scientific reports

OPEN

A new approach for solving fuzzy non-linear equations using higher order iterative method

Srilakshmi Katuri & Prashanth Maroju

This manuscript presents a novel multi-step, tenth-order iterative method for solving fuzzy nonlinear equations, which frequently emerge in a variety of applications such as optimization, decision-making, control theory, and chemical engineering problems. One of the principal challenges in solving these equations lies in the computational demands of computing and inverting the Jacobian matrix at each iteration. The primary advantage of the proposed iterative method is that it obviates the need for Jacobian matrix calculations, thereby markedly reducing the computational complexity associated with solving fuzzy nonlinear Equations. We conduct a thorough convergence analysis and establish that our method achieves tenth-order convergence. The effectiveness and robustness of the developed approach are illustrated through comprehensive numerical examples and real-life application problems, complete with graphical representations. Furthermore, we compare our method with existing tenth-order iterative methods to demonstrate the superior efficiency of our approach.

Keywords Fuzzy, Dual fuzzy non-linear equations, Jacobian matrix, Iterative methods

Over the past two decades, Fuzzy Nonlinear Equations (FNE) have found extensive applications in various fields, including engineering, health sciences, mathematics, statistics, and social sciences[1,2]. The concept of fuzziness was first introduced and explored by Zadeh in 1965[3]. The term 'fuzzy' denotes something unclear or vague. In real-world scenarios where uncertainty is prevalent, fuzzy equations serve as ideal mathematical models to address such problems[4]. One of the primary applications of fuzzy arithmetic is in dealing with the parameters of nonlinear equations, which are partially or entirely represented by fuzzy arithmetic[5]. We consider nonlinear equations in the form of

$$f(x) = 0, \tag{1.1}$$

with all the coefficients are fuzzy numbers. The FNE as follows,

$$\begin{aligned} &\tilde{a}_1\tilde{x}^3 + \tilde{b}_1\tilde{x}^2 + \tilde{c}_1\tilde{x} - \tilde{d}_1 = \tilde{e}_1 \\ &\tilde{a}_1 e^{\tilde{x}} + \tilde{b}_1 = \tilde{d}_1, \end{aligned} \tag{1.2}$$

where $\tilde{\mathrm{a}}_1, \tilde{\mathrm{b}}_1, \tilde{\mathrm{c}}_1, \tilde{\mathrm{d}}_1$ and $\tilde{\mathrm{e}}_1$ are fuzzy numbers. Standard analytic methods, as discussed by Buckley and Qu[6], are not capable of solving these types of Eq. (1.2). This challenge has motivated us to explore various iterative approaches for solving nonlinear equations with fuzzy coefficients. Solving systems using fuzzy coefficients is a prominent area of study within fuzzy mathematics. To achieve efficient and accurate solutions for fuzzy nonlinear problems, researchers continuously improve existing techniques by developing new algorithms. One notable method is the famous Newton method, introduced by Abbasbandy and Asady[7]. The simplest variant of the Newton method, developed by Abbasbandy et al.[8], is

$$\tilde{x}_{n+1} = \tilde{x}_n - \frac{\tilde{f}(\tilde{x}_n)}{\tilde{f}'(\tilde{x}_n)} \tag{1.3}$$

Newton's method, which has a quadratic convergence order, converges very quickly when the initial guess is near the root. However, a significant drawback is the need to calculate the Jacobian matrix at each iteration. Many researchers have addressed this by developing methods to solve FNEs that require the Jacobian matrix

Department of Mathematics, School of Advanced Sciences, VIT AP University, Amaravathi, Andhra Pradesh, India. ✉email: maroju.prashanth@gmail.com

calculation at every step. Abbasbandy and Jafarian proposed a gradient-based method, also known as the Steepest Descent method[9]. Broyden's method, studied by Amirah et al.[10], and the Modified Newton method applied by Sulaiman[11], are notable approaches for solving FNEs. Additionally, Sandip Moi[12–17] introduced semi-analytical and collocation methods for solving fuzzy integro-differential equations. The literature also includes third-order iterative methods developed by Thota S et al.[18], and fourth-order methods by Kansal et al.[19–21]. Fifth-order iterative methods have been introduced by Maroju et al.[22–24], and sixth-order methods have been discussed by Sharma et al.[25–27]. Eighth-order iterative methods were proposed in[28] for solving nonlinear equations.

Our research aims to introduce a new iterative method for solving FNEs. As noted, all current methods are based on Newton's method. Therefore, reducing the computational complexity of calculating and inverting the Jacobian matrix at each iteration is vital. In this work, we propose a fifth-order iterative method[29] for solving FNEs. We further develop this method to improve the convergence order to the tenth, ensuring it converges to a solution more efficiently than existing methods.

Sect. Mathematical preliminaries outlines the basic definitions of fuzzy sets. In Sect. Development of tenth order iterative method, we develop an iterative method for solving FNE. Section Convergence analysis presents the convergence analysis of our proposed method. Section Numerical application problems includes the solution of several numerical examples and application problems, accompanied by graphical representations. Finally, Sect. Conclusion concludes with our findings and insights.

Mathematical preliminaries

In this section, we discuss some definitions and preliminaries.

Definition

For the set X and the membership function $\eta_{\tilde{\nu}}$. The representation of the mapping $\tilde{\nu} : X \to [0,1]$. A fuzzy set is said to fuzzy number if it follows the below conditions[30],

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

Definition

The triangular fuzzy number, which is denoted by the triplet $\tilde{\nu}$=(a_1, b_1, c_1). Where, $a_1 \leqslant b_1 \leqslant c_1$ and its membership function defined as follows[30],

$$\eta_{\tilde{\nu}}(x_0) = \begin{cases} \frac{x_0 - a_1}{b_1 - a_1} & a_1 \leqslant x_0 \leqslant b_1 \\ 1 & x_0 = b_1 \\ \frac{c_1 - x_0}{c_1 - b_1} & b_1 \leqslant x_0 \leqslant c_1 \\ 0 & Otherwise \end{cases} \tag{2.1}$$

Also, α - cut of $\tilde{\nu}$ for $0 \leqslant \alpha \leqslant 1$ can be defined as

$$\tilde{\nu}(\alpha) = [\underline{\nu}(\alpha), \bar{\nu}(\alpha)] = [a_1 + (b_1 - a_1)\alpha, c_1 + (b_1 - c_1)\alpha]. \tag{2.2}$$

Definition

As discussed above, fuzzy numbers may be transformed into an interval through parametric form. So, for any arbitrary fuzzy number $\tilde{x}(\alpha) = [\underline{x}(\alpha), \bar{x}(\alpha)], \tilde{y}(\alpha) = [\underline{y}(\alpha), \bar{y}(\alpha)]$ and scalar k, we have the interval based fuzzy arithmetic as[31]

1. $\tilde{x} = \tilde{y}$ if and only if $\underline{x}(\alpha) = \underline{y}(\alpha)$ and $\bar{x}(\alpha) = \bar{y}(\alpha)$.
2. $\tilde{x} + \tilde{y}$ = $[\underline{x}(\alpha) + \underline{y}(\alpha), \bar{x}(\alpha) + \bar{y}(\alpha)]$.
3. $\tilde{x} - \tilde{y}$ = $[\underline{x}(\alpha) - \bar{y}(\alpha), \bar{x}(\alpha) - \underline{y}(\alpha)]$
4. $\tilde{x} \times \tilde{y}$ = $[min(S), max(S)]$, where $S = \{\underline{x}(\alpha) \times \underline{y}(\alpha), \underline{x}(\alpha) \times \bar{y}(\alpha), \bar{x}(\alpha) \times \underline{y}(\alpha), \bar{x}(\alpha) \times \bar{y}(\alpha)\}$
5. $k\tilde{x} = \begin{cases} \langle kx(\alpha), kx(\alpha)\rangle, & k < 0 \\ \langle kx(\alpha), kx(\alpha)\rangle, & k \geq 0 \end{cases}$

Development of tenth order iterative method

In the fuzzy environment, the nonlinear equation $f(x) = 0$ modify to incorporate the uncertainty, so it becomes as fuzzy nonlinear equation of the form

$$\tilde{f}(\tilde{x}_L, \tilde{x}_C, \tilde{x}_U) = (\tilde{c}_L, \tilde{c}_C, \tilde{c}_U), \tag{3.1}$$

simply it can be represented as

$$\tilde{f}(\tilde{x}) = \tilde{c}, \tag{3.2}$$

to simplify (3.2), assume $\tilde{\beta} = (\tilde{\beta}_L, \tilde{\beta}_C, \tilde{\beta}_U)$ be the fuzzy root and $(\underline{\tilde{\alpha}}, \bar{\tilde{\alpha}})$ be 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

$$\tilde{f}\left(\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}}\right)+\left(\tilde{x}-(\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}})\right)\tilde{f}'(\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}})+\frac{\left(\tilde{x}-(\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}})\right)^2}{2!}\tilde{f}''(\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}})+\cdots=\tilde{c} \tag{3.3}$$

from (3.3),

$$\tilde{x}=(\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}})-\frac{\tilde{f}\left((\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}})\right)}{\tilde{f}'((\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}}))}-\frac{\left(\tilde{x}-(\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}})\right)^2\tilde{f}''((\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}}))}{2\tilde{f}'((\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}}))}+\tilde{c} \tag{3.4}$$

Each higher-order term, such as $\left(\tilde{x}-(\underset{\sim}{\tilde{\alpha}},\tilde{\bar{\alpha}})\right)^n$, is computed using fuzzy arithmetic operations, and the coefficients are unaffected by the fuzziness. In this section, we develop a higher-order iterative method for solving FNE. We begin by considering the fifth-order iterative method developed by[29].

$$\begin{aligned}\tilde{y}_n&=\tilde{x}_n-\frac{\tilde{f}(\tilde{x}_n)}{\tilde{f}'(\tilde{x}_n)}\\ \tilde{x}_{n+1}&=\tilde{y}_n-\frac{5\tilde{f}''(\tilde{x}_n)+3\tilde{f}''(\tilde{y}_n)}{\tilde{f}''(\tilde{x}_n)+7\tilde{f}''(\tilde{y}_n)}\times\frac{\tilde{f}(\tilde{y}_n)}{\tilde{f}'(\tilde{x}_n)}\end{aligned} \tag{3.5}$$

To enhance the order of convergence of our proposed method, we incorporate a third step in the form of the Newton method, transforming it into a three-step approach.

$$\begin{aligned}\tilde{y}_n&=\tilde{x}_n-\frac{\tilde{f}(\tilde{x}_n)}{\tilde{f}'(\tilde{x}_n)}\\ \tilde{z}_n&=\tilde{y}_n-\frac{5\tilde{f}''(\tilde{x}_n)+3\tilde{f}''(\tilde{y}_n)}{\tilde{f}''(\tilde{x}_n)+7\tilde{f}''(\tilde{y}_n)}\times\frac{\tilde{f}(\tilde{y}_n)}{\tilde{f}'(\tilde{x}_n)}\\ \tilde{x}_{n+1}&=\tilde{z}_n-\frac{\tilde{f}(\tilde{z}_n)}{\tilde{f}'(\tilde{z}_n)}\end{aligned} \tag{3.6}$$

Convergence analysis

In this section, we discuss the order of convergence of our proposed method. The triangular fuzzy number $\tilde{\nu}$=(a_1,b_1,c_1) where, $a_1\leqslant b_1\leqslant c_1$ can be represented with an ordered pair of functions through an α-cut approach as $\tilde{\nu}(\alpha)=[\underline{\nu}(\alpha),\bar{\nu}(\alpha)]=[a_1+(b_1-a_1)\alpha,c_1+(b_1-c_1)\alpha]$. The $\alpha-$cut form is known as parametric form or single parametric form of fuzzy numbers[31]. It may noted that lower bound and upper bounds of fuzzy numbers satisfies the following conditions.

- $\underline{\nu}(\alpha)$is a bounded monotonic increasing left continuous function,
- $\bar{\nu}(\alpha)$ is a bounded monotonic decreasing left continuous function,
- $\underline{\nu}(\alpha)\leq\bar{\nu}(\alpha),0\leq\alpha\leq1$.

In generally, the fuzzy nonlinear equation can be represented as

$$\tilde{a}_1\tilde{x}_1+\tilde{a}_2\tilde{x}_2+\tilde{a}_3\tilde{x}_3+\tilde{a}_4\tilde{x}_4+\cdots+\tilde{a}_n\tilde{x}_n=(\tilde{c}_1,\tilde{c}_2,\tilde{c}_3,\cdots,\tilde{c}_n) \tag{4.1}$$

the above equation can be represented as

$$\sum_{i=1}^{n}\tilde{a}_i\tilde{x}_i=\tilde{c}_i,\quad i=1,2,3,4,\ldots \tag{4.2}$$

Where $\tilde{a}_i$, $\tilde{c}_i$ are triangular fuzzy numbers and $\tilde{x}_i$ is the fuzzy variable, $\tilde{x}_i\geq0$. Using the parametric form of fuzzy number we may write $\tilde{a}_i,\tilde{x}_i\,\tilde{c}_i$ as $\tilde{a}_i(\alpha)=[\underset{\sim}{\tilde{a}}_i(\alpha),\tilde{\bar{a}}_i(\alpha)]$,$\tilde{x}_i(\alpha)=[\underset{\sim}{\tilde{x}}_i(\alpha),\tilde{\bar{x}}_i(\alpha)]$,$\tilde{c}_i(\alpha)=[\underset{\sim}{\tilde{c}}_i(\alpha),\tilde{\bar{c}}_i(\alpha)]$. Substituting all expressions in (4.2). We get

$$\sum_{i=1}^{n}[\underset{\sim}{\tilde{a}}_i(\alpha),\tilde{\bar{a}}_i(\alpha)][\underset{\sim}{\tilde{x}}_i(\alpha),\tilde{\bar{x}}_i(\alpha)]=[\underset{\sim}{\tilde{c}}_i(\alpha),\tilde{\bar{c}}_i(\alpha)],\quad i=1,2,3,4,\ldots \tag{4.3}$$

By applying standard rule of fuzzy arithmetic, the Eq. (4.3) can be written as the following two crisp equations.

$$\sum_{i=1}^{n}\underset{\sim}{\tilde{a}}_i(\alpha)\underset{\sim}{\tilde{x}}_i(\alpha)=\underset{\sim}{\tilde{c}}_i(\alpha) \tag{4.4}$$

and

$$\sum_{i=1}^{n} \tilde{\tilde{a}}_i(\alpha)\tilde{\tilde{x}}_i(\alpha) = \tilde{\tilde{c}}_i(\alpha) \tag{4.5}$$

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 ten for both lower and upper bound conditions. The fuzzy error equation is given by: $\tilde{e}_{n+1} = \frac{1}{4}\tilde{c}_2^5\left(7\tilde{c}_2^2 - 2\tilde{c}_3\right)^2 \tilde{e}_n^{10} + O\left[\tilde{e}_n\right]^{11}$.*

Proof Let $\tilde{\beta}$ be a root of the fuzzy equation $\tilde{f}(\tilde{x}) = \tilde{c}$, Then the error in nth iteration, $\tilde{e_n} = \tilde{x}_n - \tilde{\beta}$ by using Taylor series expansion, we get

$$\tilde{f}(\tilde{x}_n) = \tilde{f}'(\tilde{\beta})\left(\tilde{e}_n + \tilde{c}_2\tilde{e}_n^2 + \tilde{c}_3\tilde{e}_n^3 + \tilde{c}_4\tilde{e}_n^4 + \tilde{c}_5\tilde{e}_n^5 + \cdots + \tilde{c}_{11}\tilde{e}_n^{11} + O(\tilde{e}_n^{12})\right) \tag{4.6}$$

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,

$$\tilde{c}_k = \frac{1}{k!}\frac{\tilde{f}^{(k)}(\tilde{\beta})}{\tilde{f}'(\tilde{\beta})}, \quad k = 2, 3, 4, \ldots, \tag{4.7}$$

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

$$\tilde{f}'\left(\tilde{x}_n\right) = \tilde{f}'(\tilde{\beta})\left(1 + 2\tilde{c}_2\tilde{e}_n + 3\tilde{c}_3\tilde{e}_n^2 + 4\tilde{c}_4\tilde{e}_n^3 + 5\tilde{c}_5\tilde{e}_n^4 + \cdots + 11\tilde{c}_{11}\right)\tilde{e}_n^{10} + O\left(\tilde{e}_n^{11}\right) \tag{4.8}$$

using (4.6) and (4.8) equations. We obtain

$$\begin{aligned}
\frac{\tilde{f}\left(\tilde{x}_n\right)}{\tilde{f}'\left(\tilde{x}_n\right)} =& \tilde{e}_n - \tilde{c}_2\tilde{e}_n^2 + \left(2\tilde{c}_2^2 - 2\tilde{c}_3\right)\tilde{e}_n^3 + \left(-4\tilde{c}_2^3 + 7\tilde{c}_2\tilde{c}_3 - 3\tilde{c}_4\right)\tilde{e}_n^4 \\
&+ 2\Big(4\tilde{c}_2^4 - 10\tilde{c}_2^2\tilde{c}_3 + 3\tilde{c}_3^2 + 5\tilde{c}_2\tilde{c}_4 - 2\tilde{c}_5\Big)\tilde{e}_n^5 \\
&+ \Big(-16\tilde{c}_2^5 + 52\tilde{c}_2^3\tilde{c}_3 - 33\tilde{c}_2\tilde{c}_3^2 - 28\tilde{c}_2^2\tilde{c}_4 + 17\tilde{c}_3\tilde{c}_4 + 13\tilde{c}_2\tilde{c}_5 - 5\tilde{c}_6\Big)\tilde{e}_n^6 \\
&+ 2\Big(16\tilde{c}_2^6 - 64\tilde{c}_2^4\tilde{c}_3 + 63\tilde{c}_2^2\tilde{c}_3^2 - 9\tilde{c}_3^3 + 36\tilde{c}_2^3\tilde{c}_4 - 46\tilde{c}_2\tilde{c}_3\tilde{c}_4 + 6\tilde{c}_4^2 \\
&\quad - 18\tilde{c}_2^2\tilde{c}_5 + 11\tilde{c}_3\tilde{c}_5 + 8\tilde{c}_2\tilde{c}_6 - 3\tilde{c}_7\Big)\tilde{e}_n^7 - 64\tilde{c}_2^7 + 304\tilde{c}_2^5\tilde{c}_3 - 408\tilde{c}_2^3\tilde{c}_3^2 \\
&+ 135\tilde{c}_2\tilde{c}_3^3 - 176\tilde{c}_2^4\tilde{c}_4 + 348\tilde{c}_2^2\tilde{c}_3\tilde{c}_4 - 75\tilde{c}_3^2\tilde{c}_4 - 64\tilde{c}_2\tilde{c}_4^2 + 92\tilde{c}_2^3\tilde{c}_5 - 118\tilde{c}_2\tilde{c}_3\tilde{c}_5 \\
&\quad + 31\tilde{c}_4\tilde{c}_5 - 44\tilde{c}_2^2\tilde{c}_6 + \Big(27\tilde{c}_3\tilde{c}_6 + 19\tilde{c}_2\tilde{c}_7\Big)\tilde{e}_n^8 + 2\Big(64\tilde{c}_2^8 - 352\tilde{c}_2^6\tilde{c}_3 + 600\tilde{c}_2^4\tilde{c}_3^2 \\
&- 324\tilde{c}_2^2\tilde{c}_3^3 + 27\tilde{c}_3^4 + 208\tilde{c}_2^5\tilde{c}_4 - 560\tilde{c}_2^3\tilde{c}_3\tilde{c}_4 + 279\tilde{c}_2\tilde{c}_3^2\tilde{c}_4 + 120\tilde{c}_2^2\tilde{c}_4^2 - 52\tilde{c}_3\tilde{c}_4^2 - 112\tilde{c}_2^4\tilde{c}_5 \\
&\quad + 222\tilde{c}_2^2\tilde{c}_3\tilde{c}_5 - 48\tilde{c}_3^2\tilde{c}_5 - 82\tilde{c}_2\tilde{c}_4\tilde{c}_5 + 10\tilde{c}_5^2 + 56\tilde{c}_2^3\tilde{c}_6 - 72\tilde{c}_2\tilde{c}_3\tilde{c}_6\Big) + 19\tilde{c}_4\tilde{c}_6 - 26\tilde{c}_2^2\tilde{c}_7 \\
&\quad + 16\tilde{c}_3\tilde{c}_7\tilde{e}_n^9 - 256\tilde{c}_2^9 + 1600\tilde{c}_2^7\tilde{c}_3 - 3312\tilde{c}_2^5\tilde{c}_3^2 + 2520\tilde{c}_2^3\tilde{c}_3^3 - 513\tilde{c}_2\tilde{c}_3^4 - 960\tilde{c}_2^6\tilde{c}_4 + 3280\tilde{c}_2^4\tilde{c}_3\tilde{c}_4 \\
&- 2664\tilde{c}_2^2\tilde{c}_3^2\tilde{c}_4 + 297\tilde{c}_3^3\tilde{c}_4 - 768\tilde{c}_2^3\tilde{c}_4^2 + 768\tilde{c}_2\tilde{c}_3\tilde{c}_4^2 - 48\tilde{c}_4^3 + 528\tilde{c}_2^5\tilde{c}_5 - 1424\tilde{c}_2^3\tilde{c}_3\tilde{c}_5 + 711\tilde{c}_2\tilde{c}_3^2\tilde{c}_5 \\
&\quad + 612\tilde{c}_2^2\tilde{c}_4\tilde{c}_5 - 266\tilde{c}_3\tilde{c}_4\tilde{c}_5 - 105\tilde{c}_2\tilde{c}_5^2 - 272\tilde{c}_2^4\tilde{c}_6 + 540\tilde{c}_2^2\tilde{c}_3\tilde{c}_6 - 117\tilde{c}_3^2\tilde{c}_6 - 200\tilde{c}_2\tilde{c}_4\tilde{c}_6 + 49\tilde{c}_5\tilde{c}_6 \\
&\quad + 132\tilde{c}_2^3\tilde{c}_7 - 170\tilde{c}_2\tilde{c}_3\tilde{c}_7 + 45\tilde{c}_4\tilde{c}_7\tilde{e}_n^{10} + O\left[\tilde{e}_n\right]^{11}.
\end{aligned}$$

Hence, we get

$$\begin{aligned}
\tilde{y}_n =& \tilde{c}_2\tilde{e}_n^2 + \left(-2\tilde{c}_2^2 + 2\tilde{c}_3\right)\tilde{e}_n^3 + \left(4\tilde{c}_2^3 - 7\tilde{c}_2\tilde{c}_3 + 3\tilde{c}_4\right)\tilde{e}_n^4 + \left(-8\tilde{c}_2^4 + 20\tilde{c}_2^2\tilde{c}_3 - 6\tilde{c}_3^2 - 10\tilde{c}_2\tilde{c}_4 + 4\tilde{c}_5\right)\tilde{e}_n^5 \\
&+ \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) + 5\tilde{c}_6\right)\tilde{e}_n^6 \\
&- 2\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 + \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 \\
&+ \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 + \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)\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 - 2\left(64\tilde{c}_2^8 - 352\tilde{c}_2^6\tilde{c}_3 + 27\tilde{c}_3^4 + 208\tilde{c}_2^5\tilde{c}_4 + 8\tilde{c}_2^4\left(75\tilde{c}_3^2 - 14\tilde{c}_5\right)\right. \\
&- \left(48\tilde{c}_3^2\tilde{c}_5 + 10\tilde{c}_5^2 + 19\tilde{c}_4\tilde{c}_6 + 56\tilde{c}_2^3\left(-10\tilde{c}_3\tilde{c}_4 + \tilde{c}_6\right) + \tilde{c}_2\left(279\tilde{c}_3^2\tilde{c}_4 - 82\tilde{c}_4\tilde{c}_5 - 72\tilde{c}_3\tilde{c}_6\right)\right) \\
&\left. + \tilde{c}_2^2\left(-324\tilde{c}_3^3 + 120\tilde{c}_4^2 + 222\tilde{c}_3\tilde{c}_5 - 26\tilde{c}_7\right) + \tilde{c}_3\left(-52\tilde{c}_4^2 + 16\tilde{c}_7\right)\right)\tilde{e}_n^9 + 256\tilde{c}_2^9 - 1600\tilde{c}_2^7\tilde{c}_3 + 960\tilde{c}_2^6\tilde{c}_4 \\
&- 297\tilde{c}_3^3\tilde{c}_4 + 48\tilde{c}_4^3 + 48\tilde{c}_2^5\left(69\tilde{c}_3^2 - 11\tilde{c}_5\right) + 266\tilde{c}_3\tilde{c}_4\tilde{c}_5 + 117\tilde{c}_3^2\tilde{c}_6 - 49\tilde{c}_5\tilde{c}_6 + \tilde{c}_2^4\left(-3280\tilde{c}_3\tilde{c}_4 + 272\tilde{c}_6\right) \\
&+ 36\tilde{c}_2^2\left(74\tilde{c}_3^2\tilde{c}_4 - 17\tilde{c}_4\tilde{c}_5 - 15\tilde{c}_3\tilde{c}_6\right) - 45\tilde{c}_4\tilde{c}_7 - \left(4\tilde{c}_2^3\left(630\tilde{c}_3^3 - 192\tilde{c}_4^2 - 356\tilde{c}_3\tilde{c}_5 + 33\tilde{c}_7\right)\right) \\
&+ \left(\tilde{c}_2\left(513\tilde{c}_3^4 - 711\tilde{c}_3^2\tilde{c}_5 + 5\left(21\tilde{c}_5^2 + 40\tilde{c}_4\tilde{c}_6\right)\right) + \tilde{c}_3\left(-768\tilde{c}_4^2 + 170\tilde{c}_7\right)\right)\tilde{e}_n^{10} + O\left[\tilde{e}_n\right]^{11}.
\end{aligned}$$

Now we get,

$$\begin{aligned}
\frac{5\tilde{f}''(\tilde{x}_n)+3\tilde{f}''(\tilde{y}_n)}{\tilde{f}''(\tilde{x}_n)+7\tilde{f}''(\tilde{y}_n)}\times\frac{\tilde{f}(\tilde{y}_n)}{\tilde{f}'(\tilde{x}_n)} =& \tilde{c}_2\tilde{e}_n^2+\left(-2\tilde{c}_2^2+2\tilde{c}_3\right)\tilde{e}_n^3+\left(4\tilde{c}_2^3-7\tilde{c}_2\tilde{c}_3+3\tilde{c}_4\right)\tilde{e}_n^4 \\
&+\left(-\frac{23\tilde{c}_2^4}{2}+21\tilde{c}_2^2\tilde{c}_3-6\tilde{c}_3^2-10\tilde{c}_2\tilde{c}_4+4\tilde{c}_5\right)\tilde{e}_n^5 \\
&+\frac{1}{4}\left(155\tilde{c}_2^5-315\tilde{c}_2^3\tilde{c}_3+116\tilde{c}_2^2\tilde{c}_4-68\tilde{c}_3\tilde{c}_4+\tilde{c}_2\left(149\tilde{c}_3^2-52\tilde{c}_5\right)+20\tilde{c}_6\right)\tilde{e}_n^6 \\
&+\frac{1}{8}\left(-955\tilde{c}_2^6+2492\tilde{c}_2^4\tilde{c}_3-844\tilde{c}_2^3\tilde{c}_4+\tilde{c}_2^2\left(-1637\tilde{c}_3^2+296\tilde{c}_5\right)\right) \\
&+\left(8\tilde{c}_2\left(103\tilde{c}_3\tilde{c}_4-16\tilde{c}_6\right)+4\left(45\tilde{c}_3^3-44\tilde{c}_3\tilde{c}_5+12\left(-2\tilde{c}_4^2+\tilde{c}_7\right)\right)\right)\tilde{e}_n^7 \\
&+\frac{1}{16}\left(5147\tilde{c}_2^7-17453\tilde{c}_2^5\tilde{c}_3+6652\tilde{c}_2^4\tilde{c}_4+1500\tilde{c}_3^2\tilde{c}_4+12\tilde{c}_2^3\left(1339\tilde{c}_3^2-179\tilde{c}_5\right)\right) \\
&-\left(496\tilde{c}_4\tilde{c}_5-432\tilde{c}_3\tilde{c}_6+\tilde{c}_2^2\left(-8756\tilde{c}_3\tilde{c}_4+720\tilde{c}_6\right)\right) \\
&+\left(\tilde{c}_2\left(-3837\tilde{c}_3^3+1136\tilde{c}_4^2+2104\tilde{c}_3\tilde{c}_5-304\tilde{c}_7\right)\right)\tilde{e}_n^8 \\
&-\frac{25019\tilde{c}_2^8}{32}+\frac{53007}{16}\tilde{c}_2^6\tilde{c}_3-\frac{857\tilde{c}_3^4}{8}-\frac{11811}{8}\tilde{c}_2^5\tilde{c}_4+120\tilde{c}_3^2\tilde{c}_5-20\tilde{c}_5^2 \\
&+\tilde{c}_2^4\left(-\frac{66075\tilde{c}_3^2}{16}+535\tilde{c}_5\right)+\tilde{c}_2^3\left(\frac{5317\tilde{c}_3\tilde{c}_4}{2}-163\tilde{c}_6\right)-38\tilde{c}_4\tilde{c}_6 \\
&+\tilde{c}_2\left(-\frac{7693}{8}\tilde{c}_3^2\tilde{c}_4+181\tilde{c}_4\tilde{c}_5+160\tilde{c}_3\tilde{c}_6\right)+2\tilde{c}_3\left(65\tilde{c}_4^2-16\tilde{c}_7\right) \\
&+\tilde{c}_2^2\left(\frac{13053\tilde{c}_3^3}{8}-363\tilde{c}_4^2-\frac{2789\tilde{c}_3\tilde{c}_5}{4}+53\tilde{c}_7\right)\tilde{e}_n^9 \\
&+\frac{1}{64}\left(115035\tilde{c}_2^9-579087\tilde{c}_2^7\tilde{c}_3+293100\tilde{c}_2^6\tilde{c}_4+\tilde{c}_2^5\left(925371\tilde{c}_3^2-122764\tilde{c}_5\right)\right) \\
&+\tilde{c}_2^4\left(-707772\tilde{c}_3\tilde{c}_4+41680\tilde{c}_6\right)+8\tilde{c}_2^2\left(51411\tilde{c}_3^2\tilde{c}_4-7382\tilde{c}_4\tilde{c}_5-6778\tilde{c}_3\tilde{c}_6\right) \\
&+\tilde{c}_2\left(86121\tilde{c}_3^4-78580\tilde{c}_3^2\tilde{c}_5+16\left(461\tilde{c}_5^2+880\tilde{c}_4\tilde{c}_6\right)-32\tilde{c}_3\left(2550\tilde{c}_4^2-377\tilde{c}_7\right)\right) \\
&-2\tilde{c}_2^3\left(266897\tilde{c}_3^3-55824\tilde{c}_4^2-109568\tilde{c}_3\tilde{c}_5+6136\tilde{c}_7\right)-4\left(9171\tilde{c}_3^3\tilde{c}_4-960\tilde{c}_4^3\right) \\
&-\left(5320\tilde{c}_3\tilde{c}_4\tilde{c}_5-2340\tilde{c}_3^2\tilde{c}_6+784\tilde{c}_5\tilde{c}_6+720\tilde{c}_4\tilde{c}_7\right)\tilde{e}_n^{10}+O\left[\tilde{e}_n\right]
\end{aligned}$$

Using above equ. in proposed method (3.6). We get,

$$\tilde{e}_{n+1}=\frac{1}{4}\tilde{c}_2^5\left(7\tilde{c}_2^2-2\tilde{c}_3\right)^2\tilde{e}_n^{10}+O\left[\tilde{e}_n\right]^{11}$$

Then we conclude that proposed method (3.6) is tenth order convergence. □

Numerical application problems

In this section, we solve some application problems by using our proposed method and compared it with existing tenth order method developed by Kashmita[32] to validate the computational efficiency. The proposed method is represented as PM and existing tenth order method represented as KPM. The computational procedure was coded in 11th Gen Intel(R) 8 RAM system Mathematica 9 software. The example application problems considered as follows.

Example 5.1 We consider the Vander Waal's equation. The equation of Vander Waal's helps to interpret the behavior of both real and ideal gases, which ultimately leads to the equation of the following:

$$\left(P+\frac{A_1n^2}{x^2}\right)(x-nA_2)=nRT \tag{5.1}$$

using the particular values of the parameter, we are able to obtain the fuzzy nonlinear equation that is as follows:

$$A_1x^3(s)+A_2x^2(s)+A_3x(s)=A_4 \tag{5.2}$$

where A_1=(5,7,9), A_2=(3,4,5), A_3=(2,3,4), A_4=(2,3,4). Where, the variable *x*, *R*, *P*, *n*, *T*, A_1, A_2, A_3 and A_4 represents the volume of the gas, pressure, general gas constant, number of moles, temperature, and generic parameters.

The Parametric form of (5.2) is as follows,

$$\begin{cases}(5+2s)\underline{x}^3(s)+(3+s)\underline{x}^2(s)+(2+s)\underline{x}(s)-(2+s)=0,\\(9-2s)\bar{x}^3(s)+(5-s)\bar{x}^2(s)+(4-s)\bar{x}(s)-(4-s)=0\end{cases} \tag{5.3}$$

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

$$\begin{cases}7\underline{x}^3(1)+4\underline{x}^2(1)+3\underline{x}(1)=3\\7\bar{x}^3(1)+4\bar{x}^2(1)+3\bar{x}(1)=3\end{cases}\text{ and }\begin{cases}5\underline{x}^3(0)+3\underline{x}^2(0)+2\underline{x}(0)=2\\9\underline{x}^3(0)+5\underline{x}^2(0)+4\underline{x}(0)=4\end{cases} \tag{5.4}$$

We choose the initial guess, $\underline{x}(0) = 0.454737, \bar{x}(0) = 0.475493$ and $\underline{x}(1) = \bar{x}(1) = 0.468209$ and choose the error tolerance 10^{-5} after some significant iterations. We obtained the following numerical solution of (5.2) for the different $\alpha-$ cut values represented by the values of parameter s and the lower bound and upper solution of (5.2) given in Tables 1 and 2. We represent the numerical solution of (5.2) by using our proposed method in Fig. 1.

Example 5.2 The process of converting nitrogen-hydrogen input into ammonia is referred to as Conversion of a fraction. In this case, we consider the temperature and pressure values as triangular fuzzy numbers, which leads to the fuzzy nonlinear equation that is presented below:

$$A_1x^4(s) + A_2x^3(s) + A_3x^2(s) - A_4x(s) = A_5 \tag{5.5}$$

Where, $A_1 = (1, 2.4, 3.8)$, $A_2 = (7.6, 7.8, 8.0)$, $A_3 = (14.4, 14.5, 14.6)$, $A_4 = (2.3, 2.4, 2.5)$, $A_5 = (1.4, 1.6, 1.8)$ are triangular fuzzy numbers.

From this, we write (5.5),

$$(1, 2.4, 3.8)x^4(s) + (7.6, 7.8, 8.0)x^3(s) + (14.4, 14.5, 14.6)x^2(s) + (2.3, 2.4, 2.5)x(s) = (1.4, 1.6, 1.8) \tag{5.6}$$

the parametric form of (5.6),

$$\begin{cases} (1 + 1.4s)\underline{x}^4(s) + (7.6 + 0.2s)\underline{x}^3(s) + (14.4 + 0.1s)\underline{x}^2(s) + (2.3 + 0.1s)\underline{x}(s) - (1.4 + 0.2s) = 0, \\ (3.8 - 1.4s)\bar{x}^4(s) + (8 - 0.2s)\bar{x}^3(s) + (14.6 - 0.1s)\bar{x}^2(s) + (2.5 - 0.1s)\bar{x}(s) - (1.8 - 0.2s) = 0 \end{cases} \tag{5.7}$$

For $s = 0$ and $s = 1$. We get,

$$\begin{cases} 2.4\underline{x}^4(1) + 7.8\underline{x}^3(1) + 14.5\underline{x}^2(1) + 2.4\underline{x}(1) = 1.6 \\ 2.4\bar{x}^4(1) + 7.8\bar{x}^3(1) + 14.5\bar{x}^2(1) + 2.4\bar{x}(1) = 1.6 \end{cases} \tag{5.8}$$

and

$$\begin{cases} \underline{x}^4(0) + 7.6\underline{x}^3(0) + 14.4\underline{x}^2(0) + 2.3\underline{x}(0) = 1.4 \\ 3.8\underline{x}^4(0) + 8\underline{x}^3(0) + 14.6\underline{x}^2(0) + 2.5\underline{x}(0) = 1.8 \end{cases} \tag{5.9}$$

Method	$\underline{x}(s)$	n	$\underline{x}_n$	$f(\underline{x}_n)$	$\lvert\underline{x}_{n+1} - \underline{x}_n\rvert$	CPU Time
PM	$s = 0$	1	0.454737	6.4×10^{-62}	8.1×10^{-63}	0.014
		2	0.454737	0	0	0.014
KPM	$s = 0$	3	0.454737	1.6×10^{-60}	2.1×10^{-61}	0.016
		4	0.454737	0	0	0.016
PM	$s = 0.2$	1	0.458277	2.22045×10^{-16}	0	0.014
		2	0.458277	0	0	0.014
KPM	$s = 0.2$	3	0.458277	2.22045×10^{-16}	0	0.016
		4	0.458277	0	0	0.016
PM	$s = 0.4$	1	0.461302	2.22045×10^{-16}	0	0.014
		4	0.461302	0	0	0.014
KPM	$s = 0.4$	3	0.461302	2.22045×10^{-16}	0	0.016
		4	0.461302	0	0	0.016
PM	$s = 0.6$	1	0.463917	6.66134×10^{-16}	5.55112×10^{-17}	0.014
		2	0.463917	2.22045×10^{-16}	0	0.014
KPM	$s = 0.6$	1	0.463917	6.66134×10^{-16}	5.55112×10^{-17}	0.016
		4	0.463917	2.22045×10^{-16}	0	0.014
PM	$s = 0.8$	1	0.466199	6.66134×10^{-16}	5.55112×10^{-17}	0.014
		2	0.466199	0	0	0.014
KPM	$s = 0.8$	3	0.466199	6.66134×10^{-16}	5.55112×10^{-17}	0.016
		4	0.466199	0	0	0.016
PM	$s = 1$	1	0.468209	1.4×10^{-16}	1.3×10^{-17}	0.014
		2	0.468209	0	0	0.014
KPM	$s = 1$	3	0.468209	4.1×10^{-15}	3.6×10^{-15}	0.016
		4	0.468209	0	0	0.016

Table 1. Comparison of PM and KPM for Lower bound values of example 5.1.

Method	$\bar{x}(s)$	n	$\bar{x}_n$	$\bar{f}(\bar{x}_n)$	$\lvert\bar{x}_{n+1} - \bar{x}_n\rvert$	CPU Time
PM	$s = 0$	1	0.475493	2.7×10^{-61}	1.8×10^{-62}	0.014
		2	0.475493	0	0	0.014
KPM	$s = 0$	3	0.475493	7.0×10^{-60}	4.7×10^{-61}	0.016
		4	0.475493	0	0	0.016
PM	$s = 0.2$	1	0.474317	2.22045×10^{-16}	0	0.014
		2	0.474317	0	0	0.014
KPM	$s = 0.2$	3	0.474317	2.22045×10^{-16}	0	0.016
		4	0.474317	0	0	0.016
PM	$s = 0.4$	1	0.473022	2.22045×10^{-16}	0	0.014
		4	0.473022	0	0	0.014
KPM	$s = 0.4$	3	0.473022	2.22045×10^{-16}	0	0.016
		4	0.473022	0	0	0.016
PM	$s = 0.6$	1	0.471588	2.22045×10^{-16}	0	0.014
		4	0.471588	0	0	0.014
KPM	$s = 0.6$	1	0.471588	2.22045×10^{-16}	0	0.016
		4	0.471588	0	0	0.016
PM	$s = 0.8$	1	0.469994	2.22045×10^{-16}	0	0.014
		2	0.469994	0	0	0.014
KPM	$s = 0.8$	3	0.469994	2.22045×10^{-16}	0	0.014
		4	0.469994	0	0	0.016
PM	$s = 1$	1	0.468209	2.5×10^{-19}	2.2×10^{-20}	0.014
		2	0.468209	0	0	0.014
KPM	$s = 1$	3	0.468209	6.1×10^{-15}	5.4×10^{-15}	0.016
		4	0.468209	0	0	0.016

Table 2. Comparison of PM and KPM for Upper bound values of example 5.1.

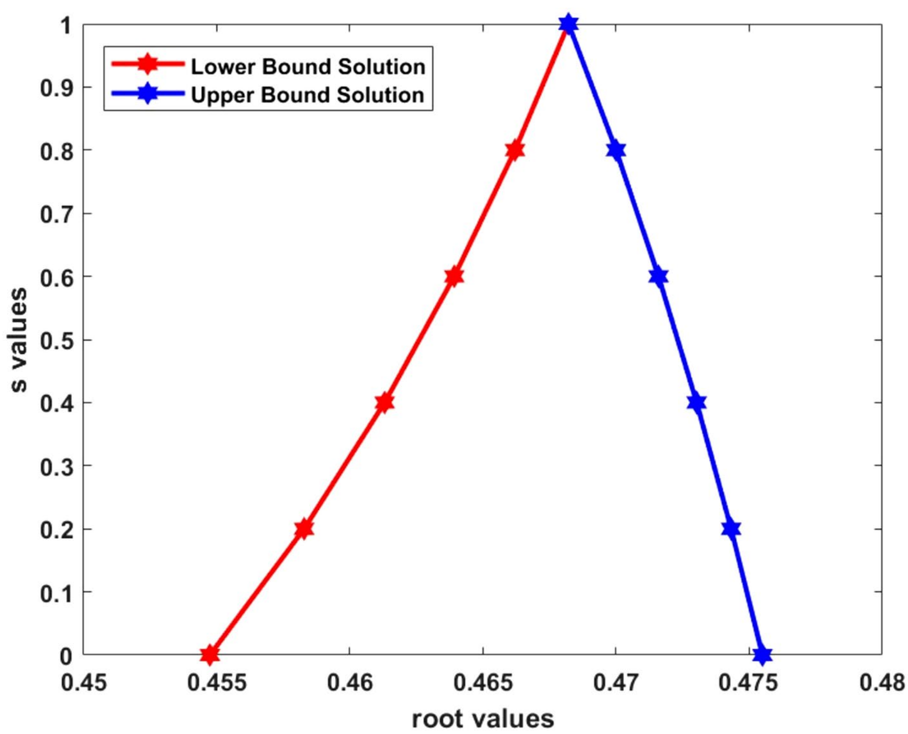

Fig. 1. numerical solution for example 5.1.

Method	$\underline{x}(s)$	n	$\underline{x}_n$	$\underline{f}(\underline{x}_n)$	$\lvert\underline{x}_{n+1}-\underline{x}_n\rvert$	CPU Time
PM	$s=0$	1	0.231369	1.38778×10^{-16}	0	0.014
		2	0.231369	0	0	0.014
KPM	$s=0$	3	0.231369	1.38778×10^{-16}	0	0.016
		4	0.231369	0	0	0.016
PM	$s=0.2$	1	0.234559	1.52656×10^{-16}	2.77556×10^{-17}	0.014
		2	0.234559	2.22045×10^{-16}	2.77556×10^{-17}	0.014
KPM	$s=0.2$	3	0.234559	1.52656×10^{-16}	2.77556×10^{-17}	0.016
		4	0.234559	2.22045×10^{-16}	2.77556×10^{-17}	0.016
PM	$s=0.4$	1	0.237679	2.08167×10^{-16}	2.77556×10^{-17}	0.014
		2	0.237679	5.55112×10^{-17}	0	0.014
KPM	$s=0.4$	3	0.237679	2.08167×10^{-16}	2.77556×10^{-17}	0.016
		2	0.237679	5.55112×10^{-17}	0	0.016
PM	$s=0.6$	1	0.240733	3.60822×10^{-16}	2.77556×10^{-17}	0.014
		4	0.240733	1.38778×10^{-17}	0	0.014
KPM	$s=0.6$	1	0.240733	6.02296×10^{-15}	5.55112×10^{-16}	0.016
		4	0.240733	1.38778×10^{-17}	0	0.016
PM	$s=0.8$	1	0.243722	4.2466×10^{-15}	3.88578×10^{-16}	0.014
		2	0.243722	0	0	0.014
KPM	$s=0.8$	3	0.243722	1.02349×10^{-13}	9.35363×10^{-15}	0.016
		4	0.243722	0	0	0.016
PM	$s=1$	1	0.246649	3.53328×10^{-14}	3.16414×10^{-15}	0.014
		2	0.246649	1.66533×10^{-16}	2.77556×10^{-17}	0.014
KPM	$s=1$	3	0.246649	9.18654×10^{-13}	8.26006×10^{-14}	0.016
		4	0.246649	3.05311×10^{-16}	2.77556×10^{-17}	0.016
		5	0.246649	1.66533×10^{-16}	2.77556×10^{-17}	0.016

Table 3. Comparison of PM and KPM for Lower bound values of example 5.2.

We choose the initial guess as $\underline{x}(0)=0.231369, \bar{x}(0)=0.260423$ and $\underline{x}(1)=\bar{x}(1)=0.246649$ and choose the error tolerance 10^{-5} after some significant iterations. We obtained the following numerical solution of (5.7) for the different $\alpha-$ cut values represented by the values of parameter *s* and the lower bound and upper solution of (5.7) given in Tables 3 and 4. We represent the numerical solution of (5.7) by using our proposed method in Fig. 2.

Conclusion

In this manuscript, we discuss a novel iterative approach for solving FNE, demonstrating a remarkable tenth-order convergence. In this approach, it avoids the need for calculation and inversion of the Jacobian matrix on each iteration. We provided some chemical engineering application problems such as Vander Wall's equation and the fraction of gas conversion to demonstrate the efficiency of the proposed method and compared it with the existing tenth-order iterative method. Our proposed method converges to the solution with less number of iterations and low computational time as compared to the existing method. In error analysis also proposed method also showed dominant behavior compared to existing method. We discussed the results of our proposed method in Tables 1, 2, 3 and 4. Figures 1 and 2 show the results of our introduced method, along with a graphical representation.

Method	$\bar{x}(s)$	n	$\bar{x}_n$	$\bar{f}(\bar{x}_n)$	$\lvert\bar{x}_{n+1} - \bar{x}_n\rvert$	CPU Time
PM	$s = 0$	1	0.260423	2.498×10^{-16}	0	0.014
		2	0.260423	0	0	0.014
KPM	$s = 0$	3	0.260423	3.05311×10^{-15}	0	0.016
		4	0.260423	0	0	0.016
PM	$s = 0.2$	1	0.257777	1.66533×10^{-16}	0	0.014
		2	0.257777	0	0	0.014
KPM	$s = 0.2$	3	0.257777	1.66533×10^{-16}	0	0.016
		4	0.257777	0	0	0.016
PM	$s = 0.4$	1	0.255078	2.77556×10^{-17}	0	0.014
		2	0.255078	0	0	0.014
KPM	$s = 0.4$	3	0.255078	2.77556×10^{-17}	0	0.016
		2	0.255078	0	0	0.016
PM	$s = 0.6$	1	0.252325	1.11022×10^{-16}	0	0.014
		2	0.252325	0	0	0.014
KPM	$s = 0.6$	3	0.252325	8.88178×10^{-16}	5.55112×10^{-17}	0.016
		4	0.252325	1.11022×10^{-16}	0	0.016
PM	$s = 0.8$	1	0.249516	5.82867×10^{-16}	5.55112×10^{-17}	0.014
		2	0.249516	8.32667×10^{-17}	0	0.014
KPM	$s = 0.8$	3	0.249516	1.38223×10^{-16}	1.22125×10^{-15}	0.016
		4	0.249516	8.32667×10^{-17}	0	0.016
PM	$s = 1$	1	0.246649	7.41074×10^{-15}	6.66134×10^{-16}	0.014
		2	0.246649	3.05311×10^{-16}	2.77556×10^{-17}	0.014
		3	0.246649	1.66533×10^{-16}	2.77556×10^{-17}	0.014
KPM	$s = 1$	4	0.246649	1.32006×10^{-13}	1.18794×10^{-14}	0.016
		5	0.246649	3.05311×10^{-16}	2.77556×10^{-17}	0.016
		6	0.246649	1.66533×10^{-16}	2.77556×10^{-17}	0.016

Table 4. Comparison of PM and KPM for Upper bound values of example 5.2.

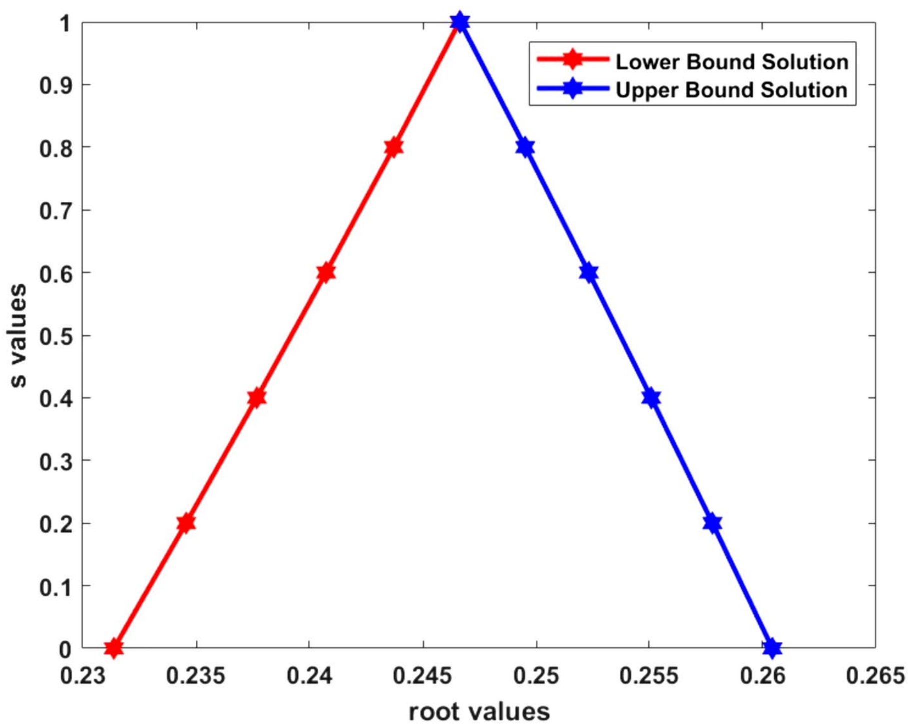

Fig. 2. numerical solution for example 5.2.

Data availability

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Received: 28 August 2024; Accepted: 7 April 2025
Published online: 15 April 2025

References

1. Said Solaiman, O. & Hashim, I. Efficacy of optimal methods for nonlinear equations with chemical engineering applications. *Math. Problems Eng.* **1**, 1728965 (2019).
2. Kwak, D. B., Noh, J. H., Lee, K. S. & Yook, S. J. Cooling performance of a radial heat sink with triangular fins on a circular base at various installation angles. *Int. J. Therm. Sci.* **1**(120), 377–85 (2017).
3. Zadeh, L. A. The concept of a linguistic variable and its application to approximate reasoning-I. *Inf. Sci.* **8**(3), 199–249 (1975).
4. Jafari, R., Yu, W., Razvarz, S. & Gegov, A. Numerical methods for solving fuzzy equations: A survey. *Fuzzy Sets Syst.* **1**(404), 1–22 (2021).
5. Ibrahim, S. M., Mamat, M. & Ghazali, P. L. Shamanskii method for solving parameterized fuzzy nonlinear equations. *Int. J. Optim. Control Theories Appl. (IJOCTA).* **11**(1), 24–9 (2021).
6. Buckley, J. J. & Yunxia, Q. Solving fuzzy equations: a new solution concept. *Fuzzy Sets Syst.* **39**(3), 291–301 (1991).
7. Abbasbandy, S. & Asady, B. Newton's method for solving fuzzy nonlinear equations. *Appl. Math. Comput.* **159**(2), 349–56 (2004).
8. Abbasbandy, S. & Ezzati, R. Newton's method for solving a system of fuzzy nonlinear equations. *Appl. Math. Comput.* **175**(2), 1189–99 (2006).
9. Abbasbandy, S. & Jafarian, A. Steepest descent method for solving fuzzy nonlinear equations. *Appl. Math. Comput.* **174**(1), 669–75 (2006).
10. Ramli, A., Abdullah, M. L. & Mamat, M. Broyden s method for solving fuzzy nonlinear equations. *Adv. Fuzzy Syst.* **2010**(1), 763270 (2010).
11. Sulaiman, I. M., Mamat, M., Malik, M., Nisar, K. S. & Elfasakhany, A. Performance analysis of a modified Newton method for parameterized dual fuzzy nonlinear equations and its application. *Res. Phys.* **1**(33), 105140 (2022).
12. Moi, S., Biswas, S. & Sarkar, S. P. A new collocation method for fuzzy singular integro-differential equations. *Int. J. Appl. Comput. Math.* **8**, 65 (2022).
13. Moi, S., Biswas, S. & Sarkar, S. P. A Lagrange spectral collocation method for weakly singular fuzzy fractional Volterra integro-differential equations. *Soft Comput.* **27**(8), 4483–4499 (2023).
14. Moi, S., Biswas, S. & Sarkar, S. P. Finite-difference method for fuzzy singular integro-differential equation deriving from fuzzy non-linear differential equation. *Granular Computi.* **8**, 503–524 (2023).
15. Moi, S., Biswas, S. & Pal, S. Second-order neutrosophic boundary-value problem. *Complex Intell. Syst.* **7**, 1079–1098 (2021).
16. Moi, S., Biswas, S. & Pal, S. Neutrosophic linear differential equation with a new concept of neutrosophic derivative. In neutrosophic operational research: methods and applications. (2020).
17. Biswas, S. & Roy, T. K. A semianalytical method for fuzzy integro-differential equations under generalized seikkala derivative. *Soft Comput.* **23**, 7959–7975 (2019).
18. Thota, S. & Shanmugasundaram, P. On new sixth and seventh order iterative methods for solving non-linear equations using homotopy perturbation technique. *BMC. Res. Notes* **15**(1), 267 (2022).
19. Kansal, M., Cordero, A., Bhalla, S. & Torregrosa, J. R. New fourth-and sixth-order classes of iterative methods for solving systems of nonlinear equations and their stability analysis. *Numerical Algorithms.* **87**, 1017–60 (2021).
20. Ababneh, O. New iterative methods for solving nonlinear equations and their basins of attraction. *WSEAS Trans. Math. Link Disabl.* **21**, 9–16 (2022).
21. Nadeem, G. A., Aslam, W. & Ali, F. An optimal fourth-order second derivative free iterative method for nonlinear scientific equations: 10.48129/kjs. 18253. *Kuwait J. Sci.* **10**, 50 (2023).
22. Sivakumar, P. & Jayaraman, J. Some new higher order weighted Newton methods for solving nonlinear equation with applications. *Math. Comput. Appl.* **24**(2), 59 (2019).
23. Maroju, P., Magreñán, Á. A., Sarría, Í. & Kumar, A. Local convergence of fourth and fifth order parametric family of iterative methods in Banach spaces. *J. Math. Chem.* **58**, 686–705 (2020).
24. Abdul-Hassan, N. Y., Ali, A. H. & Park, C. A new fifth-order iterative method free from second derivative for solving nonlinear equations. *J. Appl. Math. Comput.* **1**, 1 (2022).
25. Alqahtani, H. F., Behl, R. & Kansal, M. Higher-order iteration schemes for solving nonlinear systems of equations. *Mathematics* **7**(10), 937 (2019).
26. Sharma, E. & Panday S. Efficient sixth order iterative method free from higher derivatives for nonlinear equations. J. Math. Comput. Sci.. 2022 Mar 1;12:Article-ID.
27. Padilla, J. J., Chicharro, F. I., Cordero, A., Hernández-Díaz, A. M. & Torregrosa, J. R. A Class of Efficient Sixth-Order Iterative Methods for Solving the Nonlinear Shear Model of a Reinforced Concrete Beam. *Mathematics* **12**(3), 499 (2024).
28. Alshomrani, A. S., Behl, R. & Maroju, P. Local convergence of parameter based method with six and eighth order of convergence. *J. Math. Chem.* **58**, 841–853 (2020).
29. Fang, L., Sun, L. & He, G. An efficient Newton-type method with fifth-order convergence for solving nonlinear equations. *Comput. Appl. Math.* **27**, 269–274 (2008).
30. Farahmand Nejad, & Maryam, et al. Gröbner basis approach for solving fuzzy complex system of linear equations. New Mathematics and Natural Computation 1-18 (2023).
31. Behera, D. & Chakraverty, S. New approach to solve fully fuzzy system of linear equations using single and double parametric form of fuzzy numbers. *Sadhana* **40**, 35–49 (2015).
32. Devi, K. & Maroju, P. Local convergence study of tenth-order iterative method in Banach spaces with basin of attraction. *AIMS Math* **9**, 6648–6667 (2024).

Additional information

Correspondence and requests for materials should be addressed to P.M.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.