1

Fuzzy Equations

1.1 Introduction

Fuzzy equations are a widespread problem in many applied fields, such as production planning, optimization decision, and artificial intelligence, in which establishing general and operable solving methods has received a remarkable amount of attention in academic circles. The fuzzy equation can be regarded as a generalized form of the fuzzy polynomial. Compared with normal fuzzy systems, fuzzy equations are easier to apply because the uncertainties are direct fuzzy parameters of the fuzzy equations.

1.2 Fuzzy Equations

The fuzzy method is a highly favorable tool for uncertain nonlinear system modeling. Fuzzy models approximate uncertain nonlinear systems with several linear piecewise systems (Takagi-Sugeno method) [182]. Mamdani models use fuzzy rules to achieve a good level of approximation of uncertainties [122]. When the parameter of an equation is changeable in the manner of a fuzzy set, this equation becomes a fuzzy equation [51]. When the parameters or the states of the differential equations are uncertain, they can be modeled with fuzzy differential equations (FDEs).

In comparison with normal systems, fuzzy equations are considered to be very non-complex. It is feasible for these equations to apply directly for nonlinear control. The approach of fuzzy control is associated with the design of appropriate nonlinear functions in the fuzzy equation. Fuzzy control with fuzzy equations requires the solution of the fuzzy equation. Several approaches are incorporated. In [70], the parametric form of fuzzy numbers is utilized and the original fuzzy equations using crisp linear systems are restored. In [52], the extension principle is implemented, which suggests that the coefficients can be either real or complex fuzzy numbers. However, the validation of the solution is not assured. In [8], the Newton technique is proposed. In [16], the solution

COPYRIGHTED MATERIAL

Modeling and Control of Uncertain Nonlinear Systems with Fuzzy Equations and Z-Number,
First Edition. Wen Yu and Raheleh Jafari.
© 2019 by The Institute of Electrical and Electronics Engineers, Inc. Published 2019 by John Wiley & Sons, Inc.

of fuzzy equations is extracted using a fixed point methodology. The numerical solutions associated with fuzzy equations can be extracted using the iterative technique [108], interpolation technique [192], and Runge–Kutta technique [154]. They can also be implemented on partial differential equations (PDEs) and FDEs. In [185] the methodology of the Euler numerical technique is used to resolve the FDEs. The extension of classical fuzzy set theory in [83] results in obtaining the numerical solutions to FDEs. Some numerical approaches, such as the Nyström method [113] and the Runge–Kutta method [151] can also be implemented for resolving FDEs. The Laplace transform was utilized for second order FDEs in [20]. Several researchers have implemented the finite element technique, which has been used in the area of mechanics for solving a few specific PDEs [82]. Investigations by researchers revealing double non-traveling wave solutions associated with two systems of nonlinear PDEs are mentioned in [75]. The results of feedback control in reference to the wave equation are illustrated in [74], whereas open loop control concerning the wave equation is demonstrated in [115].

Both neural networks and fuzzy logic are considered to be the universal estimators that can estimate any nonlinear function to any notable precision [61][162][163]. Current outcomes demonstrate that the fusion methodology of these two different techniques appears to be highly efficient for nonlinear system identification [195]. Neural networks can also be implemented for resolving fuzzy equations [93][94]. A generalized fuzzy quadratic equation is resolved by utilizing neural networks, which is mentioned in [47]. In [103], the outcomes of [47] are elaborated into the fuzzy polynomial equation. Neural networks have been utilized in order to extract the solution of dual fuzzy equations, as illustrated in [95]. A matrix pattern associated with neural learning is proposed in [132]. However, these techniques are not general as they cannot resolve general fuzzy equations with neural networks. Also, they cannot generate fuzzy coefficients directly with neural networks [181]. In [67], a static neural network is proposed in order to resolve FDEs. In [15], the authors illustrate that the solutions to ordinary differential equations (ODEs) can be estimated with the help of neural networks. In [194], the neural approximations of ODEs to dynamic systems is implemented. In [118], dynamics neural networks are implemented for the approximation of the first order ODE. In [66], a feed forward neural network is suggested in order to resolve an elliptic PDE in 2D. The other methodology for solving a class of first order PDEs on the basis of multilayer neural networks is demonstrated in [79]. In [91], the neural network approach is used for solving strongly degenerate parabolic and Burgers–Fisher equations. The investigations of [129] laid down an unsupervised neural network to resolve the nonlinear Schrödinger equation. In [179], by employing a feed forward neural network, the controlled heat problem is solved.

1.3 Algebraic Fuzzy Equations

Some investigations have been carried out on algebraic fuzzy equations, but the existing methodologies calculate the roots of an algebraic fuzzy equation analytically. Also, there exists no analytical solution related to algebraic fuzzy equations having degree greater than 3 [48]. Thus utilization of numerical methodologies for such equations is essential. In [30], the author presented an algebraic fuzzy equation with degree n, including fuzzy coefficients and crisp variables, which is stated by

$$a_n x^n + \cdots + a_1 x + a_0 = 0 \tag{1.1}$$

where $0, a_0, a_1, \dots, a_n \in E$ and $a_n \neq 0$. That is $P_n(x) = 0$, where $P_n(x) = \sum_{j=0}^{n} a_j x^j$ is mentioned as a polynomial of degree n. In order to determine the roots of the stated algebraic fuzzy equation an algorithm on the basis of Gauss–Newton technique produces a series that can converge under the condition that the modal value function includes a root or else the series can diverge. In addition, if the equation contains more than one root then the roots can be obtained via different initial vectors. In order to solve the algebraic fuzzy equation, the root and fuzzy zero are assumed to be unknowns, therefore using a series they can be determined. In the case that a fuzzy zero is provided, then only the root of the algebraic fuzzy equation should be extracted.

Consider the equations below

$$F(X) - B = 0, \quad F(X) = B \tag{1.2}$$

where B is an interval or fuzzy value and $F(X)$ is taken to be some interval or fuzzy function. These equations are not deemed to be equivalent. However, the major problem is linked to the conventional interval or fuzzy extension of the usual equation, which results in the interval or fuzzy equation $F(X) - B = 0$. An interval exists on the left hand side of this elongated equation, whereas a real valued zero exists on the right hand side. As it is not possible for the interval to be equal to the real value, this result is termed as "the interval equation right hand side problem". Minimal problems will occur while dealing with interval or fuzzy equations as $F(X) = B$, but in many issues its roots are termed as inverted intervals, i.e. in such a manner that $\overline{x} < \underline{x}$ [174].

The linear equation considered in [174] is mentioned as

$$ax = b \tag{1.3}$$

where its algebraically equivalent forms are denoted as

$$x = \frac{b}{a} \tag{1.4}$$

$$ax - b = 0 \tag{1.5}$$

where a and b are intervals. Suppose $[a] = (\underline{a}, \overline{a})$ and $[b] = (\underline{b}, \overline{b})$ are intervals. Therefore considering the case $[a] > 0$, $[b] > 0$, i.e. $\underline{a}, \overline{a} > 0$ and $\underline{b}, \overline{b} > 0$, the interval extension of Equation (1.3) is given by $(\underline{a}, \overline{a})(\underline{x}, \overline{x}) = (\underline{b}, \overline{b})$. This can be illustrated as $(\underline{ax}, \overline{ax}) = (\underline{b}, \overline{b})$. It is evident that the equivalence of the right as well as left hand sides of the equation is feasible only if $\underline{ax} = \underline{b}$ and $\overline{ax} = \overline{b}$, which are illustrated as

$$\underline{x} = \frac{\underline{b}}{\underline{a}}, \quad \overline{x} = \frac{\overline{b}}{\overline{a}}. \tag{1.6}$$

The interval extension of (1.4) can be regarded as

$$\underline{x} = \frac{\underline{b}}{\overline{a}}, \quad \overline{x} = \frac{\overline{b}}{\underline{a}}. \tag{1.7}$$

The suggested methodology can be utilized only in the uncomplicated cases of linear equations. Generally, choosing the suitable interval or fuzzy extensions in the nonlinear case is a complicated task.

Example 1.1 Assume $[a] = (3, 4)$ and $[b] = (1, 2)$. Hence from Equation (1.6) we can extract $\underline{x} = 0.333$, $\overline{x} = 0.5$, and from Equation (1.7) we can extract $\underline{x} = 0.25$, $\overline{x} = 0.666$. ■

In [14], the decomposition methodology has been utilized for quadratic, cubic, and generalized higher order polynomial equations, and negative, or non-integral powers and random algebraic equations. The algebraic equations can be dealt with by using the decomposition methodology, which it supplies a crucial methodology in order to calculate the roots of polynomial equations, usually resulting in a very fast convergence. This methodology generally converges towards a precise solution.

In [33], a novel algorithm on the basis of the Adomian methodology is demonstrated in order to solve algebraic equations. This modernized algorithm computes the superior estimations related to the exact solution of algebraic equations, when compared with the standard Adomian methodology. A nonlinear equation is considered as follows

$$F(x) = 0 \tag{1.8}$$

which can be transformed to

$$x = F_0(x) + c_0 \tag{1.9}$$

where F_0 is taken to be a nonlinear function; c_0 is a constant. The Adomian methodology calculates x as a series

$$x = \sum_{i=0}^{\infty} x_i. \tag{1.10}$$

The decomposition of the nonlinear function is illustrated below

$$F(x) = \sum_{i=0}^{\infty} A_i \tag{1.11}$$

where A_i is considered to be an Adomian polynomial stated by

$$A_n(x_0, \dots, x_n) = \left(\frac{1}{n!}\right)\left(\frac{d^n}{d\lambda^n}\right) F\left(\sum x_i \lambda^i\right)\bigg|_{\lambda=0}. \tag{1.12}$$

Substituting (1.10) and (1.11) into (1.9) results in

$$\sum_{i=0}^{\infty} x_i = \sum_{i=0}^{\infty} A_i + c_0 \tag{1.13}$$

where each term of the series $x = \sum_{i=0}^{\infty} x_i$, at par with Adomian method, can be computed using the relations

$$\begin{aligned} x_0 &= c_0 \\ x_1 &= A_0 \\ x_2 &= A_1 \\ &\cdot \quad \cdot \\ &\cdot \\ &\cdot \\ x_n &= A_{n-1} \end{aligned} \tag{1.14}$$

In calculating x by utilizing any software, since n increases, the number of terms in the expression for A_n increases and this results in the dissemination of round off errors. Also, the factor $\frac{1}{n!}$ mentioned in the formula related to A_n makes it minute, hence its contribution to x is not taken into account, and therefore the primary few terms related to the series $\sum_{i=0}^{\infty} x_i$ state the preciseness of the estimated solution. By taking this concept into consideration, [33] proposed a novel algorithm on the basis of the Adomian methodology in order to improve the preciseness significantly.

1.4 Numerical Methods for Solving Fuzzy Equations

1.4.1 Newton Method

In 1669, Isaac Newton introduced a novel algorithm [140] for solving a polynomial equation that was demonstrated on the basis of an example as $y^3 - 2y - 5 = 0$. To obtain a precise root of this equation, initially, a starting value should be assumed, where $y \approx 2$. By assuming $y = 2 + p$ and substituting it into the original equation, the following is obtained $p^3 + 6p^2 + 10p - 1 = 0$. As p is presumed to be minute, $p^3 + 6p^2$ is neglected in comparison with $10p - 1$. Also the previous equation generates $p \approx 0.1$, so a superior approximation of the root is $y \approx 2.1$. The repetition of this process is feasible and $p = 0.1 + q$ is extracted, and the substitution delivers $q^3 + 6.3q^2 + 11.23q + 0.061 = 0$, hence $q \approx -0.061/11.23 = -0.0054\ldots$, so a novel approximation of the root is $y \approx 2.0946$. The process needs to be repeated until the expected number of digits is achieved. In his methodology, Newton did not distinctly utilize the hypothesis of derivative, but only applied it to polynomial equations.

In [3], some effective numerical algorithms are demonstrated in order to solve the nonlinear equation $f(x) = 0$ on the basis of the Newton–Raphson methodology. The modified Adomian decomposition methodology is implemented for developing the numerical algorithms.

In [7], the Newton methodology is proposed in association with fuzzy nonlinear equations instead of standard analytical methodologies, as they are not appropriate throughout. The primary intention is to extract a solution for fuzzy nonlinear equation $F(x) = c$. Initially, the cited researchers mentioned the fuzzy nonlinear equation in parametric form as illustrated below

$$\begin{cases} \underline{F}(\underline{x}, \overline{x}, \alpha) = \underline{c}(\alpha) \\ \overline{F}(\underline{x}, \overline{x}, \alpha) = \overline{c}(\alpha) \end{cases} \tag{1.15}$$

so they resolved it by utilizing Newton's methodology.

In [77], iterative methods are illustrated to obtain a simple root δ, i.e. $f(\delta) = 0$ as well as $f'(\delta) \neq 0$, of a nonlinear equation $f(x) = 0$. The authors used the construction of some higher order modifications of Newton's method in order to resolve nonlinear equations. This construction maximizes the convergence order of prevailing iterative methodologies by one, two, or three units. This can be implemented in any iteration formula as well as per iteration. The resulting methodologies sum up only one additional function evaluation in order to maximize the order. This makes the computational effectiveness superior. This scheme can be employed for improving any prevailing iteration formula.

In [28], the investigators found the solution of

$$A_1x \oplus A_2x^2 \oplus \cdots \oplus A_nx^n = A_0 \tag{1.16}$$

where A_i and X^j belong to a fuzzy set (for $i = 1, \ldots n$ $j = 0, 1, \ldots, n$). The fuzzy quantities are demonstrated in parametric form. The primary initiative is based on the conversion of the polynomial fuzzy coefficients into parametric form, thereby implementing Newton's technique on each limit. In the final phase, for finding the root, which is considered to be a fuzzy number, the α level sets of fuzzy coefficients on each limit are computed numerically.

The advantage of Newton's methodology is due to the convergence speed, once an adequately precise approximation is known. A drawback of this methodology is that a precise initial approximation of the solution is required to validate convergence. Another drawback of Newton's technique is due to the calculation and inversion of the Jacobian matrix $J(x)$ at each iteration.

The rapid convergence of Newton's technique is possible when an appropriate initial value is achieved. However, it is difficult to extract this kind of value, and this technique is also relatively expensive to implement [54]. Broyden's methodology is suggested to resolve this kind of equation. Broyden's methodology leans toward superlinear convergence. This methodology is selected since it is a superior alternative in comparison to Newton's technique, and it also minimizes the amount of computation at each iteration without markedly decreasing the speed of convergence. It substitutes the matrix A_{k-1}, whose inverse is directly evaluated at each iteration, in place of the Jacobian matrix J, and it minimizes the arithmetic operation $O(n^3)$ to $O(n^2)$ [54]. Instead of utilizing standard analytical methods, such as Buckley and Qu methods, which are not appropriate for resolving a system of fuzzy nonlinear equations taking into consideration that the coefficient is the fuzzy number, Broyden's technique is suggested for resolving fuzzy nonlinear equations. In [161], Broyden's technique is implemented in order to solve fuzzy nonlinear equations. Initially, fuzzy nonlinear equations are displayed in the parametric form, and they are resolved by utilizing the Broyden technique.

In [192], a new technique based on the Newton and Broyden methods is suggested for solving dual fuzzy nonlinear equations. The fuzzy nonlinear equations are transformed into parametric form and are then solved with Newton's method for initial iteration and Broyden's method for remainder of the iterations. The fuzzy coefficients are demonstrated in parametric form. This method reduces the calculation of the Jacobian matrix in every iteration.

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. The steepest descent method uses only first order information and does not deal with approximating second derivatives.

1.4.2 Steepest Descent Method

In [9], a numerical solution associated with fuzzy nonlinear equation $F(x) = 0$ is suggested using the steepest descent technique, where the fuzzy quantities are demonstrated in parametric form. The equation is represented by parametric form, as mentioned below

$$\begin{cases} \underline{F}(\underline{x}, \overline{x}, \alpha) = 0 \\ \overline{F}(\underline{x}, \overline{x}, \alpha) = 0 \end{cases}. \tag{1.17}$$

The function $K : R^2 \to R$ is stated as

$$K(\underline{x}, \overline{x}) = [\underline{F}(\underline{x}, \overline{x}, \alpha), \overline{F}(\underline{x}, \overline{x}, \alpha)]^2. \tag{1.18}$$

The technique of steepest descent characterizes a local minimum for two variable function K. The technique of steepest descent is stated as:

1) Find K at an initial approximation $X_0^\alpha = (\underline{x}_0^\alpha, \overline{x}_0^\alpha)$.
2) Determine a direction from $X_0^\alpha = (\underline{x}_0^\alpha, \overline{x}_0^\alpha)$ that causes a decrease in the value of K.
3) Shift a suitable amount in this direction and consider the new value $X_1^\alpha = (\underline{x}_1^\alpha, \overline{x}_1^\alpha)$.
4) Repeat sequence 1 via 3 with X_0^α replaced by X_1^α.

The steepest descent technique converges only linearly to the solution, but in general it converges even for weak initial approximations [53]. Even though the steepest descent technique does not need a superior initial value, its drawback is due to its low convergence speed. Genetic algorithms have been found to provide a rapid convergence to a near optimum solution in many types of problems. The genetic algorithm method has better training performance than the steepest descent method.

1.4.3 Adomian Decomposition Method

The Adomian decomposition method was initially laid down by George Adomian in [13]. In [2], the standard Adomian decomposition is implemented on the simple iteration technique in order to resolve the equation $f(x) = 0$, where $f(x)$ is a nonlinear function, and the convergence related to the series solution is proved. Initially, the nonlinear equation is transformed into canonical form, then the Adomian technique computes the solution, which is at par with the series form. As practically all the terms associated with the series are not possible to determine, hence the estimation of the solution is obtained from the truncated series. Therefore, the convergence related to the truncated series is usually very rapid.

Babolian et al. [31] modified the standard Adomian technique mentioned in [2] in order to solve nonlinear equation $f(x) = 0$ to acquire a sequence of approximations related to the solution with approximate superlinear convergence. They employed Cherruault's definition [59] and took into consideration the order of convergence related to the technique [32].

In [149], a potential numerical algorithm in order to solve fuzzy polynomial equations $\sum_{i=1}^{n} a_i x^i = c$ on the basis of Newton's technique is demonstrated, where x and c are considered to be fuzzy numbers, and all coefficients are taken to be fuzzy numbers. The modified Adomian decomposition methodology is implemented for the construction of the numerical algorithm. Initially, the fuzzy polynomials are illustrated in parametric form and then resolved using the Adomian decomposition technique.

In [189], the Shanks transformation is employed on the Adomian decomposition technique in order to resolve nonlinear equations so as to improvise the preciseness of the approximate solutions. The numerical results demonstrate that the implementation of this technique in similar conditions generates more appropriate solutions to the nonlinear equations when compared with those extracted from the Adomian decomposition technique. The Shanks transform is an effective approach that can speed up the convergence rate of the series.

In [131], an effective extension of Newton's method to the fuzzy polynomial is proposed using a modified Adomian decomposition technique in the form of $\sum_{i=1}^{n} b_i x^i = d$, where x, d, and b are fuzzy numbers. The fuzzy polynomials are written in a parametric form and then are resolved by the Adomian decomposition method.

The advantage of the Adomian decomposition technique is that it can provide analytical approximations to solutions of nonlinear equations without supposing that the system has weak nonlinearities. The major drawback of the Adomian decomposition technique is the complex and difficult procedure needed to compute the Adomian polynomials. The ranking method is simple and inexpensive.

1.4.4 Ranking Method

The ranking methodology was primarily laid down by Delgado et al. [64]. In [170], the researcher obtained the real roots of the polynomial equation, which has been demonstrated as follows

$$C_1 x + C_2 x^2 + \cdots + C_n x^n = C_0 \tag{1.19}$$

where $x \in R$ as well as $C_0, C_1, \ldots, C_n$ are taken to be fuzzy numbers. In [170] the fuzzy polynomial equation is converted to a system of crisp polynomial equations. This conversion takes place using the ranking method on the basis of three parameters: walue, ambiguity and fuzziness. The obtained system of crisp polynomial equations is resolved numerically.

In [144], the conceptual content of a ranking method is suggested in order to extract the real roots associated with a dual fuzzy polynomial equation, which has been illustrated as follows

$$A_1 x \oplus A_2 x^2 \oplus A_n x^n = B_1 x \oplus B_2 x^2 \oplus B_n x^n + d \tag{1.20}$$

where $x \in R$ as well as $A_1, \ldots, A_n, B_1, \ldots, B_n, d$ are denoted as fuzzy numbers. The dual fuzzy polynomial equations are converted to a system of crisp dual polynomial equations. This conversion is carried out by utilizing ranking methodology on the basis of three parameters, namely value, ambiguity, and fuzziness.

In [145], the real roots corresponding to the polynomial equation, $A_1 x + A_2 x^2 + \cdots + A_n x^n = A_0$, is obtained by utilizing the ranking method considering fuzzy numbers, where $x \in R$ as well as $A_0, A_1, \ldots, A_n$ are denoted as fuzzy numbers. In the referred paper, the ranking methodology is utilized for real roots of the dual polynomial equations as mentioned below

$$A_1 x + A_2 x^2 + \cdots + A_n x^n = B_1 x + B_2 x^2 + \cdots + B_n x^n + d \tag{1.21}$$

where $x \in R, A_1, \dots, A_n, B_1, \dots, B_n$ as well as d are considered to be fuzzy numbers.

In [146], the ranking technique is implemented in order to obtain the real roots of an interval type 2 dual fuzzy polynomial equation $A_1x + A_2x^2 + \cdots + A_nx^n = B_1x + B_2x^2 + \cdots + B_nx^n + d$, where $x \in R$ and the coefficients $A_1, \dots, A_n, B_1, \dots, B_n$ as well as d are termed as interval type 2 fuzzy numbers. The type 2 dual fuzzy polynomial equation is converted into a system at par with the crisp type 2 dual fuzzy polynomial equation. The conversion is done by the ranking of the fuzzy numbers on the basis of three parameters, namely value, ambiguity, and fuzziness.

It has been revealed that solutions in correspond to three parameters value, ambiguity, and fuzziness are not sufficient to generate solutions. Hence in [147], a novel ranking methodology is suggested in order to eradicate the intrinsic weakness. The novel ranking methodology, which is incorporated with four parameters is then implemented in the interval type 2 fuzzy polynomials, covering the interval type 2 of fuzzy polynomial equations, dual fuzzy polynomial equations as well as the system of fuzzy polynomials. The effectiveness of the novel ranking methodology is numerically considered in the triangular fuzzy numbers as well as the trapezoidal fuzzy numbers.

The main disadvantage of the ranking method is that it can be applied only when membership functions are known. Approximation methods such as fuzzy neural networks are also effective tools for overcoming the limitations of the other numerical methods. The major advantage of using fuzzy neural networks is training the large amount of data sets, quick convergence, and high accuracy.

1.4.5 Intelligent Methods

1.4.5.1 Genetic Algorithm Method

A genetic algorithm for resolving the fuzzy equation $P(x) = y$ is demonstrated in [46], where x and y are considered to be k sampled real fuzzy numbers and P is taken to be a fuzzy function relying on x. The motivation is to obtain a suitable value of the fuzzy argument x in such a manner that the calculated value of the polynomial, $P(x)$, is very much adjacent to the supplied target value y. The presented genetic algorithm utilizes a distinct demonstration of the fuzzy numbers, which permits the implementation of simple genetic operators. The algorithm is self-sufficient for finding multiple solutions associated with the fuzzy equations. However, no method has been utilized for an identical problem involved in the area of neural networks that can be taken possession of. Because of the distinct discrete criteria of the fuzzy arithmetic, the single realistic approach for resolving this problem is to design a dedicated genetic algorithm [127].

A genetic algorithms methodology for resolving the linear and quadratic fuzzy equations $Ax = B$ as well as $Ax^2 + Bx = C$, where A, B, C and x are considered to be fuzzy numbers, is mentioned in [124]. The methodology based on the genetic algorithms primarily begins with a set of random fuzzy solutions. After that, in each generation of genetic algorithms, the solution candidates converge to the superior fuzzy solution. In the suggested methodology the final obtained solution is not only restricted to fuzzy triangular but it can also be a fuzzy number. In this methodology, in order to obtain the best fuzzy solution associated with a fuzzy equation, initially a solution is required to be converted to its chromosome demonstration. The solution candidates for the fuzzy equations are transformed to their level sets demonstration, which are computable by genetic algorithms.

In [120], genetic algorithms are implemented for solving fuzzy equations without stating membership functions related to fuzzy numbers. The extension principle, interval arithmetic, α cut operations, or a penalty technique were not used in order to deal with the problem. An important matter for using genetic algorithms in order to extract a better solution associated with the problem is the parameter settings, which include the probability of crossover, the probability of mutation, and the number of generations. The fuzzy concept related to the genetic algorithm scheme is different, but generates superior solutions in comparison with classical fuzzy techniques.

In [125], a genetic algorithm is used for resolving nonlinear 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 a genetic algorithm is applied for finding the minimum. The method is extended to systems of nonlinear equations.

The genetic algorithm represents the most consistent results in terms of accuracy and convergence but it is computationally very expensive. The modified Adomian provides acceptable results and converges rapidly to the numerical solution. The Adomian decomposition method is less expensive than the genetic algorithm method.

1.4.5.2 Neural Network Method

In [49] neural network was employed for solving the fuzzy linear equation

$$AX = C \tag{1.22}$$

where A, B, and X are considered to be triangular fuzzy numbers. Taking into account certain values of A and C, Equation (1.22) generates no solution for X [52]. The training of a neural network in order to solve Equation (1.22) was mentioned by the researchers in [49], considering that zero is not at par with the support of A. The investigation was carried out considering neural network solutions termed as Y and X^*. When there are no restrictions concerned with the weights of the network, then the neural network output will be Y. The non-existence of the relationship between Y and X was proved and validated by utilizing computer analysis. X^* is the solution of the neural network, taking into consideration that certain sign restrictions are set on the weights. X^* is illustrated to be an approximation, which is named a new solution of fuzzy equations. It has been displayed by using $X \leq X^*$.

The evolutionary algorithm, as well as a neural network in combination, has been utilized for solving the fuzzy equation, which was referred to in [50] as follows

$$AX \oplus B = C \tag{1.23}$$

where A, B, C and X are termed as triangular fuzzy numbers. The first solution X_c related to Equation (1.23) is stated to be the classical solution that utilizes α-cut and interval arithmetic for obtaining X_c.

Example 1.2 Assume $[A] = (1, 2, 3)$, $[B] = (-3, -2, -1)$ and $[C] = (3, 4, 5)$. Employing the intervals in the fuzzy equation generates

$$\begin{aligned} &(1+\alpha)\underline{X}_c^\alpha + (-3+\alpha) = (3+\alpha) \\ &(3-\alpha)\overline{X}_c^\alpha + (-1-\alpha) = (5-\alpha) \end{aligned} \tag{1.24}$$

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

$$\begin{aligned} \underline{X}_c^\alpha &= \frac{6}{1+\alpha} \\ \overline{X}_c^\alpha &= \frac{6}{3-\alpha} \end{aligned} \tag{1.25}$$

can be extracted. However $[\underline{X}_c^\alpha, \overline{X}_c^\alpha]$ does not state a fuzzy number as $\underline{X}_c^\alpha (\overline{X}_c^\alpha)$ is a decreasing (increasing) function of α. Occasionally X_c prevails and sometimes will not exist. ■

By the fuzzification of the crisp solution $(c - b)/a, a \neq 0$, the other solution is extracted. $(C - B)/A$ represents the fuzzified solution, taking into account that zero is not at par with the support of A. For the evaluation of the fuzzified solution two approaches have been suggested. The primary approach generates the solution X_e by utilizing the extension principle, and the secondary approach generates the solution X_I by the means of α-cut and interval arithmetic. X_e can be achieved as shown below

$$X_e = \min\{\Pi(a, b, c) | (c - b)/a = x\} \tag{1.26}$$

where $\Pi(a,b,c) = \min\{A(a), B(b), C(c)\}$. For obtaining α-cut of X_e the process is described as follows

$$\underline{X}_e^\alpha = \min\left\{ \frac{c-b}{a} \middle| \, |a \in [A]^\alpha, b \in [B]^\alpha, c \in [C]^\alpha \right\}$$
$$\overline{X}_e^\alpha = \max\left\{ \frac{c-b}{a} \middle| \, |a \in [A]^\alpha, b \in [B]^\alpha, c \in [C]^\alpha \right\} \tag{1.27}$$

where $[X_e]^\alpha = (\underline{X_e}^\alpha, \overline{X_e}^\alpha)$. The solution X_I can be calculated as follows

$$[X_I]^\alpha = ([C]^\alpha - [B]^\alpha)/[A]^\alpha. \tag{1.28}$$

The original fuzzy equation may or may not be solved by $X_e(X_I)$. Taking into account some fuzzy equations, X_e is mathematically too complex to extract, so in [50] an evolutionary algorithm was implemented for estimating their α-cuts. The method in the paper can be normalized for interacting with fuzzy problems, evolutionary algorithms, and neural networks. There are disadvantages in the method that were mentioned in [50]. The method is exclusively meant for symmetric fuzzy numbers, and it computes just the upper bound and the lower bound of the fuzzy numbers, avoiding the center part.

An architecture of the fuzzy neural network that is suggested in order to obtain a real root of the fuzzy polynomials is illustrated in the form [10]

$$A_1 x + \cdots + A_n x^n = A_0 \tag{1.29}$$

where $x \in R$ as well as $A_0, A_1, \ldots, A_n \in E$. A learning algorithm associated with the cost function in order to adjust the crisp weights has been suggested. The methodology mentioned in [10] has drawbacks. It was solely capable of extracting a crisp solution of fuzzy polynomials, and this neural network cannot extract a fuzzy solution.

In [103], the researchers obtained the approximate solution of the following fuzzy polynomial having degree n

$$A_1 x + \cdots + A_n x^n = A_0 \tag{1.30}$$

where $A_0, A_1, \ldots, A_n, x \in E$. They laid down two types of neural networks for approximating the solution of Equation (1.30), namely feedforward (static) as well as recurrent (dynamic) models. The corresponding algorithm of both neural networks is based on the least mean square. The difference between the two neural networks is that the dynamic neural network has superior robustness than the static neural network. The technique, illustrated in [103], is sufficient to find an approximate solution at par with the special case of a fuzzy equation, but not the generalized case.

The general fuzzy equation known as a dual fuzzy equation [192] was illustrated in [95]. Normal fuzzy equations have fuzzy numbers solely on one side of the equation. However, dual fuzzy equations have fuzzy numbers on both sides of the equation. Since it is not possible to move the fuzzy numbers from one side to the other [108], dual fuzzy equations are more generalized and complex. In [95], the existence of the solutions related to the dual fuzzy equations is analyzed, and is incorporated with the controllability problem of fuzzy control [58]. Afterward, two kinds of neural networks for the approximation of the solutions related to dual fuzzy equations were demonstrated, namely the static and dynamic models.

In [133], an architecture of fuzzy neural networks was suggested for solving dual fuzzy polynomial equations. A learning algorithm of fuzzy weights of two layers of feed-forward fuzzy neural networks is used whose input–output relations are defined by the extension principle.

In [102], a dynamic neural network is proposed for solving a dual fuzzy polynomial and is demonstrated as follows

$$a_1x + \cdots + a_nx^n = b_1x + \cdots + b_nx^n + d \tag{1.31}$$

where $a_1, \dots, a_n$, $b_1, \dots, b_n$ and d belong to a fuzzy set. The neural network is trained by a back propagation type learning algorithm that has five layers where connection weights are crisp numbers. The important advantage of this methodology is that it can greatly reduce the size of calculations and generate high accuracy of the numerical solution.

Fuzzy linear regression analysis has become popular with investigators and is a standard model for analyzing data vagueness phenomena. It is utilized to generate a suitable linear relation between a dependent variable and various independent variables in a fuzzy environment.

1.4.5.3 Fuzzy Linear Regression Model

Generally, there exist two techniques in fuzzy regression analysis namely a linear programming based technique [155][172][183][184] and a fuzzy least squares technique [171][65]. The primary technique relies on diminishing fuzziness at par with optimal criteria. The secondary technique utilizes least square errors at par with fitting criteria. As illustrated in [191], the benefit of the primary technique is its simplicity of programming as well as calculation, whereas in the fuzzy least squares technique it is its minimal degree of fuzziness between the observed and approximated values. Currently, the least total square error of the spread values is utilized as the fitting criteria as well as an advanced mathematical programming methodology in which the predictability of the primary technique can be improved and the calculation complication of the secondary technique can be minimized [137].

Fuzzy linear regression was initially proposed by Tanaka et al. [184]. The main intention was to minimize the total spread of the fuzzy parameters relating to the support of the approximated values that enclose the support of the observed values at par with a certain α level. Even though this concept was later modified by Tanaka et al. [183], their model is deemed to be very responsive to data. It can generate infinite solutions as well as the spread of the approximated values, since it becomes wider as more data are piled up in the model.

In [172], a fuzzy linear regression model is generated in the form of $Y_i = A_0 + A_1 x_i$ at par with the fuzzy output as well as fuzzy parameters taking into consideration the mathematical programming problem by utilizing three indices concerned with the equalities between fuzzy numbers. Three patterns of multi-objective programming problems in order to extract fuzzy linear regression models are laid down related to the three indices. Linear programming relies on an interactive decision making method in order to extract the convenient solution at par with the decision making for formulating the multi-objective programming problem. The technique implied in [172] can generate an infinite number of solutions via repeated observations. Therefore, the mentioned technique is able to generate crisp coefficients. By repeated observations this technique results in redundant constraints. Hence, all observations cannot contribute to the computation of the model. In [172], fuzzy linear regression experiences crisp coefficients, redundant constraints, and the possibility of an infinite number of solutions. To deal with the possibility of an infinite number of solutions, the approximation point of the centers at par with the fuzzy coefficients can be computed using the available data, which is implemented into fuzzy linear regression algorithms. In [173], it was illustrated that the least squares technique can be utilized as a point approximation to center the fuzzy coefficients, and this is employed in the Tanaka technique. Two advantages are linked to the use of a point approximation to center the fuzzy coefficients. Primarily, if the researcher selects a point approximation that is distinctively defined, then the possibility of an infinite number of solutions is eliminated. Secondly, point approximations permit all data points to contribute information in the fuzzy linear algorithm. Hence, under repeated observations, the utilization of point approximations associated with the center of the fuzzy coefficients tackles some of the problems imparted by redundant constraints.

Nasrabadi et al. [138] utilized a multi-objective programming concept in order to illustrate the linear regression coefficients $Y_i = A_0 + A_1 X_i + \cdots + A_n X_{in}$, $i = 1, \ldots, m$ where $X_{ij} = (x_{ij}, r_{ij})$, $A = (a_j, \alpha_j)$ as well as $Y_i = (y_i, \beta_i)$ are considered to be symmetric fuzzy numbers. In the mentioned fuzzy regression work, powerful predictions are developed on the basis of the fuzzy number parameters.

In [134], the fuzzy linear regression $Y_i = A_0 \oplus A_1 x_{i1} \oplus A_2 x_{i2} \oplus \cdots \oplus A_n x_{in}$ as well as fuzzy polynomial regression $Y_i = A_{l0} \oplus \sum_{j=1}^{n} A_{lj} x_{ij} \oplus \sum_{j=1}^{n} \sum_{k=1}^{n} A_{ljk} x_{ij} x_{ik} \oplus \ldots$, where input units are taken to be crisp numbers and the output unit is taken to be a fuzzy number. The proposed technique relies on a neural network model in order to extract the estimation of the regression coefficients. A more generalized pattern of fuzzy polynomial regression was revealed in [148].

In [148], a polynomial fuzzy regression model at par with fuzzy independent variables, as well as fuzzy parameters based on the following form, is illustrated as

$$Y_i = A_{l0} \oplus \sum_{j=1}^{n} A_{lj} X_{ij} \oplus \sum_{j=1}^{n} \sum_{k=1}^{n} A_{ljk} X_{ij} X_{ik} \oplus \dots \tag{1.32}$$

where i denotes the different observations, $X_{i1}, X_{i2}, \dots, X_{in}$ are coefficients, and Y_i is considered to be a fuzzy number. A fuzzy neural network model is utilized to extract an estimate related to the fuzzy parameters in a statistical sense. This technique permits the development of nonlinear regression models, along with general fuzzy number inputs, outputs, and parameters. The suggested technique consists of numerous properties. Initially, it can use non-triangular fuzzy observations. Furthermore, the fuzzy neural network technique performs perfectly with respect to the sum of squared errors and accuracy of approximation.

In order to obtain a fuzzy polynomial interpolation having degree n, it is essential to extract n fuzzy coefficients. So as to find these coefficients, it is a requirement to solve $2n \times 2n$ equations, which is very complicated in terms of large values of n. Also sometimes it is barred of a fuzzy solution [71]. In [123], an innovative estimation algorithm for fuzzy polynomial interpolation by utilizing the artificial bee colony algorithm for interpolating fuzzy data is demonstrated. It is assumed that $X = \{x_1, \dots, x_n\}$ is a set of m distinct points associated with R and $F = \{y_1, \dots, y_n\}$ is the value of a triangular fuzzy function f at the point x_i, $i = 1, \dots, n$. The polynomial below of m degree is taken into account

$$p_m(x) = \sum_{j=0}^{m} a_j x^j = \sum_{j=0}^{m} (\underline{a}_j(r), \overline{a}_j(r)) x^j \tag{1.33}$$

where a_j is a trapezoidal fuzzy number at par with parametric form $(\underline{a}_j(r), \overline{a}_j(r))$ for $j = 0, 1, \dots, m$. The experimental data are considered to be $(x_1, y_1), (x_2, y_2), (x_n, y_n)$, $x_i \in X$, and $y_i \in F$ (given that $n > m + 1$).

In order to compare the efficiency of the numerical methods to approximate the solution of dual fuzzy equations the examples below are demonstrated.

Example 1.3 A water tank system contains two inlet valves q_1 and q_2, as well as two outlet valves q_3 and q_4, see Figure 1.1. The areas of the valves are uncertain, $A_1 = G(0.021, 0.023, 0.024)$, $A_2 = G(0.008, 0.018, 0.038)$, $A_3 = G(0.012, 0.013, 0.015)$, and $A_4 = G(0.038, 0.058, 0.068)$. The velocities of the flow (controlled by the valves) are $g_1 = \left(\frac{\vartheta}{10}\right) e^{\vartheta}$, $g_2 = \vartheta cos(\Pi \vartheta)$, $g_3 = cos\left(\frac{\Pi \vartheta}{8}\right)$, and $g_4 = \frac{\vartheta}{2}$. If the outlet flow is aimed to be $q = (4.088, 6.336, 36.399)$, what is the quantity of the control variable ϑ?

The mass balance of the tank is [177],

$$\rho A_1 g_1 \oplus \rho A_2 g_2 = \rho A_3 g_3 \oplus \rho A_4 g_4 \oplus q \tag{1.34}$$

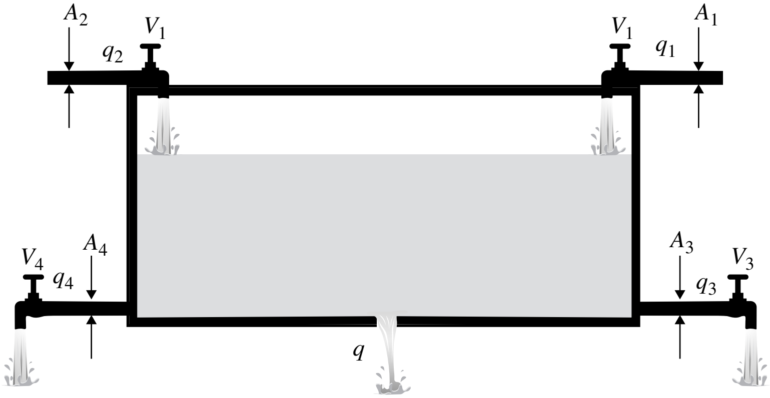

Figure 1.1 Water tank system.

Table 1.1 Approximation errors of the water tank.

k	**Newton**	**Descent**	**Genetic**	**Decomposition**	**Ranking**	**NN**
1	0.186	0.168	0.334	0.140	0.310	0.439
2	0.296	0.260	0.247	0.223	0.238	0.323
3	0.361	0.326	0.130	0.180	0.119	0.217
⋮	⋮	⋮	⋮	⋮	⋮	⋮
119	0.079	0.052	0.045	0.034	0.029	0.003
120	0.075	0.049	0.039	0.030	0.025	0.002

where ρ is the density of water. The exact solution is $\vartheta_0 = 2$ [177]. To approximate the solution, six popular methods are used: the Newton method, the steepest descent method, the genetic algorithm method, the Adomian decomposition method, the ranking method, and the neural network method. The errors of these methods are shown in Table 1.1. Corresponding error plots are demonstrated in Figure 1.2.

It can be seen that all six methods can approximate the solutions of the dual fuzzy equations. The neural network method is more suitable for solving these kinds of equations. The estimated errors of the neural network based algorithm are less than the other methods. The neural network method is faster and more robust when compared with the other methods. ■

Example 1.4 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 1.3. The positions are not exact, $L_1 = F(0.2, 0.3, 0.5, 0.6)$, $L_2 = F(0.4, 0.6, 0.7, 0.8)$, and $L_3 = F(0.4, 0.6, 0.7, 0.8)$. The area of the rod is $A = \frac{\pi}{4}d^2$. The external forces are a function of x, $F_1 = x^7$, $F_2 = x^6\sqrt{x}$,

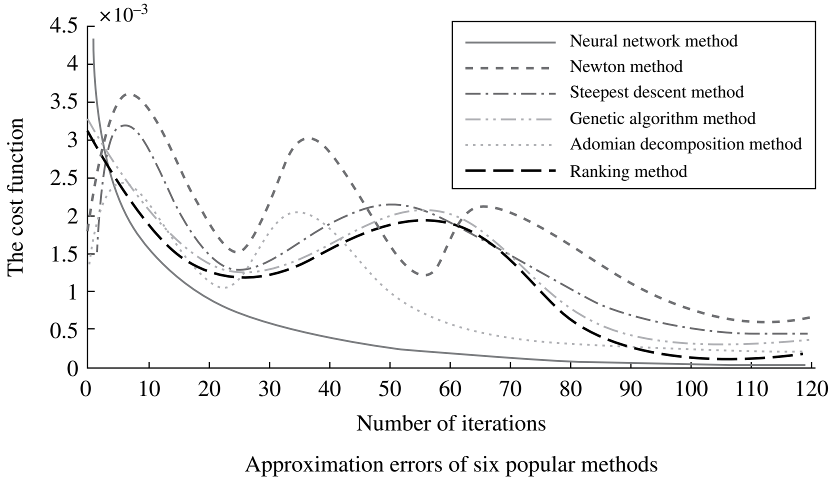

Figure 1.2 Approximation errors of the six popular methods.

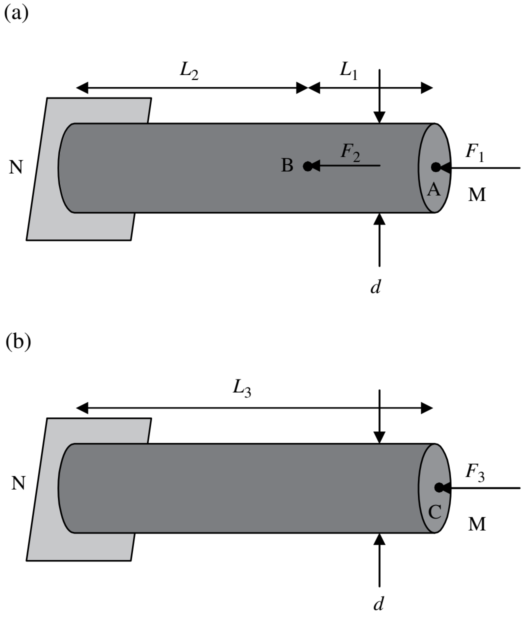

Figure 1.3 Two solid cylindrical rods.

and $F_3 = e^{2x}$. If the desired deformation at the point N is desired to be $N^* = F(0.000563, 0.000822, 0.001003, 0.001211)$, what is the quantity of the control force x?

Table 1.2 Approximation errors of the solid cylindrical rod.

k	Newton	Steepest descent	Genetic algorithm	Adomian decomposition	Ranking	Neural network
1	0.150	0.201	0.486	0.267	0.600	0.788
2	0.229	0.299	0.574	0.33	0.498	0.500
3	0.311	0.184	0.407	0.239	0.379	0.310
⋮	⋮	⋮	⋮	⋮	⋮	⋮
89	0.109	0.0801	0.0699	0.0594	0.0500	0.0098
90	0.0960	0.072	0.0600	0.0490	0.0411	0.0071

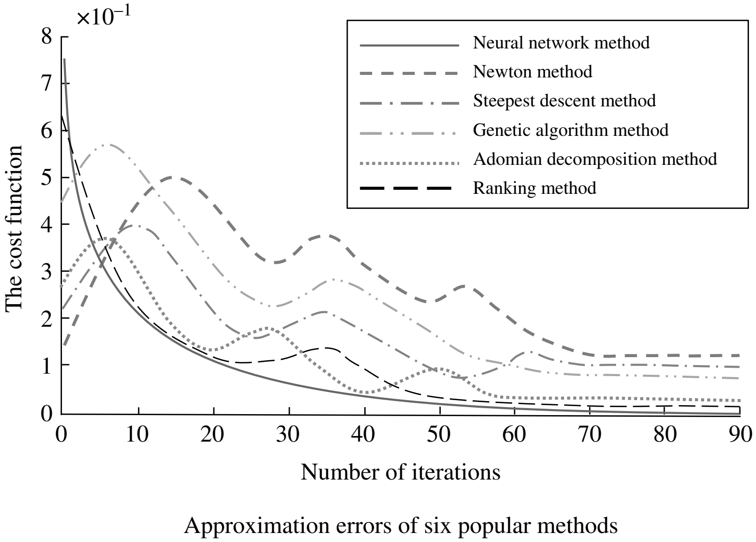

Figure 1.4 Approximation error of the six popular methods.

The tension relation is [42],

$$\frac{L_1F_1}{AE} \oplus \frac{L_2(F_1+F_2)}{AE} = \frac{L_3F_3}{AE} \oplus N^* \tag{1.35}$$

where $d = 0.02$ and $E = 70 \times 10^9$. The exact solution is $x = 4$. To approximate the solution, six popular methods are used: the Newton method, the steepest descent method, the genetic algorithm method, the Adomian decomposition method, the ranking method, and the neural network method. The errors of these methods are shown in Table 1.2. Neural network method is more robust than the other methods. Furthermore, the estimated error of the neural network is less when compared with other methods. Corresponding error plots are demonstrated in Figure 1.4.

1.5 Summary

In this chapter, some of the numerical methodologies are demonstrated as solutions of fuzzy equations and dual fuzzy equations. This review illustrates that the fuzzy roots of fuzzy equations can be obtained with different algorithms. However, in a few cases there exist no fuzzy roots in a particular fuzzy equation. The solution of the fuzzy polynomial by the ranking methodology is proposed for solving the fuzzy polynomial equation, which converts to a crisp system of polynomial equations, and therefore the system is easily solvable. For obtaining the real roots of the system in the case that there is no exact solution, iteration methodologies can be utilized for estimating the solution. Using a modified Adomian decomposition methodology, the fuzzy roots can be obtained by proposing fuzzy polynomials in parametric form and solving by the Adomian decomposition methodology. Obtaining a solution using the Newton methodology demonstrates that the fuzzy roots of fuzzy equations can be obtained at the initial step with high accuracy. By contrast, with the fuzzy neural network, the fuzzy root of the fuzzy equation can be obtained by proposing a learning algorithm. This review provides input for those showing interest in the field of fuzzy equations.