May 2013. Vol. 3, No. 3 ISSN2305-8269
International Journal of Engineering and Applied Sciences

www.eaas-journal.org

# MULTI- SOLUTIONS OF FUZZY POLYNOMIALS: A REVIEW

[1]Nurhakimah Ab. Rahman, [2] Lazim Abdullah
[1, 2] Department of Mathematics,
Faculty of Science and Technology, Universiti Malaysia Terengganu, Malaysia.
e-mail : [1]hamikahrun@yahoo.com, [2]lazim_m@umt.edu.my

***Abstract-***

*There have been many numerical methods for the solution of fuzzy polynomials. However, these methods have their own algorithm that will make it an efficient method to find real root of fuzzy polynomials. In this study, a current review on the solution of fuzzy polynomials in the form of* $A_1x + A_2x^2 + \ldots + A_nx^n = A_0$ *will be presented. Related articles appearing in the international journals from 1997-2011 (searched via ScienceDirect and ProQuest) are gathered and analyzed. Hopefully this review will provide an input in the study of fuzzy polynomials.*

*Keywords: Adomian decomposition method, Fuzzy polynomial, Fuzzy neural network, Newton-Raphson method, Ranking method.*

## 1. INTRODUCTION

Polynomials play a major role in various areas such as pure and applied mathematics, engineering, and social sciences. This paper interested in finding roots of fuzzy polynomial like $A_1x + A_2x^2 + \ldots + A_nx^n = A_0$ where $x^i, A_j \in E^1$ for $i = 1,2,\ldots,n$ $j = 0,1,\ldots,n$. (if exists). The set of all the fuzzy numbers is denoted by $E^1$. The applications of fuzzy polynomials are considered by [1]. The concept of fuzzy numbers and arithmetic operation with this numbers were first introduced by Zadeh[2].

Many researchers have studied on methods for the solution of fuzzy polynomials. Buckley and Eslami[3] have consider neural net solutions to fuzzy problems. Otadi [4,5] have proposed architecture of fuzzy neural networks with crisp weights for fuzzy input vector and fuzzy target. Abbasbandy and Asady [6] have considered Newton's method for solving fuzzy nonlinear equations. Linear and nonlinear fuzzy equations are solved by [5,6,7,8,9,10]. Abbasbandy and Otadi [5] have create $FNN_2$ (fuzzy neural network with fuzzy set input signals and real number weights) equivalent to the system of fuzzy polynomials. To find the real roots of a system of fuzzy polynomials, they introduced a learning algorithm from the cost function for adjusting of crisp weights. In [6,8] the authors have suggested numerical solving method for fuzzy nonlinear equations instead of standard analytical techniques which are not suitable everywhere. They wrote fuzzy nonlinear equation in parametric form and then solve it by Newton's method. Abbasbandy and Alavi [7] have proposed an efficient method for solving a system of $n$ fuzzy linear equations with $n$ variables. The original system with matrix $A$ is replaced by two $n \times n$ crisp systems. The solution vector be symmetric solution if the right hand side vector be symmetric. Asady et al. in [9] have proposed a general model for solving a system of $m$ fuzzy linear equations with $n(m \leq n)$ variables. The original system is replaced by a crisp linear system which is solved by least square method.

Seeing to the so many studies about solution of fuzzy polynomial, this motivates us to make a review on the numerical solutions of fuzzy polynomials. Based on the international journals so far, it shown that no study has been conducted about the review for the solutions of fuzzy polynomials. Specifically, the main aim of this paper is to review some numerical methods in the solution of fuzzy polynomial. There are Newton-Raphson method, ranking method, modified Adomian decomposition method and the last one is numerical solution of fuzzy polynomial by fuzzy neural network. The

ISSN2305-8269
International Journal of Engineering and Applied Sciences

www.eaas-journal.org

structure of this paper is organized as follows: Section 2 contains chronology solutions of fuzzy polynomials by some well-known authors. Section 3 represents some review of multi-solutions of fuzzy polynomials. Then, a simple observation of the overall of this study is given in Section 4 and Conclusion comes in Section 5.

## 2. CHRONOLOGY SOLUTIONS OF FUZZY POLYNOMIALS

In the beginning of 1980s, the Adomian decomposition method has been applied to a wide class of functional equations [11,12]. Adomian gives the solution as of infinite series usually converging to an accurate solution. Then, in 1994 Abbaoui and Cherruault [13] applied the standard Adomian decomposition on simple iteration method to solve the equation $f(x)=0$ where $f(x)$ is a nonlinear function, and proved the convergence of the series solution. In 1997, Buckley and Eslami have showed how neural nets can be used to compute solutions to the fuzzy linear equation $AX=C$. Then they concerned with neural net solutions to $AX^2+BX=C$ [3].

In 2003, Abbasbandy has improve Newton-Raphson method to solve the nonlinear equation $f(x)=0$ based on modified Adomian's method [14]. After one year, Abbasbandy and Asady then applied Newton's method for solving fuzzy nonlinear equation $f(x)=c$ [6]. In 2005, Abbasbandy [15] has extended Newton's method for a system of equation by modification of Adomian decomposition method. In the same year, Kajani et al. also have applied Newton's method in solving dual fuzzy nonlinear equations $f(x)=g(x)+c$ [16].

In 2006, Abbasbandy and Otadi investigate numerical solution of fuzzy polynomials by fuzzy neural network with crisp weights for fuzzy input vector and fuzzy target [4]. They also proposed numerical solution of system of fuzzy polynomial equation for fuzzy neural network in 2008 [5]. In 2007, Allahviranloo et al. [17] applied the fixed point method for solving fuzzy nonlinear equations. Rouhparvar [18] also proposed a new method for solving fuzzy polynomial equation based on ranking method.

In 2009, Allahviranloo and Asari [19] solve fuzzy polynomials with fuzzy coefficients and fuzzy variable, numerically. The fuzzy quantities are presented in parametric form. They first convert the polynomial fuzzy coefficients into parametric form then apply Newton method in each limit. In order to find root, which is also fuzzy number, they have calculate level sets of fuzzy coefficients in each limits. The latest study is by Noor'ani Ahmed et al. [20] which solving dual fuzzy polynomial equation by ranking method.

## 3.REVIEW OF MULTI-SOLUTIONS

In this section, some of numerical solutions in solving fuzzy polynomials will be presented. This paper is simplified and combined based on a review of the eleven international journals of fuzzy polynomials. The following multi-solutions will be presented based on its chronology.

### 3.1 Modified adomian decomposition method

Since the beginning of the 1980s, the Adomian decomposition method has been applied to a wide class of functional equations [11,12]. Adomian gives the solution as infinite series that usually converging to an accurate solution. Then, in 1994 Abbaoui and Cherruault [13] applied the standard Adomian decomposition on simple iteration method to solve the equation $f(x)=0$ , where $f(x)$ is a nonlinear function, and proved the convergence of the series solution.

In 2003, Abbasbandy [14] has improve Newton-Raphson method to solve the nonlinear equation $f(x)=0$ based on modified Adomian's method. Then he extended Newton's method for a system of equation by using modified Adomian decomposition method in 2005 [15]. The purpose is to introduce an efficient extension of Newton's method by modified Adomian decomposition method for solving fuzzy polynomials (if exists) $\sum_{i=1}^{n} a_i x^i = c$ where $x, c$ are fuzzy numbers and all coefficients are fuzzy numbers.

Hence, this study considers the fuzzy polynomial equation

 ISSN2305-8269
International Journal of Engineering and Applied Sciences

www.eaas-journal.org

$$P_n(x)=\sum_{i=1}^{n} a_i x^i = c \qquad (1)$$

where $x, c$ and all coefficients are fuzzy numbers.
Let

$P_n(x)=\left(\underline{P}_n(\underline{x},\overline{x};r),\overline{P}_n(\underline{x},\overline{x};r)\right)$ for $r\in[0,1]$ with

$$\underline{P}_n(\underline{x},\overline{x};r)= \min\left\{\underline{P}_n(u)\middle| u\in[\underline{x}(r),\overline{x}(r)], a_i\in[\underline{a}_i(r),\overline{a}_i(r)]\right\}$$

$$\overline{P}_n(\underline{x},\overline{x};r)= \max\left\{\underline{P}_n(u)\middle| u\in[\underline{x}(r),\overline{x}(r)], a_i\in[\underline{a}_i(r),\overline{a}_i(r)]\right\}$$

The parametric form for any $r\in[0,1]$ is as follows:

$$\begin{cases}\underline{P}_n(\underline{x},\overline{x};r)=\underline{c}(r)\\ \overline{P}_n(\underline{x},\overline{x};r)=\overline{c}(r)\end{cases} \qquad (2)$$

where $c=(\underline{c}(r),\overline{c}(r))$ for $r\in[0,1]$. The problem (2) can reformulated in an equivalent form as

$$\begin{cases}\underline{F}(\underline{x},\overline{x};r)=0\\ \overline{F}(\underline{x},\overline{x};r)=0\end{cases} \qquad (3)$$

Suppose that $(\alpha,\beta)$ is the solution of (3),

$$\begin{cases}\underline{F}(\alpha,\beta;r)=0\\ \overline{F}(\alpha,\beta;r)=0\end{cases}$$

The Newton's method is given by

$$\begin{cases}\underline{x}_{n+1}(r)=\underline{x}_n(r)+h_n(r)\\ \overline{x}_{n+1}(r)=\overline{x}_n(r)+k_n(r)\end{cases} \qquad (4)$$

where $n=0,1,2,\ldots$. For initial guess, one can use the trapezoidal fuzzy number $x_0=\left(\underline{x}(1),\overline{x}(1),\underline{x}(1)-\underline{x}(0),\overline{x}(0)-\overline{x}(1)\right)$ and in parametric form

$$\underline{x}_0(r)=\underline{x}(1)+(\underline{x}(1)-\underline{x}(0))(r-1),$$
$$\overline{x}_0(r)=\overline{x}(1)+(\overline{x}(0)-\overline{x}(1))(1-r)$$

The iteration (4) will converge to $(\alpha,\beta)$ if the starting point $(\underline{x}_0(r),\overline{x}_0(r))$ is close enough to $(\alpha,\beta)$ for $0\le r\le 1$, local convergence property [12]. If we use Taylor's expansion of $\underline{F}(\underline{x},\overline{x};r)$ and $\overline{F}(\underline{x},\overline{x};r)$ to a higher order and we are looking for $h(r)$ and $k(r)$ such as

$$\left[\underline{F}-h\underline{F}_{\underline{x}}-k\underline{F}_{\overline{x}}+\frac{1}{2}\left(h^2\underline{F}_{\underline{x}\underline{x}}+2hk\underline{F}_{\underline{x}\overline{x}}+k^2\underline{F}_{\overline{x}\overline{x}}\right)\right](\underline{x},\overline{x};r)\approx 0$$

$$\left[\overline{F}-h\overline{F}_{\underline{x}}-k\overline{F}_{\overline{x}}+\frac{1}{2}\left(h^2\overline{F}_{\underline{x}\underline{x}}+2hk\overline{F}_{\underline{x}\overline{x}}+k^2\overline{F}_{\overline{x}\overline{x}}\right)\right](\underline{x},\overline{x};r)\approx 0$$

given

$$\begin{bmatrix}h(r)\\k(r)\end{bmatrix}=\begin{bmatrix}c_1\\c_2\end{bmatrix}+N\left(\begin{bmatrix}h(r)\\k(r)\end{bmatrix}\right)=\begin{bmatrix}c_1\\c_2\end{bmatrix}+\begin{bmatrix}N_1(h,k)\\N_2(h,k)\end{bmatrix} \qquad (5)$$

where $c_1=\frac{\underline{F}}{\underline{F}_{\underline{x}}}(\underline{x},\overline{x};r)$ and $c_2=\frac{\overline{F}}{\overline{F}_{\overline{x}}}(\underline{x},\overline{x};r)$ are constants and $N$ is a vector quadratic polynomial and for approximating $h(r)$ and $k(r)$, then we can apply the multivariable Adomian decomposition method [21].

The Adomian decomposition technique considers representing the solution of (5) as a series

$$h=\sum_{n=0}^{\infty}h_n,\; k=\sum_{n=0}^{\infty}k_n \qquad (6)$$

and the nonlinear functions are decomposed as

$$N_i(h,k)=\sum_{n=0}^{\infty}A_{in}(h_0,h_1,\ldots,h_n,k_0,k_1,\ldots,k_n), \qquad (7)$$

where the $A_{in}$'s are Adomian's polynomials given by Abbasbandy [14],

$$A_{in}=\frac{1}{n!}\frac{d^n}{d\lambda^n}\left[N_i\left(\sum_{j=0}^{\infty}\lambda^j h_j,\sum_{j=0}^{\infty}\lambda^j k_j\right)\right]_{\lambda=0}$$

Upon substituting (6), (7) in the (5) yields
$h_0=c_1, h_{n+1}=A_{1n}, k_0=c_2, k_{n+1}=A_{2n}$,
for $n=0,1,\ldots$ multivariable polynomials $A_{in}$ are generated by practical formulae presented in Abbaoui et al. [21], for $i=1,2,\ldots$, we have

$$A_{i0}=N_i(h_0,k_0)$$

$$A_{in}=\sum_{\varphi}\frac{h_1^{p_1}}{p_1!}\cdots\frac{h_n^{p_n}}{p_n!}\cdot\frac{k_1^{q_1}}{q_1!}\cdots\frac{k_n^{q_n}}{q_n!}\cdot\frac{\partial^{\varphi1+\varphi2}}{\partial h^{\varphi1}\partial k^{\varphi2}}N_i(h_0,k_0)$$

where $\varphi$ stands for $(p_1+2p_2+\ldots+np_n)+(q_1+2q_2+\ldots+nq_n)=n$, and

ISSN2305-8269
International Journal of Engineering and Applied Sciences

www.eaas-journal.org

$$\varphi_1 = p_1 + p_2 + \dots + p_n$$
$$\varphi_2 = q_1 + q_2 + \dots + q_n$$

As a simple conclusion, fuzzy polynomials have to be written in parametric form and then solve it by Adomian decomposition method.

Otadi & Mosleh [22]

### 3.2 Newton-Raphson Method

In 2009 Allahviranloo and Asari have solve fuzzy polynomials with fuzzy coefficients and fuzzy variable, numerically. The fuzzy quantities are presented in parametric form. They first convert the polynomial fuzzy coefficients into parametric form then apply Newton method on each limit. In order to find root, which is also fuzzy number, they have calculate level sets of fuzzy coefficients on each limits.

This work interested in finding solution of

$$A_1 x + A_2 x^2 + \dots + A_n x^n = A_0 \quad (8)$$

where $A_i, x^j \in E^1$ for $i = 1,2,\dots,n$ $\quad j = 0,1,\dots,n$

Let $f(x) = \sum_{j=1}^{n} A_j x^j = A_0$ . $f(x)$ is called fuzzy polynomial of degree at most $n$.

Therefore Eq. (8) can be written such as $f(x) = A_0$ . We use of parametric form of the $f(x)$, so Eq. (8) can be in the form of the following formulas

$$f^l(x,h) = \sum_{x^j \ge 0} A_j^l(h) x^j + \sum_{x^j < 0} A_j^u(h) x^j = A_0^l(h)$$
$$f^u(x,h) = \sum_{x^j \ge 0} A_j^u(h) x^j + \sum_{x^j < 0} A_j^l(h) x^j = A_0^u(h) \quad (9)$$

Assume $f^l(x,h)$ and $f^u(x,h)$ has at least two continuous derivatives for all $x$ in some intervals about the roots $\alpha^l(h)$ and $\alpha^u(h)$ respectively.

By using Eq. (9), $[f(x)]_h$ has such form as follows

$$f^l(x,h) = \sum_{x^j \ge 0} A_j^l(h) x^j + \sum_{x^j < 0} A_j^u(h) x^j - A_0^l(h)$$
$$f^u(x,h) = \sum_{x^j \ge 0} A_j^u(h) x^j + \sum_{x^j < 0} A_j^l(h) x^j - A_0^u(h) \quad (10)$$

From (10), we obtain

$$f'^l(x,h) = \sum_{x^j \ge 0} jA_j^l(h) x^{j-1} + \sum_{x^j < 0} jA_j^u(h) x^j$$
$$f'^u(x,h) = \sum_{x^j \ge 0} jA_j^u(h) x^{j-1} + \sum_{x^j < 0} jA_j^l(h) x^{j-1}$$

Now, by applying Newton's formula we have

$$x_{n+1}^l(h) = x_n^l(h) - \frac{f^l(x_n,h)}{f'^l(x_n,h)}$$
$$= x_n^l(h) - \frac{\sum_{x^j \ge 0} A_j^l(h) x^j + \sum_{x^j < 0} A_j^u(h) x^j - A_0^l(h)}{\sum_{x^j \ge 0} jA_j^l(h) x^{j-1} + \sum_{x^j < 0} jA_j^u(h) x^j}$$

$$x_{n+1}^u(h) = x_n^u(h) - \frac{f^l(x_n,h)}{f'^u(x_n,h)}$$
$$= x_n^u(h) - \left[ \sum_{x^j \ge 0} A_j^u(h) x^j + \sum_{x^j < 0} A_j^l(h) x^j - A_0^u(h) \right]$$

By applying Newton's method for parametric Eq. (10) with initial points $x_0^l(h) = m + \alpha(h-1)$ and $x_0^u(h) = m + \beta(1-h)$, we can obtain roots of $f^l(x,h)$ and $f^u(x,h)$ respectively. Two sequences $\{x_{n+1}^l(h)\}$ and $\{x_{n+1}^u(h)\}$ must convergent to $\alpha^l(h)$ and $\alpha^u(h)$ respectively.

However in 2005, Kajani et al. [16] have applied Newton's method for solving a dual fuzzy nonlinear system. This is because according to fuzzy arithmetic, dual fuzzy nonlinear system cannot be replaced by a fuzzy nonlinear system. Therefore Newton's method was applied for solving a dual fuzzy nonlinear system.

Usually there is no inverse element for an arbitrary fuzzy number $x \in E$, there exists no element $y \in E$ such that $x + y = 0$. Actually for all non-crisp fuzzy number $x \in E$ , we have $x + (-x) \ne 0$. Therefore the fuzzy nonlinear equation system $H(x) = G(x) + c$ cannot be equivalently replaced by the fuzzy nonlinear equation $H(x) - G(x) = c$ which had been investigated.

 ISSN2305-8269
International Journal of Engineering and Applied Sciences

www.eaas-journal.org

So, aim of this study is to obtain a solution for duality of fuzzy nonlinear system $H(x)=G(x)+c$ whose parametric form is as follows:

$$\begin{cases}\underline{H}(\underline{x},\overline{x},r)=\underline{G}(\underline{x},\overline{x},r)+\underline{c}(r)\\ \overline{H}(\underline{x},\overline{x},r)=\overline{G}(\underline{x},\overline{x},r)+\overline{c}(r)\end{cases}$$

where $\forall r\in[0,1]$.

Kajani et al. [16]

### 3.3 Fuzzy Neural Network

In 2006, Abbasbandy and Otadi [4] have investigate numerical solution of fuzzy polynomials by fuzzy neural network with crisp weights for fuzzy input vector and fuzzy target. They have proposed architecture of fuzzy neural networks with crisp weights for fuzzy input vector and fuzzy target. The input-output relation of each unit is defined by the extension principle of Zadeh [23]. Output from the fuzzy neural network, which is also fuzzy number, is numerically calculated by interval arithmetic [24] for crisp weights and level sets of fuzzy inputs. Then, a cost function for the level sets of fuzzy output and fuzzy target is defined. A crisp learning algorithm is derived from the cost function to find the real root of the polynomials.

Therefore this study interested in finding solution

$$A_1x+A_2x^2+.....+A_nx^n=A_0 \qquad (11)$$

for $x\in R$, when $A_i\in E,\ for\ i=1,....,n$. A FNN$_2$ (fuzzy neural network with fuzzy set input signals and real number weights) solution to Eq. (11) is given in Figure 1. The input neurons make no change in their inputs, so the input to the output neuron is $A_1x+A_2x^2+.....+A_nx^n$ and the output, in the output neuron, equals its input, so

$$Y=A_1x+A_2x^2+.....+A_nx^n$$

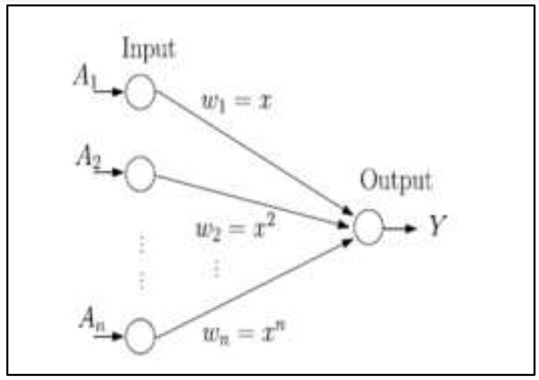


*Figure 1: Fuzzy neural network to solve fuzzy polynomial*

How is the FNN2 going to solve the fuzzy polynomials? The training data is $(A_1,......,A_n)$ for input and target (desired) output is $A_0$. Hence a learning algorithm from the cost function for adjusting weights is proposed. Let the h-level sets of the target output $A_0$ are denoted by

$$[A_0]_h=\left[[A_0]_h^L,[A_0]_h^U\right],\ h\in[0,1] \qquad (12)$$

where $A_0^L(h)$ denotes the left-hand side and $A_0^U(h)$ denotes the right-hand side of the h-level sets of the desired output. A cost function to be minimized is defined for each h-level sets as follows:

$$e(h)=e^L(h)+e^U(h)$$

$$e^L(h)=\frac{1}{2}\left(A_0^L(h)-Y^L(h)\right)^2$$

$$e^U(h)=\frac{1}{2}\left(A_0^U(h)-Y^U(h)\right)^2$$

hence $e^L(h)$ denotes the error between the left-hand sides of the h-level sets of the desired output, and $e^U(h)$ denotes the error between the right-hand sides of the h-level sets of the desired and the computed output. Then the error function for the training pattern is

$$e=\sum_h he(h) \qquad (13)$$

Theoretically this cost function satisfies the following equation if we use infinite number of h-level sets in Eq. (13) $e\rightarrow 0$ if and only if

ISSN2305-8269
International Journal of Engineering and Applied Sciences

www.eaas-journal.org

$Y \to A_0$. The weights are updated by the following rules

$$\Delta w_1(t) = -\eta \sum_h h \frac{\partial e(h)}{\partial w_1} + \alpha \cdot \Delta w_1(t-1) \quad (14)$$

where $\eta$ is a learning constant, $\alpha$ is a momentum constant and $t$ indexes the number of adjustments. The derivatives in Eq. (14) can be written as follows:

$$\frac{\partial e(h)}{\partial w_1} = \frac{\partial e^L(h)}{\partial w_1} + \frac{\partial e^U(h)}{\partial w_1}$$

$$\frac{\partial e^L(h)}{\partial w_1} = \frac{\partial e^L(h)}{\partial Y^L} + \frac{\partial Y^L(h)}{\partial w_1}$$

$$\frac{\partial e^U(h)}{\partial w_1} = \frac{\partial e^U(h)}{\partial Y^U} + \frac{\partial Y^U(h)}{\partial w_1}$$

$$\frac{\partial e^L(h)}{\partial Y^L} = -\left(A_0^L(h) - Y^L(h)\right),$$

$$\frac{\partial e^U(h)}{\partial Y^U} = -\left(A_0^U(h) - Y^U(h)\right)$$

If $w_1 \geq 0$

$$\frac{\partial Y^L(h)}{\partial w_1} = A_1^L(h), \quad \frac{\partial Y^U(h)}{\partial w_1} = A_1^U(h)$$

otherwise

$$\frac{\partial Y^L(h)}{\partial w_1} = A_1^U(h), \quad \frac{\partial Y^U(h)}{\partial w_1} = A_1^L(h)$$

Therefore if $w_1 \geq 0$

$$\Delta w_1(t) = \eta \sum_h h \begin{bmatrix} \left(A_0^L(h) - Y^L(h)\right) A_1^L(h) + \\ \left(A_0^U(h) - Y^U(h) A_1^U(h)\right) \end{bmatrix} + \alpha \cdot \Delta w_1(t-1)$$

otherwise

$$\Delta w_1(t) = \eta \sum_h h \begin{bmatrix} \left(A_0^L(h) - Y^L(h)\right) A_1^U(h) + \\ \left(A_0^U(h) - Y^U(h) A_1^L(h)\right) \end{bmatrix} + \alpha \cdot \Delta w_1(t-1)$$

We can adjust other weights by $w_i = w_1^i \;\; for \;\; i = 1,2,\ldots,n$. The fuzzy polynomials may have no real root for crisp $w_i, 1 \leq i \leq n$. In this case, there is no hope in making the error measure close to zero.

Abbasbandy & Otadi [4]

Two years later in 2008, Abbasbany et al. [5] solving systems of fuzzy polynomials based on fuzzy neural network (FNN). Previous paper by Abbasbandy and Otadi [4] was aimed at solving a fuzzy polynomial equation. However in this work, they consider to solve systems of fuzzy polynomials in the form of:

$$\begin{cases} A_{11}xy + A_{12}x^2y^2 + \ldots + A_{1n}x^ny^n = A_{10} \\ A_{21}xy + A_{22}x^2y^2 + \ldots + A_{2n}x^ny^n = A_{20} \end{cases}$$

where $x, y \in R$ and $A_{10}, A_{11}, \ldots, A_{1n}, A_{20}, A_{21}, \ldots\ldots, A_{2n}$ are fuzzy numbers.

Abbbasbandy et al. [5]

### 3.4 Ranking Method

In 2007, Rouhparvar [18] has proposed a new method in solving fuzzy polynomial equation based on ranking method. The ranking method was firstly introduced by Delgado et al. [25,26]. This work interested in finding real roots of polynomial equation like

$$A_1x + A_2x^2 + \ldots. + A_nx^n = A_0 \quad (15)$$

that $x \in R$ and $A_0, A_1, \ldots\ldots, A_n$ are fuzzy numbers by a ranking method of fuzzy numbers.

They introduced three real indices called Value, Ambiguity and Fuzziness to obtain simple fuzzy numbers that could be used to represent more arbitrary fuzzy numbers. Therefore these parameters were used for transform fuzzy polynomial equation to system of crisp polynomial equations. By solving this system, real roots of fuzzy polynomial equation can be found.

Let $x$ be a real solution for (15), then

ISSN2305-8269

International Journal of Engineering and Applied Sciences

www.eaas-journal.org

$$\begin{cases} V\left(C_1x + C_2x^2 + \ldots + C_nx^n\right) = V(C_0) \\ A\left(C_1x + C_2x^2 + \ldots + C_nx^n\right) = A(C_0) \\ F\left(C_1x + C_2x^2 + \ldots + C_nx^n\right) = F(C_0) \end{cases}$$

implies for

$$\begin{cases} V(C_1)x + \ldots + V(C_n)x^n = V(C_0) \\ A(C_1)|x| + \ldots + A(C_n)|x^n| = A(C_0) \\ F(C_1)|x| + \ldots + F(C_n)|x^n| = F(C_0) \end{cases} \quad (16)$$

To find the real solution of (15), it is enough to solve the above system of crisp polynomial equations with crisp methods. Even though there is a system of crisp polynomial equations but it is only enough to solve one equation of (16) and finding its real roots, then real roots of that equation satisfy in both others equations which are the real solutions of system (16). Also, for solving (16) can consider two states:

Suppose $x \geq 0$, then

$$\begin{cases} V(C_1)x + \ldots + V(C_n)x^n = V(C_0) \\ A(C_1)x + \ldots + A(C_n)x^n = A(C_0) \\ F(C_1)x + \ldots + F(C_n)x^n = F(C_0) \end{cases}$$

that we obtain positive real roots. When $x < 0$ then

$$\begin{cases} V(C_1)x + \ldots + V(C_n)x^n = V(C_0) \\ -A(C_1)x + \ldots + (-1)^n A(C_n)x^n = A(C_0) \\ -F(C_1)|x| + \ldots + (-1)^n F(C_n)x^n = F(C_0) \end{cases}$$

where we obtain negative real roots.

Let $A_i = (a_i, b_i, \alpha_i, \beta_i)$, $i = 0,1,\ldots,n$ trapezoidal fuzzy number. It is easy to imply that

$$V(A_i) = \frac{a_i + b_i}{2} + \frac{\beta_i - \alpha_i}{6}$$

$$A(A_i) = \frac{b_i - a_i}{2} + \frac{\beta_i + \alpha_i}{6}$$

$$F(A_i) = \frac{\beta_i + \alpha_i}{4}$$

For any triangular fuzzy number $A_i = (m_i, \alpha_i, \beta_i)$, $i = 0,1,\ldots,n$ fuzzy polynomial equation is transformed to system of crisp polynomial equations. Then we obtain

$$V(A_i) = m_i + \frac{\beta_i - \alpha_i}{6}$$

$$A(A_i) = \frac{\beta_i + \alpha_i}{6}$$

$$F(A_i) = \frac{\beta_i + \alpha_i}{4}$$

Rouhparvar[18]

According to the above work by Rouhparvar, then on 2010 Noor'ani et al. presents a system of fuzzy polynomials by ranking method. In this study, a linear ranking method was applied to order trapezoidal and triangular fuzzy numbers. Then system of fuzzy polynomials is transformed to system of crisp polynomials based on Value, Ambiguity and Fuzziness. Overall, this study is similarly with Rouhparvar [18]. What sets it apart is, this study solves systems of fuzzy polynomials while Rouhparvar [18] is a study in solving fuzzy polynomial equation. Hence this study interested in finding real roots of system of fuzzy polynomials in the form

$$\begin{cases} A_{11}xy + A_{12}x^2y^2 + \ldots + A_{1n}x^ny^n = A_{10} \\ A_{21}xy + A_{22}x^2y^2 + \ldots + A_{2n}x^ny^n = A_{20} \end{cases}$$

where $x, y \in R$

and $A_{10}, A_{11}, \ldots, A_{1n}, A_{20}, A_{21}, \ldots, A_{2n}$ are fuzzy numbers, by a ranking method of fuzzy numbers.

One year after that, on 2011 Noor'ani et al. solving dual fuzzy polynomial equation by ranking method. In this study, the concept of ranking method is proposed to find real roots of a dual fuzzy polynomial equation by ranking fuzzy numbers. Hence this study interested in finding real roots of dual fuzzy polynomials in the form

$$A'_1x + A'_2x^2 + \ldots + A'_nx^n = A''_1x + A''_2x^2 + \ldots + A''_nx^n + A_0$$

where $x \in R$ and $A'_1, \ldots, A'_n, A''_1, \ldots, A''_n, A_0$ are fuzzy numbers, by a ranking method of fuzzy numbers.

Noor'ani et al.[20]

## 4. OBSERVATIONS

In this paper, 11 journal articles which appeared in the period from 1997-2011, solving the fuzzy polynomial equation are reviewed. According to those articles, it shown that there are four popular numerical methods that always investigated by some authors. They are Newton-Raphson method, Adomian decomposition method, fuzzy neural network, and ranking method. However, it can be conclude that the most popular method is ranking method. This is according to the previous studies that using ranking method as a solution of equation, systems as well as dual fuzzy polynomials.

## 5. CONCLUSIONS

In this study, some of numerical methods have been presented as a solution of fuzzy polynomial. This review shows that the real roots of fuzzy polynomial can be found with different algorithms. However sometimes there are no real roots exists in certain fuzzy polynomial. Solution of fuzzy polynomial by ranking method was introduced to solve fuzzy polynomial equation that transform to a system crisp of polynomial equations, so that it system easy solvable. For finding the real roots of it system whereas has no exact solution, iteration methods can be used for approximation solution. By modified Adomian decomposition method, the real roots can be found by wrote fuzzy polynomials in parametric form and then solved it by Adomian decomposition method. Solution by Newton-Raphson method shows that the real roots of fuzzy polynomial will be found on initial step with almost high accuracy. Differently with fuzzy neural network, the real root of fuzzy polynomial can be found by introduce a learning algorithm of crisp weights of two-layer feedforward fuzzy neural network. So that input-output relations were defined by the extension principle. Hopefully, this simple review will provide an input for those interested in the study of fuzzy polynomials .

## References

[1] S. Abbasbandy and M. Amirfakhrian, Numerical approximation of fuzzy function by fuzzy polynomials, Applied Mathematics and Computation 174 (2006), 669-675.

[2] L.A. Zadeh, Fuzzy sets, Information Control 8 (1986), 338-353.

[3] J.J. Buckley and E. Eslami, Neural net solution of fuzzy problems: the quadratic equation, Fuzzy Sets and Systems. 86 (1997), 289-298.

[4] S. Abbasbandy and M. Otadi, Numerical solution of fuzzy polynomials by fuzzy neural network, Applied Mathematics and Computation 181 (2006), 1084-1089.

[5] S. Abbasbandy, M. Otadi and M. Mosleh, Numerical solution of a system of fuzzy polynomials by fuzzy neural network, Information Science 178 (2008), 1948-1960.

[6] S. Abbasbandy and B. Asady, Newton's method for solving fuzzy nonlinear equations, Applied Mathematics and Computation. 159 (2004), 379-356.

[7] ] S. Abbasbandy and M. Alavi, A method for solving fuzzy linear systems, Iran Journal Fuzzy Sets. 2 (2005), 37-43.

[8] S. Abbasbandy and R. Ezzati, Newton's method for solving fuzzy nonlinear equations, Applied Mathematics and Computation. 175 (2006), 1189-1199.

[9] B. Asady, S. Abbasbandy and M. Alavi, Fuzzy general linear systems, Applied Mathematics and Computation. 169 (2005), 34-40.

[10] J.J. Buckley and Y. Qu, Solving linear and neural net solution of fuzzy problems: the quadratic equation, Fuzzy Sets and Systems 86 (1997), 289-298.

[11] G. Adomian, Nonlinear stochastic system and approximations to physics. Kluwer, Dordrecht, 1989.

[12] G. Adomian and R. Rach, On the solution of algebraic equations by the decomposition method. Journal of Mathematical Analysis and Applications 105(1) (1985), 141-166.

[13] K. Abbaoui and Y. Cherruault, Convergence of Adomian's method applied to nonlinear equations. Mathematical Computer and Modelling 20(9) (1994), 69-73.

[14] S. Abbasbandy, Improving Newton-Raphson method for nonlinear equations by modified Adomian decomposition method. Applied Mathematics and Computation, 145 (2003), 887-893.

[15] S. Abbasbandy, Extended Newton's method for a system of nonlinear equations by modified Adomian decomposition method. Applied Mathematics and Computation 170 (2005), 648-656.

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

[17] T. Allahviranloo, M. Otadi and M. Mosleh, Iterative method for fuzzy equations. Soft Computing 12 (2007), 935-939.

[18] H. Rouhparvar, Solving fuzzy polynomial equation by ranking method. First Joint Congress on Fuzzy and Intelligent Systems, Ferdowsi University of Mashhad, Iran, 2007.

[19] T. Allahviranloo and S. Asari, Numerical solution of fuzzy polynomials by Newton-Raphson method. Journal of Applied Mathematics, Islamic Azad University of Lahijan, Vol. 7, No. 4(27) (2011), 17-24.

[20] A. Noor'ani, J. Kavikumar, M. Mustafa and S. Nor, Solving dual fuzzy polynomial equation by ranking method. Far East Journal Mathematics and Science 51, no. 2 (2011), 151-163.

[21] K. Abbaoui, Y. Cherruault and V. Seng, Practical formulae for the calculuse of multivariable Adomian polynomials. Mathematical Computer and Modelling 22(1) (1995), 89-93.

[22] M. Otadi and M. Mosleh, Solution of fuzzy polynomial equations by modified Adomian decomposition method. Soft Computing 15 (2011), 187-192.

[23] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning: Parts 1-3, Information Sciences 8, vol. 9 (1975), 199-249, 301-357, 43-80.

[24] G. Alefeld and J. Herzberger, Introduction to interval computations, Academic Press, New York, 1983.

[25] M. Delgado, M.A. Vila and W. Voxman, On a canonical representation of fuzzy Numbers, Fuzzy Sets and Systems 93 (1998), 125-135

[26] M. Delgado, M.A. Vila and W. Voxman, A fuzziness measure for fuzzy numbers: Application, Fuzzy Sets and Systems 94 (1998), 205-216.

[27] A. Noor'ani, J. Kavikumar, M. Mustafa and S. Nor, Solving a system of fuzzy polynomials by ranking method, Far East Journal of Applied Mathematics, 43(1) (2010), 7-20.