Numerical methods for solving fuzzy equations: A Survey

Raheleh Jafari[a,*], Wen Yu[b], Sina Razvarz[b], Alexander Gegov[c]

[a] *Centre for Artificial Intelligence Research (CAIR), University of Agder, Grimstad, Norway*
[b] *Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute), Mexico City, Mexico*
[c] *School of Computing, University of Portsmouth, Buckingham Building, Portsmouth PO1 3HE, UK*

Abstract

In this paper, we study different numerical methods for solving fuzzy equations, dual fuzzy equations, fuzzy differential equations (FDEs) and fuzzy partial differential equations (PDEs). In this study, conditions that guarantee the existence of the roots of these equations are discussed. Also, this paper provides some discussion about the rates of convergence of each of the numerical methods. Finally, some numerical examples are given to illustrate the efficiency of these methods.

Keywords: nonlinear systems, fuzzy number, fuzzy solution

1. Introduction

The study of fuzzy equations has attracted the interest of many researchers in the past few years [1][2][3]. Fuzzy equations are known as perfect mathematical modeling of real-world problems whereby uncertainty exists. Fuzzy equations are the equations whose parameters can be varied from the form of the fuzzy set [4]. When the parameters or states of the differential equations are vague, they can as well be modeled with FDEs.

The solutions of the fuzzy equations can be implemented directly for modeling as well as nonlinear control. Some of the problems related to applying

*Corresponding author
Email address: raheleh.jafari@uia.no (Raheleh Jafari)

finite-dimensional state models in designing control laws for distributed-mass systems are discussed in [5]. In [6] Newton's method is proposed for solving fuzzy nonlinear equations. In [7] the fixed point method for solving fuzzy nonlinear systems is suggested. The analytical solution of a fuzzy heat equation under generalized Hukuhara partial differentiability by fuzzy Fourier transform is investigated in [8]. In [9] the uniqueness and stability of the solution for fuzzy Poisson equation are discussed using the fuzzy maximum principle. The numerical solutions of the fuzzy equations can be obtained using the iterative method [10], the interpolation method [11] and the Runge-Kutta method [12].

Some numerical methods, like the Nystrom method [13] and the Runge-Kutta method [14] can be used to solve FDE. In [15] the Euler method is used to obtain the approximate solution of the fuzzy initial value problem. In [16] the Laplace transform method is used to obtain the solutions of second-order FDE. In [17] the compound $(\frac{G'}{G})$-expansion method is proposed to construct the multiple non-traveling wave solutions of nonlinear PDEs. In [18] positive or negative solutions to first-order fully fuzzy linear differential equations under generalized differentiability are studied. In [19] concreted solutions to fuzzy linear fractional differential equations under Riemann-Liouville H-differentiability is studied.

Artificial neural networks can also be used for solving fuzzy equations. In [20] fuzzy quadratic equations are solved using artificial neural networks. In [21] fuzzy polynomial equations are solved using artificial neural networks. In [22] dual fuzzy equations are solved using artificial neural networks. In [23] a method based on fuzzy neural network is proposed for approximate solution of fully fuzzy matrix equations. Nevertheless, these methods can not solve general fuzzy equations with artificial neural networks. Furthermore, they can not produce fuzzy coefficients directly with the artificial neural networks [24]. In [25] artificial neural network method is proposed for solving FDEs with initial conditions. In [26] an unsupervised adaptive network-based fuzzy inference system model is proposed for solving differential equations. In [27] neural algorithms are used for solving differential equations. In [28] artificial neural networks are used for solving PDEs. In [29] multi-layer artificial neural networks are used to solve a class of first-order PDEs. In [30] an unsupervised artificial neural network is proposed for solving differential equations. In [31] artificial neural network is used for finding the solution of boundary control problem for the heat equation.

In this paper, a survey is given of recent numerical methods for solving fuzzy equations, dual fuzzy equations, FDE and fuzzy PDE. In this study, it is discussed in detail that the roots of these equations can be obtained with different methods. Conditions that guarantee the existence of the roots of these equations are discussed. Furthermore, the advantages of numerical methods in terms of precision are illustrated. The remaining of the article is organized as follows. In Section 2, some basic definitions used in the rest of the paper are given. Section 3 discusses some numerical methods for finding the solutions of fuzzy equations and dual fuzzy equations. Section 4 discusses some numerical techniques for finding the solutions of FDEs and fuzzy PDEs. Section 5 presents numerical examples with comparative analysis. Section 6 concludes the paper.

2. Mathematical preliminaries

The following definitions are used in this paper.

Definition 1 (fuzzy variable). If x is: 1) normal, there exists $\zeta_0 \in \mathbb{R}$ in such a manner that $x(\zeta_0) = 1$; 2) convex, $x\left[\lambda\zeta + (1-\lambda)\xi\right] \geq \min\{x(\zeta), x(\xi)\}$, $\forall \zeta, \xi \in \mathbb{R}, \forall \lambda \in [0,1]$; 3) upper semi-continuous on $\mathbb{R}$, $x(\zeta) \leq x(\zeta_0) + \varepsilon$, $\forall \zeta \in N(\zeta_0)$, $\forall \zeta_0 \in \mathbb{R}$, $\forall \varepsilon > 0$, $N(\zeta_0)$ is a neighborhood; or 4) $x^+ = \{\zeta \in \mathbb{R}, x(\zeta) > 0\}$ is compact, then x is a fuzzy variable, and the fuzzy set is defined as E, $x \in E : \mathbb{R} \to [0,1]$.

The fuzzy variable x can also be represented as

$$x = A\left(\underline{x}, \bar{x}\right) \tag{1}$$

where $\underline{x}$ is the lower-bound variable, $\bar{x}$ is the upper-bound variable, and A is a continuous function. The membership functions are utilized to implicate the fuzzy variable x. The best known membership functions are the triangular function

$$x(\zeta) = F(a,b,c) = \begin{cases} \frac{\zeta-a}{b-a} & a \le \zeta \le b \\ \frac{c-\zeta}{c-b} & b \le \zeta \le c \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and trapezoidal function

$$x(\zeta) = F(a,b,c,d) = \begin{cases} \frac{\zeta-a}{b-a} & a \le \zeta \le b \\ \frac{d-\zeta}{d-c} & c \le \zeta \le d \\ 1 & b \le \zeta \le c \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The fuzzy variable x that contains the dimension of ζ is dependent on the membership functions, where (2) includes three variables and (3) includes four variables. To demonstrate the consistency of operations, the application initially lies within the $\alpha-$level operation of the fuzzy number.

Definition 2 (fuzzy number). A fuzzy number x associates with a real value with α-level as

$$[x]^\alpha = \{a \in \mathbb{R} : x(a) \ge \alpha\} \tag{4}$$

where $0 < \alpha \le 1$, $x \in E$.

If $x, y \in E$, $\lambda \in \mathbb{R}$, the fuzzy operations are as follows:

Sum,

$$[x \oplus y]^\alpha = [x]^\alpha + [y]^\alpha = [\underline{x}^\alpha + \underline{y}^\alpha, \bar{x}^\alpha + \bar{y}^\alpha] \tag{5}$$

subtract,

$$[x \ominus y]^\alpha = [x]^\alpha - [y]^\alpha = [\underline{x}^\alpha - \underline{y}^\alpha, \bar{x}^\alpha - \bar{y}^\alpha] \tag{6}$$

or multiply,

$$\underline{z}^\alpha \le [x \odot y]^\alpha \le \bar{z}^\alpha \text{ or } [x \odot y]^\alpha = A(\underline{z}^\alpha, \bar{z}^\alpha) \tag{7}$$

where $\underline{z}^\alpha = \underline{x}^\alpha \underline{y}^1 + \underline{x}^1 \underline{y}^\alpha - \underline{x}^1 \underline{y}^1$, $\bar{z}^\alpha = \bar{x}^\alpha \bar{y}^1 + \bar{x}^1 \bar{y}^\alpha - \bar{x}^1 \bar{y}^1$, and $\alpha \in [0,1]$.

Therefore, $[x]^0 = x^+ = \{\zeta \in \mathbb{R}, x(\zeta) > 0\}$. Since $\alpha \in [0,1]$, $[x]^\alpha$ is a bounded interval such that $\underline{x}^\alpha \leq [x]^\alpha \leq \bar{x}^\alpha$. The α-level of x between $\underline{x}^\alpha$ and $\bar{x}^\alpha$ is given as

$$[x]^\alpha = A\left(\underline{x}^\alpha, \bar{x}^\alpha\right) \tag{8}$$

2.1. Applying the solutions of fuzzy equations to nonlinear systems

Consider the following unknown discrete-time nonlinear system

$$\bar{x}_{k+1} = \bar{f}\left(\bar{x}_k, u_k\right), \quad y_k = \bar{g}\left(\bar{x}_k\right) \tag{9}$$

where $u_k \in \Re^u$ is the input vector, $\bar{x}_k \in \Re^l$ is an internal state vector, and $y_k \in \Re^m$ is the output vector. $\bar{f}$ and $\bar{g}$ are general nonlinear smooth functions $\bar{f}, \bar{g} \in C^\infty$. Denoting $Y_k = \left(y_{k+1}^T, y_k^T, \cdots\right)^T$, $U_k = \left(u_{k+1}^T, u_k^T, \cdots\right)^T$. If $\frac{\partial Y}{\partial \bar{x}}$ is non-singular at $\bar{x} = 0$, $U = 0$, this leads to the following model

$$y_k = \Phi(y_{k-1}^T, y_{k-2}^T, \cdots u_k^T, u_{k-1}^T, \cdots) \tag{10}$$

where $\Phi(\cdot)$ is an unknown nonlinear difference equation representing the plant dynamics, u_k and y_k are measurable scalar input and output. The nonlinear system (9) is a NARMA model. We can also regard the input of the nonlinear system as

$$x_k = (y_{k-1}^T, y_{k-2}^T, \cdots u_k^T, u_{k-1}^T, \cdots)^T \tag{11}$$

the output as y_k.

Many nonlinear systems as in (9) can be rewritten as the following linear-in-parameter model,

$$y_k = \sum_{i=1}^{n} a_i f_i\left(x_k\right) \tag{12}$$

or

$$y_k + \sum_{i=1}^{m} b_i g_i(x_k) = \sum_{i=1}^{n} a_i f_i\left(x_k\right) \tag{13}$$

where a_i and b_i are linear parameters, $f_i\left(x_k\right)$ and $g_i(x_k)$ are nonlinear functions. The variables of these functions are measurable input and output. A famous example of this kind of model is the robot manipulator

$$M\left(q\right)\ddot{q} + C\left(q, \dot{q}\right)\dot{q} + B\dot{q} + g\left(q\right) = \tau \tag{14}$$

(14) can be rewritten as

$$\sum_{i=1}^{n} Y_i \left(q, \dot{q}, \ddot{q}\right) \theta_i = \tau \tag{15}$$

To identify or control the linear-in-parameter systems (12), (13) or (15), the normal least square or adaptive methods can be applied directly.

In this paper, we consider the uncertain nonlinear systems, *i.e.*, the parameters a_i, b_i or θ_i are not fixed (not crisp). They are uncertain in the sense of fuzzy logic. The uncertain nonlinear systems are modeled by linear-in-parameter models with fuzzy parameters. These models are called fuzzy equations.

Remark. There are several extensions of an equation to a fuzzy equation where the coefficients are fuzzy intervals [32]. To extend an equation to a fuzzy equation, the interval definitions [33] apply to all α-cuts. To calculate the membership function of the set of solutions, the α-cut of the solution sets can be defined by a transformation of the α-cuts of the fuzzy coefficients, that is, the α-cuts of the function (transformation) is the function of the α-cuts of its fuzzy arguments [34].

For the uncertain nonlinear system (9), we use the following two types of fuzzy equations to model it

$$y_k = a_1 f_1(x_k) \oplus a_2 f_2(x_k) \oplus ... \oplus a_n f_n(x_k) \tag{16}$$

or

$$\begin{aligned} & a_1 f_1(x_k) \oplus a_2 f_2(x_k) \oplus ... \oplus a_n f_n(x_k) \\ & = b_1 g_1(x_k) \oplus b_2 g_2(x_k) \oplus ... \oplus b_m g_m(x_k) \oplus y_k \end{aligned} \tag{17}$$

Because a_i and b_i are fuzzy numbers, we use the fuzzy operation $\oplus$. (17) has more general form than (16), it is called dual fuzzy equation.

In a special case, $f_i(x_k)$ has polynomial form,

$$a_1 x_k \oplus ... \oplus a_n x_k^n = b_1 x_k \oplus ... \oplus b_n x_k^n \oplus y_k \tag{18}$$

(18) is called dual fuzzy polynomial. If we use the dual fuzzy polynomial (18) to model a nonlinear function

$$z_k = f(x_k) \tag{19}$$

so the object is to minimize error between the two output y_k and z_k. Since y_k is a fuzzy number and z_k is a crisp number, we use the maximum of all points as the modeling error

$$\max_k |y_k - z_k| = \max_k |y_k - f(x_k)| = \max_k |\beta_k| \tag{20}$$

where $y_k = F\left(a\left(k\right), b\left(k\right), c\left(k\right)\right)$, $\beta_k = F\left(\beta_1, \beta_2, \beta_3\right)$, which are defined in (2).

In [35], we use the dual fuzzy equation (17) to model the uncertain nonlinear system (9). The controller design process is to find u_k, such that the output of the plant y_k can follow desired output y_k^*, or the trajectory tracking error is minimized

$$\min_{u_k} \|y_k - y_k^*\| \tag{21}$$

This control object can be considered as: finding a solution u_k for the following dual fuzzy equation

$$\begin{aligned} &a_1 f_1(x_k) \oplus a_2 f_2(x_k) \oplus ... \oplus a_n f_n(x_k) \\ &= b_1 g_1(x_k) \oplus b_2 g_2(x_k) \oplus ... \oplus b_m g_m(x_k) \oplus y_k^* \end{aligned} \tag{22}$$

where $x_k = [y_{k-1}^T, y_{k-2}^T, \cdots u_k^T, u_{k-1}^T, \cdots]^T$.

The uncertain nonlinear system can also be modeled by PDEs, such as

$$\frac{\partial^2 \zeta(x,t)}{\partial t^2} + \frac{2}{t}\frac{\partial \zeta(x,t)}{\partial t} = F(x, \zeta(x,t), \frac{\partial \zeta(x,t)}{\partial x}, \frac{\partial^2 \zeta(x,t)}{\partial x^2}) \tag{23}$$

in which t and x are independent variables, ζ is the dependent variable, F is a nonlinear function of x, ζ, ζ_x and ζ_{xx}, also the initial conditions for the PDE (23) are illustrated as below

$$\zeta(x,0) = f(x), \quad \zeta_t(x,0) = g(x) \tag{24}$$

The following FDE can be used to model the uncertain nonlinear system (9),

$$\frac{d}{dt}x = f(x,u) \tag{25}$$

where x is the fuzzy variable that corresponds to the state x_k in (9), $f(t,x)$ is a fuzzy vector function that relates to $f_1(x_k, u)$, and $\frac{d}{dt}x$ is the fuzzy derivative.

Definition 3 (fuzzy derivative). The fuzzy derivative of f at x_0 is expressed as

$$\frac{d}{dt}f(x_0) = \lim_{h\to 0}\frac{1}{h}[f(x_0+h)\ominus_{gH} f(x_0)] \tag{26}$$

where $\ominus_{gH}$ is the Hukuhara difference [36], defined by

$$x \ominus_{gH} y = z \Longleftrightarrow \begin{cases} \quad 1)\ x = y \oplus z \\ \text{or } 2)\ y = x \oplus (-1)z \end{cases} \tag{27}$$

The $\alpha-$level of the fuzzy derivative is

$$f(x,\alpha) = [\underline{f}(x,\alpha), \overline{f}(x,\alpha)] \tag{28}$$

where $x \in E$ for each $\alpha \in [0,1]$.

If we apply the $\alpha-$level (8) to $f(x,\alpha)$ in (28)

$$[x \ominus_{gH} y]^\alpha = [\min\{\underline{x}^\alpha - \underline{y}^\alpha, \bar{x}^\alpha - \bar{y}^\alpha\}, \max\{\underline{x}^\alpha - \underline{y}^\alpha, \bar{x}^\alpha - \bar{y}^\alpha\}] \tag{29}$$

then, we obtain two functions: $\underline{f}\left[u, \underline{x}(\zeta,\alpha), \bar{x}(\zeta,\alpha)\right]$ and $\overline{f}\left[u, \underline{x}(\zeta,\alpha), \bar{x}(\zeta,\alpha)\right]$. Thus, the fuzzy differential equation (25) can be equivalent to the following four ordinary differential equations (ODE)

$$\begin{cases} \frac{d}{dt}\underline{x}(\alpha) = \underline{f}\left[u, \underline{x}(\zeta,\alpha), \overline{x}(\zeta,\alpha)\right] \\ \frac{d}{dt}\overline{x}(\alpha) = \overline{f}\left[u, \underline{x}(\zeta,\alpha), \bar{x}(\zeta,\alpha)\right] \end{cases}$$
$$or \begin{cases} \frac{d}{dt}\underline{x}(\alpha) = \overline{f}\left[u, \underline{x}(\zeta,\alpha), \overline{x}(\zeta,\alpha)\right] \\ \frac{d}{dt}\overline{x}(\alpha) = \underline{f}\left[u, \underline{x}(\zeta,\alpha), \overline{x}(\zeta,\alpha)\right] \end{cases} \tag{30}$$

So for whatever purposes, such as modeling and control of nonlinear systems, or analysis of uncertainty dynamic, we need solutions of the algebraic fuzzy equations and the FDEs. Since it is impossible to obtain analytical solutions, numerical methods are used to solve these fuzzy equations.

3. Numerical methods for solving algebraic fuzzy equations

There are not any analytical solution for algebraic fuzzy equations with degree greater than 3. Therefore, numerical methods are required for finding the roots of such equations. In this section, five different important techniques are illustrated to solve fuzzy equations and dual fuzzy equations.

3.1. Newton technique

In 1671, Isaac Newton proposed a new algorithm [37] to resolve a polynomial equation that was represented based on an example like $z^3 - 2z - 5 = 0$. To find an exact root of the mentioned equation, at first an initial value is presumed, such that $z \approx 2$. By presuming $z = 2 + p$ and replacing it into the original equation, the outcome is acquired as $p^3 + 6p^2 + 10p - 1 = 0$. Since p is supposed to be small, $p^3 + 6p^2$ is neglected compared to $10p - 1$. The previous equation produces $p \approx 0.1$, therefore an excellent approximation of the root is $z \approx 2.1$. The repeat of this procedure is easy and $p = 0.1 + c$ is obtained. The replacement produces $c^3 + 6.3c^2 + 11.23c + 0.061 = 0$, so $c \approx -0.061/11.23 = -0.0054...$, hence a new estimation of the root is $z \approx 2.0946$. It is essential to repeat the procedure until the expected number of digits is obtained. In his technique, Newton did not obviously apply the concept of derivative, he only implemented it on polynomial equations.

Newton's technique is suggested in [38] for solving fuzzy nonlinear equations instead of standard analytical techniques since they are not suitable everywhere. However, just a positive root of the fuzzy nonlinear equation has been acquired in [38], even though a negative solution can also exist. We believe that the root of this problem is the interpretation of interval as well as fuzzy extensions.

Example 3.1.1. Let us consider the following fuzzy equation

$$a_1 z^2 + a_2 z = a_3 \tag{31}$$

where $a_1 = (3, 3, 4, 5), a_2 = (1, 2, 3), a_3 = (1, 1, 2, 3)$. The positive fuzzy solution of fuzzy equation (31) acquired in [38] is shown in Figure 1.

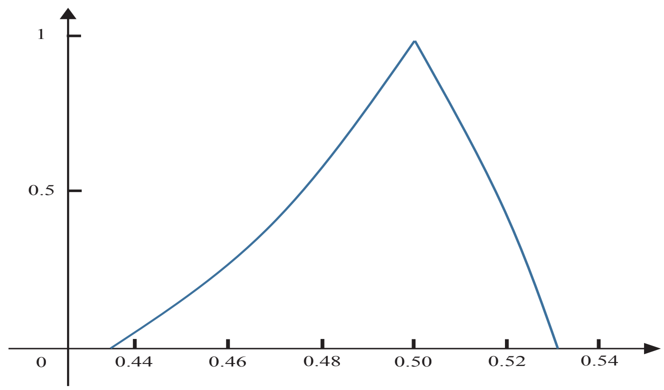

Fig. 1. Positive fuzzy solution acquired in [38]

Even though it is declared in [38] that (31) does not have negative fuzzy root, we find such root. For $\alpha = 0$, the fuzzy equation (31) can be represented as

$$(3,5)z^2 + (1,3)z = (1,3) \tag{32}$$

For $z > 0$, i.e., $\underline{z}, \overline{z} > 0$, from (32) we get $\underline{z} = 0.4343$, $\overline{z} = 0.5307$. However, in the case of $z < 0$, i.e., $\underline{z}, \overline{z} < 0$, from (32) we get $\underline{z} \cong -0.629$, $\overline{z} \cong -0.98$, and since $\underline{z} > \overline{z}$, so a negative root does not exist [38]. For clarifying the source of this problem, we take into consideration a simple interval linear equation $a_1 z = a_2$, where a_1, a_2 are intervals. Applying conventional interval arithmetic rules [39] we obtain $(\underline{a}_1\underline{z}, \overline{a}_1\overline{z}) = (\underline{a}_2, \overline{a}_2)$, hence, $\underline{z} = \frac{\underline{a}_2}{\underline{a}_1}, \overline{z} = \frac{\overline{a}_2}{\overline{a}_1}$. Let $a_1 = (3,4), a_2 = (1,2)$, from $\underline{z} = \frac{\underline{a}_2}{\underline{a}_1}, \overline{z} = \frac{\overline{a}_2}{\overline{a}_1}$ we obtain $\underline{z} = 0.333, \overline{z} = 0.5$, for $a_1 = (1,2), a_2 = (3,4)$, we obtain $\underline{z} = 3, \overline{z} = 2$, for $a_1 = (3,4), a_2 = (0.7, 0.8)$, we obtain $\underline{z} = 0.23, \overline{z} = 0.2$. We can see that the solution of interval equation $a_1 z = a_2$ exists just in some particular conditions. For solving this problem in the case of nonlinear interval and fuzzy equations, we suggest interval extended zero technique [40]. In order to use interval extended zero technique we present (31) on each $\alpha-$cut as below

$$(\underline{a}_1, \overline{a}_1)(\underline{z}, \overline{z})^2 + (\underline{a}_2, \overline{a}_2)(\underline{z}, \overline{z}) - (\underline{a}_3, \overline{a}_3) = (-w, w) \tag{33}$$

such that w is taken to be the undefined parameter and also index α is deleted for the easiness. Applying conventional interval arithmetic rules to (33), the following is extracted

$$(\underline{a}_1\underline{z}+\underline{a}_2,\overline{a}_1\overline{z}+\overline{a}_2)(\underline{z},\overline{z})-(\underline{a}_3,\overline{a}_3)=(-w,w) \tag{34}$$

Using interval extended zero technique the positive and negative roots of fuzzy equation (34) can be acquired. For negative case from (34) we have,

$$\underline{a}_1\overline{z}^2+\overline{a}_2\underline{z}-\overline{a}_3=-w, \quad \overline{a}_1\underline{z}^2+\underline{a}_2\overline{z}-\underline{a}_3=w \tag{35}$$

The sum of two equations in (35) leads to the below relation

$$\underline{a}_1\overline{z}^2+\overline{a}_2\underline{z}-\overline{a}_3+\overline{a}_1\underline{z}^2+\underline{a}_2\overline{z}-\underline{a}_3=0 \tag{36}$$

Let z_k be the real valued solution of (36) in such a way that it is taken to be the natural top boundary for negative $\underline{z}$, i.e., $\underline{z} \leq z_k$ and bottom boundary for negative $\overline{z}$, i.e., $z_k \leq \overline{z}$. For $\underline{z}=\overline{z}=z_k$ we have

$$z_k=\frac{-(\underline{a}_2+\overline{a}_2)-\sqrt{(\underline{a}_2+\overline{a}_2)^2+4(\underline{a}_1+\overline{a}_1)(\underline{a}_3+\overline{a}_3)}}{2(\underline{a}_1+\overline{a}_1)} \tag{37}$$

The interval solution of (35) can be obtained as

$$z_{min}=\frac{-\overline{a}_2-\sqrt{\overline{a}_2{}^2+4\overline{a}_1\overline{a}_3}}{2\underline{a}_1}, \quad z_{max}=\frac{-\underline{a}_2-\sqrt{\underline{a}_2{}^2+4\underline{a}_1\underline{a}_3}}{2\overline{a}_1} \tag{38}$$

From (36) we have

$$\underline{z}=\underline{g}(\overline{z})=\frac{-\overline{a}_2-\sqrt{\overline{a}_2{}^2+4\overline{a}_1(\underline{a}_3+\overline{a}_3-\underline{a}_1\overline{z}^2-\underline{a}_2\overline{z})}}{2\overline{a}_1} \tag{39}$$

and

$$\overline{z}=\overline{g}(\underline{z})=\frac{-\underline{a}_2-\sqrt{\underline{a}_2{}^2+4\underline{a}_1(\underline{a}_3+\overline{a}_3-\overline{a}_1\underline{z}^2-\overline{a}_2\underline{z})}}{2\overline{a}_1} \tag{40}$$

In general, the interval solution of the above constraint satisfaction problem can be presented as below,

$$[\underline{z}]=[z_{\min},z_k]\cap[\underline{z}_1^*,\underline{z}_2^*], \quad [\overline{z}]=[z_k,z_{\max}]\cap[\overline{z}_1^*,\overline{z}_2^*] \tag{41}$$

where $\underline{z}_1^*=\min \underline{g}(\overline{z})$, $\underline{z}_2^*=\max \underline{g}(\overline{z})$ $(z_k \leq \overline{z} \leq z_{\max})$; $\overline{z}_1^*=\min \overline{g}(\underline{z})$, $\overline{z}_2^*=\max \overline{g}(\underline{z})$ $(z_{\max} \leq \underline{z} \leq z_k)$ Using the suggested technique, the positive solution of fuzzy equation (34) can be acquired as well.

Newton's technique is comparatively costly, as the computation of the Hessian on the first iteration is required. Therefore, the analytic explanation for the second derivative is usually complex or intractable, need lots of calculation. Steepest descent technique applies merely first-order information and never deals with estimating second derivatives.

3.2. Steepest descent technique

In [41], the steepest descent method is used for obtaining the solution of fuzzy nonlinear equation $F(y) = 0$, where the fuzzy quantities are shown in parametric form. The equation is presented by parametric form as follows

$$\begin{cases} \underline{F}(\underline{y}^{\alpha}, \overline{y}^{\alpha}) = 0 \\ \overline{F}(\underline{y}^{\alpha}, \overline{y}^{\alpha}) = 0 \end{cases} \tag{42}$$

The function $H : \Re^2 \to \Re$ is defined as

$$H(\underline{y}, \overline{y}) = [\underline{F}(\underline{y}^{\alpha}, \overline{y}^{\alpha}), \overline{F}(\underline{y}^{\alpha}, \overline{y}^{\alpha})]^2 \tag{43}$$

Steepest descent method determines a local minimum for two-variable function H. Steepest descent method can be illustrated as follows:

1. Evaluate H at an initial estimation $Y_0^{\alpha} = (\underline{y}_0^{\alpha}, \overline{y}_0^{\alpha})$.

2. Define a direction from $Y_0^{\alpha} = (\underline{y}_0^{\alpha}, \overline{y}_0^{\alpha})$ which decreases the value of H.

3. Move a proper amount in this direction and name the recent value $Y_1^{\alpha} = (\underline{y}_1^{\alpha}, \overline{y}_1^{\alpha})$.

4. Repeat steps 1 through 3 using Y_0^{α} substituting Y_1^{α}.

Using the same fuzzy equation (31) as in example 3.1.1, the steepest descent technique [41] gets the positive fuzzy solution. The positive fuzzy solution of the fuzzy equation (31) acquired in [41] is shown in Figure 2.

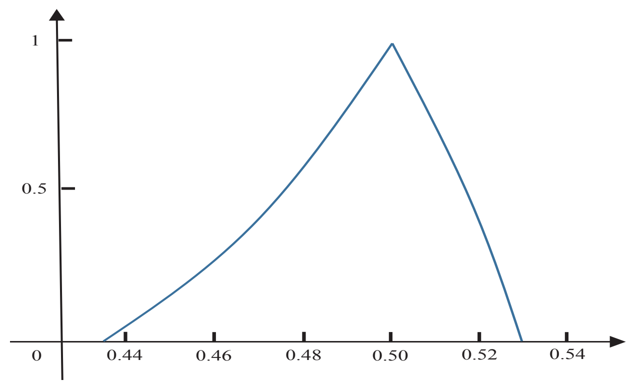

Fig. 2. Positive fuzzy solution acquired in [41]

Even though it is declared in [41] that (31) does not have negative fuzzy root, interval extended zero technique makes it feasible to obtain both the positive and negative roots of fuzzy equation (31). The positive fuzzy solution of the fuzzy equation (31) acquired with the use of interval extended zero technique is shown in Figure 3. The negative fuzzy solution of the fuzzy equation (31) acquired with the use of interval extended zero technique is shown in Figure 4.

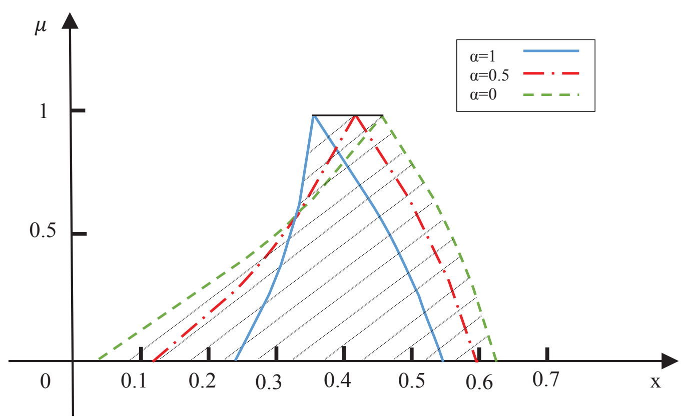

Fig. 3. Positive fuzzy solution acquired with the use of interval extended zero technique

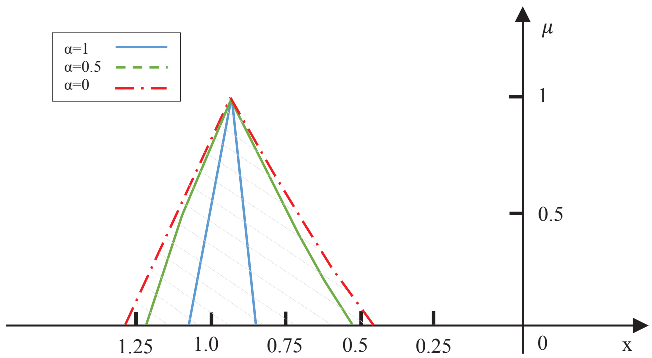

Fig. 4. Negative fuzzy solution acquired with the use of interval extended zero technique

The steepest descent method approaches merely linearly to the solution, however, generally, it approaches even for weak initial estimations [42]. Although the steepest descent method does not require a good initial value, its disadvantage is having a slow convergence speed. The genetic algorithm provides a fast convergence to nearly optimal solutions in many kinds of problems. Genetic algorithm technique has higher training performance compared with the steepest descent technique.

3.3. Genetic algorithm technique

Resolving fuzzy equations can be considered as one of the basic problems in fuzzy set theory. Let us take into consideration the algebraic expression cz^2+dz, where c and d are real parameters, also z is a real variable. By substituting the fuzzy variable Z, fuzzy numbers C, and D into cz^2+dz for z, c, and d, respectively, we obtain CZ^2+DZ. There exist two main traditional fuzzy techniques to evaluate the fuzzy expression CZ^2+DZ. The first technique to obtain the value of CZ^2+DZ is utilizing the extension principle and the second technique is utilizing interval arithmetic and $\alpha-$cuts. Evaluating fuzzy algebraic expression CZ^2+DZ utilizing interval arithmetic and $\alpha-$cuts yields a larger fuzzy set compared to utilizing the extension principle. Another drawback of traditional fuzzy techniques is that resolving algebraic fuzzy equations is too complex because of the shortage of inverse operators and also the multiple incidences of parameters in an expression may cause in a high inaccuracy [43]. In [44] the genetic algorithm method is used for resolving fuzzy algebraic equations without using membership functions for fuzzy numbers. Furthermore, the presented genetic algorithm does not use the extension principle, interval arithmetic, the $\alpha-$cut operations for fuzzy calculations, and the penalty approach for constraint violations. The suggested genetic algorithm technique simulates a fuzzy number by spreading it into specified partition points. Afterward, the genetic algorithm is implemented for evolving the values in each partition point. Consequently, the final values present the membership function of that fuzzy number. The fuzzy concept of the genetic algorithm in [44] is different, however, generates good results compared with the traditional fuzzy approaches.

In [45] a genetic algorithm is proposed for solving the fuzzy equation $S(p) = q$, such that p and q are k-sampled real fuzzy numbers, also S is a fuzzy function depends on p. As it is nearly impossible to obtain the exact solution of the fuzzy equation $S(p) = q$, hence it is more rational to find a fuzzy number $\tilde{p}$ in such a way that $S(\tilde{p})$ is close enough to q. The fitness function of the chromosome p which is a candidate solution of the fuzzy equation $S(p) = q$ is stated as

$$fit(p) = d(S(p), q) \tag{44}$$

where d is the measure of the difference between $S(p)$ and q. The fitness function (44) is minimized using the genetic algorithm. It is clear that the exact root p^* of the fuzzy equation $S(p) = q$ yields to $fit(p^*) = 0$. The genetic algorithm presented in [45] finds multiple solutions of the fuzzy equation and the program runs for a maximum number of generations (1000) several times. An acceptable solution is obtained in 600-700 generations. Three different solutions are found in [45] for fuzzy equation, however, the third solution is found with low accuracy. The presented genetic algorithm in [45] has slow convergent speed. To improve the performance of the genetic algorithm proposed in [45], an unsupervised clustering mechanism can be applied to the evolving population for creating subpopulations of individuals as well as developing different solutions during the same evolution procedure. Furthermore, the implementation of a fast and flexible parallel genetic algorithm could be a good idea to find good solutions fast [46].

Although genetic algorithm shows effectiveness in terms of solution accuracy and convergence, it is computationally expensive. The ranking method results are found to converge very quickly and are more accurate compared to the genetic algorithm method. The ranking method is computationally inexpensive.

3.4. Ranking technique

The ranking technique is introduced by Delgado et al [47]. They proposed three parameters named value, ambiguity, and fuzziness to obtain fuzzy numbers which can be used to present more arbitrary fuzzy numbers. The value, ambiguity, and fuzziness of a fuzzy number v with parametric form $(\underline{v}(\alpha), \overline{v}(\alpha))$ are defined as follows:

$$\begin{aligned} Val(v) &= \int_0^1 k(\alpha)[\underline{v}(\alpha) + \overline{v}(\alpha)]d\alpha \\ Amb(v) &= \int_0^1 k(\alpha)[\overline{v}(\alpha) - \underline{v}(\alpha)]d\alpha \\ Fuzz(v) &= \int_0^{\frac{1}{2}} [\overline{v}(\alpha) - \underline{v}(\alpha)]d\alpha + \int_{\frac{1}{2}}^1 [\underline{v}(\alpha) - \overline{v}(\alpha)]d\alpha \end{aligned} \tag{45}$$

where $k : [0,1] \to [0,1]$ is a reducing function.

In [48] the ranking fuzzy numbers technique is proposed to find the real roots of the following fuzzy polynomial equation

$$a_1 y + a_2 y^2 + ... + a_n y^n = a_0 \tag{46}$$

such that $y \in \Re$ and $a_0, a_1, ..., a_n$ are fuzzy numbers. The fuzzy polynomial equation (46) is transformed into a system of crisp polynomial equations using three parameters value, ambiguity, and fuzziness. The provided system of crisp polynomial equations is solved numerically. However, the ranking fuzzy numbers technique proposed in [48] based on three parameters value, ambiguity and fuzziness is quite inefficient to produce answers.

In [49], a new ranking technique is proposed which overcomes the drawback of the ranking fuzzy numbers technique proposed in [48]. The ranking technique proposed in [49] has four parameters named value, ambiguity, fuzziness, and vagueness. With the new parameter vagueness, the process of fuzzy polynomials in generating real roots is more effective and precise. The vagueness of the fuzzy number v with parametric form $(\underline{v}(\alpha), \overline{v}(\alpha))$ is defined as follows:

$$Vag(v) = \int_0^{\frac{1}{2}} [\underline{v}(\alpha) + \overline{v}(\alpha)] d\alpha + \int_{\frac{1}{2}}^{1} [\underline{v}(\alpha) + \overline{v}(\alpha)] d\alpha \qquad (47)$$

The proposed technique in [49] is successfully applied in the interval type-2 fuzzy polynomials, interval type-2 fuzzy polynomial equations, dual fuzzy polynomial equations as well as system of fuzzy polynomials.

The major drawback of the ranking technique is that it can be implemented only if membership functions are known. Approximation techniques like fuzzy neural networks are powerful tools that can overcome the limitations of other numerical techniques. The main advantages of fuzzy neural networks are their ability to train the great amount of data sets, rapid convergence and excellent precision.

3.5. Neural network technique

Both artificial neural networks and fuzzy logic are universal estimators that can approximate any nonlinear function to any desired degree of accuracy [50]. In [51] artificial neural network is used to solve the following fuzzy linear equation

$$A_1 Y = A_2 \qquad (48)$$

where A_1, A_2 and Y are triangular fuzzy numbers. For certain values of A_1 and A_2, (48) has no solution for Y [52]. In [51] two solutions for the artificial neural network are generated, X and Y^*. X is the output of the artificial neural network when there are no restrictions on the weights in the network. Y^* is the output of the artificial neural network when there are certain sign restrictions on the weights. Y^* is the new solution to fuzzy equations and $Y \leq Y^*$, whenever Y exists.

In [53] evolutionary algorithms and artificial neural networks are used to solve the following fuzzy equation,

$$A_1 Y + A_2 = A_3 \tag{49}$$

where A_1, A_2, A_3 and Y are triangular fuzzy numbers. Three solution techniques for solving the fuzzy equation (49) are introduced. The first solution type (Y_c), is named classical solution which uses α-cuts and interval arithmetic to obtain Y_c.

Example 3.5.1. Suppose $[A_1] = (1, 2, 3)$, $[A_2] = (-3, -2, -1)$ and $[A_3] = (3, 4, 5)$. Applying the intervals into the fuzzy equation (49), we get

$$\begin{aligned} (1+\alpha)\underline{Y}_c^\alpha + (-3+\alpha) = (3+\alpha) \\ (3-\alpha)\overline{Y}_c^\alpha + (-1-\alpha) = (5-\alpha) \end{aligned} \tag{50}$$

where $[Y_c]^\alpha = (\underline{Y_c}^\alpha, \overline{Y_c}^\alpha)$. Therefore,

$$\begin{aligned} \underline{Y}_c^\alpha = \tfrac{6}{1+\alpha} \\ \overline{Y}_c^\alpha = \tfrac{6}{3-\alpha} \end{aligned} \tag{51}$$

Nevertheless, $[\underline{Y}_c^\alpha, \overline{Y}_c^\alpha]$ is not a fuzzy number since $\underline{Y}_c^\alpha(\overline{Y}_c^\alpha)$ is a decreasing (increasing) function of α. Y_c may sometimes exist and sometimes may not exist.

The other solution is generated from fuzzifying the crisp solution $(a_3 - a_2)/a_1, a_1 \neq 0$. $(A_3 - A_2)/A_1$ is the fuzzified solution, such that zero does not belong to the support of A_1. To evaluate the fuzzified solution $(A_3 - A_2)/A_1$, two methods are proposed. The first technique is using the extension principle to produce Y_e. The second technique is using α-cut and interval arithmetic to produce Y_I. Y_e is obtained as follows:

$$Y_e(y) = \min\{\Pi(a_1, a_2, a_3) | (a_3 - a_2)/a_1 = y\} \tag{52}$$

where $\Pi(a_1, a_2, a_3) = \min\{A_1(a_1), A_2(a_2), A_3(a_3)\}$. The α-cut of Y_e are obtained as follows:

$$\begin{aligned} \underline{Y}_e^\alpha = \min\{\tfrac{a_3-a_2}{a_1} | a_1 \in [A_1]^\alpha, a_2 \in [A_2]^\alpha, a_3 \in [A_3]^\alpha\} \\ \overline{Y}_e^\alpha = \max\{\tfrac{a_3-a_2}{a_1} | a_1 \in [A_1]^\alpha, a_2 \in [A_2]^\alpha, a_3 \in [A_3]^\alpha\} \end{aligned} \tag{53}$$

where $[Y_e]^\alpha = (\underline{Y_e}^\alpha, \overline{Y_e}^\alpha)$. The solution Y_I is obtained as

$$[Y_I]^\alpha = ([A_3]^\alpha - [A_2]^\alpha)/[A_1]^\alpha \tag{54}$$

The fuzzy equation (49) can be solved by $Y_e(Y_I)$ if after substituting α-cuts of A_1, A_2, A_3 and $Y_e(Y_I)$ into (49) the resulting equation is valid.

Example 3.5.2. Suppose $[A_1] = (1, 2, 3)$, $[A_2] = (-3, -2, -1)$ and $[A_3] = (3, 4, 5)$. Since $\frac{a_3 - a_2}{a_1}$ is an increasing function of a_3 but a decreasing function of both a_1 and a_2 (supposing $a_1 > 0, a_3 > 0, a_2 < 0$), hence

$$\begin{aligned} \underline{Y}_e^\alpha &= \tfrac{4+2\alpha}{3-\alpha} \\ \overline{Y}_e^\alpha &= \tfrac{8-2\alpha}{1+\alpha} \end{aligned} \tag{55}$$

In this example $Y_e = Y_I$ which does not satisfy in (49).

For some fuzzy equations, Y_e is computationally too difficult to be obtained, so in [53] an evolutionary algorithm is proposed to approximate its α-cuts. However, the proposed technique in [53] is defined for only symmetric fuzzy numbers. It only calculates the upper bound and lower bound of the fuzzy numbers without taking into consideration the center part.

4. Numerical methods for solving fuzzy differential equations

Because of the nonlinear nature of the PDEs, analytical techniques cannot be used and solutions must be obtained with numerical techniques. In this section, five different important techniques are illustrated to solve FDEs and fuzzy PDEs.

4.1. Predictor-corrector technique

The predictor-corrector technique is extensively used for solving initial value problems. Three numerical techniques named Adams-Bashforth, Adams-Moulton and predictor-corrector are proposed in [54] to solve fuzzy ODEs. Predictor-corrector is generated from the combination of Adams-Bashforth and Adams-Moulton techniques. The Adams-Bashforth two-step technique is defined as

$$\begin{aligned} w_0 &= a_0, \quad w_1 = a_1 \\ w_{j+1} &= w_j + \frac{k}{2}[3g(t_j, w_j) - g(t_{j-1}, w_{j-1})], \quad j = 1, 2, ..., N-1 \end{aligned} \tag{56}$$

where $p = t_0 \leq t_1 \leq ... \leq t_N = q$, and $k = \frac{(q-p)}{N}$. The Adams-Moulton two-step technique is defined as

$$\begin{aligned} w_0 &= a_0, \quad w_1 = a_1 \\ w_{j+1} &= w_j + \frac{k}{12}[5g(t_{j+1}, w_{j+1}) + 8g(t_j, w_j) - g(t_{j-1}, w_{j-1})] \end{aligned} \tag{57}$$

for $j = 1, 2, ..., N-1$. The convergence order of the techniques proposed in [54] is $O(h^m)$ which is higher than the convergence order of the Euler technique that is $O(h)$ [55]). The following example is presented in [54] which uses the Adams-Bashforth, Adams-Moulton and predictor-corrector techniques to solve fuzzy ODEs in the setting of Hukuhara or Seikkala differentiability.

Example 4.1.1. Consider the following initial value problem

$$\begin{aligned} \tfrac{d}{dt}w &= -w + t + 1 \\ w(0) &= (0.96, 1, 1.01) \end{aligned} \tag{58}$$

In [54] it is presented that the exact solution at $t = 0.1$ is

$$w(0.1) = (0.1 + 0.96e^{-0.1}, 0.1 + e^{-0.1}, 0.1 + 1.01e^{-0.1}) \tag{59}$$

The exact solution (59) is acquired by supposing that the solution takes the form

$$w(t) = t + (0.96, 1, 1.01)e^{-t} \tag{60}$$

However, this function is not Hukuhara differentiable as it has a decreasing length of the support. The Hukuhara differentiable function has an increasing length of the support. The correct exact solution is illustrated in [56].

Lemma 1. If $s(t) = (\beta(t), \gamma(t), \varphi(t))$ is triangular number valued function and if s is Hukuhara differentiable, so $\frac{d}{dt}s = (\frac{d}{dt}\beta, \frac{d}{dt}\gamma, \frac{d}{dt}\varphi)$.

Consider the following initial value problem

$$\begin{aligned} \tfrac{d}{dt}w &= g(t, w) \\ w(t_0) &= w_0 \end{aligned} \tag{61}$$

with $w_0 = (\underline{w}_0, w_0^1, \overline{w}_0) \in E, w(t) = (\underline{s}, s^1, \overline{s}) \in E, g : [t_0, t_0 + b] \times E \to E, g(t, (\underline{s}, s^1, \overline{s})) = (\underline{g}(t, \underline{s}, s^1, \overline{s}), g^1(t, \underline{s}, s^1, \overline{s}), \overline{g}(t, \underline{s}, s^1, \overline{s}))$, and using Lemma 1, (61) can be transformed into the following system of ODE

$$\begin{cases} \frac{d}{dt}\underline{s} = \underline{g}(t, \underline{s}, s^1, \overline{s}) \\ \frac{d}{dt}s^1 = g^1(t, \underline{s}, s^1, \overline{s}) \\ \frac{d}{dt}\overline{s} = \overline{g}(t, \underline{s}, s^1, \overline{s}) \\ \underline{s}(0) = \underline{w}_0, s^1(0) = w_0^1, \overline{s}(0) = \overline{w}_0 \end{cases} \tag{62}$$

Theorem 1. Let us consider the initial value problem (61) with $w_0 = (\underline{w}_0, w_0^1, \overline{w}_0) \in E, g : [t_0, t_0+b] \times E \to E, g(t, (\underline{s}, s^1, \overline{s})) = (\underline{g}(t, \underline{s}, s^1, \overline{s}), g^1(t, \underline{s}, s^1, \overline{s}), \overline{g}(t, \underline{s}, s^1, \overline{s}))$ such that $\underline{g}, g^1, \overline{g}$ are Lipschitz continuous (real-valued) functions. Therefore, the solution of (61) is triangular-valued function $w(t) = (\underline{s}(t), s^1(t), \overline{s}(t)) : [t_0, t_0 + b] \to E$, also the initial value problem (61) is equivalent to the system of ODE (62).

To correct the example 4.1.1, using Theorem 1, the problem (58) is transformed into

$$\begin{cases} \frac{d}{dt}\underline{s} = -\overline{s} + t + 1 \\ \frac{d}{dt}s^1 = -s^1 + t + 1 \\ \frac{d}{dt}\overline{s} = -\underline{s} + t + 1 \\ \underline{s}(0) = 0.96, s^1(0) = 1, \overline{s}(0) = 1.01 \end{cases} \tag{63}$$

having solution $\underline{s}(t) = t - 0.025e^t + 0.985e^{-t}, s^1(t) = t + 1.0e^{-t}, \overline{s}(t) = t + 0.025e^t + 0.985e^{-t}$. Therefore the solution of (58) is

$$w(t) = (t - 0.025e^t + 0.985e^{-t}, t + 1.0e^{-t}, t + 0.025e^t + 0.985e^{-t}) \tag{64}$$

The predictor-corrector method is efficient as it uses the information from prior steps. The disadvantage of the predictor-corrector method is that the number of iterations is long which may lead to slow convergence. Moreover, this method is too hard to program. Adomian decomposition technique is simple and easy to use.

4.2. Adomian decomposition technique

The Adomian decomposition technique was first proposed by Adomian in the early 1980s. It has been applied to a wide class of linear and non-linear ODEs, PDEs and integral equations. In [57] the Adomian decomposition technique is proposed for obtaining the numerical solution of hybrid FDEs. The Adomian decomposition technique considers the estimate solution of a nonlinear equation as an infinite series that often approaches the exact solution.

In [58] the fuzzy solution of the second-order homogeneous fuzzy PDEs is obtained using the Adomian decomposition technique. Using Seikkala derivative in [58], the Seikkala solution of a fuzzy heat equation with specific fuzzy boundary and initial conditions is obtained. Seikkala solution is based on Seikkala derivative presented in [59].

Definition 4 (Seikkala derivative). Let I be a real interval and $U : I \to E$ be a fuzzy process. The Seikkala derivative $\frac{d}{dt}U(t)$ of a fuzzy process u is defined as

$$SDU(t) = [\frac{d}{dt}U(t)]^{\alpha} = [\frac{d}{dt}\underline{U}(t,\alpha), \frac{d}{dt}\overline{U}(t,\alpha)], \quad t \in I \tag{65}$$

Consider the following differential equation

$$Ak + Bk + Ck = f \tag{66}$$

where A is the highest order derivative that is supposed to be easily invertible, B is a linear differential operator of order less than A, C presents the nonlinear terms, also f is the source term. It has been assumed in the Adomian decomposition technique that the unknown function k can be decomposed as follows

$$k = \sum_{m=0}^{\infty} k_m \tag{67}$$

The nonlinear operator Ck is defined as

$$Ck = \sum_{m=0}^{\infty} B_m(k_0, ..., k_m) \tag{68}$$

where $B_m(k_0, ..., k_m)$ are the appropriate Adomian's polynomials that are presented as

$$B_h = \frac{1}{h!}(\frac{d^h}{d\varphi^h}F(\sum_{m=0}^{\infty} k_m\varphi^m))|_{\varphi=0} \tag{69}$$

The terms of series $k = \sum_{m=0}^{\infty} k_m$ are computed using the following iterated approach

$$\begin{aligned} k_0 &= A^{-1}u \\ k_m &= -A^{-1}B(k_m) - A^{-1}(B_{m-1}) \end{aligned} \tag{70}$$

The Adomian decomposition method is reliable and promising. It can be used for all kinds of differential equations, linear or nonlinear, homogeneous or non-homogeneous. However, the effectiveness and precision of the Adomian decomposition method depend on the convergence and the rate of convergence of the series solution. Adomian decomposition method produces a series solution that may have a slow rate of convergence over broader areas. Moreover, the Adomian decomposition method series solution can be divergent if the solution of the problem is oscillatory. To overcome these disadvantageous, the Adomian decomposition method should be modified in order to deal with problems with oscillatory solutions in nature. For this, the Laplace transform is proposed with Adomian decomposition technique for solving such problems [60].

4.3. Taylor technique

In [61], the 2nd Taylor method is proposed for solving linear and nonlinear FDEs. The convergence order of the 2nd Taylor method is $O(h^2)$ which is higher than the convergence order of the Euler method that is $O(h)$ [55].

Solving numerically the FDEs by the Taylor approach of order p is illustrated in [62]. The algorithm successfully solves linear and nonlinear fuzzy Cauchy problems with the convergence order of $O(h^p)$.

The main disadvantage of the Taylor series method is the calculation of higher derivatives. The procedure becomes more difficult as the order increases. Runge-Kutta technique is often considered to be the most efficient one-step method. The Runge-Kutta formulas simplifies the Taylor techniques, while not remarkably increasing the error.

4.4. Runge-Kutta technique

In [63] the four-stage order Runge-Kutta technique is proposed for solving linear and nonlinear FDEs. Even though this work is important, it has the drawback that, while investigating the convergence of their four-stage order Runge-Kutta technique, the authors practically work on the convergence of the ODEs system that happens when solving numerically. In [14], RungeKutta s-stage method is proposed for a more general category of problems.

consider the following fuzzy initial value problem

$$\begin{aligned} &\tfrac{d}{dt}w = g(t,w,v) \\ &w(t_0) = w_0 \in E \end{aligned} \tag{71}$$

where w is the unknown fuzzy function, $t \in [t_0, T]$, and v is considered as a vector of triangular fuzzy numbers. Moreover, g is a continuous fuzzy function that its fuzziness is because of the existence of v, meaning that if v is taken to be a vector of real numbers, consequently g will become a crisp function. In [14], the RungeKutta s-stage technique for the solution of (71) is defined as follows:

$$w_{n+1} = w_n + \Psi(t_n, w_n, h) \tag{72}$$

where

$$\begin{aligned} &\Psi(t_n, w_n, v, h) = \textstyle\sum_{i=1}^{s} \lambda_i q_i \\ &q_1 = g(t_n, w_n, v) \\ &q_i = g(t_n + \gamma_i h, w_n + h \textstyle\sum_{j=1}^{i-1} \zeta_{ij} q_j, v), \quad i = 2, ..., s \end{aligned} \tag{73}$$

where $h = (T - t_0)/N$ also the following conditions are hold

$$\sum_{i=1}^{s} \lambda_i = 1, \quad \gamma_i = \sum_{j=1}^{i-1} \zeta_{ij}, \quad i = 1, 2, ..., s \tag{74}$$

It is clear that constants q_i are fuzzy numbers. Furthermore, convergence for s-stage RungeKutta technique is proved in [14].

The main advantages of Runge-Kutta methods are that they are easy to use and also they are stable. The main disadvantages of Runge-Kutta methods are that they need comparatively large computer time. Also, in particular, they are not suitable for systems of differential equations with a mix of fast and slow state dynamics. Artificial neural networks are relatively easy to implement and computationally fast.

4.5. Neural network technique

Numerical solutions of FDEs and fuzzy PDEs by utilizing fuzzy artificial neural networks is more modern than the previous subjects because it only goes back to 2010. In [64] fuzzy artificial neural network method is used for finding the approximate solution of fuzzy PDEs. The proposed method is based on substituting each u in the input vector $u = (u_1, u_2, ..., u_n), u_i \in [a, b]$ by a polynomial of first degree $P(u) = \epsilon(u+1), \epsilon \in (0, 1)$. Therefore, the input vector will be $(P(u_1), P(u_2), ..., P(u_n)), P(u_i) \in (a, b)$.

The proposed technique in [64] selects the training points over the open interval (a, b) without training the neural network in the range of first and end points which cause in decreasing the computational error. This technique can handle efficiently all kinds of fuzzy PDEs and produce a precise approximate solution.

Fully fuzzy neural networks have disadvantages such as having long computation time and complicated learning algorithm. In order to reduce the complexity of the learning algorithm and computation time in [65] a partially fuzzy neural network is proposed for finding the solutions of FDEs. In the proposed partially fuzzy neural network the connection weights to output unit are fuzzy numbers while connection weights and biases to hidden units are real numbers.

5. Comparisons

In this section, several application examples have been established to compare the efficiency of numerical methods to approximate the solution of dual fuzzy equations and FDEs.

Example 1. The water tank system has two inlet valves W_1, W_2, and two outlet valves W_3, W_4, see Figure 5. The areas of the valves are uncertain as the triangle function (2), $A_1 = F(0.019, 0.022, 0.025)$, $A_2 = F(0.009, 0.019, 0.037)$,

$A_3 = F(0.011, 0.014, 0.016)$, $A_4 = F(0.041, 0.059, 0.071)$. The velocities of the flow (controlled by the valves) are $f_1 = (\frac{v}{10})e^v$, $f_2 = vcos(\Pi v)$, $f_3 = cos(\frac{\Pi v}{8})$, $f_4 = \frac{v}{2}$. If the outlet flow is aimed to be $d = (4.091, 6.341, 36.388)$, what is the quantity of the control variable v.

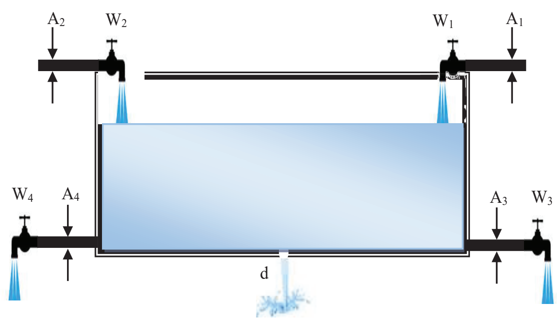

Fig. 5. Water tank system

The mass balance of the tank is [66]:

$$\rho A_1 f_1 \oplus \rho A_2 f_2 = \rho A_3 f_3 \oplus \rho A_4 f_4 \oplus d \tag{75}$$

where ρ is the density of the water. The exact solution is $v_0 = 2$ [66]. To approximate the solution, we use five popular techniques: Newton technique, Steepest descent technique, Genetic algorithm technique, Ranking technique, and Fuzzy neural network technique. The errors of these techniques are demonstrated in Table 1. We can see that all five techniques can estimate the solutions of the dual fuzzy equations. Fuzzy neural network technique is more appropriate for solving these type of equations. In Table 1, k is the number of iterations. The small estimation errors can be acquired by making the number of iteration larger. By increasing the number of iterations the estimated errors of the fuzzy neural networks are less than the other techniques. Fuzzy neural network technique is more robust in comparison with the other techniques. Corresponding error plots are shown in Figure 6.

*Table*1. *Estimation errors*

k	Newton	Steepest descent	Genetic algorithm	Ranking	Fuzzy neural network
1	0.18764	0.16932	0.33339	0.31115	0.43884
2	0.29598	0.26112	0.24813	0.23793	0.32382
3	0.36201	0.32701	0.13123	0.11704	0.21802
⋮	⋮	⋮	⋮	⋮	⋮
119	0.07886	0.05198	0.04601	0.02888	0.00322
120	0.07499	0.04892	0.03887	0.02493	0.00275

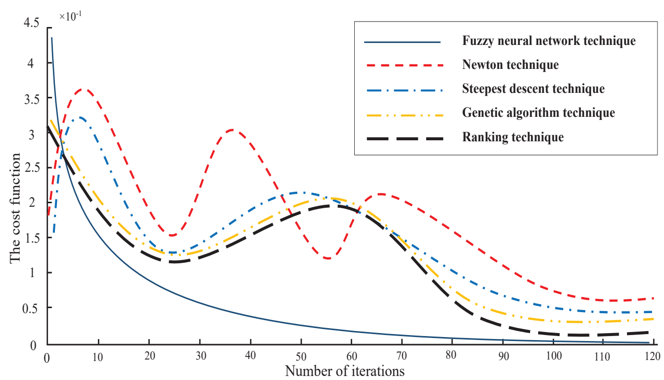

Fig. 6. Estimation errors of five popular techniques

Example 2. The deformation of a solid cylindrical rod depends on the stiffness E, the forces on it f, the positions of the forces L, and the diameter of the rod d, see Figure 7. The positions are not exact, they satisfy the trapezoidal function (3), $L_1 = F(0.2, 0.3, 0.5, 0.6)$, $L_2 = F(0.4, 0.6, 0.7, 0.8)$, $L_3 = F(0.4, 0.6, 0.7, 0.8)$. The area of the rod is $A = \frac{\pi}{4}d^2$. The external forces are the function of v, $f_1 = v^7$, $f_2 = v^6\sqrt{v}$, $f_3 = e^{2v}$. If the the desired deformation at the point M is aimed to be $M^* = F(0.000563, 0.000822, 0.001003, 0.001211)$, what is the quantity of the control force v.

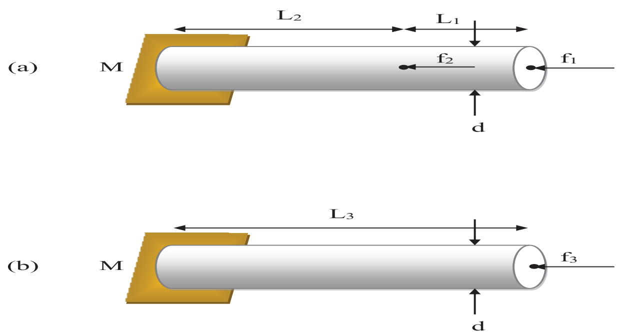

Fig. 7. Two solid cylindrical rods

According to the tension relations we have [67]

$$\frac{L_1 f_1}{AE} \oplus \frac{L_2(f_1 + f_2)}{AE} = \frac{L_3 f_3}{AE} \oplus M^* \tag{76}$$

where $d = 0.02$, $E = 70 \times 10^9$. The exact solution is $v = 4$. To approximate the solution, we use five popular techniques: Newton technique, Steepest descent technique, Genetic algorithm technique, Ranking technique, and Fuzzy neural network technique. The errors of these techniques are demonstrated in Table 2. Fuzzy neural network technique is more robust than the other techniques. Furthermore, the estimated error of the fuzzy neural network is less when compared with other techniques. Corresponding error plots are shown in Figure 8.

Table2. Estimation errors

k	Newton	Steepest descent	Genetic algorithm	Ranking	Fuzzy neural network
1	0.1508	0.2013	0.4865	0.6004	0.7883
2	0.2296	0.2996	0.5743	0.4987	0.5002
3	0.3119	0.1844	0.4076	0.3791	0.3101
⋮	⋮	⋮	⋮	⋮	⋮
89	0.1099	0.08014	0.06995	0.05001	0.00985
90	0.09607	0.07201	0.06001	0.04112	0.00711

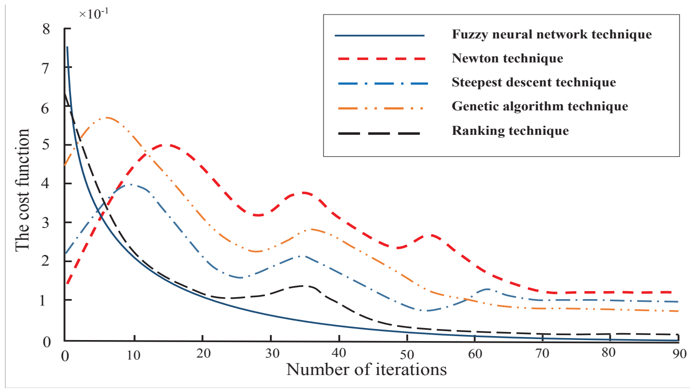

Fig. 8. Estimation errors of five popular techniques

Example 3. The vibration mass system shown in Figure 9 is modeled as,

$$\frac{d}{dt}u(t) = \frac{\tilde{c}}{\tilde{m}}x(t), \quad u(t) = \frac{d}{dt}x(t) \tag{77}$$

where the spring constant is $\tilde{c} = 1$, and the mass is $\tilde{m} = (0.75, 1.125)$. If the initial position is $x(0) = (0.75 + 0.25\alpha, 1.125 - 0.125\alpha)$, $\alpha \in [0, 1]$, hence the exact solutions of (77) are [68]

$$x(t, \alpha) = \left[(0.75 + 0.25\alpha)e^t, (1.125 - 0.125\alpha)e^t\right] \tag{78}$$

where $t \in [0, 1]$.

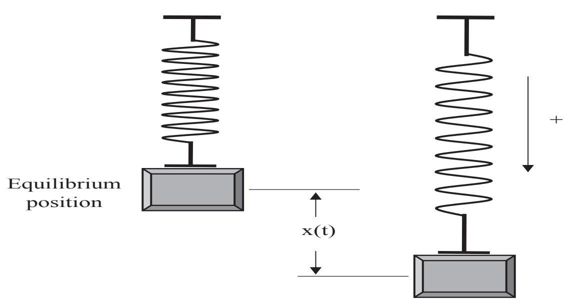

Fig. 9. Vibration mass

To approximate the solution (78), we use five popular techniques: Predictor-corrector technique, Adomian decomposition technique, Taylor technique, Runge-Kutta technique, and Fuzzy neural network technique. The errors of these techniques are demonstrated in Table 3. Corresponding solution plots are shown in Figure 10.

Table3. Estimation errors

α	Predictor-corrector	Adomian decomposition	Taylor	Runge-Kutta	Fuzzy neural network
0	[0.2059,0.4378]	[0.0931,0.1411]	[0.0611,0.1097]	[0.0412,0.0895]	[0.0212,0.0611]
0.2	[0.2229,0.4568]	[0.1028,0.1512]	[0.0713,0.1187]	[0.0611,0.1089]	[0.0314,0.0711]
0.4	[0.1962,0.4281]	[0.0829,0.1312]	[0.0509,0.0988]	[0.0209,0.0689]	[0.0111,0.0512]
0.6	[0.1861,0.4181]	[0.0723,0.1211]	[0.0411,0.0879]	[0.0209,0.0688]	[0.0009,0.0411]
0.8	[0.2469,0.4789]	[0.1229,0.1709]	[0.1011,0.1489]	[0.0709,0.1188]	[0.0507,0.0909]
1	[0.2569,0.2569]	[0.1429,0.1429]	[0.1111,0.1111]	[0.0812,0.0812]	[0.0611,0.0611]

All five techniques are appropriate for solving FDEs. The leaning process of the fuzzy neural network technique is more rapid than the other techniques. Furthermore, the robustness of fuzzy neural network technique is better in comparison with the other techniques.

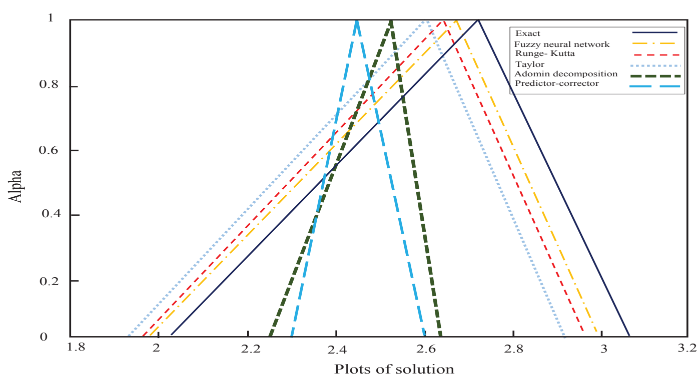

Fig. 10. Comparison plot of five popular techniques and the exact solution

Example 4. A tank with a heating system is demonstrated in Figure 11, where $\tilde{R} = 0.5$ and the thermal capacitance is $\tilde{C} = 2$. The temperature is x. The model is [69],

$$\frac{d}{dt}x(t) = -\frac{1}{\tilde{R}\tilde{C}}x(t) \tag{79}$$

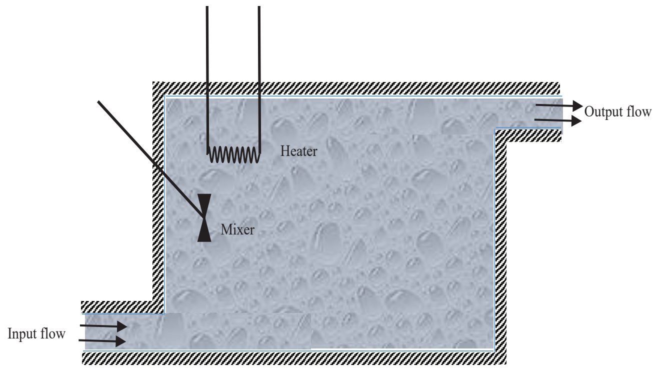

Fig. 11. Thermal system

where $t \in [0, 1]$ and x is the amount of sinking in each moment. If the initial position is $x(0) = (\alpha - 1, 1 - \alpha)$ and $\alpha \in [0, 1]$, so the exact solutions of (79) are

$$x(t, \alpha) = [(\alpha - 1)e^t, (1 - \alpha)e^t] \tag{80}$$

To approximate the solution (80), we use five popular techniques: Predictor-corrector technique, Adomian decomposition technique, Taylor technique, Runge-Kutta technique, and Fuzzy neural network technique. The errors of these techniques are demonstrated in Table 4. The lower and upper bounds of absolute errors are displayed in Figure 12 and Figure 13, respectively. The approximation errors of the fuzzy neural network technique is smaller than the other techniques.

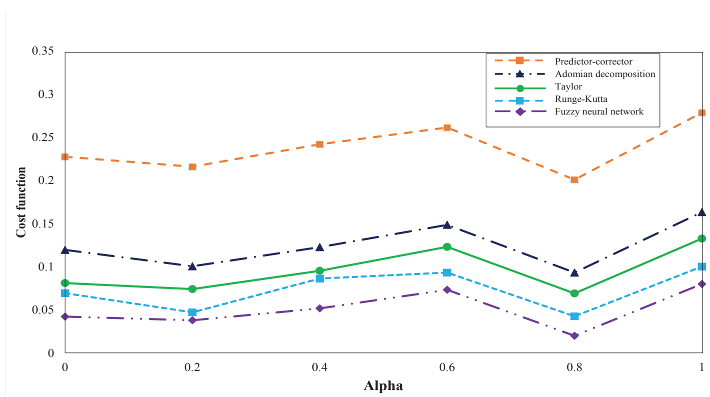

Fig. 12. The lower bounds of absolute errors

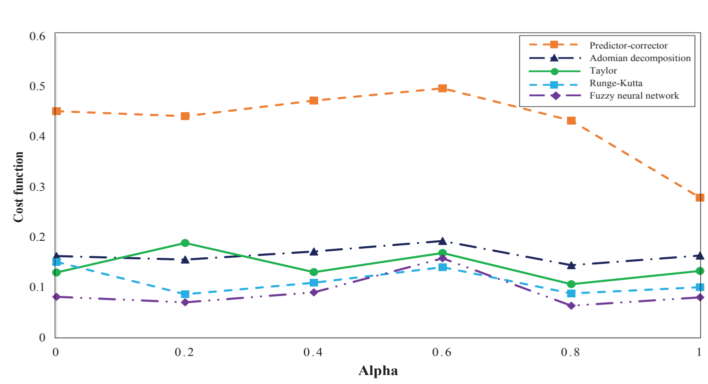

Fig. 13. The upper bounds of absolute errors

Table4. Estimation errors

α	Predictor-corrector	Adomian decomposition	Taylor	Runge-Kutta	Fuzzy neural network
0	[0.2281,0.4512]	[0.1188,0.1619]	[0.0809,0.1287]	[0.0686,0.1512]	[0.0418,0.0809]
0.2	[0.2161,0.4412]	[0.1011,0.1549]	[0.0739,0.1879]	[0.0469,0.0858]	[0.0379,0.0711]
0.4	[0.2421,0.4723]	[0.1231,0.1709]	[0.0949,0.1311]	[0.0859,0.1088]	[0.0521,0.0911]
0.6	[0.2608,0.4959]	[0.1479,0.1919]	[0.1229,0.1679]	[0.0928,0.1412]	[0.0729,0.1569]
0.8	[0.2011,0.4322]	[0.0929,0.1438]	[0.0688,0.1059]	[0.0431,0.0881]	[0.0212,0.0641]
1	[0.2791,0.2791]	[0.1629,0.1629]	[0.1331,0.1331]	[0.1011,0.1011]	[0.0812,0.0812]

6. Conclusions

In this paper, we have presented an overview of the most common numerical solution strategies for the fuzzy equations, dual fuzzy equations, FDEs, and fuzzy PDEs. The existence of solutions for these equations is discussed in detail. Research in this area continues to develop new types of numerical techniques and strategies. Emphasis is given to recent developments in solving strategies in the last two decades, which indicates their significant progress.

References

[1] R. Jafari, W. Yu., Uncertainty nonlinear systems control with fuzzy equations, IEEE International Conference on Systems, Man, and Cybernetics

(2015) 2885–2890doi:10.1109/SMC.2015.502.

[2] R. Jafari, S. Razvarz., Solution of fuzzy differential equations using fuzzy sumudu transforms, Mathematical and Computational Applications 23 (2018) 1–15.

[3] R. Jafari, S. Razvarz, A. Gegov., A new computational method for solving fully fuzzy nonlinear systems, Computational Collective Intelligence: 10th International Conference, ICCCI 2018, Bristol, UK, September 5-7, 2018, Proceedings, Part I, Lecture Notes in Computer Science, Springer 11055 (2018) 503–512. doi:10.1007/978-3-319-98443-8_46.

[4] J. Buckley, Y. Hayashi., Can fuzzy neural nets approximate continuous fuzzy functions, Fuzzy Sets Syst. 61 (1) (1994) 43–51. doi:https://doi.org/10.1016/0165-0114(94)90283-6.

[5] J. Gibson, An analysis of optimal modal regulation: convergence and stability, SIAM J. Control. Optim. 19 (5) (1981) 686–707. doi:https://doi.org/10.1137/0319044.

[6] S. Abbasbandy, R. Ezzati., Newton's method for solving a system of fuzzy nonlinear equations, Appl. Math. Comput. 175 (2) (2006) 1189–1199. doi:https://doi.org/10.1016/j.amc.2005.08.021.

[7] T. Allahviranloo, M. Otadi, M. Mosleh., Iterative method for fuzzy equations, Soft Computing 12 (10) (2008) 935–939. doi:https://doi.org/10.1007/s00500-007-0263-y.

[8] Z. Gouyandeh, T. Allahviranloo, S. Abbasbandy, A. Armand., A fuzzy solution of heat equation under generalized hukuhara differentiability by fuzzy fourier transform, Fuzzy Sets Syst. 309 (2017) 81–97.

[9] R. G. Moghaddam, T. Allahviranloo., On the fuzzy poisson equation, Fuzzy Sets Syst.doi:https://doi.org/10.1016/j.fss.2017.12.013.

[10] M. Kajani, B. Asady, A. Vencheh., An iterative method for solving dual fuzzy nonlinear equations, Appl. Math. Comput. 167 (1) (2005) 316–323. doi:https://doi.org/10.1016/j.amc.2004.06.113.

[11] M. Waziri, Z. Majid., A new approach for solving dual fuzzy nonlinear equations using broyden's and newton's methods, Advances in Fuzzy Systems 2012 (2012) 1–5. doi:http://dx.doi.org/10.1155/2012/682087.

[12] S. Pederson, M. Sambandham., The runge-kutta method forhybrid fuzzy differential equation, Nonlinear Anal. Hybrid Syst. 2 (2) (2008) 626–634. doi:https://doi.org/10.1016/j.nahs.2006.10.013.

[13] A. Khastan, K. Ivaz., Numerical solution of fuzzy differential equations by nystr$\ddot{o}$m method, Chaos, Solitons. Fractals. 41 (2) (2009) 859–868. doi:https://doi.org/10.1016/j.chaos.2008.04.012.

[14] S. Palligkinis, G. Papageorgiou, I. Famelis., Runge-kutta methods for fuzzy differential equations, Applied Mathematics and Computation 209 (1) (2009) 97–105. doi:https://doi.org/10.1016/j.amc.2008.06.017.

[15] S. Tapaswini, S. Chakraverty., Euler-based new solution method for fuzzy initial value problems, Int. J. Artificial. Intell. Soft. Comput. 4 (1) (2014) 58–79. doi:https://doi.org/10.1504/IJAISC.2014.059288.

[16] M. Ahmadi, N. Kiani, N. Mikaeilvand., Laplace transform formula on fuzzy nth-order derivative and its application in fuzzy ordinary differential equations, Soft Comput 18 (12) (2014) 2461–2469. doi:https://doi.org/10.1007/s00500-014-1224-x.

[17] S. Guo, L. Mei, Y. Zhou., The compound $\frac{G'}{G}$-expansion method and double non-traveling wave solutions of (2+1)-dimensional nonlinear partial differential equations, COMPUT. MATH. APPL. 69 (8) (2015) 804–816. doi:https://doi.org/10.1016/j.camwa.2015.02.016.

[18] M. Chehlabi, T. Allahviranloo, Positive or negative solutions to first-order fully fuzzy linear differential equations under gener-

alized differentiability, Appl. Soft. Comput. 70 (2018) 359–370. doi:https://doi.org/10.1016/j.asoc.2018.05.040.

[19] M. Chehlabi, T. Allahviranloo, Concreted solutions to fuzzy linear fractional differential equations, Appl. Soft. Comput. 44 (2016) 108–116. doi:https://doi.org/10.1016/j.asoc.2016.03.011.

[20] J. Buckley, E. Eslami., Neural net solutions to fuzzy problems: The quadratic equation, Fuzzy Sets Syst. 86 (3) (1997) 289–298. doi:https://doi.org/10.1016/S0165-0114(95)00412-2.

[21] A. Jafarian, R. Jafari, A. Khalili, D. Baleanud., Solving fully fuzzy polynomials using feed-back neural networks, International Journal of Computer Mathematics 92 (4) (2015) 742–755. doi:https://doi.org/10.1080/00207160.2014.907404.

[22] R. Jafari, W. Yu., Fuzzy control for uncertainty nonlinear systems with dual fuzzy equations, Journal of Intelligent and Fuzzy Systems 29 (3) (2015) 1229–1240. doi:10.3233/IFS-151731.

[23] M. Mosleh, Evaluation of fully fuzzy matrix equations by fuzzy neural network, Appl. Math. Model. 37 (9) (2013) 6364–6376. doi:https://doi.org/10.1016/j.apm.2013.01.011.

[24] A. Tahavvor, M. Yaghoubi., Analysis of natural convection from a column of cold horizontal cylinders using artificial neural network, Appl. Math. Model. 36 (7) (2012) 3176–3188. doi:https://doi.org/10.1016/j.apm.2011.10.003.

[25] S. Effati, M. Pakdaman., Artificial neural network approach for solving fuzzy differential equations, Inform. Sci. 180 (8) (2010) 1434–1457. doi:https://doi.org/10.1016/j.ins.2009.12.016.

[26] H. Yazdi, R. Pourreza., Unsupervised adaptive neural-fuzzy inference system for solving differential equations, Appl. Soft. Comput. 10 (1) (2010) 267–275. doi:https://doi.org/10.1016/j.asoc.2009.07.006.

[27] H. Lee, I. Kang., Neural algorithms for solving differential equations, Journal of Computational Physics 91 (1) (1990) 110–131. doi:https://doi.org/10.1016/0021-9991(90)90007-N.

[28] M. Dissanayake, N. Phan-Thien., Neural-network based approximations for solving partial differential equations, International Journal for Numerical Methods in Biomedical Engineering 10 (3) (1994) 195–201. doi:10.1002/cnm.1640100303.

[29] S. He, K. Reif, R. Unbehauen., Multilayer neural networks for solving a class of partial differential equations, Neural Networks. 13 (3) (2000) 385–396. doi:https://doi.org/10.1016/S0893-6080(00)00013-7.

[30] C. Montelora, C. Saloma., Solving the nonlinear schrodinger equation with an unsupervised neural network: Estimation of error in solution, Opt. Commun. 222 (1-6) (2003) 331–339. doi:https://doi.org/10.1016/S0030-4018(03)01570-0.

[31] N. Sukavanam, V. Panwar., Computation of boundary control of controlled heat equation using artificial neural networks, Int. Commun. Heat Mass Transfer. 30 (8) (2003) 1137–1146. doi:https://doi.org/10.1016/S0735-1933(03)00179-9.

[32] W. Lodwick, D. Dubois., Interval linear systems as a necessary step in fuzzy linear systems, Fuzzy Sets Syst. 281 (2015) 227–251.

[33] S. Shary, Solving the tolerance problem for interval linear equations, Interval Comput. 2 (1994) 4–22.

[34] H. Nguyen, A note on the extension principle for fuzzy sets, J. Math. Anal. Appl. 64 (1978) 369–380.

[35] R. Jafari, W. Yu, X. Li., Fuzzy differential equations for nonlinear system modeling with bernstein neural networks, IEEE Access 4 (2016) 9428–94367.

[36] B. Bede, L. Stefanini., Generalized differentiability of fuzzy-valued functions, Fuzzy Sets Syst. 230 (2013) 119–141.

[37] I. Newton, The method of uxions and infinite series, London: Henry Woodfall. Retrieved from https://archive.org/details/methodoffluxions00newt. 3 (1671) 43–47. doi:ISBN1498167489,9781498167482.

[38] S. Abbasbandy, B. Asady., Newton's method for solving fuzzy nonlinear equations, Appl. Math. Comput. 159 (2) (2004) 349–356. doi:https://doi.org/10.1016/j.amc.2003.10.048.

[39] R. Moore, Interval analysis, Prentice-Hall, Englewood Cliffs.

[40] P. Sevastjanov, L. Dymova, P. Bartosiewicz., A new approach to normalization of interval and fuzzy weights, Fuzzy Sets Syst. 198 (2012) 34–45. doi:https://doi.org/10.1016/j.fss.2012.01.003.

[41] S. Abbasbandy, A. Jafarian., Steepest descent method for solving fuzzy nonlinear equations, Appl. Math. Comput. 174 (1) (2006) 669–675. doi:https://doi.org/10.1016/j.amc.2005.04.092.

[42] J. F. R.L. Burden, A. Burden, Numerical analysis, seventh ed., PWS-Kent, Boston,doi:ISBN-13:978-1305253667.

[43] S. Okada, M. Gen., Fuzzy multiple choice knapsack problem, Fuzzy Sets Syst. 67 (1994) 71–80.

[44] F. Lin, simulating fuzzy numbers for solving fuzzy equations with constraints using genetic algorithm, International Journal of Innovative Computing, Information and Control 6 (1) (2010) 239–253.

[45] O. Brudaru, F. Leon, O. Buzatu., Genetic algorithm for solving fuzzy equations, 8th International Symposium on Automatic Control and Computer Science, Iasi,doi:ISBN973-621-086-3.

[46] E. Cantu-Paz, Implementing fast and flexible parallel genetic algorithms, Practical Handbook of Genetic Algorithms, CRC Press, Boca Raton, 3.

[47] M. Delgado, M. Vila, W. Voxman., On a canonical representation of fuzzy numbers, Fuzzy Sets Syst. 93 (1) (1998) 125–135. doi:https://doi.org/10.1016/S0165-0114(96)00144-3.

[48] H. Rouhparvar, Solving fuzzy polynomial equation by ranking method, First Joint Congress on Fuzzy and Intelligent Systems, Ferdowsi University of Mashhad, Iran, (2007) 1–6.

[49] R. N. Ab, A. Lazim, A. A. Termimi, A. Noorani., Solutions of interval type-2 fuzzy polynomials using a new ranking method, AIP Conference Proceedings 1682 (1) (2015) 0200121–0200129. doi:https://doi.org/10.1063/1.4932421.

[50] G. Cybenko, Approximation by superposition of sigmoidal activation function, Math. Control, Sig Syst 2 (4) (1989) 303–314. doi:https://doi.org/10.1007/BF02551274.

[51] J. Buckley, E. Eslami, Y. Hayashi., Solving fuzzy equations using neural nets, Fuzzy Sets Syst. 86 (3) (1997) 271–278. doi:https://doi.org/10.1016/S0165-0114(96)00008-5.

[52] J. Buckley, Y. Qu., Solving linear and quadratic fuzzy equations, Fuzzy Sets Syst. 35 (1) (1990) 43–59. doi:https://doi.org/10.1016/0165-0114(90)90099-R.

[53] J. Buckley, T. Feuring, Y. Hayashi., Solving fuzzy equations using evolutionary algorithms and neural nets, Soft comput. 6 (2) (2002) 116–123. doi:https://doi.org/10.1007/s005000100147.

[54] T. Allahviranloo, N. Ahmadi, E. Ahmadi., Numerical solution of fuzzy differential equations by predictor-corrector method, Inform. Sci. 177 (7) (2007) 1633–1647. doi:https://doi.org/10.1016/j.ins.2006.09.015.

[55] M. Ma, M. Friedman, A. Kandel., Numerical solutions of fuzzy differential equations, Fuzzy Sets Syst. 105 (1) (1999) 133–138.

[56] B. Bede, Note on numerical solutions of fuzzy differential equations by predictor corrector method, Inform. Sci. 178 (7) (2008) 1917–1922. doi:https://doi.org/10.1016/j.ins.2007.11.016.

[57] M. Paripour, E. Hajilou, H. Heidari., Application of adomian decomposition method to solve hybrid fuzzy differential equations, Journal of Taibah University for Science 9 (1) (2015) 95–103. doi:10.1016/j.jtusci.2014.06.002.

[58] U. Pirzada, D. Vakaskar., Solution of fuzzy heat equations using adomian decomposition method, Int. J. Adv. Appl. Math. and Mech. 3 (1) (2015) 87–91.

[59] S. Seikkala, On the fuzzy initial value problem, Fuzzy Sets Syst. 24 (1987) 319–330.

[60] N. Singh, M. Kumar., Adomian decomposition method for solving higher order boundary value problems, Mathematical Theory and Modeling 2 (1) (2011) 11–22.

[61] S. Abbasbandy, T. Allahvinloo., Numerical solutions of fuzzy differential equation, Math. Comput. Appl. 7 (1) (2002) 41–52. doi:10.3390/mca7010041.

[62] S. Abbasbandy, T. Allahvinloo., Numerical solutions of fuzzy differential equations by taylor method, Computational Methods in Applied Mathematics 2 (2) (2002) 113–124. doi:10.2478/cmam-2002-0006.

[63] S. Abbasbandy, T. Allahvinloo., Numerical solution of fuzzy differential equation by runge-kutta method, Nonlinear Studies 11 (1) (2004) 117–129.

[64] E. Hussian, M. Suhhiem., Numerical solution of fuzzy partial differential equations by using modified fuzzy neural networks, British Journal of Mathematics and Computer Science 12 (2) (2016) 1–20. doi:10.9734/BJMCS/2016/20504.

[65] M. Mosleh, M. Otadi., Simulation and evaluation of fuzzy differential equations by fuzzy neural network, Appl. Soft. Comput. 12 (9) (2012) 2817–2827. doi:https://doi.org/10.1016/j.asoc.2012.03.041.

[66] V. Streeter, E. Wylie, E. Benjamin., Fluid mechanics, 4th Ed, Mc. Graw-Hill Book Company.

[67] F. Beer, E. Johnston., Mechanics of materials, 2nd Edition, mcgraw-Hilly.

[68] M. Hazewinkel, Oscillator harmonic, Springer, ISBN.

[69] R. Pletcher, J. Tannehill, D. Anderson., Computational fluid mechanics and heat transfer, Taylor and Francis.