Fuzzy Control of Uncertain Nonlinear Systems with Numerical Techniques: A Survey

Raheleh Jafari[1], Sina Razvarz[2], Alexander Gegov[3], and Wen Yu[2]

[1] Centre for Artificial Intelligence Research (CAIR), University of Agder, Grimstad, Norway
raheleh.jafari@uia.no

[2] Departamento de Control Automático, CINVESTAV-IPN (National Polytechnic Institute), Mexico City, Mexico
srazvarz@yahoo.com
yuw@ctrl.cinvestav.mx

[3] School of Computing, University of Portsmouth, Buckingham Building, Portsmouth PO1 3HE, UK
alexander.gegov@port.ac.uk

Abstract. This paper provides an overview of numerical methods in order to solve fuzzy equations (FEs). It focuses on different numerical methodologies to solve FEs, dual fuzzy equations (DFEs), fuzzy differential equations (FDEs) and partial fuzzy differential equations (PFDEs). The solutions which are produced by these equations are taken to be the controllers. This paper also analyzes the existence of the roots of FEs and some important implementation problems. Finally, several examples are reviewed with different methods.

Keywords: fuzzy equations, solutions, numerical methods

1 Introduction

Fuzzy numbers have been used in many studies to deal with uncertainties in recent years [1–16]. Several approaches used the parametric form of fuzzy numbers for uncertainties in crisp dynamic systems [17]. In [18], the extension principle is implemented and suggests that the coefficients can be real or complex fuzzy numbers, where the system uncertainties are represented by fuzzy coefficients.

FEs are the equations whose parameters can be changed from the form of the fuzzy set [19]. The solutions of the FEs can be applied directly for modeling and nonlinear control. The results of the feedback control in reference to the wave equation is described in [20], while the open loop control corresponding to the wave equation is shown in [21]. However, the solutions are not easy to obtain. [22] introduced Newton's method. In [23], the fixed-point technique is used in order to obtain the solution of FEs. The numerical solutions of the FEs can be obtained applying the iterative method [24], the interpolation method [25] and Runge-Kutta method [26]. It can also be implemented to the differential equations. [27] portrays Euler's numerical technique methodology to solve the

FDE. The extension of classical fuzzy set theory in [28] results in obtaining a numerical solution of FDE. The wave solutions associated with two nonlinear PDE systems have been investigated in [29]. In [30] a static neural network is suggested to solve FDE. In [31] neural network is used in order to obtain the solution of the ordinary differential equation (ODE).

In this work, a survey is presented on the numerical solutions of the FEs, DFEs, FDE, and PFDE. The privileges of numerical techniques regarded to the precision are explained. The study of prior works demonstrates detail explanations in order to obtain numerical solutions for these equations. This paper is structured as follows. In Section 2, some basic definitions and notions used in the rest of the paper are given. In section 3, some numerical methods in order to find the numerical solutions of the FEs and the DFEs are presented. Section 4 describes some numerical methods in order to find the numerical solutions of the FDEs and PFDEs. In Section 5, some examples are provided with different methods in order to compare the efficiency of the numerical methods to approximate the solution of DFEs and FDEs. Conclusions are included in Section 6.

2   Mathematical preliminaries

The following definitions are used in this paper.

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

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 function is the triangular function

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

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

$$[x]^\alpha = \{a \in R : x(a) \geq \alpha\} \tag{3}$$

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

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

$$[x \oplus y]^\alpha = [x]^\alpha + [y]^\alpha = [\underline{x}^\alpha + \underline{y}^\alpha, \bar{x}^\alpha + \bar{y}^\alpha] \tag{4}$$

subtract,

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

or multiply,

$$\underline{z}^\alpha \leq [x \odot y]^\alpha \leq \overline{z}^\alpha \, or \, [x \odot y]^\alpha = A\left(\underline{z}^\alpha, \overline{z}^\alpha\right) \tag{6}$$

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

Therefore, $[x]^0 = x^+ = \{\zeta \in R, x(\zeta) > 0\}$. 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

$$[x]^\alpha = A\left(\underline{x}^\alpha, \bar{x}^\alpha\right) \tag{7}$$

3 Numerical techniques for solving fuzzy equations

In most cases, the analytical solution for FEs may not be found. Instead of using analytical methods that are not suitable for solving FEs and DFEs, numerical methods are proposed to solve these equations. In this section, we describe three numerical methods that are among the most important techniques.

3.1 Newton method

In 1671, Isaac Newton introduced a novel algorithm [32] for solving a polynomial equation which was demonstrated on the basis of an example as $x^3 - 2x - 5 = 0$. To obtain a precise root of the mentioned equation, initially, a starting value should be assumed, here $x \approx 2$. By assuming $x = 2 + q$ and substituting it into the original equation, the result is obtained as $q^3 + 6q^2 + 10q - 1 = 0$. As q is presumed to be minute, $q^3 + 6q^2$ is neglected in comparison with $10q - 1$. The previous equation generates $q \approx 0.1$, so a superior approximation of the root is $x \approx 2.1$. The repetition of this process is feasible and $q = 0.1 + a$ is extracted. The substitution gives $a^3 + 6.3a^2 + 11.23a + 0.061 = 0$, henceforth $a \approx -0.061/11.23 = -0.0054...$, so a novel approximation of the root is $x \approx 2.0946$. It is the requirement to repeat the process till the expected number of digits is achieved. In his methodology, Newton did not explicitly utilize the concept of the derivative, he just applied it on polynomial equations.

In [33] Newton's methodology is proposed for fuzzy nonlinear equations in lieu of standard analytical methodologies, as they are not appropriate everywhere. The primary intention is to extract a solution for fuzzy nonlinear equation $G(x) = b$. Primarily the cited researchers have mentioned fuzzy nonlinear equation in parametric form as illustrated below

$$\begin{cases} \underline{G}(\underline{x}^\alpha, \overline{x}^\alpha) = \underline{b}^\alpha \\ \overline{G}(\underline{x}^\alpha, \overline{x}^\alpha) = \overline{b}^\alpha \end{cases}$$

so they resolved it by utilizing Newton's methodology. Also, the convergence of the method is proved.

Newton's method is relatively expensive, since the calculation of the Hessian on the first iteration is needed. Accordingly, the analytic expression for the second derivative is often complicated or intractable, requiring a lot of computation.

3.2   Genetic algorithm method

A genetic algorithm is shown to solve the FE, $R(x) = y$ in [34], where x and y are considered real fuzzy numbers sampled with k, it is also considered that R is a fuzzy function relying on x. The motivation is to obtain an adequate value of the fuzzy argument x in such a way that the calculated value of the polynomial, $R(x)$, is very adjacent to the provided objective value y. The genetic algorithm presented uses a different demonstration of the fuzzy numbers that allows the implementation of simple genetic operators. The algorithm is self-sufficient to find multiple solutions associated with FEs. Unfortunately, no method has been used for an identical problem involved in the area of neural networks that can be taken over. Due to the different discrete criteria of fuzzy arithmetic, the only realistic approach to solve this problem is to design a dedicated genetic algorithm [35].

In [36] genetic algorithm is used for resolving non-linear equations of the form $g(x) = 0$, where x and $g(x)$ may be real, complex or vector quantities. At first $g(x) = 0$ is transformed into a minimization problem, then genetic algorithm is applied for finding the minimum. The method is extended for systems of non-linear equations.

The genetic algorithm represents the most consistent results in terms of accuracy and convergence but it is expensive in computational costs.

3.3   Neural network method

Approximation methods such as fuzzy neural networks are also effective tools to overcome the limitations of the other numerical methods. The major advantage of using fuzzy neural networks is training a large amount of data sets, quick convergence and high accuracy.

In [37] an architecture related to the fuzzy neural network is suggested in order to obtain a real root at par with fuzzy polynomials which is illustrated in the form mentioned below

$$C_1 x + ... + C_n x^n = C_0 \tag{8}$$

where $x \in \Re$ as well as $C_0, C_1, ..., C_n$ are fuzzy numbers. A learning algorithm associated with the cost function in order to adjust the crisp weights has been suggested. The methodology mentioned in [37] has drawbacks. It is solely capable of extracting a crisp solution of fuzzy polynomials, and this neural network cannot extract a fuzzy solution.

In [38] an architecture of fuzzy neural networks is suggested for solving dual fuzzy polynomial equations. A learning algorithm of fuzzy weights of two-layer feedforward fuzzy neural networks is used whose input-output relations are defined by extension principle.

4   Numerical techniques for solving fuzzy differential equations

Analytical methods can not be applied for solving FDEs. Therefore, numerical methods become essential, especially for PDEs. In this section three different important methods are described.

4.1   Taylor method

In [39], an approach on the basis of the 2nd Taylor technique is illustrated in order to resolve linear as well as nonlinear FDEs. The convergence order of the Euler technique in [40] is $O(h)$, whereas the convergence order in [39] is $O(h^2)$. The better solutions are extracted by [39].

In [41] the Taylor method of order p is utilized for solving FDEs. The algorithm is explained by resolving some linear and nonlinear fuzzy Cauchy problems. The convergence order of the Taylor method is $O(h^p)$.

The drawback of Taylor series technique is the computation of higher derivatives, that by increasing the order the calculation process becomes increasingly complicated. However, Runge-Kutta method is generally considered to be the most effective one-step technique.

4.2   Runge-Kutta method

In [42] an effective s-stage Runge-Kutta technique is employed for extracting the numerical solution of FDE. In that paper, Runge-Kutta method is applied for a more generalized category of problems and a convergence definition as well as error definitions are given at par with FDEs theory. Furthermore, convergence related to s-stage Runge-Kutta method is analyzed. This technique, when compared with developed Euler technique, performs superior. Although Euler technique is suitable, it is embedded with the disadvantage that, when analyzing the convergence of Euler technique [40], the authors generally investigate on the convergence of the ODEs system which happens while resolving numerically.

In [43] a numerical algorithm in order to solve linear as well as nonlinear fuzzy ODE on the basis of Seikkala derivative of fuzzy process is suggested. A numerical technique on the basis of the Runge-Kutta methodology of order five is elaborately investigated and this is carried on by going through an analysis of complete error. This technique with $O(h^5)$ outperforms than improved Euler's technique with $O(h^2)$.

In [44] a numerical algorithm in order to solve FDEs on the basis of Seikkala's derivative of a fuzzy process is suggested. A numerical technique based on a

Runge-Kutta Nystrom technique of order three is employed for solving the initial value problem, also it is illustrated that this methodology is superior in comparison with the Euler method by considering the convergence order of Euler methodology ($O(h)$) as well as Runge-Kutta Nystrom methodology ($O(h^3)$).

The major advantage of Runge-Kutta technique is that it is easy to apply. The main drawback of Runge-Kutta technique is that it needs more computer time when compared with multi-step techniques, also it does not easily yield desirable global approximations of the truncation error. The neural network is comparatively simple as well as computational rapid. Due to the superior estimation abilities of neural networks, the estimated solution for FDE is extremely near to the exact solution.

4.3   Neural network method

In [45] a modified technique is proposed in order to obtain the numerical solutions of fuzzy PDEs by utilizing fuzzy artificial neural networks. Utilizing modified fuzzy neural network ensures that the training points get selected over an open interval without training the network in the range of first and end points. This novel technique is on the basis of substituting each x in the training set (where $x \in [a, b]$) by the polynomial $Q(x) = \epsilon(x + 1)$ in such a manner that $Q(x) \in (a, b)$, by selecting an appropriate $\epsilon \in (0, 1)$. Also, it can be suggested that the proposed methodology can deal efficiently with all types of fuzzy PDEs as well as to generate precise estimated solution entirely for all domain and not only at the training set. Hence, one can utilize the interpolation methodologies (to be mentioned as curve fitting methodology) in order to obtain the estimated solution at points in the midst of the training points or at points outside the training set.

In [46] a novel technique on the basis of learning algorithm associated with the fuzzy neural network as well as Taylor series is laid down for extracting numerical solution of FDEs. A fuzzy neural network on the basis of the semi-Taylor series (in concerned to the function e^x) for the first (and second) order FDE is utilized. It is possible to use the same approach for solving high order FDE as well as fuzzy PDE. A fuzzy trial solution related to the fuzzy initial value problem is presented as an addition of two parts. The primary part suffices the fuzzy initial condition and it includes Taylor series, also contains no fuzzy adjustable parameters. The secondary part includes a feed-forward fuzzy neural network having fuzzy adjustable parameters (the fuzzy weights). Therefore by development, the fuzzy primary condition is sufficed and the training of the fuzzy network is carried out in order to suffice the FDE. The preciseness of this technique relies on the Taylor series that is selected for the trial solution. This selection is not distinct, hence, the preciseness is different from one problem to another problem. The suggested technique gives more precise estimations. Superior outcomes will be possible if more neurons or more training points are used. In addition, after resolving a FDE the solution is achievable at any arbitrary point in the training interval (even in the midst of training points).

5   Numerical Examples

In this section, two application examples are established to compare the performance of numerical techniques in order to estimate the solution of DFEs and FDEs.

Example 1. The water tank system contains two inlet valves V_1, V_2, as well as two outlet valves V_3, V_4, see Figure 1. The areas of the valves are uncertain as the triangle function (2), $C_1 = G(0.021, 0.023, 0.024)$, $C_2 = G(0.008, 0.018, 0.038)$, $C_3 = G(0.012, 0.013, 0.015)$, $C_4 = G(0.038, 0.058, 0.068)$. The velocities of the flow (controlled by the valves) are $g_1 = (\frac{\vartheta}{10})e^{\vartheta}$, $g_2 = \vartheta cos(\Pi\vartheta)$, $g_3 = cos(\frac{\Pi\vartheta}{8})$, $g_4 = \frac{\vartheta}{2}$. If the outlet flow is aimed to be $z = (4.088, 6.336, 36.399)$, what is the quantity of the control variable ϑ? The mass balance of the tank is [47]:
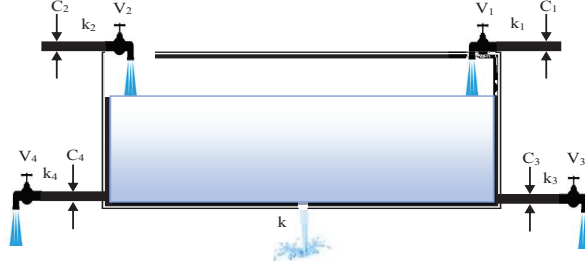

Fig. 1. Water tank system

$$\rho C_1 g_1 \oplus \rho C_2 g_2 = \rho C_3 g_3 \oplus \rho C_4 g_4 \oplus z$$

where ρ is considered to be the density of the water. The exact solution is taken to be $\vartheta_0 = 2$ [47]. To approximate the solution, we use three popular methods: Newton method, Genetic algorithm method, and Neural network method. The errors of these methods are shown in Table 1. It can be seen that all three methods can approximate the solutions of the dual fuzzy equations. Neural network method is more suitable for solving these kind of equations. In this table k is the number of iterations. The small approximation errors can be obtained by making the number of iteration larger. By increasing the number of iterations the estimated errors of the neural networks based algorithm are less than the other methods. Neural network method is more robust when compared with the other methods.

Example 2. The vibration mass system displayed in Fig. 2 is modeled by,

$$\frac{d}{dt}u(t) = \frac{c}{m}x(t), \quad u(t) = \frac{d}{dt}x(t) \tag{9}$$

Table 1. Approximation errors

k	Newton	Genetic algorithm	Neural network
1	0.18635	0.33463	0.43967
2	0.29602	0.24791	0.32375
3	0.36175	0.13012	0.21763
$\vdots$	$\vdots$	$\vdots$	$\vdots$
119	0.07982	0.04563	0.00316
120	0.07526	0.03952	0.00286

where the spring constant is considered to be $c = 1$, as well as the mass is $m = (0.73, 1.123)$. If the initial position is taken to be $x(0) = (0.73 + 0.23\alpha, 1.123 - 0.123\alpha)$, $\alpha \in [0, 1]$, so the exact solutions of (9) are [48],

$$x(t, \alpha) = \left[(0.73 + 0.23\alpha)e^t, (1.123 - 0.123\alpha)e^t \right] \tag{10}$$

where $t \in [0, 1]$. In order to estimate the solution (10), we utilize three popu-
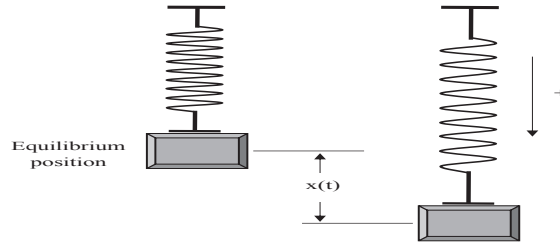
Fig. 2. Vibration mass

lar techniques: Taylor technique, Runge-Kutta technique, and Neural network technique. The errors related to these techniques are demonstrated in Table 2. Corresponding solution plots are displayed in Fig. 3. All three methods are suitable for resolving the FDEs. The leaning procedure of the neural network method is more quickly than the other methods. Also, the robustness of neural network method is better when compared with the other methods.

6   Conclusion

In this review, recent numerical methods are considered to solve FEs. It addresses numerical methodologies to solve FEs, DFEs, FDEs, and PFDEs. Research in

Table 2. Approximation errors

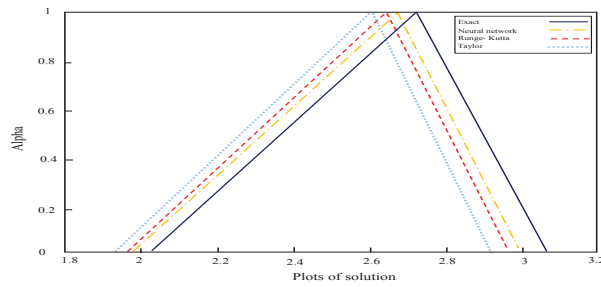| α	Taylor	Runge-Kutta	Neural network
0	[0.0604,0.1088]	[0.0407,0.0889]	[0.0209,0.0606]
0.2	[0.0701,0.1191]	[0.0609,0.1092]	[0.0308,0.0704]
0.4	[0.0511,0.0993]	[0.0211,0.0692]	[0.0102,0.0501]
0.6	[0.0403,0.0880]	[0.0211,0.0691]	[0.0011,0.0409]
0.8	[0.1009,0.1493]	[0.0712,0.1192]	[0.0512,0.0913]
1	[0.1104,0.1104]	[0.0806,0.0806]	[0.0602,0.0602]

Fig. 3. Comparison plot of three popular methods and the exact solution

this field continues to grow with new types of numerical methods and new strategies. Emphasis is given to current developments in the solving strategies in the last two decades, which demonstrates their significant improvements.

References

1. Jafari, R., Yu, W.: Uncertainty nonlinear systems control with fuzzy equations. In: IEEE International Conference on Systems, Man, and Cybernetics, 2885-2890doi:10.1109/SMC.2015.502, (2015).

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

3. Razvarz, S., Jafari, R., Yu, W.: Numerical Solution of Fuzzy Differential Equations with Z-numbers using Fuzzy Sumudu Transforms. In: IEEE International Conference on Industrial Technology (ICIT), Toronto, ON, 2017, pp. 890–895. doi: 10.1109/ICIT.2017.7915477. (2017)

4. Jafari, R., Yu, W.: Uncertainty nonlinear systems control with fuzzy equations. In: IEEE International Conference on Industrial Technology (ICIT), Toronto, ON, 2017, pp. 890–895. doi: 10.1109/ICIT.2017.7915477. (2015).

5. Jafari, R., Yu. W., Li, X.: Solving fuzzy differential equation with Bernstein neural networks. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, pp. 001245–001250. doi: 10.1109/SMC.2016.7844412. (2016).

6. Jafari. R., Yu, W.: Artificial neural network approach for solving strongly degenerate parabolic and burgers-fisher equations. In: 12th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, pp. 1–6. doi: 10.1109/ICEEE.2015.7357914, (2015).
7. Jafarian, A., Measoomy nia, S., Jafari, R.: Solving Fuzzy Equations Using Neural Nets with a New Learning Algorithm, Journal of Advances in Computer Research, 3 (4), pp. 33–45, (2012).
8. Jafari, R., Razvarz, S., Gegov, A., Paul, S.: Fuzzy Modeling for Uncertain Nonlinear Systems Using Fuzzy Equations and Z-Numbers. In: Advances in Computational Intelligence Systems. UKCI 2018. Advances in Intelligent Systems and Computing, 840, pp.96–107 (2019).
9. Jafari, R., Razvarz, S., Gegov, A., Paul, S., Keshtkar, S.: Fuzzy Sumudu Transform Approach to Solving Fuzzy Differential Equations With Z-Numbers, In: Advanced Fuzzy Logic Approaches in Engineering Science, pp.1–31, doi: 10.4018/978-1-5225-5709-8.ch002. (2019).
10. Jafari, R., Razvarz, S., Gegov, A.: Solving Differential Equations with Z-Numbers by Utilizing Fuzzy Sumudu Transform. In: IntelliSys 2018. Advances in Intelligent Systems and Computing, Springer, Cham, 869, pp. 1125–1138. https://doi.org/10.1007/978-3-030-01057-7.82. (2019).
11. Jafari, R., Razvarz, S., Gegov, A., Paul, S.: Modeling and Control of Uncertain Nonlinear Systems, In: International Conference on Intelligent Systems (IS), Funchal - Madeira, Portugal, pp. 168–173. doi: 10.1109/IS.2018.8710463. (2018).
12. Jafari, R., Razvarz, S., Gegov, A.: Fuzzy Differential Equations for Modeling and Control of Fuzzy Systems. In: 13th International Conference on Theory and Application of Fuzzy Systems and Soft Computing. ICAFS 2018. Advances in Intelligent Systems and Computing, Springer, Cham, 896. pp 732–740. doi: https://doi.org/10.1007/978-3-030-04164-9-96. (2019).
13. Jafari, R., Razvarz, S., Gegov, A.: A novel technique to solve fully fuzzy nonlinear matrix equations. In: 13th International Conference on Applications of Fuzzy Systems and Soft Computing. Advances in Intelligent Systems and Computing, Springer, (2018).
14. Jafari, R., Razvarz, S., Gegov, A.: A New Computational Method for Solving Fully Fuzzy Nonlinear Systems. In: Computational Collective Intelligence. ICCCI 2018. Lecture Notes in Computer Science, Springer. 11055, pp 503–512. doi: https://doi.org/10.1007/978-3-319-98443-8-46. (2018).
15. Razvarz, S., Jafari, R., Gegov, A., Yu, W., Paul, S.: Neural network approach to solving fully fuzzy nonlinear systems. In: Fuzzy Modelling and Control Nova Science Publishers. pp. 46–68. (2018).
16. Yu, W., Jafari, R.: Fuzzy Modeling and Control of Uncertain Nonlinear Systems (IEEE Press Series on Systems Science and Engineering) 1st Edition, 208 pages, Wiley-IEEE Press, ISBN: 978-1119491552, (2019).
17. Friedman, N., Ming, M., Kandel, A.: Fuzzy linear systems. Fuzzy Sets Syst, **96**, pp. 201–209, doi:https://doi.org/10.1016/S0165-0114(96)00270-9, (1998).
18. Buckley, J., Qu, Y.: Solving linear and quadratic fuzzy equations. Fuzzy Sets Syst, **35**, pp. 43–59, doi:https://doi.org/10.1016/0165-0114(90)90099-R, (1990).
19. Buckley, J., Hayashi. Y.: Can fuzzy neural nets approximate continuous fuzzy functions. Fuzzy Sets Syst, **61**, pp. 43–51, doi:https://doi.org/10.1016/0165-0114(94)90283-6, (1994).
20. Gibson, J.: An analysis of optimal modal regulation: convergence and stability. SIAM J. Control. Optim, **19**, pp. 686–707, doi:https://doi.org/10.1137/0319044, (1981).

21. Kröner, A., Kunisch, K.: A minimum effort optimal control problem for the wave equation. Comput. Optim. Appl, **57**, pp. 241–270, doi:https://doi.org/10.1007/s10589-013-9587-y, (2014).
22. Abbasbandy, S., Ezzati, R.: Newton's method for solving a system of fuzzy nonlinear equations. Appl. Math. Comput, **175**, pp. 1189–1199, doi:https://doi.org/10.1016/j.amc.2005.08.021, (2006).
23. Allahviranloo, T., Otadi, M., Mosleh, M.: Iterative method for fuzzy equations. Soft Computing, **12**, pp. 935–939, doi:https://doi.org/10.1007/s00500-007-0263-y, (2008).
24. Kajani, M., Asady, B., Vencheh, A.: An iterative method for solving dual fuzzy nonlinear equations. Appl. Math. Comput, **167**, pp. 316–323, doi:https://doi.org/10.1016/j.amc.2004.06.113, (2005).
25. Waziri, M., Majid, Z.: A new approach for solving dual fuzzy nonlinear equations using broyden's and newton's methods. Advances in Fuzzy Systems, pp. 1–5, doi:http://dx.doi.org/10.1155/2012/682087, (2012).
26. Pederson, S., Sambandham, M.: The runge-kutta method forhybrid fuzzy differential equation. Nonlinear Anal. Hybrid Syst, **2**, pp. 626–634, doi:https://doi.org/10.1016/j.nahs.2006.10.013, (2008).
27. Tapaswini, S., Chakraverty, S.: Euler-based new solution method for fuzzy initial value problems. Int. J. Artificial. Intell. Soft. Comput, **4**, pp. 58–79, doi:https://doi.org/10.1504/IJAISC.2014.059288, (2014).
28. Hüllermeier, E.: An approach to modeling and simulation of uncertain dynamical systems. Internat. J. Uncertainty Fuzzyness Knowledge-Based Syst, **5**, pp. 117–137, doi:https://doi.org/10.1142/S0218488597000117, (1997).
29. Guo, S., Mei, L., Zhou, Y.: 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**, pp. 804–816, doi:https://doi.org/10.1016/j.camwa.2015.02.016, (2015).
30. Effati, S., Pakdaman, M.: Artificial neural network approach for solving fuzzy differential equations. Inform. Sci, **180**, pp. 1434–1457, doi:https://doi.org/10.1016/j.ins.2009.12.016, (2010).
31. Agatonovic-Kustrin, S., Beresford, R.: Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. Journal of Pharmaceutical and Biomedical Analysis, **22**, pp. 717–727, doi:https://doi.org/10.1016/S0731-7085(99)00272-1, (2000).
32. Newton, I.: The method of fluxions and infinite series. London: Henry Woodfall. Retrieved from https://archive.org/details/methodoffluxions00newt, **3**, pp. 43–47, doi:ISBN1498167489,9781498167482, (1671).
33. Abbasbandy, S., Asady, B.: Newton's method for solving fuzzy nonlinear equations. Appl. Math. Comput, **159** pp. 349–356, doi:https://doi.org/10.1016/j.amc.2003.10.048, (2004).
34. Brudaru, O., Leon, F., Buzatu, O.: Genetic algorithm for solving fuzzy equations. In: 8th International Symposium on Automatic Control and Computer Science, Iasi,doi:ISBN973-621-086-3.
35. Michalewicz, Z.: Genetic algorithms+data structures=evolution program. In: 2nd ed., Artificial Intelligence, Springer Verlag, Berlin, doi:10.1007/978-3-662-03315-9.
36. Mastorakis, N.: Solving non-linear equations via genetic algorithms. In: 6th WSEAS Int. Conf. on EVOLUTIONARY COMPUTING, Lisbon, Portugal, pp. 24–28, (2005).

37. Abbasbandy, S., Otadi, M.: Numerical solution of fuzzy polynomials by fuzzy neural network. Appl. Math. Comput, **181**, pp. 1084–1089, doi:https://doi.org/10.1016/j.amc.2006.01.073, (2006).
38. Mosleh, M., Otadi, M.: A new approach to the numerical solution of dual fully fuzzy polynomial equations. Int. J. Industrial Mathematic, **2**, pp. 129–142, (2010).
39. Abbasbandy, S., Allahvinloo, T.: Numerical solutions of fuzzy differential equation. Math. Comput. Appl, **7** pp. 41–52, doi:10.3390/mca7010041, (2002).
40. Ma, M., Friedman, M., Kandel, A.: Numerical solutions of fuzzy differential equations. Fuzzy Sets Syst, **105**, pp. 133–138, (1999).
41. Abbasbandy, S., Allahvinloo, T.: Numerical solutions of fuzzy differential equations by taylor method. Computational Methods in Applied Mathematics, **2**, pp. 113–124, doi:10.2478/cmam-2002-0006, (2002).
42. Palligkinis, S., Papageorgiou, G., Famelis, I.: Runge-kutta methods for fuzzy differential equations. Applied Mathematics and Computation, **209**, pp. 97–105, doi:https://doi.org/10.1016/j.amc.2008.06.017, (2009).
43. Jayakumar, T., Maheskumar, D., Kanagarajan, K.: Numerical solution of fuzzy differential equations by runge-kutta method of order five. Appl. Math. Sci, **6**, pp. 2989–3002, (2012).
44. Kanagarajan, K., Sambath, M.: Runge-kutta nystrom method of order three for solving fuzzy differential equations. Computational Methods in Applied Mathematics, **10**, pp. 195–203, doi:https://doi.org/10.2478/cmam-2010-0011, (2010).
45. Hussian, E., Suhhiem, M.: Numerical solution of fuzzy partial differential equations by using modified fuzzy neural networks. British Journal of Mathematics and Computer Science, **12**, pp. 1–20, doi:10.9734/BJMCS/2016/20504, (2016).
46. Hussian, E., Suhhiem, M.: Numerical solution of fuzzy differential equations based on taylor series by using fuzzy neural networks. advance in mathematics, **11**, pp. 4080–4092, (2015).
47. Streeter, V., Wylie, E., Benjamin, E.: Fluid mechanics. In: 4th Ed, McGraw-Hill Book Company.
48. Hazewinkel, M.: Oscillator harmonic. In: Springer, ISBN.