamat, M.Y. Waziri, A. Shadi, M.A. Zaini, I. Sumiati, Derivative free Levenberg-Marquardt method for solving fuzzy nonlinear equation, *IOP Conf. Ser. Mater. Sci. Eng.* 1115 (2021) 1–7.
- [35] Y.J. Cho, N.J. Huang, S.M. Kang, Nonlinear equations for fuzzy mappings in probabilistic normed spaces, *Fuzzy Sets Syst.* 110 (1) (2000) 115–122.
- [36] M. Mosleh, Solution of dual fuzzy polynomial equations by modified Adomian decomposition method, *Fuzzy Inf. Eng.* 5 (1) (2013) 45–56.
- [37] T. Allahviranloo, S. Asari, Numerical solution of fuzzy polynomial by Newton-Raphson method, *Islamic Azad Univ. Lahijan* 4 (2011) 17–44.
- [38] R. Jafari, W. Yu, Uncertain nonlinear system control with fuzzy differential equations and Z-numbers, in: 2017 IEEE International Conference on Industrial Technology (ICIT), 2017, pp. 890–895.
- [39] J. Ma, G. Feng, An approach to H control of fuzzy dynamic systems, *Fuzzy Sets Syst.* 137 (3) (2003) 367–386.
- [40] J. Bhattacharya, Some results on certain properties of intuitionistic fuzzy sets, *J. Fuzzy Ext. Appl.* 2 (4) (2021) 377–387.
- [41] T.Y. Chen, Signed distanced-based TOPSIS method for multiple criteria decision analysis based on generalized interval-valued fuzzy numbers, *Int. J. Inf. Technol. Decis. Mak.* 10 (06) (2011) 1131–1159.
- [42] A. Elmoasry, M. Shams, N. Yaqoob, N. Kausar, Y.U. Gaba, N. Rafiq, Numerical Scheme for Finding Roots of Interval-Valued Fuzzy Nonlinear Equation with Application in Optimization, *J. Funct. Spaces* 2021 (2021) 6369129.
- [43] K. Hussain, A. Amourah, J. Salah, A.F. Jameel, N. Anakira, Autonomous block method for uncertainty analysis in first-order real-world models using fuzzy initial value problem, *AIMS Math.* 10 (2025) 9614–9636.
- [44] S. Abbasbandy, B. Asady, Newton's method for solving fuzzy nonlinear equations, *Appl. Math. Comput.* 159 (2) (2004) 349–356.
- [45] J. Shokri, Numerical method for solving fuzzy nonlinear equations, *Appl. Math. Sci.* 2 (24) (2008) 1191–1203.