Advances in Fuzzy Sets and Systems
© 2023 Pushpa Publishing House, Prayagraj, India
http://www.pphmj.com
http://dx.doi.org/10.17654/0973421X23001
Volume 28, 2023, Pages 1-19 **P-ISSN: 0973-421X**

FUZZY MODEL USING BERNSTEIN NEURAL NETWORK IN AN UNCERTAIN NON-LINEAR DIFFERENTIAL EQUATION

Jitendra Binwal[1], Arvind Maharshi[2] and Anita Mundra[1]

[1]Department of Mathematics
School of Liberal Arts and Sciences
Mody University of Science and Technology
India

[2]School of Engineering and Technology
Mody University of Science and Technology
India

Abstract

Nonlinear systems can also be simulated using fuzzy models, differential equations, and algebraic systems. Fuzzy systems are excellent models for uncertainty of nonlinear systems because nonlinear systems uncertainties can be converted into the fuzzy set concept. Fuzzy models employ many linear piecewise systems to

Received: September 29, 2022; Accepted: December 20, 2022
2020 Mathematics Subject Classification: 00A71, 03E72, 68Q10.
Keywords and phrases: BNN, bisection method, harmonic Newton's method, fuzzy model, non-linear differential equations.

How to cite this article: Jitendra Binwal, Arvind Maharshi and Anita Mundra, Fuzzy model using Bernstein neural network in an uncertain non-linear differential equation, Advances in Fuzzy Sets and Systems 28 (2023), 1-19. http://dx.doi.org/10.17654/0973421X23001

Published Online: January 10, 2023

approximate nonlinear systems with fuzzier outcomes. The fuzzy polynomial is an extended version of the fuzzy equation. The fuzzy equations are simpler to use than the standard fuzzy systems. There are a variety of ways to build fuzzy equations such as an interpolation technique, an iterative approach, and a Runge-Kutta technique that can be used to extract the statistical solution connected with fuzzy equations. In this work, it is studied that the previous research used fuzzy equations with *Z*-number coefficients to represent nonlinear systems with unknown parameters. A fuzzy model is suggested using Bernstein Neural Network (BNN) in uncertain non-linear differential equation in this research. The two methods bisection method and harmonic Newton's method are used to solve the uncertain non-linear differential equations. Few non-linear differential equations are used in the work to acquire findings. The results reveal that the bisection method produces accurate roots that are approximately zero.

1. Introduction

The fuzzy technique is an excellent tool for modeling uncertain nonlinear systems. Various linear systems are utilized to approximate uncertain nonlinear systems in fuzzy models such as the Takagi-Sugeno method. Mamdani models utilize fuzzy rules to obtain a high degree of uncertainty estimate [1, 2]. The use of fuzzy differential equations (FDE) for fuzzy control needs a solution. Various numerical methods are used to get the required outcome as the Nystrom technique is one example of a numerical approach [3]. In the second order FDE, the Laplace transform was applied [4]. FDEs may also be resolved using neural networks. The solution to an "ordinary differential equation (ODE)" can be calculated with a neural network [5, 6]. Various strategies have been used to extract the roots of the equations. The need for effective numerical algorithms in mathematical modeling as an alternative to traditional ways of using mathematics contributes significantly to the development of successful statistical methods. For regulating these systems, the fractional order model leads to the use of various control techniques and their extensions. The Lagrange and Euler-Lagrange equations have been extended to fractional order systems using fractional calculus [7].

In the frequency domain, approximate frequency diagrams have addressed fractional order system identification in conditions of dynamic evaluation and identification [8]. The order and limitations of a "fractional order system" can be estimated using an adaptive method. In the context of application control, it is necessary to evaluate the impact of any uncertainty or undetermined phrase in the dynamic of many systems. Adaptive updating of neural network coefficients is utilized to approximate an unknown component in a "dynamic fractional order system". Lyapunov stability criteria are used to verify the convergence of system of collapsed rings in the occurrence of fractional order rules and neural network estimators. After assessing the unknown component in system dynamics, differential calculus provides the optimum control rules for "fractional order systems". Because of the occurrence of a right-side operator and the absence of an online solution, optimum fractional order control cannot be used to generate a control signal and must instead be paired with an optimal controller and estimator utilizing step-by-step progression and predictive control principles [9].

The fuzzy polynomial is an extended version of the fuzzy equation. The fuzzy equations are simpler to use than the standard fuzzy systems. There are a variety of ways to build fuzzy equations such as an interpolation technique, an iterative approach, and a Runge-Kutta technique that can be used to extract the statistical solution connected with fuzzy equations [10-12].

2. Bernstein Neural Network

Output layer, Input node x, and a Bernstein basis function-based functional extension block are all combined in a single-layer BNN. The hidden layer had been eliminated utilizing the Bernstein, which converts the input form into a higher-dimensional space.

BNN is represented by $B_n(x)$, here x is the input data, $n = 0, 1, 2, \ldots.$

The Bernstein basis function has the following general form:

$$B_{j,n}(x) = C_n^j x^j (1-x)^{n-j}, \quad j = 0, 1, \ldots, n. \tag{1}$$

The symmetric and recursive formulae for the Bernstein are given as follows:

$$B_{j,n}(x) = B_{n-j,n}(1-x), \tag{2}$$

$$B_{j,n}(x) = (1-x)B_{j,n-1}(x) + xB_{j-1,n-1}(x). \tag{3}$$

Regard an m-dimensional input vector $x = (x_1, x_2, x_3, \ldots, x_m)$ as an improved form acquired by utilizing Bernstein polynomials:

$$[B_0(x_1), B_1(x_1), \ldots, B_n(x_1); B_0(x_2), B_1(x_2), \ldots, B_n(x_2); \ldots,$$
$$B_0(x_m), B_1(x_m), \ldots B_n(x_m)]. \tag{4}$$

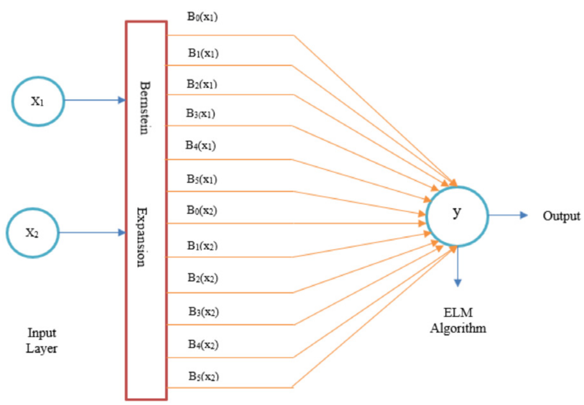

Figure 1. Structure of a Bernstein neural network [13].

BNN can be used to convert an m-dimensional input vector to an n-dimensional one, as shown in Figure 1, which can be applied to a single-layer neural network $(n > m)$. To solve differential equations, this BNN model can be used [13].

3. Review of Literature

The following work extends the previous review of the fuzzy model using BNN in uncertain non-linear differential equations. Nassajian et al. [14] explained new methods for optimum control of fractional order systems when an unknown component is present, where the fractional order derivative is between 0 and 1. For the unknown term in the system dynamics, a neural network might be employed. Adaptive and real-time updates are made to neural network coefficients. The system requirements for achieving a homogenous fractional order system are considered while presenting the updating laws. Fractional differential calculus may also be used to formulate optimum control rules for fractional order systems. To get a control signal, a step-by-step sequence and the analytical control notion are utilized to combine the ideal controller and estimator, which is non-causal. There is no unknown term in the system dynamic that cannot be solved by this approach. Direct Lyapunov methods are used to verify that the closed loop system is uniformly bounded. The final simulation results demonstrate the efficiency of the suggested method.

Gegov and Yu [15] focused on numerical approaches for solving fuzzy equations (FEs) discussed in the study. For example, it focuses on several mathematical approaches for explaining Partial Fuzzy Derivative Equations (PFDEs), Fuzzy Derivative Equations, and Dual Fuzzy Equations. These equations generate solutions, and those solutions are regarded as controllers. The presence of FE roots and certain significant implementation issues are also discussed.

Jafari et al. [16] reviewed the approaches for modeling and controlling unpredictable nonlinear systems. The main criterion for highlighting the work is the use of multiple strategies for solving FDEs that relate to the fuzzy controllability problem. The controllers are the equalities that create these solutions. Currently, numerical approaches have emerged as better ways to address these kinds of issues. The use of neural networks in dealing with fuzzy systems adds to the complexity of the problem-solving process.

Razvarz et al. [17] explored that *Z*-numbers used to represent the coefficients and variables of a fuzzy equation are used in this work to illustrate the uncertainty property. Nonlinear system modeling with unknown parameters is well-suited to this fuzzy equation modification. Uncertain nonlinear systems are represented here via fuzzy equations. It is the goal of modeling uncertain nonlinear systems to determine the coefficients of fuzzy equation.

Yu et al. [18] examined that *Z*-numbers can be defined by combining the fuzzy set theory into FDEs. On the other side, FDEs are very complex to solve. FDE solutions are assessed utilizing two kinds of neural networks. In this case, the *Z*-numbers represent the uncertainty. The FDE is first converted into four ordinary differential equations (ODEs) with Hukuhara differentiability. Neural models can be built once the ODE framework is established. The neural networks are trained using a modified back propagation approach for *Z*-number variables. It is possible to approximate their solutions with high accuracy using BNN when solving FDEs based on *Z*-numbers.

4. Background of the Study

The uncertainty of the linear-in parameters may be used to simulate a wide variety of nonlinear systems. *Z*-number coefficients are used to represent unknown parameters. The models of uncertain nonlinear systems are represented using fuzzy equations. This means that fuzzy equations are considered controllers, whereas desired references are outputs. The assumption of the existence of a solution is given. Neuronal networks with two different structures are used to approximate fuzzy *Z*-number coefficients [19].

5. Formulation of the Problem

A variety of strategies for converting nonlinear equations (NEs) into optimization problems would be addressed in further detail. Nonlinear equations are among the most difficult to solve, and as a result, these get less attention than other forms of equations. Various nonlinear optimization problems need several variables while keeping to severe constraints. Nonlinearity in the goal function is often the cause of the emerging multimodal objective function.

6. Research Methodology

The goal of this paper is to find the most effective approach for finding solutions of nonlinear equations. Most NEs cannot be solved in a finite number of steps. Nonlinear equations are explained using the bisection method. The consequence is that the faster the rate of convergence, the faster the equation would approach its approximate root or solution. As a result, the harmonic Newton's approach was suggested as the optimum method for solving nonlinear equations $F(x) = 0$! comprising of one variable as of its maximum rate of convergence.

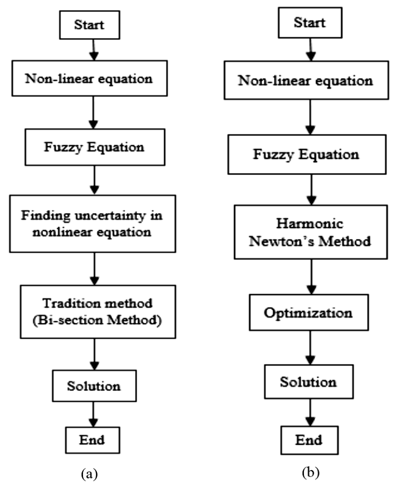

Figure 2. Solution of nonlinear fuzzy equation utilizing (a) Bisection method, (b) Harmonic Newton's method.

Figure 2 illustrates a flow diagram of a nonlinear equation using two methods, which are referred to as the bisection method and the harmonic Newton's method. The bisection technique is a generic approach that may be used to determine the nonlinear equation with fuzzy numbers when there are many variables.

- **Fuzzy equations and *Z*-number with nonlinear system modeling**

Z-numbers and fuzzy equations can be implemented using notions from *Z*-numbers and discrete-time nonlinear systems.

The following is a description of a nonlinear discrete-time system:

$$\overline{v}_{k+1} = \overline{p}[\overline{v}_k, w_k], \ z_k = \overline{q}[\overline{v}_k]. \tag{5}$$

Here, $w_k \in \Re^u$! is an input vector, $\bar{v}_k \in \Re^l$ is deemed as an internal state vector, $z_k \in \Re^m$ is the output vector, $\bar{p}$ and $\bar{q}$ are deemed as comprehensive nonlinear smooth functions, and $\bar{p}, \bar{q} \in C^\infty$. Set $Z_k = [z_{k+1}^T, z_k^T, \ldots]^T$ and $W_k = [w_{k+1}^T, w_k^T, \ldots]^T$. Presume that $\frac{\partial z}{\partial \bar{v}}$ is non-singular at $\bar{v}_k = 0, w_k = 0,$ which would expand the subsequent standard [19]:

$$z_k = \Gamma[z_{k-1}^T, z_{k-2}^T, \ldots, w_k^T, w_{k-1}^T \ldots]. \tag{6}$$

Here, w_k and z_k are shown as the quantifiable scalar input and output separately. $\Gamma(\cdot)$ is a nonlinear variation equivalence explaining the plant dynamics. The input of the system is shown to be as

$$v_k = [z_{k-1}^T, z_{k-2}^T, \ldots, w_k^T, w_{k-1}^T, \ldots]^T. \tag{7}$$

According to equation (7), a linear-in-parameter model for nonlinear systems may be simplified as follows:

$$z_k = \sum_{i=1}^n c_i g_i(x_k), \tag{8}$$

where $g_i(x_k)$ is a nonlinear function and c_i's are taken to be linear parameters.

Definition 1. A *Z-number* includes the dual number $Z = [S(\zeta), p]$. Real-valued uncertain variables are restricted by the main module $S(\zeta)$. There are many ways to describe the second module of *S*. *P* is dependability on reliability, conviction, and probability. The number Z^+ is known as the *Z*-number, $S(\zeta)$ is a fuzzy number, and *p* is the probability distribution of ϑ.

The number Z^+ contains additional data when compared with Z^-. It utilizes the meaning of Z^+ number, i.e., $Z = [S, p]$, here *p* is a probability distribution and *S* is a fuzzy number. The membership functions are often used to demonstrate fuzzy numbers. The best often used membership functions are the triangle function defined as

$$\mu_s = \psi(a, b, c) = f(x) = \begin{cases} \dfrac{\zeta - a}{b - a}, & a \le \zeta \le b \\ \dfrac{c - \zeta}{c - b}, & b \le \zeta \le c, \end{cases} \quad \text{otherwise } \mu_s = 0 \qquad (9)$$

and the trapezoidal function defined as

$$\mu_s = \psi(a, b, c, d) = \begin{cases} \dfrac{\zeta - a}{b - a}, & a \le \zeta \le b \\ \dfrac{d - \zeta}{d - c}, & c \le \zeta \le d \\ 1, & b \le \zeta \le c, \end{cases} \quad \text{otherwise } \mu_s = 0. \qquad (10)$$

The probability is illustrated as $P = \int_R^0 \mu_s(\zeta) p(\zeta) d\zeta,$ where R is shown to be the limitation on P and p is the probability density of ζ. For a discrete Z-number, $P(s) = \sum_{i=1}^{n} \mu_s(\vartheta_i) p(\vartheta_i).$ The space of discrete fuzzy sets is characterized as $\widetilde{E}$. $\widetilde{E}_{[a, b]}$ implies the space of discrete fuzzy sets of $[a, b] \subset R.$

The space of discrete Z-numbers is defined by

$$\widetilde{Z} = \{Z = (A, p) | S \in \widetilde{E}, p \in \widetilde{E}_{[0,1]}\}. \qquad (11)$$

- **Method for solving a fuzzy non-linear equation**

Fuzzy nonlinear equations are the parameters of fuzzy numbers. A generic bisection approach has been suggested in parametric form for the mathematical explanation of a method of fuzzy NEs.

- **Pseudo Code**

In the work, a researcher writes pseudo code for the suggested methods, i.e., bisection method and harmonic Newton’s method as given below in Table 1 and Table 2, respectively.

Table 1. Pseudo code of bisection method

```
Pseudo Code of Bisection Method
a = Input ('Enter function with right hand side zero's:');
f = inline(a);
x1 = Input ('Enter the first value of guess interval:');
xu= Input ('Enter the end value of guess interval:');
tol= Input ('Enter the allowed error:');
if f(xu)*f(xl)<0
else
   print ('The guess is incorrect! Enter new guesses\n');
   xl= input ('Enter the first value of guess interval:\n');
   xu=input ('Enter the end value of guess interval:\n');
end
fis = read fis ('fuzzy rules')
for i =2:1000
xr=(xu+xl)/2;
if f(xu)*f(xr)<0
   xl=xr;
else
   xu=xr;
end
if f(xl)*f(xr)<0
   xu=xr;
else
   xl = xr;
end
x new (1) = 0;
x new (i) = xr;
if abs ((x new (i) -x new (i-1))/x new (i)) <tol, break, end
end
```

```
x new (1) = 0;
x new (i) = xr;
if abs ((x new (i) -x new (i-1))/x new (i)) <tol, break, end
end
str = ['The required root from bisection method is: ', num2str(xr), '']
syms a(t) b(t) c(t) d e x
eqn1 = d*diff(a(t), 2) == c(t)/e*a(t);
eqn2 = d*diff(b(t), 2) == c(t)/e*b(t) - d*x;
eqn3 = a(t)^2 + b(t)^2 == e^2;
eqns = [eqn1 eqn2 eqn3];
vars = [a(t); b(t); c(t)];
orig Vars = length(vars);
M = incidence matrix (eqns, vars);
[eqns, vars] = reduce Differential Order (eqns, vars);
Is Low Index DAE (eqns, vars);
[DAEs, DAEvars] = reduced index (eqns, vars);
[DAEs, DAEvars] = reduce redundancies (DAEs, DAEvars);
Is Low Index DAE (DAEs, DAEvars);
pDAEs = symvar (DAEs);
pDAEvars = symvar(DAEvars);
extra Params = setd iff (pDAEs, pDAEvars);
f = dae Function (DAEs, DAE vars, x,d,e);
x = 9.81;
d = 1;
e = 1;
F = @(t,Y,YP) f(t,Y,YP,x,d,e);
DAE vars;
y0est = [e*sin(pi/6); -e*cos(pi/6); 0; 0; 0; 0; 0];
yp0est = zeros (7,1);
opt = odeset ('Rel Tol', 10.0^(-7),'AbsTol',10.0^(-7));
[y0,yp0] = decic(F,0,y0est,[],yp0est,[],opt);
[tSol, ySol] = ode15i(F,[0 0.5],y0,yp0,opt);
for k = 1:orig Vars
  S{k} = char (DAEvars(k));
end
```

Table 2. Pseudo code of harmonic Newton method

```
Pseudo code of Harmonic Newton's method
Tolerance = 0.001;
X=0:0.1:5;
fis = read fis ('fuzzy rules')
N max= 20;
guess = 2;
new =guess - (R(guess)/Rd(guess));
eps=abs(new-guess);
i=1;
while eps> = tolerance && i <= N max
        new1 = new-(R(new)/Rd(new));
        eps=abs(new1-new);
        new=new1;
        i=i+1;
end
L=new;
disp 'root using newton-hamilton is')
disp (L)
main1
```

• **Bisection method**

An equation of the form $f(x) = 0$ whose solution is in the range of (a, b) is to be determined. The function $f(x)$ must be continuous and might be algebraic or transcendental. There must be at least one root between a and b if $f(b)$ and $f(a)$ are of opposite signs.

As an initial estimate, it is assumed that the root is $x_0 = \frac{a+b}{2}$ (midpoint of the ends of the range). x_0 and b are the roots of $f(x_0)$, which is negative and each one of these statements is accurate. For this approach, the interval is repeatedly divided into half, with each iteration selecting the half that likewise meets this sign criterion [20]. The relation may be utilized to calculate the number of iterations required $\left|\frac{b-a}{2^k}\right| \leq \varepsilon$. The usual formulation is as follows:

$$x_k = \frac{a+b}{2}. \tag{12}$$

• **Harmonic Newton's method**

Consider the problem of determining a function's actual zero $F : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$. A real solution (α) of the nonlinear equation system $F(x) = 0$ of n variables can be found as a fixed point of various functions $G : \mathbb{R}^n \to \mathbb{R}^n$ by the method of the fixed-point iteration technique:

$$x_{k+1} = G(x_k), \quad k = 0, 1, \ldots. \tag{13}$$

Here, x_k represents the original estimate. Newton's technique is the most well-known fixed-point approach, given by

$$x_{k+1} = x_k - J_F(x_k)^{-1} F(x_k), \quad k = 0, 1, 2, \ldots. \tag{14}$$

Note that $J_F(x_k)$ is the Jacobian matrix of the function F assessed in x_k, $F : D \subseteq \mathbb{R}^n \to \mathbb{R}^n$ is an appropriately differentiable function and α is a zero method of NEs [21].

7. Results Analysis

The performance of the results is the primary focus of this section. The softwares which are used at the time of implementation of the work are described below.

- Software: MATLAB is used as the software in the work. MATLAB is the preferred platform for engineers and scientists when analyzing and creating systems and solutions that have the potential to change the game. At its center is the MATLAB language, a matrix-based language that allows for the most intuitive expression of computer mathematics. MATLAB R2021a is a MATLAB version that is taken into consideration in this work.

Consider the equation:

$$x^3 - x - 1 = 0. \tag{15}$$

The root of equation (15) lies between 1 and 2. The approximate root of equation (15) found by using the bisection method is 1.3203 and by using harmonic Newton's method is 1.3247.

Next, consider

$$x^2 - 3 = 0. \tag{16}$$

A root of equation (16) lies between 1 and 2. The harmonic Newton method yields the value of 1.3653 as a root for equation (16), whereas the bisection method gives 1.3753.

For

$$x^3 + 4x^2 - 10 = 0, \tag{17}$$

a root lies between 3 and 4. The approximate root of equation (17) determined using the bisection method is 3.2813, whereas the approximate root found using the harmonic Newton method is 0.155.

A root of

$$x^2 - 1 = 0, \tag{18}$$

lies between 0 and 3. The approximate root of equation (18) found by using the bisection method is 1.002 and by using harmonic Newton method is 1.

A root of

$$10 - x^2 = 0 \tag{19}$$

lies between -2 and 5. The approximate root of equation (19) found by using the bisection method is 3.168 and by using harmonic Newton method is 3.1623.

Consider, next

$$x^3 + x - 1 = 0. \tag{20}$$

The approximate root of equation (20) found by using the bisection method is 3.5156 and by using harmonic Newton's method is 3.4495.

A root of

$$x^2 - 2x - 5 = 0, \tag{21}$$

lies between 0 and 1. By utilizing the bisection method, the estimated root of equation (21) is determined to be 0.7382, and by applying the harmonic Newton method, the root is found to be 0.7391.

For the equation

$$x^3 - 4x - 9 = 0 \tag{22}$$

a root lies between 2 and 3. The approximate root of equation (22) is 2.7031 using the bisection method and 2.7065 through the harmonic Newton method.

A root of

$$x^3 - 5 = 0, \tag{23}$$

is situated between 1 and 2. Bisection method and harmonic Newton method provide 1.7031 and 1.71, respectively, as the approximate roots of equation (23).

Finally, consider

$$x^4 - x^3 - x^2 - x = 0. \tag{24}$$

A root of equation (24) lies between 0 and 1. The approximate root of equation (24) found by using the bisection method is 1.832 and by using harmonic Newton method is 1.8393.

Table 3 shows the roots of fuzzy equations from 15 to 24 which are obtained by utilizing the bisection method and the graph of the values is shown in Figure 3.

Table 3. Approximate roots by bisection method

Equation No.	Bisection Method
15	1.3203
16	1.7344
17	1.3753
18	1.002
19	3.168
20	0.68359
21	3.5156
22	2.7031
23	1.7031
24	1.832

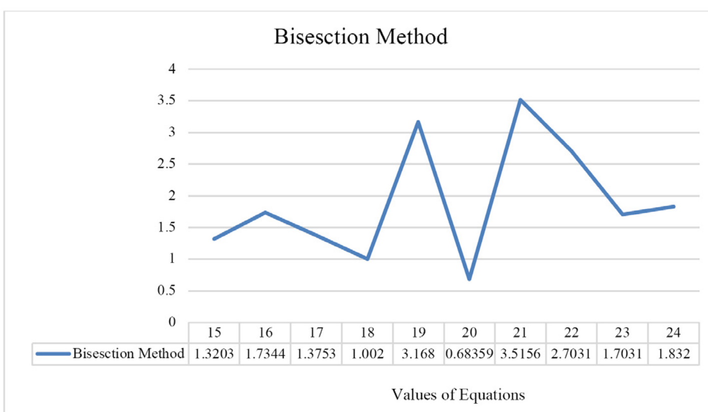

Figure 3. Graph of roots by bisection method.

The results of the harmonic Newton's method for solving fuzzy equations from 15 to 24 are listed in Table 4, and their corresponding graphs are shown in Figure 4.

Table 4. Approximate roots by harmonic Newton's method

Equation No.	Harmonic Newton's Method
15	1.3247
16	1.7321
17	1.3653
18	1
19	3.1623
20	0.6823
21	3.4495
22	2.7065
23	1.71
24	1.8393

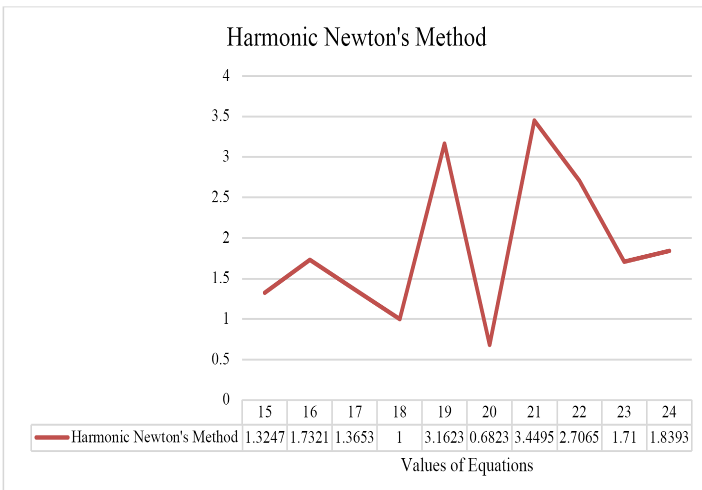

Figure 4. Graph of roots by harmonic Newton's method.

The overall results show that the values which are obtained by the bisection method are better than the values which are obtained by harmonic Newton's method.

8. Conclusion and Future Work

Solutions to fuzzy equations might be studied using a wide variety of methods. Previous studies have shown that fuzzy equations with *Z*-number coefficients can be useful for modeling uncertain nonlinear systems. Fuzzy equations are used to describe uncertain nonlinear equations. In the paper, a fuzzy model using BNN is proposed to solve uncertain non-linear differential equations. The bisection method and harmonic Newton's method are used at the time of implementation in MATLAB to obtain the results. Ten uncertain non-linear differential equations are taken in the work and the roots of these equations are found by utilizing the bisection method and harmonic Newton's method. The roots of the equation from 15 to 24 are found by using the bisection method. These are 1.3203, 1.7344, 1.3753, 1.002, 3.168, 0.68359, 3.5156, 2.7031, 1.7031 and 1.832, respectively, which are approximately zero. The roots of the equations from 15 to 24 have been determined using harmonic Newton's method. These are 1.3247, 1.7321, 1.3653, 1, 3.1623, 0.6823, 3.4495, 2.7065, 1.71, and 1.8393, respectively, which are all nearly zero. It can be observed from the results, that the bisection method gives the roots of equations approximately zero which means bisection is better than harmonic Newton's method. In future, the research can be carried by using other mathematical tools such as Newton's Raphson method and Euler's Modified method to get the root nearer to zero.

References

[1] Yang, Gengjiao, Fei Hao, Lin Zhang and Bohu Li, Exponential stability of discrete-time positive switched TS fuzzy systems with all unstable subsystems, Science China Information Sciences 64 (2021), 1-3.

[2] Kwang Y. Lee and Mohamed A. El-Sharkawi, eds., Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems, Vol. 39, John Wiley & Sons, 2008.

[3] M. Chehlabi and Tofigh Allahviranloo, Positive or negative solutions to first-order fully fuzzy linear differential equations under generalized differentiability, Applied Soft Computing 70 (2018), 359-370.

[4] M. B. Ahmadi, N. A. Kiani and N. Mikaeilvand, Laplace transform formula on fuzzy *n*th-order derivative and its application in fuzzy ordinary differential equations, Soft Comput. 18 (2014), 2461-2469.

[5] Muhammad Asif Zahoor Raja, Jabran Mehmood, Zulqurnain Sabir, A. Kazemi Nasab and Muhammad Anwaar Manzar, Numerical solution of doubly singular nonlinear systems using neural networks-based integrated intelligent computing, Neural Computing and Applications 31(3) (2019), 793-812.

[6] S. Agatonovic-Kustrin and R. Beresford, Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research, Journal of Pharmaceutical and Biomedical Analysis 22 (2000), 717-727.

[7] Ricardo Almeida and Delfim FM Torres, Necessary and sufficient conditions for the fractional calculus of variations with Caputo derivatives, Communications in Nonlinear Science and Numerical Simulation 16(3) (2011), 1490-1500.

[8] Milan R. Rapaić and Alessandro Pisano, Variable-order fractional operators for adaptive order and parameter estimation, IEEE Transactions on Automatic Control 59(3) (2013), 798-803.

[9] Andrzej Dzieliński and Dominik Sierociuk, Observer for discrete fractional order state-space systems, IFAC Proceedings 39(11) (2006), 511-516.

[10] M. Tavassoli Kajani, B. Asady and A. Hadi Vencheh, An iterative method for solving dual fuzzy nonlinear equations, Applied Mathematics and Computation 167(1) (2005), 316-323.

[11] Mohammed Yusuf Waziri and Zanariah Abdul Majid, A new approach for solving dual fuzzy nonlinear equations using Broyden's and Newton's methods, Advances in Fuzzy Systems 2012 (2012).

[12] S. Pederson and M. Sambandham, The Runge-Kutta method for hybrid fuzzy differential equations, Nonlinear Analysis: Hybrid Systems 2(2) (2008), 626-634.

[13] Hongli Sun, Muzhou Hou, Yunlei Yang, Tianle Zhang, Futian Weng and Feng Han, Solving partial differential equation based on Bernstein neural network and extreme learning machine algorithm, Neural Processing Letters 50(2) (2019), 1153-1172.

[14] Gholamreza Nassajian and Saeed Balochian, Optimal Control Based on Neuro Estimator for Fractional Order Uncertain Non-linear Continuous-time Systems, Neural Processing Letters 52(1) (2020), 221-240.

[15] Alexander Gegov and Wen Yu, Fuzzy control of uncertain nonlinear systems with numerical techniques: a survey, In UK Workshop on Computational Intelligence, Springer, Cham, 2019, pp. 3-14.

[16] Raheleh Jafari, Sina Razvarz and Alexander Gegov, Fuzzy differential equations for modeling and control of fuzzy systems, In International Conference on Theory and Applications of Fuzzy Systems and Soft Computing, Springer, Cham, 2018, pp. 732-740.

[17] Sina Razvarz, Raheleh Jafari, Alexander Gegov and Satyam Paul, Fuzzy modeling for uncertain nonlinear systems using fuzzy equations and *Z*-numbers, In UK Workshop on Computational Intelligence, Springer, Cham, 2018, pp. 96-107.

[18] Wen Yu, Sina Razvarz and Raheleh Jafari, Numerical solution of fuzzy differential equations with *Z*-numbers using fuzzy Sumudu transforms, Advances in Science, Technology and Engineering Systems 3(1) (2018), 66-75.

[19] Sina Razvarz and Mohammad Tahmasbi, Fuzzy equations and *Z*-numbers for nonlinear systems control, Procedia Computer Science 120 (2017), 923-930.

[20] Okorie Charity Ebelechukwu, Ben Obakpo Johnson, Ali Inalegwu Michael and Akuji Terhemba Fidelis, Comparison of some iterative methods of solving nonlinear equations, International Journal of Theoretical and Applied Mathematics 4(2) (2018), 22.

[21] Javad Shokri, Numerical method for solving fuzzy nonlinear equations, Applied Mathematical Sciences 2(24) (2008), 1191-1203.