Journal of Applied Mathematics and Computational Mechanics 2025, 24(1), 18-29
www.amcm.pcz.pl p-ISSN 2299-9965
DOI: 10.17512/jamcm.2025.1.02 e-ISSN 2353-0588

AN EFFICIENT HIGHER ORDER NUMERICAL ALGORITHM FOR SOLVING REAL LIFE APPLICATION PROBLEMS IN AN UNCERTAIN ENVIRONMENT USING TRIANGULAR FUZZY NUMBER

Srilakshmi Katuri, Prashanth Maroju

Department of Mathematics, School of Advanced Sciences, VIT-AP University Amaravati
Andhra Pradesh, India
katurisrilakshmi09@gmail.com, maroju.prashanth@gmail.com

Received: 4 December 2024; Accepted: 12 March 2025

Abstract. Many researchers have proposed numerical approaches for solving fuzzy nonlinear equations (FNE). Most of the methods used are based on the Newton algorithm. The main difficulty in solving these FNEs is obtaining and inverting the Hessian matrix. In this article, we propose the multi step second derivative free iterative method for solving FNE. The main advantage of our method is that it avoids calculating and inverting the Hessian matrix in each iteration, leading to a significant decrease in computational expenses. We solve some numerical examples and application problems related to the fraction of conversion of gases in an uncertain environment with graphical representations. We compare our results with the existing eighth order iterative method to show the efficiency of our proposed method.

MSC 2010: *94D05, 65H04, 65H05, 65H20*
Keywords: *fuzzy nonlinear equations, Jacobian matrix, derivative free method, higher order Iterative method, fuzzy number*

1. Introduction

Let us consider the nonlinear equations in the form of

$$f(x) = 0. \tag{1}$$

Equation (1) describes issues in geophysics, engineering, optimization, astronomy, natural science, medicine, and computer science. The presence of unknown parameters complicates interpreting these modeled equations in real-world circumstances. Fuzzy equations are typically considered the most effective mathematical models for resolving uncertainty-related situations [1, 2]. The most difficult aspect in addressing this problem is when the coefficients are presented in fuzzy numbers rather than crisp ones. As a result, the solutions rely on the fuzzy equations roots [3,4]. The fuzzy nonlinear equations are of the form

$$\begin{aligned} &\tilde{a}_1\tilde{x}^3+\tilde{b}_1\tilde{x}^2+\tilde{c}_1\tilde{x}-\tilde{d}_1=\tilde{e}_1,\\ &\tilde{a}_1e^{\tilde{x}}+\tilde{b}_1=\tilde{d}_1. \end{aligned} \tag{2}$$

where $\tilde{a}_1$, $\tilde{b}_1$, $\tilde{c}_1$, $\tilde{d}_1$ and $\tilde{e}_1$ are fuzzy numbers. The conventional analytical methods are inefficient for solving FNE due to the imprecise nature of the coefficients. Many researchers developed numerical approaches for solving these FNE. The Newton algorithm, developed in [5,6], is a well-known and desirable foundational algorithm

$$\tilde{x}_{n+1}=\tilde{x}_n-\frac{\tilde{f}(\tilde{x}_n)}{\tilde{f}'(\tilde{x}_n)}. \tag{3}$$

It has a quadratic convergence rate and shows rapid convergence when the starting approximation approaches the solution. However, its main disadvantage is the need to calculate the Jacobian matrix at each iteration. To solve FNE, numerous scholars have devised iterative approaches such as the Steepest Descent method and Broyden's method, as mentioned in [7,8]. The Modified Newton approach for solving FNE, provided in [9], similarly tackles the difficulty of finding the Jacobian matrix at every iteration.

Several third-order iterative methods for solving nonlinear equations have been developed, including Chebyshev's and Halley's methods, as explained in [10]. Researchers have created fourth-order iterative algorithms, as detailed in [11–13]. Fifth-order iterative approaches for nonlinear equations have been introduced in [14–16]. In addition, [17–20] present sixth-order iterative approaches for solving nonlinear equations.

The primary goal of our research is to provide an iterative method for solving fuzzy nonlinear equations. Our eighth-order method improves on Sharma et al.'s fourth-order method for solving nonlinear equations [21]. This novel approach has been used in our research to address FNE. Our method's speedy convergence to the solution removes the computational burden of computing the Jacobian matrix. Our methodology outperforms Siva Kumar et al.'s eighth-order iterative method [22] in terms of efficiency and reliability while solving FNE.

In this article, Section 2 includes fundamental explanations of fuzzy concepts. In Section 3, we present an iterative approach for solving FNE. Section 4 presents the convergence study of our suggested method. Section 5 and 6 provide numerical examples and application problems along with graphical depiction. The paper concludes with a brief summary in Section 7.

2. Mathematical preliminaries

For all sets of X. The membership function $\eta_{\tilde{v}}$ is a representation of the mapping that occurs from the set X to the unit interval [0,1] for the function $\tilde{v}$.

Definition 1. The following conditions must be met for a fuzzy set to be considered as a fuzzy number [23]:

1. $\eta_{\tilde{v}}$ is normal i.e., there exist s_0 such that $\eta_{\tilde{v}}(s_0) = 0$.
2. $\eta_{\tilde{v}}$ is convex i.e., $\eta_{\tilde{v}}(\lambda s_1 + (1-\lambda)s_2) \geq min\{\eta_{\tilde{v}}(s_1), \eta_{\tilde{v}}(s_2)\}, \forall s_1, s_2 \in \mathbb{R}, \forall \lambda \in [0,1]$.
3. $\eta_{\tilde{v}}$ is upper semi continuous i.e., for all $s \in [0,1]$, the subset $\tilde{v}(s) = \{x \in \mathbb{R} : \eta_{\tilde{v}}(x) \geq s\}$ is closed.
4. $\tilde{v}_0$ is compact at 0-level.

Definition 2. One of the most common types of fuzzy numbers is the triangular fuzzy number, which is denoted by the triplet $\tilde{v} = (a_1, b_1, c_1)$. Where $a_1 \leqslant b_1 \leqslant c_1$ and its membership function are defined as follows [23],

$$\eta_{\tilde{v}}(x_0) = \begin{cases} \dfrac{x_0 - a_1}{b_1 - a_1} & a_1 \leqslant x_0 \leqslant b_1 \\ 1 & x_0 = b_1 \\ \dfrac{c_1 - x_0}{c_1 - b_1} & b_1 \leqslant x_0 \leqslant c_1 \\ 0 & Otherwise \end{cases} \tag{4}$$

Moreover, β-cut value of $\tilde{v}$ for $0 \leqslant \beta \leqslant 1$ can be defined as

$$\tilde{v}(r) = [\underline{v}(r), \bar{v}(r)] = [a_1 + (b_1 - a_1)\beta, c_1 + (b_1 - c_1)\beta]. \tag{5}$$

3. Development of proposed method

In the fuzzy environment, the nonlinear equation (1) is modified to incorporate the uncertainty, so the fuzzy nonlinear equation is of the form

$$\tilde{f}(\tilde{x_L}, \tilde{x_C}, \tilde{x_U}) = (\tilde{c_L}, \tilde{c_C}, \tilde{c_U}), \tag{6}$$

simply (6) can be represented as

$$\tilde{f}(\tilde{x}) = \tilde{c}, \tag{7}$$

to simplify (7), assume $\tilde{\beta} = (\tilde{\beta_L}, \tilde{\beta_C}, \tilde{\beta_U})$ is the fuzzy root and $(\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}})$ is the initial guess which is close to $\tilde{\beta}$. We must modify the traditional Taylor series expansion to take fuzzy arithmetic and fuzzy derivatives as factors in order to elucidate the series within the framework of fuzzy parameters. Using Taylor series expansion of the function $\tilde{f}(\tilde{x})$, we get

$$\tilde{f}\left(\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}}\right) + \left(\tilde{x} - (\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}})\right)\tilde{f}'(\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}}) + \frac{\left(\tilde{x} - (\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}})\right)^2}{2!}\tilde{f}''(\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}}) + \cdots = \tilde{c} \tag{8}$$

from (8),

$$\tilde{x} = (\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}}) - \frac{\tilde{f}\left((\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}})\right)}{\tilde{f}'((\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}}))} - \frac{\left(\tilde{x} - (\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}})\right)^2 \tilde{f}''((\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}}))}{2\tilde{f}'((\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}}))} + \tilde{c} \tag{9}$$

Each higher-order term, such as $\left(\tilde{x} - (\underline{\tilde{\alpha}}, \tilde{\bar{\alpha}})\right)^n$, is computed using fuzzy arithmetic operations, and the coefficients are unaffected by the fuzziness. Many researchers developed higher order iterative approaches for solving nonlinear equations. In 2015, Sharma & Bahl [21] proposed the two step Newton type fourth order iterative method which is expressed as

$$\begin{aligned} \tilde{y}_n &= \tilde{x}_n - \theta\frac{\tilde{f}(\tilde{x}_n)}{\tilde{f}'(\tilde{x}_n)}, \\ \tilde{x}_{n+1} &= \tilde{x}_n - \left(A + B\frac{\tilde{f}'(\tilde{x}_n)}{\tilde{f}'(\tilde{y}_n)} + C\frac{\tilde{f}'(\tilde{y}_n)}{\tilde{f}'(\tilde{x}_n)}\right) \times \frac{\tilde{f}(\tilde{x}_n)}{\tilde{f}'(\tilde{x}_n)}. \end{aligned} \tag{10}$$

Where θ, A, B, C all are constants and the values are $\theta = \frac{2}{3}$, $A = -\frac{1}{2}$, $B = \frac{9}{8}$, $C = \frac{3}{8}$.

In this section, we introduced our eighth order iterative method using the fourth order method. We generalized (10) by adding the next step, and we get

$$\begin{aligned} \tilde{y}_n &= \tilde{x}_n - \theta\frac{\tilde{f}(\tilde{x}_n)}{\tilde{f}'(\tilde{x}_n)}, \\ \tilde{z}_n &= \tilde{x}_n - \left(A + B\frac{\tilde{f}'(\tilde{x}_n)}{\tilde{f}'(\tilde{y}_n)} + C\frac{\tilde{f}'(\tilde{y}_n)}{\tilde{f}'(\tilde{x}_n)}\right) \times \frac{\tilde{f}(\tilde{x}_n)}{\tilde{f}'(\tilde{x}_n)}, \\ \tilde{x}_{n+1} &= \tilde{z}_n - \frac{\tilde{f}(\tilde{z}_n)}{\tilde{f}'(\tilde{z}_n)}. \end{aligned} \tag{11}$$

Equation (11) summarizes our proposed method, which has an order of convergence of eight. We utilize this higher-order iterative method (11) to solve fuzzy nonlinear equations and tackle real-world application challenges in fuzzy systems.

4. Convergence analysis

Theorem: Let $\tilde{f}:\tilde{\Gamma}\subset\tilde{\mathbb{X}}\to[0,1]$ be a fuzzy mapping for the membership function $\tilde{\Gamma}$, where the nonlinear fuzzy equation $\tilde{f}(\tilde{x})=\tilde{c}$ has a fuzzy root $\tilde{\beta}\in\tilde{\Gamma}$. Assume that $\tilde{f}(\tilde{x})$ is sufficiently smooth in the neighborhood of $\tilde{\beta}$. Then, the proposed fuzzy method has a convergence order of eight if $\theta=\frac{2}{3}, A=-\frac{1}{2}, B=\frac{9}{8}, C=\frac{3}{8}$ for both lower and upper bound conditions. The fuzzy error equation is as follows,
$\tilde{e}_{n+1}=\frac{1}{81}\tilde{c}_2\left(21\tilde{c}_2^3-9\tilde{c}_2\tilde{c}_3+\tilde{c}_4\right)^2\tilde{e}_n^8+O\left[\tilde{e}_n\right]^9.$
Proof: Let $\tilde{\beta}$ be a root of the fuzzy equation $\tilde{f}(\tilde{x})=\tilde{c}$. Then the error in n^{th} iteration, $\tilde{e_n}=\tilde{x}_n-\tilde{\beta}$ by using Taylor series expansion, we get

$$\tilde{f}(\tilde{x}_n)=\tilde{f}'(\tilde{\beta})\left(\tilde{e}_n+\tilde{c}_2\tilde{e}_n^2+\tilde{c}_3\tilde{e}_n^3+\tilde{c}_4\tilde{e}_n^4+\tilde{c}_5\tilde{e}_n^5+\cdots+\tilde{c}_{11}\tilde{e}_n^{11}+O(\tilde{e}_n^{12})\right) \tag{12}$$

Here $\tilde{f}(\tilde{x}_n)$ represents the fuzzy function applied to the fuzzy variable $\tilde{x}_n$, $\tilde{f}'(\tilde{\beta})$ represents the fuzzy derivative of the function at the fuzzy root $\tilde{\beta}$ and $\tilde{c}_2,\tilde{c}_3,\tilde{c}_4,\cdots$ are fuzzy coefficients for each power of the error $\tilde{e}_n$.

Where,

$$\tilde{c}_k=\frac{1}{k!}\frac{\tilde{f}^{(k)}(\tilde{\beta})}{\tilde{f}'(\tilde{\beta})},\quad k=2,3,4,\ldots, \tag{13}$$

By using Taylor's series expand $\tilde{f}'(\tilde{x}_n)$, we get,

$$\tilde{f}'\left(\tilde{x}_n\right)=\tilde{f}'(\tilde{\beta})\left(1+2\tilde{c}_2\tilde{e}_n+3\tilde{c}_3\tilde{e}_n^2+4\tilde{c}_4\tilde{e}_n^3+5\tilde{c}_5\tilde{e}_n^4+\cdots+11\tilde{c}_{11}\right)\tilde{e}_n^{10}+O\left(\tilde{e}_n^{11}\right) \tag{14}$$

Furthermore, we have

$$\begin{aligned}\tilde{y}_n=&\frac{\tilde{e}_n}{3}+\frac{2}{3}\tilde{c}_2\tilde{e}_n^2-\frac{4}{3}\left(\tilde{c}_2^2-\tilde{c}_3\right)\tilde{e}_n^3+\frac{2}{3}\left(4\tilde{c}_2^3-7\tilde{c}_2\tilde{c}_3+3\tilde{c}_4\right)\tilde{e}_n^4-\frac{4}{3}\left(4\tilde{c}_2^4-10\tilde{c}_2^2\tilde{c}_3\right.\\&\left.+3\tilde{c}_3^2 5\tilde{c}_2\tilde{c}_4-2\tilde{c}_5\right)\tilde{e}_n^5+\frac{2}{3}\left(16\tilde{c}_2^5-52\tilde{c}_2^3\tilde{c}_3+28\tilde{c}_2^2\tilde{c}_4-17\tilde{c}_3\tilde{c}_4+\tilde{c}_2\left(33\tilde{c}_3^2-13\tilde{c}_5\right)\right.\\&\left.+5\tilde{c}_6\right)\tilde{e}_n^6-\frac{4}{3}\left(16\tilde{c}_2^6-64\tilde{c}_2^4\tilde{c}_3-9\tilde{c}_3^3+36\tilde{c}_2^3\tilde{c}_4+6\tilde{c}_4^2+9\tilde{c}_2^2\left(7\tilde{c}_3^2-2\tilde{c}_5\right)+11\tilde{c}_3\tilde{c}_5\right.\\&\left..+\tilde{c}_2\left(-46\tilde{c}_3\tilde{c}_4+8\tilde{c}_6\right)-3\tilde{c}_7\right)\tilde{e}_n^7+\frac{2}{3}\left(64\tilde{c}_2^7-304\tilde{c}_2^5\tilde{c}_3+176\tilde{c}_2^4\tilde{c}_4+75\tilde{c}_3^2\tilde{c}_4\right.\\&+\tilde{c}_2^3\left(408\tilde{c}_3^2-92\tilde{c}_5\right)-31\tilde{c}_4\tilde{c}_5-27\tilde{c}_3\tilde{c}_6+\tilde{c}_2^2\left(-348\tilde{c}_3\tilde{c}_4+44\tilde{c}_6\right)\\&\left.+\tilde{c}_2\left(-135\tilde{c}_3^3+64\tilde{c}_4^2+118\tilde{c}_3\tilde{c}_5-19\tilde{c}_7\right)\right)\tilde{e}_n^8+O\left[\tilde{e}_n\right]^9.\end{aligned}$$

Let us assume $\tilde{\xi}=\left(\tilde{A}+\tilde{B}\frac{\tilde{f}'\left(\tilde{x}_n\right)}{\tilde{f}'\left(\tilde{y}_n\right)}+\tilde{C}\frac{\tilde{f}'\left(\tilde{y}_n\right)}{\tilde{f}'\left(\tilde{x}_n\right)}\right)\times\frac{\tilde{f}\left(\tilde{x}_n\right)}{\tilde{f}'\left(\tilde{x}_n\right)}.$

By putting the values $A=-\frac{1}{2}, B=\frac{9}{8}, C=\frac{3}{8}$, we are deriving the following,

$$\begin{aligned}\tilde{\xi} &= \tilde{e}_n+\frac{1}{9}\left(-21\tilde{c}_2^3+9\tilde{c}_2\tilde{c}_3-\tilde{c}_4\right)\tilde{e}_n^4+\frac{2}{27}\left(174\tilde{c}_2^4-216\tilde{c}_2^2\tilde{c}_3+27\tilde{c}_3^2+30\tilde{c}_2\tilde{c}_4\right.\\ &\quad\left.-4\tilde{c}_5\right)\tilde{e}_n^5-\frac{2}{27}\left(623\tilde{c}_2^5-1257\tilde{c}_2^3\tilde{c}_3+321\tilde{c}_2^2\tilde{c}_4-99\tilde{c}_3\tilde{c}_4+9\tilde{c}_2\left(51\tilde{c}_3^2-5\tilde{c}_5\right)\right.\\ &\quad\left.+7\tilde{c}_6\right)\tilde{e}_n^6+\frac{4}{243}\left(8223\tilde{c}_2^6-22824\tilde{c}_2^4\tilde{c}_3-1377\tilde{c}_3^3+7734\tilde{c}_2^3\tilde{c}_4+393\tilde{c}_4^2\right.\\ &\quad\left.+9\tilde{c}_2^2\left(1656\tilde{c}_3^2-205\tilde{c}_5\right)+630\tilde{c}_3\tilde{c}_5-9\tilde{c}_2\left(669\tilde{c}_3\tilde{c}_4-29\tilde{c}_6\right)-46\tilde{c}_7\right)\tilde{e}_n^7\\ &\quad+\frac{1}{729}\left(-258960\tilde{c}_2^7+909738\tilde{c}_2^5\tilde{c}_3-364530\tilde{c}_2^4\tilde{c}_4-70875\tilde{c}_3^2\tilde{c}_4\right.\\ &\quad+13191\tilde{c}_4\tilde{c}_5+\tilde{c}_2^3\left(-874584\tilde{c}_3^2+115701\tilde{c}_5\right)+27\tilde{c}_2^2\left(17945\tilde{c}_3\tilde{c}_4-983\tilde{c}_6\right)\\ &\quad\left.+9531\tilde{c}_3\tilde{c}_6+3\tilde{c}_2\left(67905\tilde{c}_3^3-17457\tilde{c}_4^2.-30618\tilde{c}_3\tilde{c}_5+1253\tilde{c}_7\right)-2\tilde{c}_8\right)\tilde{e}_n^8\\ &\quad+O\left[\tilde{e}_n\right]^9.\end{aligned}$$

Now we are obtaining,

$$\begin{aligned}\tilde{z}_n &= \frac{1}{9}\left(21\tilde{c}_2^3-9\tilde{c}_2\tilde{c}_3+\tilde{c}_4\right)\tilde{e}_n^4-\frac{2}{27}\left(174\tilde{c}_2^4-216\tilde{c}_2^2\tilde{c}_3+27\tilde{c}_3^2+30\tilde{c}_2\tilde{c}_4-4\tilde{c}_5\right)\tilde{e}_n^5\\ &\quad+\frac{2}{27}\left(623\tilde{c}_2^5-1257\tilde{c}_2^3\tilde{c}_3+321\tilde{c}_2^2\tilde{c}_4-99\tilde{c}_3\tilde{c}_4+9\tilde{c}_2\left(51\tilde{c}_3^2-5\tilde{c}_5\right)+7\tilde{c}_6\right)\tilde{e}_n^6\\ &\quad-\frac{4}{243}\left(8223\tilde{c}_2^6-22824\tilde{c}_2^4\tilde{c}_3-1377\tilde{c}_3^3+7734\tilde{c}_2^3\tilde{c}_4+393\tilde{c}_4^2+9\tilde{c}_2^2\left(1656\tilde{c}_3^2\right.\right.\\ &\quad\left.\left.-205\tilde{c}_5\right)+630\tilde{c}_3\tilde{c}_5-9\tilde{c}_2\left(669\tilde{c}_3\tilde{c}_4-29\tilde{c}_6\right)-46\tilde{c}_7\right)\tilde{e}_n^7+\frac{1}{729}\left(258960\tilde{c}_2^7\right.\\ &\quad-909738\tilde{c}_2^5\tilde{c}_3+364530\tilde{c}_2^4\tilde{c}_4+70875\tilde{c}_3^2\tilde{c}_4+3\tilde{c}_2^3\left(291528\tilde{c}_3^2-38567\tilde{c}_5\right)\\ &\quad-13191\tilde{c}_4\tilde{c}_5-27\tilde{c}_2^2\left(17945\tilde{c}_3\tilde{c}_4-983\tilde{c}_6\right)-9531\tilde{c}_3\tilde{c}_6+\tilde{c}_2\left(-203715\tilde{c}_3^3\right.\\ &\quad\left.\left.+52371\tilde{c}_4^2+91854\tilde{c}_3\tilde{c}_5-3759\tilde{c}_7\right)+2\tilde{c}_8\right)\tilde{e}_n^8+O\left[\tilde{e}_n\right]^9.\end{aligned}$$

Substituting all above equations in (11). Finally, the error equation for the proposed method is

$$\tilde{e}_{n+1}=\frac{1}{81}\tilde{c}_2\left(21\tilde{c}_2^3-9\tilde{c}_2\tilde{c}_3+\tilde{c}_4\right)^2\tilde{e}_n^8+O\left[\tilde{e}_n\right]^9.$$

5. Mathematical formulation

In this section, we discussed the algorithmic procedure for our proposed method. The following steps that are involved in the algorithmic procedure of our developed approach that we used to solve fuzzy nonlinear equations the following steps are represented in the following flow chart (Fig. 1).

Algorithm:

1. Start.
2. Find the values of $\tilde{x}_0$ and ε.
   Here $\tilde{x}_0$ is an initial approximation,
   ε is the absolute error or desired degree of accuracy, which is also the stopping criteria.
3. Calculate $\tilde{x}_1 = f(\tilde{x}_0)$ using (11).
4. if $|\tilde{x}_1 - \tilde{x}_0| < \varepsilon$, then display $\tilde{x}_1$ is the root.
5. Otherwise, assign $\tilde{x}_0 = \tilde{x}_1$ and go to step 3.

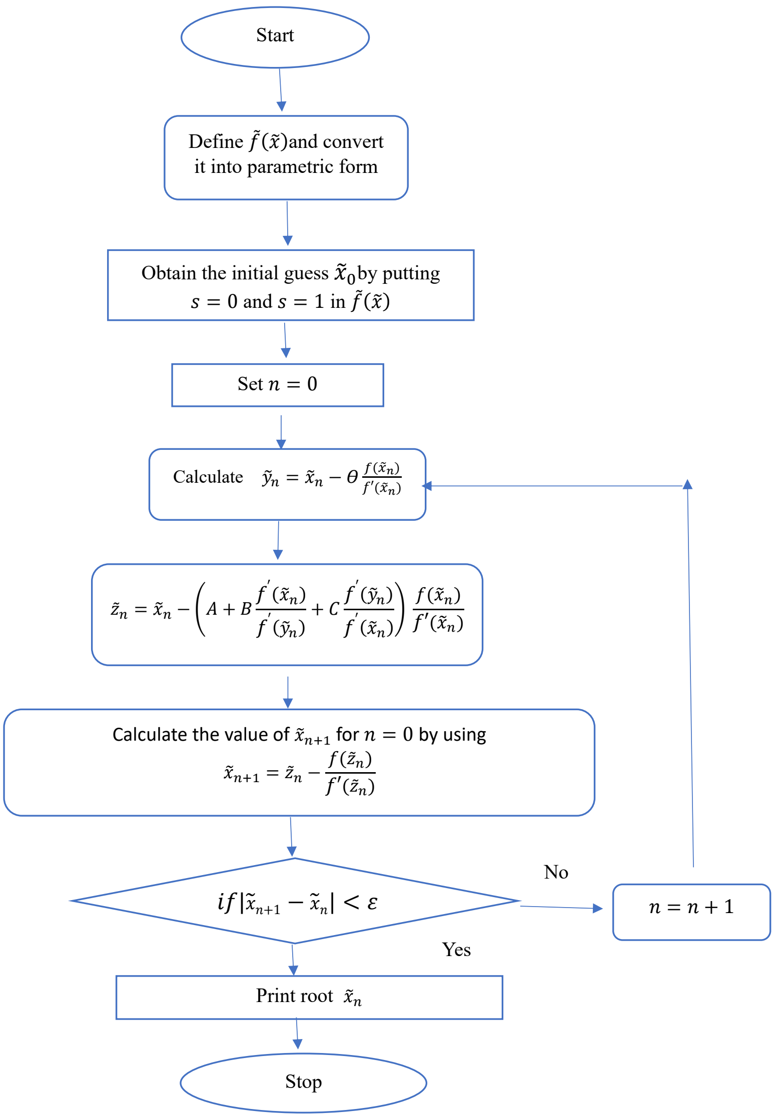


Fig. 1. Flow chart for the proposed method

6. Numerical results and discussion

In this section, we solve some example problems by using our proposed method and compare it with the existing eighth order iterative method to validate the computational efficiency. In the comparison table our proposed method represented by PM, eighth order iterative method represented by EOIM, N represents the number of iterations, and s represents the $\alpha-$ cut values. The computational procedure was coded in Mathematica 9 software. The example problems considered are as follows.

Example 1 Consider the fuzzy non linear equation,

$$(12,14,16)x^2+(8,10,12)x=(8,10,12), \tag{15}$$

the parametric representation of (15) is as follows,

$$\begin{cases} (12+2s)\underline{x}^2(s)+(8+2s)\underline{x}(s)-(8+2s)=0, \\ (16-2s)\bar{x}^2(s)+(12-2s)\bar{x}(s)-(12-2s)=0, \end{cases} \tag{16}$$

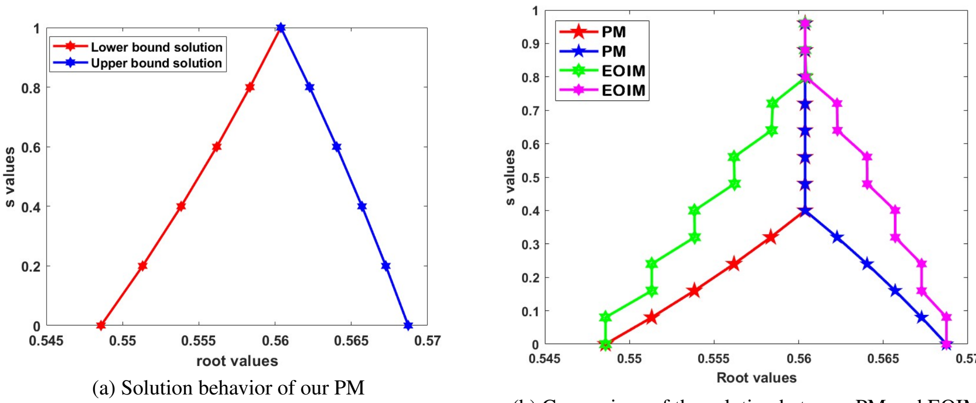
for parameter, $s=0$ and $s=1$. We get,

$$\begin{cases} 14\underline{x}^2(1)+10\underline{x}(1)=10, \\ 14\bar{x}^2(1)+10\bar{x}(1)=10. \end{cases} \text{ and } \begin{cases} 12\underline{x}^2(0)+8\underline{x}(0)=8, \\ 16\bar{x}^2(0)+12\bar{x}(0)=12. \end{cases} \tag{17}$$

We choose $\underline{x}(0)=0.548584, \bar{x}(0)=0.568729$ and $\underline{x}(1)=\bar{x}(1)=0.560374$. Therefore, the initial guess is defined as $x_0=(0.548584,0.560374,0.568729)$, and we choose an error tolerance of 10^{-5} following several significant iterations. We obtained the following numerical solution (15) for the different values of parameter s given in Table 1. Figure 2a represents the numerical solution of (15) by using our proposed method, and Figure 2b represents the comparison of the solution between the existing method and the proposed method.

Table 1. Comparison of PM with EOIM

s	N	EOIM		CPU Time	PM		CPU Time
		$\underline{x}(s)$	$\bar{x}(s)$	sec	$\underline{x}(s)$	$\bar{x}(s)$	sec
$s=0$	1	0.548587	0.568728	0.046	0.548584	0.568729	0.032
	2	0.548584	0.568729	0.046			
$s=0.2$	1	0.551319	0.567260	0.046	0.551315	0.567258	0.032
	2	0.551315	0.567258	0.046			
$s=0.4$	1	0.553853	0.565702	0.046	0.553838	0.565697	0.032
	2	0.553838	0.565697	0.046			
$s=0.6$	1	0.556208	0.564047	0.046	0.556176	0.564036	0.032
	2	0.556176	0.564036	0.046			
$s=0.8$	1	0.558401	0.562288	0.046	0.558459	0.562265	0.032
	2	0.558453	0.562263	0.046			
	3	0.558459	0.562265	0.046			
$s=1$	1	0.560449	0.560411	0.046	0.560374	0.560374	0.032
	2	0.560379	0.560378	0.046			
	3	0.560374	0.560374	0.046			



(a) Solution behavior of our PM

(b) Comparison of the solution between PM and EOIM

Fig. 2. Triangular fuzzy number plots

6.1. Application problem

The process known as "fractional conversion", which involves altering the nitrogen-hydrogen input, can produce ammonia. When the temperature and pressure inputs are treated as triangular fuzzy numbers, the resulting fuzzy nonlinear equation is given in [25]

$$\alpha_1 x^3(s) + \alpha_2 x^2(s) + \alpha_3 x(s) = \alpha_4. \tag{18}$$

This method allows for a more precise modeling of the system's behavior under varying conditions, accounting for the inherent uncertainties in the input parameters.

Here, $\alpha_1 = (7,9,11)$, $\alpha_2 = (5,7,9)$, $\alpha_3 = (3,5,7)$, $\alpha_4 = (3,5,7)$ are triangular fuzzy numbers. By replacing all the values in equation (18), we get the fuzzy nonlinear equation as follows

$$(7,9,11)x^3(s) + (5,7,9)x^2(s) + (3,5,7)x(s) = (3,5,7), \tag{19}$$

the parametric representation of (19) as follows,

$$\begin{cases} (7+2s)\underline{x}^3(s) + (5+2s)\underline{x}^2(s) + (3+2s)\underline{x}(s) - (3+2s) = 0, \\ (11-2s)\bar{x}^3(s) + (9-2s)\bar{x}^2(s) + (7-2s)\bar{x}(s) - (7-2s) = 0, \end{cases} \tag{20}$$

for $s = 0$ and $s = 1$. We get

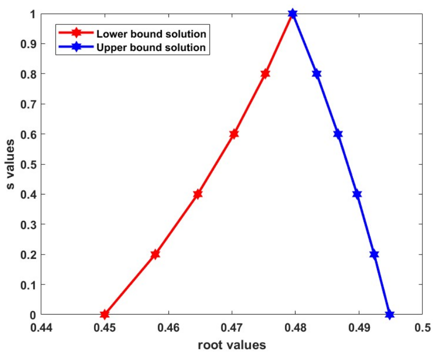
$$\begin{cases} 9\underline{x}^3(1) + 7\underline{x}^2(1) + 5\underline{x}(1) = 5, \\ 9\bar{x}^3(1) + 7\bar{x}^2(1) + 5\bar{x}(1) = 5. \end{cases} \text{and} \begin{cases} 7\underline{x}^3(0) + 5\underline{x}^2(0) + 3\underline{x}(0) = 3, \\ 11\bar{x}^3(0) + 9\bar{x}^2(0) + 7\bar{x}(0) = 7. \end{cases} \tag{21}$$

We choose the initial guess $\underline{x}(0) = 0.449968$, $\bar{x}(0) = 0.494817$ and $\underline{x}(1) = \bar{x}(1) = 0.479547$. Therefore, the initial guess is defined as $x_0 = (0.449968, 0.479547, 0.494817)$, and we choose an error tolerance of 10^{-5} following several significant iterations. We obtained the following numerical solution (19) for the different values of parameter

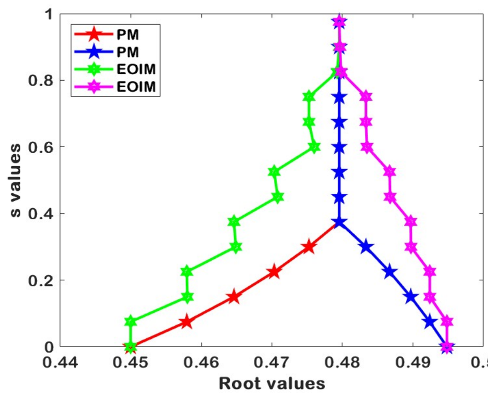
s given in Table 2. Figure 3a represents the numerical solution of (19) by using our proposed method, and Figure 3b represents the comparison of the solution between the existing method and the proposed method.

Table 2. Comparison of PM with EOIM

s	N	EOIM		CPU Time	PM		CPU Time
		$\underline{x}(s)$	$\bar{x}(s)$	sec	$\underline{x}(s)$	$\bar{x}(s)$	sec
$s=0$	1	0.449964	0.494815	0.046	0.449968	0.494817	0.032
	2	0.449968	0.494817	0.046			
$s=0.2$	1	0.458011	0.492386	0.046	0.457934	0.492379	0.032
	2	0.457934	0.492379	0.046			
$s=0.4$	1	0.464879	0.489711	0.046	0.464624	0.489683	0.032
	2	0.464624	0.489683	0.046			
$s=0.6$	1	0.470813	0.486756	0.046	0.470328	0.486684	0.032
	2	0.470328	0.486684	0.046			
$s=0.8$	1	0.475991	0.483472	0.046	0.475251	0.483328	0.032
	2	0.475252	0.483334	0.046			
	3	0.475251	0.483328	0.046			
$s=1$	1	0.479248	0.479802	0.046	0.479547	0.479547	0.032
	2	0.479548	0.479646	0.046			
	3	0.479547	0.479547	0.046			



(a) Solution behavior of our PM



(b) Comparison of the solution between PM and EOIM

Fig. 3. Triangular fuzzy number plots

7. Conclusion

In this study, we presented an iterative strategy for solving nonlinear equations, particularly when the coefficients are fuzzy. This method showed eighth-order convergence. Our results indicated that this strategy quickly converges on a solution. Tables 1 and 2 showed that our proposed approach outperforms the existing eighth--order iterative method in terms of fewer iterations and shorter computational time.

Figures 2 and 3 exhibit the solution behavior of our proposed method in Figure (a) and compared it to the existing EOIM in Figure (b). Furthermore, for application problems like fraction of conversion, our proposed solution outperforms the existing methods.

References

[1] Senol, M., Atpinar, S., Zararsiz, Z., Salahshour, S., & Ahmadian, A. (2019). Approximate solution of time-fractional fuzzy partial differential equations. *Computational and Applied Mathematics*, *38*, 1-18.

[2] El Fatini, M., Louriki, M., Pettersson, R., & Zararsiz, Z. (2021). Epidemic modeling: diffusion approximation vs. stochastic differential equations allowing reflection. *International Journal of Biomathematics*, *14*(05), 2150036.

[3] Jafari, R., Yu, W., Razvarz, S., & Gegov, A. (2021). Numerical methods for solving fuzzy equations: A survey. *Fuzzy Sets and Systems*, *404*, 1-22.

[4] Ibrahim, S.M., Mamat, M., & Ghazali, P.L. (2021). Shamanskii method for solving parameterized fuzzy nonlinear equations. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, *11*(1), 24-29.

[5] Abbasbandy, S., & Asady, B. (2004). Newton's method for solving fuzzy nonlinear equations. *Applied Mathematics and Computation*, *159*(2), 349-356.

[6] Abbasbandy, S., & Ezzati, R. (2006). Newton's method for solving a system of fuzzy nonlinear equations. *Applied Mathematics and Computation*, *175*(2), 1189-1199.

[7] Abbasbandy, S., & Jafarian, A. (2006). Steepest descent method for solving fuzzy nonlinear equations. *Applied Mathematics and Computation*, *174*(1), 669-675.

[8] Ramli, A., Abdullah, M.L., & Mamat, M. (2010). Broydens method for solving fuzzy nonlinear equations. *Advances in Fuzzy Systems*, *2010*(1), 763270.

[9] Sulaiman, I.M., Mamat, M., Malik, M., Nisar, K.S., & Elfasakhany, A. (2022). Performance analysis of a modified Newton method for parameterized dual fuzzy nonlinear equations and its application. *Results in Physics*, *33*, 105140.

[10] Magrenan Ruiz, A.A., & Argyros, I.K. (2014). Two-step Newton methods. *Journal of Complexity*, *30*(4), 533-553.

[11] Behl, R., Maroju, P., & Motsa, S.S. (2017). A family of second derivative free fourth order continuation method for solving nonlinear equations. *Journal of Computational and Applied Mathematics*, *318*, 38-46.

[12] Nadeem, G.A., Aslam, W., & Ali, F. (2023). An optimal fourth-order second derivative free iterative method for nonlinear scientific equations. *Kuwait Journal of Science*, *50*(2A).

[13] Ababneh, O. (2022). New iterative methods for solving nonlinear equations and their basins of attraction. *WSEAS Transactions on Mathematics Link Disabl.*, 21, 9-16.

[14] Abdul-Hassan, N.Y., Ali, A.H., & Park, C. (2022). A new fifth-order iterative method free from second derivative for solving nonlinear equations. *Journal of Applied Mathematics and Computing*, 1-10.

[15] Maroju, P., Magreñán, Á.A., Sarría, Í., & Kumar, A. (2020). Local convergence of fourth and fifth order parametric family of iterative methods in Banach spaces. *Journal of Mathematical Chemistry*, *58*, 686-705.

[16] Sivakumar, P., & Jayaraman, J. (2019). Some new higher order weighted Newton methods for solving nonlinear equation with applications. *Mathematical and Computational Applications*, *24*(2), 59.

[17] Alshomrani, A.S., Behl, R., & Maroju, P. (2020). Local convergence of parameter based method with six and eighth order of convergence. *Journal of Mathematical Chemistry*, *58*, 841-853.

[18] Solaiman, O.S., & Hashim, I. (2019). Two new efficient sixth order iterative methods for solving nonlinear equations. *Journal of King Saud University-Science*, *31*(4), 701-705.

[19] Maroju, P., Magreñán, Á.A., Motsa, S.S., & Sarría, Í. (2018). Second derivative free sixth order continuation method for solving nonlinear equations with applications. *Journal of Mathematical Chemistry*, *56*, 2099-2116.

[20] Kumar, A., Maroju, P., Behl, R., Gupta, D.K., & Motsa, S.S. (2018). A family of higher order iterations free from second derivative for nonlinear equations in R. *Journal of Computational and Applied Mathematics*, *330*, 676-694.

[21] Sharma, R., & Bahl, A. (2015). An optimal fourth order iterative method for solving nonlinear equations and its dynamics. *Journal of Complex Analysis*, *2015*(1), 259167.

[22] Sivakumar, P., Madhu, K., & Jayaraman, J. (2021). Optimal eighth and sixteenth order iterative methods for solving nonlinear equation with basins of attraction. *Applied Mathematics E-Notes*, *21*, 320-343.

[23] Farahmand Nejad, Maryam, et al. (2023). Gröbner basis approach for solving fuzzy complex system of linear equations. *New Mathematics and Natural Computation*, 1-18.

[24] Sariman, S.A., & Hashim, I. (2020). New optimal Newton-Householder methods for solving nonlinear equations and their dynamics. *Computers, Materials & Continua*, *65*(1), 69-85.

[25] Rafiq, N., Yaqoob, N., Kausar, N., Shams, M., Mir, N.A., Gaba, Y.U., & Khan, N. (2021). Computer-based fuzzy numerical method for solving engineering and real-world applications. *Mathematical Problems in Engineering*, *2021*(1), 6916282.