USING THE CONCEPT OF FUZZY LOGIC AND CONTROLLER SYSTEM IN INTERPOLATION AND FUNCTIONS APPROXIMATION

Awni M. Abu-Saman

Alazhar University-Gaza (AUG), Palestine

† Sabbatical leave at Stellenbosch University, South Africa

Email: a.saman@alazhar.edu.ps

Abstract:

The main aim of this contribution is to present an alternative approach for interpolation and function approximation from a given data. Instead of the traditional interpolation methods we consider and propose a numerical procedure for interpolation using the concept of fuzzy logic and membership functions. The method can be used for interpolating data resulting from physical experiments, engineering, medicine, applied sciences etc. Fuzzification will be applied to the given data according to Mamdani technique and membership functions will be chosen to satisfy the interpolation mathematical condition. The defuzzification process will be implemented to get the crisp values of the interpolation. The procedure will be implemented on the mathematical code MATLAB and its Simulink fuzzy logic features. Finally the applicability and efficiency of the procedure is illustrated by numerical examples.

2010 AMS Subject Classification: 68T37, 65N22, 26A33, 41A55.

Keywords: *Fuzzy sets, Fuzzy logic, Fuzzy Controller System (FCS), Crisp sets Crisp logic, Membership functions, interpolation.*

1. Introduction

Interpolation is at root a simple mathematical concept. It is considered to be a mathematical and statistical method by which related known data are used to estimate an unknown values. Interpolation is achieved by using other established data that are located in the domain of the unknown value.

In the mathematical field of numerical analysis, interpolation is a type of estimation, it's a method of introducing and constructing new data points based on the range of a discrete set of known or given data points. [6, 9] It's widely used to replace complicated and non-smooth functions by polynomials or splines.

There is a rich history behind interpolation and function approximation. It really began with the early studies of astronomy when the motion of heavenly bodies was determined from periodic observations. The names of many famous mathematicians are associated with interpolation: Gauss, Newton, Bessel, and others.[9]

Interpolation is widely needed and used in engineering, applied sciences and medical sciences, where often has a number of available data points, obtained by sampling or experimentation, which represent the values of a function for a limited number of values of the independent variable. It is often required to interpolate; that is, estimate the value of that function for an intermediate value of the independent variable.

There are different types of interpolation. The n^{th} degree polynomial interpolation has been used in many interpolation methods for uniform and non-uniform data points, i.e. Newton-Gregory, Lagrange, Divided difference methods, least square method … etc.. Linear interpolation uses a linear function for each sub-interval $[x_i, x_{i+1}]$. Spline interpolation is another type that uses low-degree polynomials in each of the sub-intervals, and chooses the polynomial pieces such that they fit smoothly together. The resulting function is called a spline. For example, the natural cubic spline is piecewise cubic and twice continuously differentiable. Furthermore, its second derivative is zero at the end points. B-splines are often used to numerically integrate and differentiate functions that are defined only through a set of data points. [6, 9]

Saman [1] used the concept of fuzzy logic and system controllers together with the membership functions to give more reasonable values for the loss precisions in solving a system of equations. He considered the condition number and the rate of singularity of the system of equation as an input variables and the rate of accuracy of computed solution as the output variable. Fuzzy logic allows us also to model a non-specific mathematical and scientific language notions, especially, so called vogue expressions (small rate, very small determinant, large condition number, nearly singular matrix … etc.).

The aim of this paper is to overview the main concepts of fuzzy control based on logic and its applications. Moreover, to provide a technique using fuzzy control developed by Zadeh [18-21] and Mamdani [11-13] to help in choosing and evaluating the technique that best suits the given data set: the fuzzy membership functions and the logic rules will be chosen carefully to improve the quality of interpolation compared to traditional approximation methods, and to satisfy the interpolation condition at the given data. Data sets in the examples will be chosen to be connected to known functions to facilitate computing the actual error. Examples will be given with the given data as input or behaviour parameters , and the interpolating values as output or action parameters. The input crisp data will be fuzzified, and then defuzzified according to given rules.

2. Basic Concept of Fuzzy Logic

Fuzzy logic is considered as the concept of linguistic variable, it was supposed to be the elite tool for computing with wards variables where computers couldn't process linguistic variables. It considers linguistic variables, that is, variables whose values are not numbers but words or sentences in a natural or artificial language[15]. The term fuzzy logic was introduced with the 1965 proposal of fuzzy set theory by Lotfi

Zadeh who is considered to be the father of fuzzy logic. [18-20] Fuzzy logic had, however, been studied since the 1920s, as infinite-valued logic notably by Łukasiewicz and Tarski.[1, 10, 13, 21] It provided mathematicians with an appropriate tool for modelling vagueness phenomenon and shed new light into non-digital variables and control theory for engineers.[1-3, 8, 10, 13]

Fuzzy set and crisp set are the part of the distinct set theories, where the fuzzy set implements and allow the whole interval [0 1] to be the range of their characteristic functions which we call membership functions $\mu_A(x)$, and employ infinite-valued logic. The fuzzy membership functions $\mu_A(x)$ enabled us to overcome the difficulty of having very different control actions for a small a change in the inputs. while crisp set employs bi-valued logic and normally gives different actions for a very close similar inputs. [13, 18]

$$\mu_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

Where A is a classical (crisp) set.

In logic, fuzzy logic is a form of many-valued logic in which the truth value of variables may be any real number between 0 and 1 both inclusive $0 \leq \mu_A(x) \leq 1$. It is employed to handle the concept of partial truth, where the truth value may range between completely true and completely false. By contrast, in crisp logic, the truth values of variables may only be the integer values 0 or 1.[13, 20]

Since computers can't process wards, fuzzy logic works as a bridge between words and numbers. It works in combination with a tool named membership functions $\mu_A(x)$ in a way that mimics Crisp logic.

membership functions $\mu_A(x)$ quantifies the degree of belongingness of the element x to the fuzzy set A. The type and the number of membership functions control the demanded degree of accuracy.

There is a significant difference between the classical crisp sets and fuzzy sets, the classical sets are considered as a subset of the fuzzy sets. The classical sets are a yes or no sets and defined as:

$$A = \{\, x \mid x \varepsilon\, X \; and \;\; x \; is \;\; \Delta \,\}$$

Where X is the universe of discourse and x has some property Δ.

While in the fuzzy set, every element x should be combined with a membership $\mu_A(x)$, determines it's degree of membership

$$A_{fuzzy} = \{\, (x, \mu_A(x) \mid x \epsilon\, X, \mu_A(x) \epsilon [0, 1]\}$$

To this end, replacements for basic operators and, or must be available. There are several ways to this. A common replacement is called the Zadeh operators [16], where the $\cap(x, y)$ and $\cup(x, y)$ in crisp logic are replaced by $min(x, y)$and $max(x, y)$ respectively.

Let A and B be two fuzzy sets in D with membership functions $\mu_A(x)$ and $\mu_B(x)$ respectively. Then the union and intersection sets operations are defined in terms of their membership functions $\mu_A(x)$as following,

Definition 1 [10]: The membership function of $A \cup B$ is denoted by $\mu_{A \cup B}(x)$, and defined pointwise for all $x \in D$ by,

$\mu_{A \cup B}(x)$= max$\{\mu_A(x), \mu_B(x)\}$

Definition 2 [10]: The membership function of $A \cap B$ is denoted by $\mu_{A \cap B}(x)$, and defined pointwise for all $x \in D$ by,

$\mu_{A \cup B}(x)$= min$\{\mu_A(x), \mu_B(x)\}$

Fuzzy logic is considered to be a basic control system that relies on the degrees of state of the input and the output depends on the state of the input and rate of change of this state. [5] In other words, a fuzzy logic system works on the principle of assigning a particular output (action) depending on the probability of the state of the input. An interesting property is that the behaviour of a fuzzy system is not described using algorithms and formulas, but rather as a set of rules that may be expressed in natural language.

3. Fuzzy Control System (FCS)

In 1974, fuzzy logic was applied and implemented practically for the first time by Mamdani and his colleagues [13] in steam engine control theory. fuzzy control based on fuzzy logic and fuzzy sets has been considered as one of the most active and valuable topics for research in the application of fuzzy logic and fuzzy set theory. Fuzzy control is a logical system which is much closer in spirit to human thinking and natural language than traditional logic systems.

Mamdani and his colleagues [11- 13] have developed the theory of fuzzy control systems based on the theory of fuzzy sets motivated by the pioneer research of Zadeh's [18-20] on the linguistic approach and system analysis. Mamdani's work opened the way for industrial and practical applications of fuzzy control in water quality control, automatic train operations systems, automatic transmission system, nuclear reactors, city planning and many other applications in engineering and medicine.

Fuzzy logic controllers have many advantages over the conventional controllers [7]: they are cheaper to develop, they cover a wider range of operating conditions, and they are more readily customizable in natural language terms.[13, 19]

Unlike the other logic controllers, fuzzy controller also prevents different actions for almost same behaviours. [19]

After 1990's, Fuzzy sets and logic were made more practically useful [10, 21]. Several models based on the fuzzy sets, the fuzzy transform has been proposed and widely used as a numerical tools for solving differential equations and image processing [1, 2, 3] . Fuzzy logic allows scientists to model vogue human language notions, especially, so called linguistic expressions (small, very big, more or less … etc.).

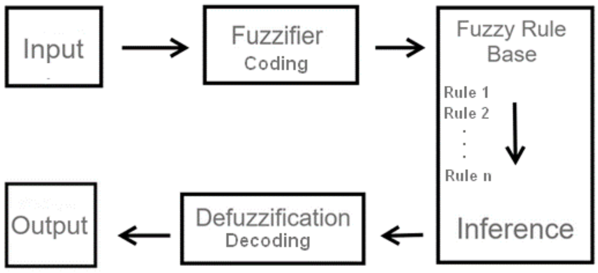

Fig. 1 Fuzzy controller system (FCS)

Numerical Steps:

Given the set of experimental data or values that satisfy a function or a mathematical equation,
Define the inputs and output variables,
Define the descriptors of the input and output variables.
Define membership functions for each of the input and output variables, such that the given data can be satisfied or nearly satisfied i.e.

$$\mu_A(x_i) = 1, i = 0,1,2,\dots,n$$

x_i are the given input data

Form the logic rules based on the mathematical problem to satisfy or nearly satisfy the given data (i.e. consistent with the interpolation condition) ,
Evaluate the Rules,
Defuzzification,
Increase the number of membership functions or their definition to get better results,
Results

4. Numerical Computations

Example One:

Given the following equally spaced crisp values $x, f(x)$ for the function $f(x) = \cos(x)$

x	-0.4	-0.2	0	0.2	0.4	0.6	0.8
$f(x)$	0.920	0.980	1.00	0.980	0.921	0.825	0.697

Table 1 Equally spaced values $x, f(x)$ for the function $\cos(x)$

MATLAB Simulink [14] features used to describe and design a fuzzy logic controller to determine the interpolating value i.e. the values of $f(x) = \cos(x), x\ \varepsilon\ [-0.4, 0.8]$. the x values are assumed as an input crisp values, and $f(x)$ as output crisp value. The x-input and y-output fuzzy descriptors are shown in Table 2, seven descriptors were used for the x values, five descriptor for the output interpolating variable. The triangular shape functions representing the input and output are shown in Fig 2 and Fig 3 respectively. The centroid defuzzification option and non-uniform triangular memberships in MATLAB [14] are used for the output. The output variable is the action that should be taken to satisfy the function *f(x)* at the input values i.e. $\mu_A(x_i) = 1, i = 0,1,2, \ldots, n$.

Table 3 and Fig. 4 show the approximated y values of cos(x) obtained for a selected values x in the given domain. the graph agree well with the graph of cos(x) in the interval [-0.4, 0.8]. The output values are considered as an action that should be taken due to the given input values. The membership functions and the fuzzy rules are designed such that the approximated y values are consistent with the neighbouring given values. According to the defined membership functions and fuzzy rules, fig 4 show the MATLAB computations for an input x= 0 .42, the output result is y=0.905, where the true value is 0.913, the error is 0.008.

The input fuzzy descriptors for x	**The output fuzzy descriptors *y***
S1X: Extremely small x values, S2X: Very small x values, SX: Small x values, M1X: Medium small x values, M2X: Medium x values, LX: Large x values, VLX: Very large x values.	VSY: Very small interpolation value, SY: Small interpolation value, SMY: Small Medium interpolation value, MY: Medium interpolation value , LY: Large interpolation value,

Table 2 The *x*-input and *y*-output fuzzy descriptors

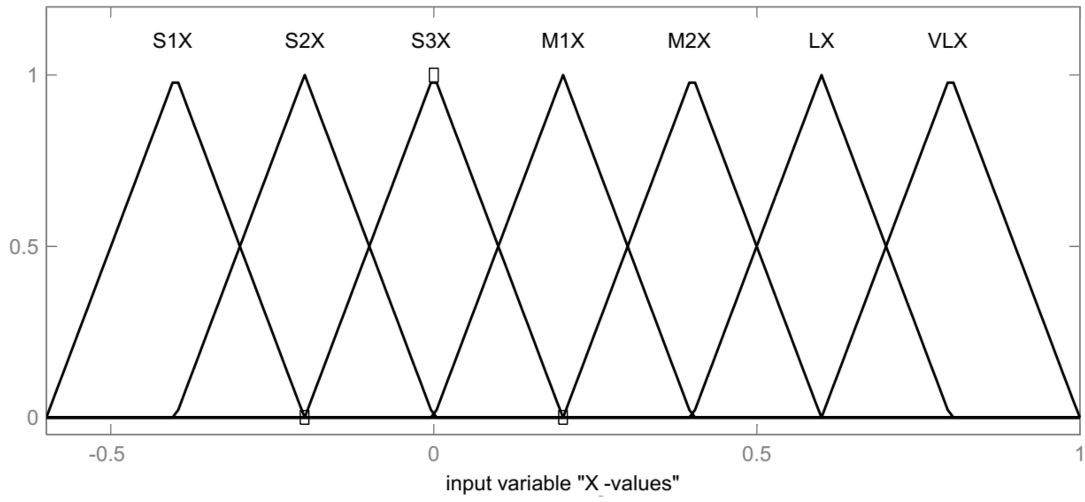

Fig. 2 The uniform triangular membership functions used to define the input.

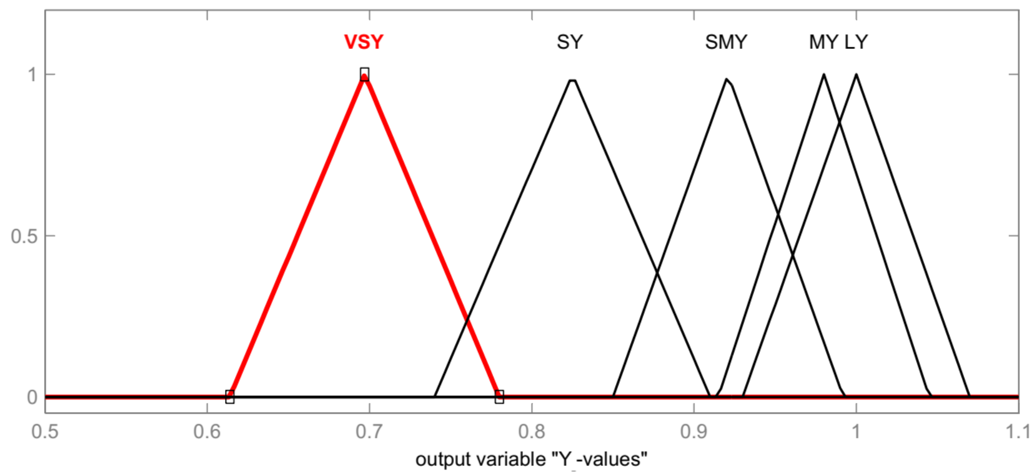

Fig. 3 The non-uniform triangular membership functions used to define the output.

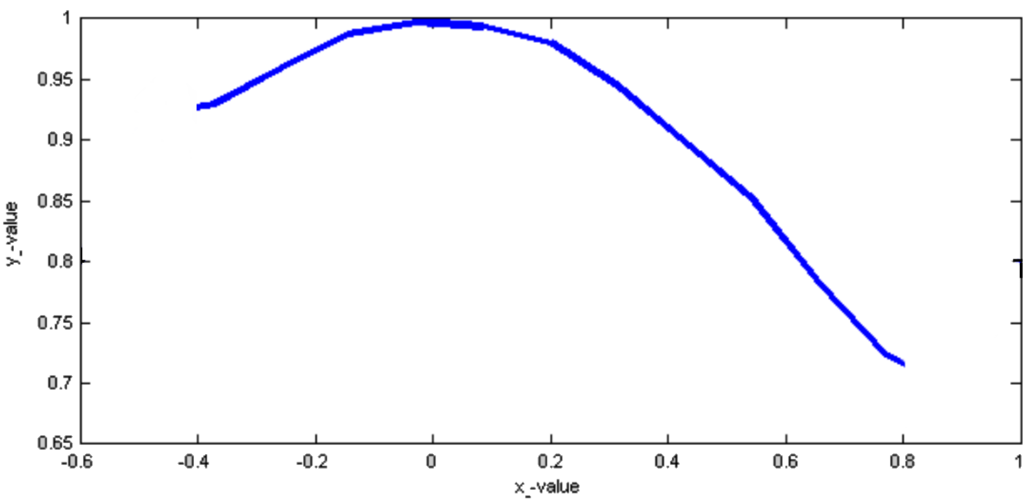

Fig. 4 A two dimensional surface figure for the input and output parameters.

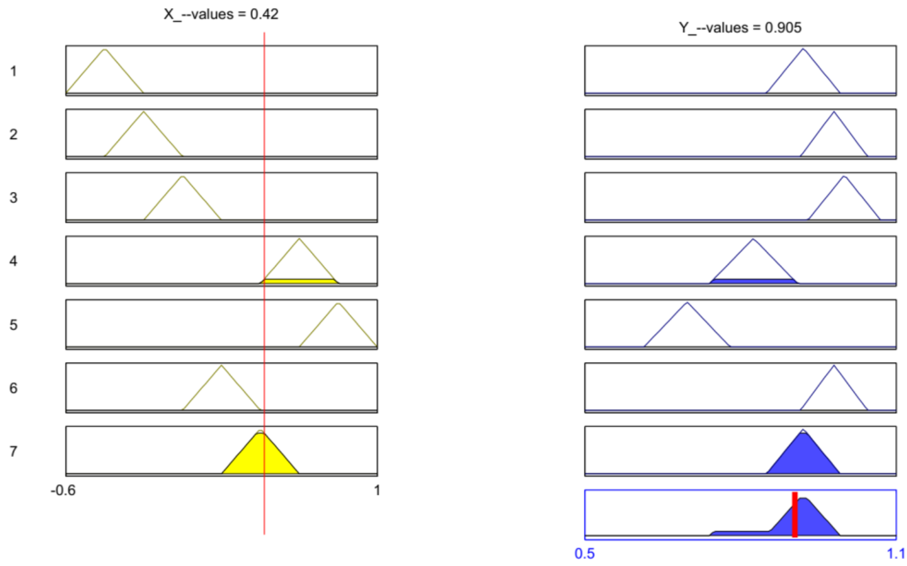

Fig. 5 MATLAB Rule figure for the interpolation at x=0.42

x	Interpolation value	Abs. actual error	x	Interpolation value	Abs. actual error
-0.35	0.936	0.003	0.39	0.924	0.001
-0.3	0.948	0.007	0.42	0.905	0.008
-0.25	0.961	0.008	0.57	0.841	0.001
-0.1	0.992	0.003	0.62	0.806	0.008
-0.15	0.986	0.003	0.69	0.766	0.005
0.1	0.992	0.003	0.7	0.762	0.003
0.15	0.986	0.003	0.73	0.748	0.003
0.25	0.961	0.008	0.75	0.738	0.006
0.3	0.948	0.007	0.77	0.725	0.007
0.35	0.936	0.003	0.79	0.708	0.004

Table 3 The interpolation crisp output, the inputs x and the absolute actual error.

s	x	f	Δf	$\Delta^2 f$	$\Delta^3 f$	$\Delta^4 f$
-1	-0.4	0.921	0.059	-0.039	-0.001	0.002
0	-0.2	0.980	0.020	-0.04	0.001	0.001
1	0.0	1.000	-0.02	-0.039	0.002	0.003
2	0.2	0.980	-0.059	-0.037	0.005	
3	0.4	0.921	-0.096	-0.032		
4	0.6	0.825	-0.128			
	0.8	0.697				

Table 4 Difference table for $f(x) = \cos(x)$

In order to calculate interpolating value $f(0.15)$ from the data in table 4, we use third degree Newton-Gregory polynomial [6, 9]

$$p_n(x_s) = f_\circ + s\Delta f_\circ + \frac{s(s-1)}{2!}\Delta^2 f + \frac{s(s-1)(s-2)}{3!}\Delta^3 f_\circ \ldots \frac{s(s-1)\ldots(s-n+1)}{n!}\Delta^n f_\circ,$$

Where $s = \frac{x-x_\circ}{h}$, with $h = \Delta x$, the uniform spacing in x-values.

In order to centre the x-value around x=0.15, we must use the four entries beginning with x = - 0.2. That makes $x_\circ$= - 0.2 and $s = \frac{0.15+0.2}{0.2} = 1.75$. Inserting the proper values into the expression for Newton-Gregory polynomial, we get

$$f(0.15) = 0.980 + \frac{(1.75)(0.75)}{2!}(0.020) + \frac{(1.75)(0.75)(-0.25)}{3!}(-0.04)$$

$$= 0.980 + 0.0131 + 0.002 = 0.995$$

The function is actually for $f(x) = \cos(x)$, so we know that the true value of $f(0.15) = 0.989$; the absolute error is 0.006. Fuzzy logic computation results indicates that the error at x = 0.15 is 0.003, which indicates that our estimate is very good compared to the true value and even for Newton-Gregory polynomial approximation.

Example Two

The data given in Table 4 is the numerical solution for the steady-state equation-flow equation, u(x, y) are the temperatures at the nodes of a gridwork constructed in the domain of interest [9]. As in example one, we used MATLAB [14] Simulink features to describe and design the interpolation with fuzzy controller method to calculate the temperatures at points other than the nodes of the grid. The *x*-values are assumed as the first input crisp values, the y values are assumed as the second input crisp values and the temperature will be considered as the output values

Six descriptors were used for the first input x values, Five descriptors were used for the second input y-values, and 10 descriptors used for the output interpolating Temperature variable. The input and output membership functions are chosen such that the given values are satisfied at most given nodes and very close to the others. The centroid defuzzification option and non-uniform triangular memberships in MATLAB [14] Simulink features are used for the output. The *x*-input and *y*-output fuzzy descriptors are indicated in table 6, and the fuzzy rules are in table 7.

X / y	**0.0**	**0.5**	**1.0**	**1.5**	**2.0**	**2.5**
0.0	0.0	5.00	10.00	15.00	20.00	25.00
0.5	5.00	7.51	10.05	12.00	15.67	20.00
1.0	10.00	10.00	10.00	10.00	10.00	10.00
1.5	15.00	12.51	9.95	7.32	4.33	0.00
2.0	20.00	15.00	10.00	5.00	0.00	-5.00

Table 5 Temperatures at the nodes of a gridwork constructed in the domain of interest for the steady-state equation.

The first input fuzzy descriptor x	**The second input fuzzy descriptor y**	**The output (Temperature) fuzzy descriptor**
VSX: Very small x values, SX: Small x values, MSX: Medium small x values, MX: Medium x values, LX: Large x values, VLX: Very large x values.	VSY: Very small y values, SY: Small y values, MY: Medium y values, LY: Large y values, VLY: Very large y values.	VVST: Extremely small T values, VST: Very small T values, S1T: Small 1 T values, ST: Small T values, M1T: Medium1 T values, M2T: medium 2 T values. M3T: Medium 3 T values, L1T: Large 1 T values, L2T: Large 2 T values, VLT: Very large T values

Table 6 The two inputs and output fuzzy descriptors

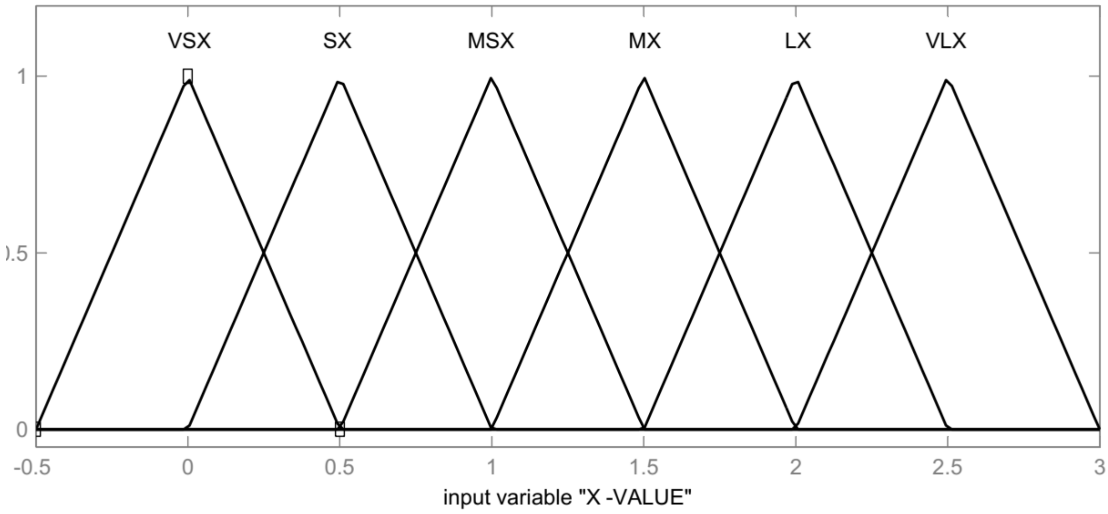

Fig. 6 The uniform triangular membership functions used to define the first input.

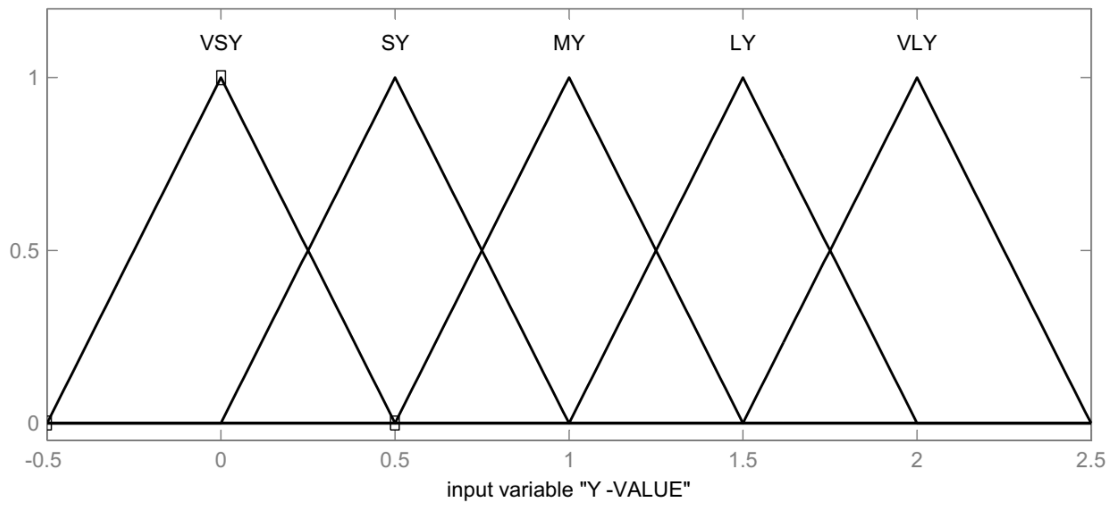

Fig. 7 The uniform triangular membership functions used to define the second input.

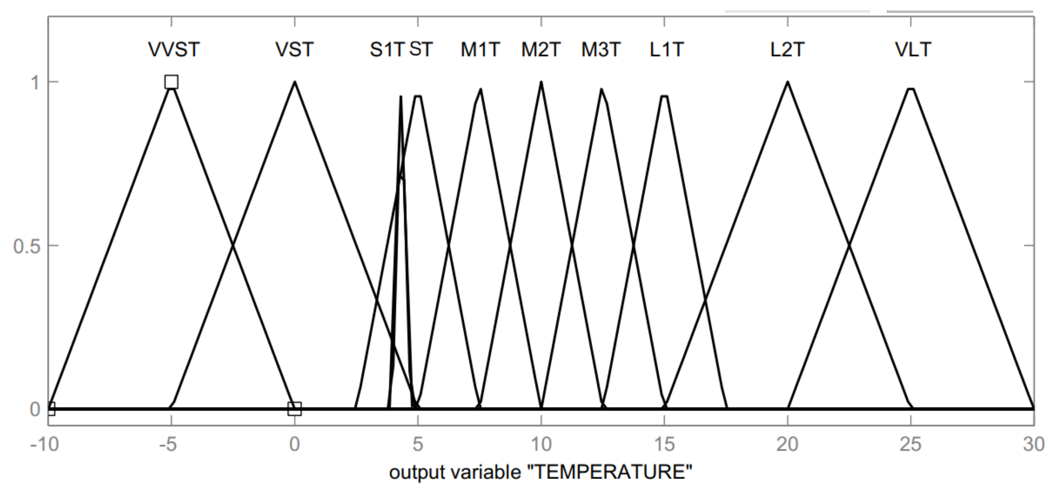

Fig. 8 The non-uniform triangular membership functions used to define the output.

x/y	VSX	SX	MSX	MX	LX	VLX
VSY	VST	ST	M2T	L1T	L2T	VLT
SY	ST	M1T	M2T	M3T	L1T	L2T
MY	M2T	M2T	M2T	M2T	M2T	M2T
LY	L1T	M3T	M2T	M1T	S1T	VST
VLY	L2T	L1T	M2T	ST	VST	VVST

Table 7 The applied fuzzy rules

x/y	0.3	0.8	1.3	1.6	1.9	2.1	2.4
0.4	4.10	8.26	12.1	15.7	16.6	19.2	23.8
0.9	8.47	9.33	10.7	11.4	11.4	14.3	14.3
1.2	12.2	11.1	8.92	8.75	8.94	5.34	3.96
1.6	15.9	11.6	7.92	4.29	2.74	-1.87	-1.17
1.8	17.1	12.2	7.19	3.83	1.45	-1.21	-2.86

Table 8 The x, y inputs and the interpolation crisp output (Temperature).

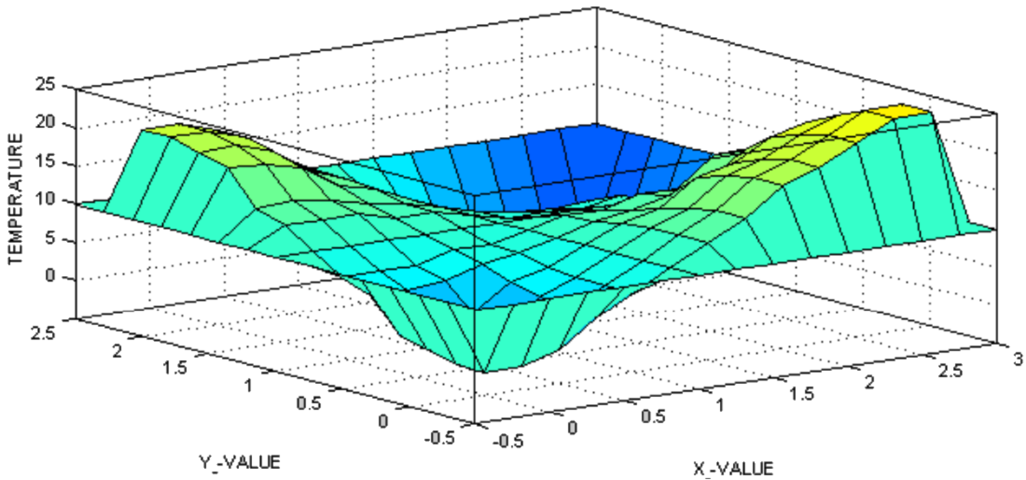

Fig 9 A surface graph for the x, y inputs and temperature output.

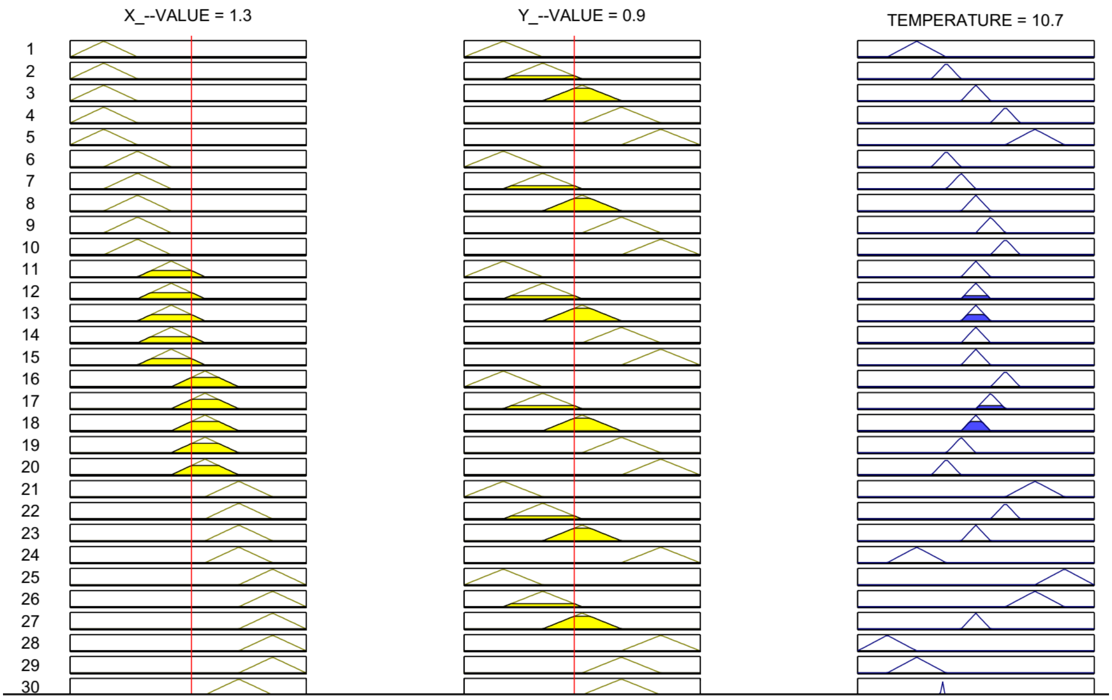

Fig 10 MATLAB rules figure for the output interpolation at x = 1.3, and y = 0.9

Fig. 9 is a three dimensional graph for *x*, *y* inputs and the approximated temperatures obtained at selected nodes in the given domain. Table 8 contains the computed result at random nodes. The membership functions and the fuzzy rules are designed such that the obtained temperature values are consistent with the neighbouring given values. Fig 10 is an example showing the MATLAB computation at the node (1.3, 0.9) according to the given membership functions and fuzzy rules.

Conclusions

Fuzzy logic is considered to be an elite approximation tool, it provides an effective means of capturing the approximate, inexact nature of the real word. It's a scientific revolution that made our linguistic wards accessible and processed easily by computers In this paper, we presented a method which used a given experimental or mathematical data, fuzzy logic, uniform and non-uniform triangular membership functions to approximate and interpolate unknown values. The proposed interpolation method is proved to be applicable and flexible rather than the other existing methods which depends mainly on numerical approximation algorithms (i.e. polynomials or splines). The presented method can introduce a new and fundamental change in dealing with mathematical fuzzy terms and vagueness phenomenon (i.e. close, very close, large, small, very small,….etc.). It provided us with a tool to avoid different actions or values for almost same input values (same behaviour). The method can be easily used and implemented for different mathematical and physical data. Results show the efficiency of fuzzy logic and fuzzy controller system (FCS) in approximating values satisfying mathematical relations. It's shown that obtained approximated values in the domain of the given values are very close to the exact values, and by using proper number and suitable memberships, the error could be even less than the error from known traditional methods. Choosing and designing suitable membership functions and rules plays a crucial role, we recommend more study and on that side.

CONFLICT OF INTERESTS

The author declare that there is no conflict of interests.

References

[1] Abu-Saman A., El-Zerii H. (2014), Numerical Solution of initial value problems using Fuzzy Transform, IJAER, vol. 9, 23, PP 13667-13675.

[2] Abu-Saman A., El-Zerii H (2016), Fuzzy Transform Algorithm for partial fractional differential equations, Journal of Mathematical and Computational Science (JMCS), vol. 6, 2, PP. 235-246.

[3] Abu-Saman A., El-Zerii H. (2015), Solving Fractional Order Differential Problems using Fuzzy Transform, Journal of informatics and mathematical Sciences JIMS, vol. 7, 3, pp. 163-171.

[4] Allahviranloo T., Ghanbari M., Hosseinzaheh A., Haghi E., and Nuraei R. (2011), A note on Fuzzy linear systems, Fuzzy Sets and Systems, vol.177, pp. 87-92.

[5] Babuška R. (2012), Fuzzy Modeling for Control, Springer Science & Business Media. ISBN 978-94-011-4868-9.

[6] Burden R. and Faires J. (2005), Numerical Analysis, Eighth Edition.

[7] Chen G., Tat Pham T. (2001), Introduction to fuzzy set, fuzzy logic, and fuzzy control system, CRC Press.

[8] Dehghan M., and Hshemi B. (2006), Iterative solution of fuzzy linear systems, Applied Mathematics and Computation, vol. 175, 1, pp. 645-674.

[9] Gerald C.F., Wheatley P. O. (2004), Applied numerical analysis, Eighth Edition.

[10] Lee C. (1990), Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I, IEEE: Transactions on systems, man, and cybernetics, vol. 20, 2, pp. 404-416

[11] Mamdani E. H. (1974), Application of fuzzy algorithms for simple dynamic plant, Proc. IEE, vol. 121, 12, pp. 1858-1588.

[12] Mamdani E. H. (1976), advances in the linguistic synthesis of fuzzy controllers, Int. J. Man Mach. Studies, vol5, 6, pp. 669-678.

[13] Mamdani E. H., and Assilian S. (1975), An experiment in linguistic synthesis with a fuzzy logic controller, International Journal of Man-Machine Studies, 7, pp.1-13.

[14] MATLAB User's Guide, The Math Works Inc., 2016a.

[15] Novák V., Perfilieva I., Močkoř J. (1999), Mathematical principles of fuzzy logic, Dordrecht: Kluwer Academic ISBN 978-0-7923-8595-0.

[16] Pelletier, Francis J. (2000), The Bulletin of Symbolic Logic. **6** (3): 342–346. doi: 10.2307/421060. JSTOR 421060. Archived (PDF) from the original on 2016-03-03.

[17] Vroman A., Dschrijver, and Kerre E. (2007), Solving systems of linear fuzzy equations by parametric functions: an improved algorithm, Fuzzy Sets and Systems, vol. 158, 14, pp. 1515-1534.

[18] Zadah L. A. (1965), Fuzzy set theory, Information and control, vol. 8, 3, pp 338-353.

[19] Zadeh L. A. (1975), The concept of a linguistic variable and its application to approximate reasoning, Information Sciences, vol. 8, 3, PP. 199-249
[20] Zadeh L. A. ; et al. (1996), Fuzzy Sets, Fuzzy Logic, Fuzzy Systems, World Scientific Press. ISBN 978-981-02-2421-9.
[21] Zimmermann Z. (1992), Fuzzy set Theory and its Applications, Kluwer Academic, Norwell, Mass, USA, 2nd edition.