A METHOD FOR APPROXIMATING SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS BY USING B-SPLINE BASED FUZZY TRANSFORMS

Purnendu Das
Department of Computer Science, Assam University Silchar, India
purnen1982@gmail.com

Abstract: Fuzzy transforms developed by Irina Perfilieva is a novel, mathematically well founded soft computing tool with many applications. In this paper, we introduce a new method for numerical solution of first order ordinary differential equation by using B-spline based fuzzy transform. By using this method we reduce the problem of solving differential equations to linear system of equations. Finally we give two examples to study the accuracy of the proposed method.

Keywords: Fuzzy Transforms, B-Spline Curve, Integral Equation, B-Spline Based F-Transforms.

Introduction: In classical Mathematics, various types of transforms are introduced (e.g. Laplace transform, Fourier transform, wavelet transform etc.) by various researchers. In 2001 Irina Perfilieva introduced fuzzy transform in her paper [3]. Latter on fuzzy transform is applied in to various fields, like numerical solution of differential equations, image processing, data mining etc. in the papers [6, 11]. Different types of approximation properties of the F-transforms are studied by I. Perfilieva in [7] and she established that F-transforms are capable to accurately approximate any continuous function. Later on B. Bede and I.J. Rudas enlarge the class of F-transforms by considering different shapes and not necessarily small support for the atoms of the fuzzy partition in [2] and they also defined a new F-transform based B-splines. The advantage of the B-spline based F-transform is that it has better smoothness property in comparison to F-transform.

In this paper we will apply B-spline based F-transform for numerical solution of differential equation. Here we develop a new technique based on B-spline F-transform to approximate the unknown function f(x), which is the solution of the differential equation. In the next section we give some basic definitions of B-spline functions and its properties.

B-Spline And B-Spline Based F-Transforms: The underlying core of the B-spline is its basis or basis functions. The original definition of the B-spline basis functions uses the idea of divided differences and is very difficult to evaluate. But in 1970 Carl De Boor established a recursive definition for the B-spline basis functions. By applying the Leibniz theorem, De Boor was able to derive the following two stage definition for B-spline basis functions.

Definition2.1: Suppose $t = (t_0, t_1, t_2, \ldots)$, be the knot vectors on the real line. Then the B-spline basis function $B_{j,d}(u)$ is defined as follows:

1. when $d = 1$

$$B_{j,1}(u) = \begin{cases} 1, & if\ \ t_j \le t_{j+1} \\ 0, & otherwise \end{cases}$$

2. when $d > 1$

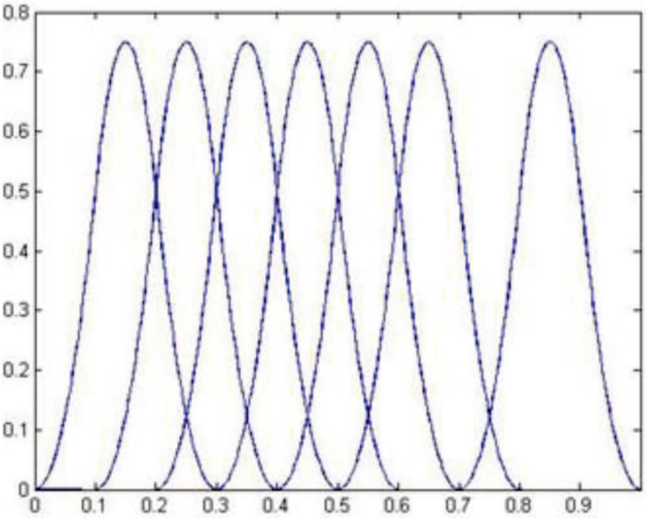
$$B_{j,d}(u) = \frac{(u - t_j)}{t_{j+d-1} - t_{j+1}} B_{j,d-1}(u) + \frac{(t_{j+d} - u)}{t_{j+d} - t_{j+1}} B_{j+1,d-1}(u)$$

where $d - 1$ is called the degree of the B-spline.

Definition2.2: Let (t_i) be a set of given knots in the real line. A quadratic B-spline basis function defined on (t_i) is of the following form:

$$B_{j,3}(u) = \begin{cases} \frac{(u-t_i)^2}{(t_{i+2}-t_i)(t_{i+1}-t_i)}, & if\ t_i \le u < t_{i+1} \\ \frac{(u-t_i)(t_{i+2}-u)}{(t_{i+2}-t_i)(t_{i+2}-t_{i+1})} + \frac{(u-t_{i+1})(t_{i+3}-u)}{(t_{i+3}-t_{i+1})(t_{i+2}-t_{i+1})}, & t_{i+1} \le u < t_{i+2} \\ \frac{(u-t_{i+3})^2}{(t_{i+3}-t_{i+1})(t_{i+3}-t_{i+2})}, & if\ t_{i+2} \le u < t_{i+3} \\ 0, & otherwise \end{cases}$$

Example2.1: Let $= (0, .1, .2, .3, .4, .5, .6, .7, .8, .9, 1)$, are the knots in the real line. Then the graph of the quadratic B-spline basis functions defined on these knots are given below:



If the knots are from t_{-2} to t_{n+3}, then for a set of n +3 control points Po , P1 , P2, ...Pn+2, one can construct a quadratic B-spline curve C(t) as follows
C(t) = $\sum_{i=-2}^{n} P_{i+2}B_{i,3}(t)$, $t_0 \le t \le t_{n+1}$

Theorem2.1 ([2]): Let $t_0 \le t_1 \dots\dots.. \le t_n$ be a sequence of ponts in a given interval [a, b]. Let $t_{-r+1} \le t_{-r+2} \dots\dots.. \le t_0$ = a and b= $t_{n+1} \le t_{n+2} \dots\dots.. \le t_{n+r}$ be some extra knots. Then the range of the B-spline basis functions is $0 \le B_{j,\ r}(x) \le 1$, $\forall\ x \in [a, b]$ and
$\sum_{i=-r+1}^{n} B_{j,\ r}(x) = 1$, $\forall\ x \in [a, b]$

Definition2.3 ([2]): Let [a, b] be an interval of real numbers and $x_1 < x_2 < \dots < x_n$ be fixed nodes within [a, b] such that x_1 = a, x_n = b and n ≥ 2 .We say that fuzzy sets A_1, A_2, ..., A_n identified with their membership functions $A_1(x)$, $A_2(x)$, ...,$A_n(x)$ and defined on [a, b] form a fuzzy partition of [a, b] if they fulfill the following conditions for i = 1,2,......n
1. $A_i(x)$ is continuous
2. $\sum_{i=1}^{n} A_i(x)$ = 1, for all x
A fuzzy partition where A_i(x) are B-spline basis functions is called B-spline fuzzy partition.

Definition2.4 ([2]): Let $f(x)$ be a continuous function defined on [a , b] and $B_{j,\ r}(x)$ for $j = -r + 1\ to\ n$ are B-spline basis functions which forms a uniform B-spline fuzzy partition of [a, b]. Then the real numbers
$F_i = \frac{\int_a^b f(x)B_{j,\ r}(x)dx}{\int_a^b B_{j,\ r}(x)dx}$, for $j = -r + 1\ to\ n$
are called continuous B-spline based fuzzy transforms of f with respect to the given B-spline basis functions.

Definition2.5 ([2]): Let *f(x)* be a continuous function on [a , b] and $B_{j,\ r}(x)$ for $j = -r+1\ to\ n$ are B-spline basis functions which forms a uniform B-spline fuzzy partition of [a, b]. Then the real numbers

$F_i = \frac{\sum_{i=1}^{m} f(x_i) B_{j,\ r}(x_i)dx}{\sum_{i=1}^{m} B_{j,\ r}(x_i)dx} \quad , \ j = -r+1\ to\ n$

(where x_i 's are given knot vectors on [a, b] and m ≥ n) are called discrete B-spline based fuzzy transforms of f with respect to the given B-spline basis functions.

Definition2.6 ([2]): Let *f(x)* be a continuous function defined on [a , b] and $B_{j,\ r}(x)$ for $j = -r+1\ to\ n$ are B-spline basis functions which forms a uniform B-spline fuzzy partition of [a, b]. Let the real numbers , $F_{-r+1}, F_{-r+2}, \dots\dots\dots.., F_0, F_1 \dots\dots\dots\dots, F_n$ are the corresponding B-spline fuzzy transform of f with respect to $B_{j,\ r}(x)$ for $j = -r+1\ to\ n$. Then the function

$f_{F,n}(x) = \sum_{j=-r+1}^{n} F_j\, B_{j,r}(x)\ ,\ \forall\, x\ \in [a,b]$

is called the inverse continuous B-spline fuzzy transform of f.

Theorem2.2 ([2]): Let *f(x)* be a continuous function defined on [a , b]. Then for any ε > 0 there exist n_ε and a B-spline fuzzy partition $B_{j,\ r}(x)$ for $j = -r+1\ to\ n$ of [a, b] such that for all $x \in$ [a ,b]

$$\left|f(x) - f_{F,n}(x)\right| \le \varepsilon$$

Proposed Method for Numerical Solution of Differential Equation: In this section, we show that how the B-spline based fuzzy transform can be used for solution of 1st order ordinary differential equation.

Consider the following differential equation:

$y'(x) = f\,(x,y)$ (1)

$y(x_1) = c$,

and here we show that this differential equation can be solved by using B-spline based fuzzy transform on a given domain $[a, b]$.

For solving the above equation we need a uniform B-spline based fuzzy partition of the domain $[a, b]$. Let $a = x_1 < x_2 < \cdots < x_n = b$ be fixed nodes within the domain. For using cubic B-spline based fuzzy partition we need four extra knots two on the right hand side of the domain $[a, b]$ and other two on the left hand side of the domain $[a, b]$. Let these extra knots are x_0, x_{-1}, x_{n+1}, x_{n+2} on the left and right side of the domain respectively. Now by using these knots $t = x_{-1}, x_0, x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}$ and taking r = 3, we can form the B-spline fuzzy partition of $[a\ ,b]$, as basis functions $B_{-1,\ 3}$(x), $B_{0,\ 3}$(x), $B_{1,\ 3}$(x), $B_{1,\ 3}$(x), $B_{2,\ 3}$(x), …, $B_{n-2,\ 3}$(x), $B_{n-1,\ 3}$(x), which are defined by using the *Definition 1.1.7*. Here we also assume that all the node points are equidistant, i.e. $x_i - x_{i-1} = h$ (say).

Now we approximate $y'(x)$ and $y''(x)$ by the following formula

$y'(x) = \frac{y(x+h) - y(x)}{h} + O(h)$ (2)

Denote $y_1(x) = y(x+h)$ as a new function and if we apply the B-spline based F-transform on both sides of equation (4), then by using these new function and by linearity of B-spline based F-transform, we obtain the relation between the B-spline fuzzy transform components of y, y_1 and y' as follows

$F_n[y'] = \frac{F_n[y_1] - F_n[y]}{h} + O(h^2)$ (3)

where, $F_n[y''] = [Y'_{-1}, Y'_0, Y'_1, Y'_2, \dots, Y'_{n-2}]$, $F_n[y_1] = [Y1_{-1}, Y1_0, Y1_1, Y1_2, \dots, Y1_{n-2}]$ and $F_n[y] = [\,Y_{-1}, Y_0, Y_1, Y_2, \dots, Y_{n-2}]$,

are the B-spline based fuzzy transform components of y'', y_1 and y respectively. Note that these vectors are two components shorter since y_1 may not be defined on $[x_{n-1}, x_n\,]$.

Now by using the definition of B-spline based fuzzy transform it can be easily proved that,

$Y1_k = Y_{k+1}$, for $k = -1, 0, 1, 2, 3, \dots, n-2.$

Indeed, the proof is simple so we omit the proof.
Therefore, we can write the components of the B-spline based F-transform of y' via components of the B-spline based F-transform of y. So, we can write the equation (6) component wise as
$Y'_k = \frac{Y_{k+1}-Y_k}{h}$, for $k = -1, 0, 1, 2, 3, \dots, n-2$.
Now, for $k = -1, 0, 1,2, 3, \dots, n-2$, we introduce the following system of linear equations.

$$\begin{aligned} Y'_{-1} &= \frac{Y_0 - Y_{-1}}{h} \\ Y'_0 &= \frac{Y_1 - Y_0}{h} \\ Y'_1 &= \frac{Y_2 - Y_1}{h} \\ &\vdots \\ Y'_{n-2} &= \frac{Y_{n-1} - Y_{n-2}}{h} \end{aligned}$$

The above system can be written in matrix form as

$$[Y'_{-1}, Y'_0, Y'_1, \dots, Y'_{n-2}]^T = D[Y_{-1}, Y_0, Y_1, \dots, Y_{n-2}, Y_{n-1}]^T \quad (4)$$

where, D is the $n \times (n+1)$ matrix, given by

$$D = \frac{1}{h}\begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & & \\ 0 & 0 & & \cdots & 0 & -1 & 1 \end{bmatrix}$$

Now, by using equation (4) and equation (1), we can write
$D\,[Y_{-1}, Y_0, Y_1, \dots, Y_{n-2}, Y_{n-1}]^T = [F_{-1}, F_0, F_1, \dots, F_{n-2}]^T$
where, $F_{-1}, F_0, F_1, \dots, F_{n-2}$ are the corresponding fuzzy transform components of $f(x, y)$.
Now we use the initial conditions and make the matrix D as $n \times n$ matrix.
The initial condition given as,
$y(x_1) = c$, $\Rightarrow Y_1 = c$
By using the above initial conditions, we make the matrix D as square matrix of order $n \times n$ and also write the system of linear equations as

$$D^c\,[Y_{-1}, Y_0, Y_1, \dots, Y_{n-2}, Y_{n-1}]^T = [\tfrac{c}{h}, F_{-1}, F_0, F_1, \dots, F_{n-2}]^T \quad (5)$$

where, $$D^c = \frac{1}{h}\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}$$

Now, we solve the system of linear equation (8) by using any numerical techniques. Note here that the solution must exist since D^c is a invertible matrix.

Note 4.1: Here for solving both the systems of linear equations we need the value of F_i, which is the B-spline based fuzzy transform components of the function $f(x, y)$, with respect to x, therefore F_i will be dependent on y also. For overcoming these difficulties we approximate F_i as

$$F'_i = \frac{\int_{x_{k-1}}^{x_{k+1}} f(x, Y_k) A_k(x)\,dx}{\int_{x_{k-1}}^{x_{k+1}} A_k(x)\,dx}$$

That means here we have assumed the value of the function y as constant for the interval (x_{k-1}, x_{k+1}). Here we omit the proof that $F_i - F_i' = O(h^2)$.
Now after evaluating all the fuzzy transform components of y, we will find an approximation of the function y by using inverse B-spline based fuzzy transform.

Numerical Examples of the Proposed Method for Solving Differential Equation: In this section we present an example of the numerical solution of 1st order ordinary differential equation by using our proposed method. We use the software "Matlab" for computing the example.

Example 4.1: Consider the following differential equation

$$\frac{dy}{dx} = 1 + xy, \quad y(0) = 1$$

We want a solution of the above differential equation on the interval [0, 1].
Now for solving the above equation we need a B-spline based fuzzy partition of the domain [0, 1]. For doing these we first take some equidistant nodes from [0, 1], and let these nodes are $t_1 = 0, t_2 = .2, t_3 = .4, t_4 = .6, t_5 = .8, t_6 = 1$. Since for cubic B-spline we need four extra knots and we take these extra knots are $t_0 = -.2, t_{-1} = -.4, t_7 = 1.2, t_8 = 1.4$. Now by using these knots $t = t_{-1}, t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8$ and taking r = 3, we can form the B-spline fuzzy partition of [0 ,1] , as basis functions $B_{-1,\ 3}$(x), $B_{0,\ 3}$(x), $B_{1,\ 3}$(x), $B_{2,\ 3}$(x), $B_{3,\ 3}$(x), $B_{4,\ 3}$(x) , $B_{5,\ 3}$(x), which are defined by using the *Definition 1.1.7*.
Since here we have taken h= .2 and n =6, thus equation (8) can be written as

$$D^c\,[Y_{-1}, Y_0, Y_1, Y_2, Y_3, Y_4, Y_5]^T = [\frac{1}{.2}, F_{-1}, F_0, F_1, F_2, F_3, F_4]^T$$

Where D^c is as given above, with h = 0.2. The above system of equation can be written in linear form as

$$Y_{-1} = 1$$
$$Y_0 = 1 + 0.2 \times F_{-1}$$
$$Y_1 = 1 + 0.2(F_{-1} + F_0)$$
$$Y_2 = 1 + 0.2(F_{-1} + F_0 + F_1)$$
$$Y_3 = 1 + 0.2(F_{-1} + F_0 + F_1 + F_2)$$
$$Y_4 = 1 + 0.2(F_{-1} + F_0 + F_1 + F_2 + F_3)$$
$$Y_5 = 1 + 0.2(F_{-1} + F_0 + F_1 + F_2 + F_3 + F_4)$$
$$F_{-1} = 0.2\ , F_0 = 1.118,\ F_1 = 1.4025, F_2 = 1.811\ ,\ F_3 = 2.889, F_4 = 3.3995$$

Now for finding the values of $Y_0, Y_1, Y_2, Y_3, Y_4, Y_5$, we need the values of $F_{-1}, F_0, F_1, F_2, F_3, F_4$, which are the B-spline based fuzzy transform components of the function $f(x, y) = 1 + xy$. By using the Definition 4.1.6 and Note4.4.1 we evaluate the values of $F_{-1}, F_0, F_1, F_2, F_3, F_4$ and the corresponding values are and by using these values we evaluate the values of $Y_{-1}, Y_0, Y_1, Y_2, Y_3, Y_4, Y_5$, and the corresponding values are $Y_{-1} = 1$, $Y_0 = 1.04$, $Y_1 = 1.3416$, $Y_2 = 1.622$, $Y_3 = 1.99$, $Y_4 = 2.56, Y_5 = 3.242$.
Now the solution of the differential equation is approximated by the following formula (inverse B-spline based fuzzy transform)

$$y(x) \approx B_{-1,\ 3}(x) + 1.04\times B_{0,\ 3}(x) + 1.3416\times B_{1,\ 3}(x) + 1.622\times B_{2,\ 3}(x) + 1.99\times B_{3,\ 3}(x) + 2.56\times B_{4,\ 3}(x) + 3.242\times B_{5,\ 3}(x)\ .$$

The graph of our approximated solution $y(x)$ and the exact solution is given below:

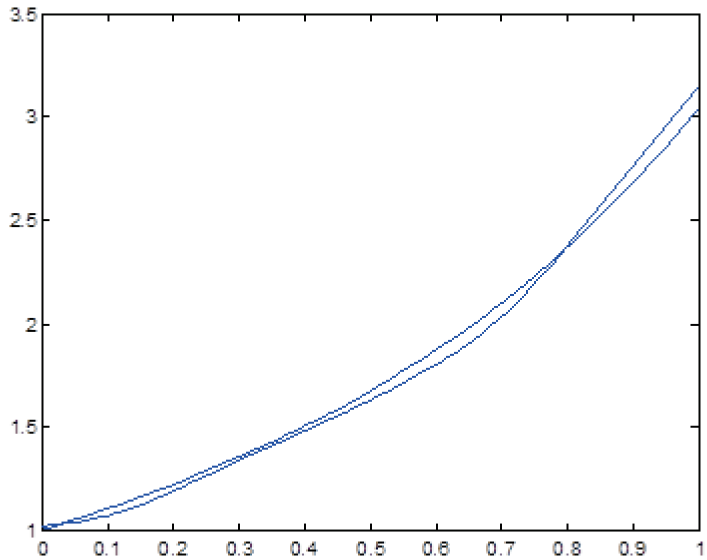


Figure 4.1: Graph of the exact solution of the differential equation and its approximated solution by using B-spline based fuzzy transform

Example 4.2: Consider the following differential equation

$$\frac{dy}{dx} = x^2 + y^2, \quad y(0) = 1$$

We want a solution of the above differential equation on the interval [0, 1].

Now for solving the above equation we need a B-spline based fuzzy partition of the domain [0, 1]. For doing these we first take some equidistant nodes from [0, 1], and let these nodes are $t_1 = 0, t_2 = .2, t_3 = .4, t_4 = .6, t_5 = .8, t_6 = 1$. Since for cubic B-spline we need four extra knots and we take these extra knots are $t_0 = -.2, t_{-1} = -.4, t_7 = 1.2, t_8 = 1.4$. Now by using these knots $t = t_{-1}, t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8$ and taking r = 3, we can form the B-spline fuzzy partition of [0, 1], as basis functions $B_{-1,\ 3}$(x), $B_{0,\ 3}$(x), $B_{1,\ 3}$(x), $B_{2,\ 3}$(x), $B_{3,\ 3}$(x), $B_{4,\ 3}$(x), $B_{5,\ 3}$(x), which are defined by using the *Definition 1.1.7.*

Since here we have taken h= .2 and n = 6, thus equation (8) can be written as

$D^c\,[Y_{-1}, Y_0, Y_1, Y_2, Y_3, Y_4, Y_5]^T = [\frac{1}{.2}, F_{-1},\ F_0, F_1, F_2, F_3, F_4]^T$

Where D^c is given as above in equation (8), with h = 0.2. The above system of equation can be written in linear form as

$$Y_{-1} = 1$$
$$Y_0 = 1 + 0.2 \times F_{-1}$$
$$Y_1 = 1 + 0.2(F_{-1} + F_0)$$
$$Y_2 = 1 + 0.2(F_{-1} + F_0 + F_1)$$
$$Y_3 = 1 + 0.2(F_{-1} + F_0 + F_1 + F_2)$$
$$Y_4 = 1 + 0.2(F_{-1} + F_0 + F_1 + F_2 + F_3)$$
$$Y_5 = 1 + 0.2(F_{-1} + F_0 + F_1 + F_2 + F_3 + F_4)$$

Now for finding the values of $Y_0, Y_1, Y_2, Y_3, Y_4, Y_5$, we need the values of $F_{-1}, F_0, F_1, F_2, F_3, F_4$, which are the B-spline based fuzzy transform components of the function $f(x, y) = x^2 + y^2$. By using the Definition 4.1.6 and Note 4.4.1 we evaluate the values of $F_{-1}, F_0, F_1, F_2, F_3, F_4$ and the corresponding values are $F_{-1} = 1.004$, $F_0 = 1.46$, $F_1 = 2.34, F_2 = 4.116$, $F_3 = 8.234, F_4 = 20.396$

And by using these values we the values of $Y_{-1}, Y_0, Y_1, Y_2, Y_3, Y_4, Y_5$, and the corresponding values are $Y_{-1} = 1, Y_0 = 1.1, Y_1 = 1.4, Y_2 = 1.9, Y_3 = 2.38, Y_4 = 3.131, Y_5 = 5.51$.
Now the solution of the differential equation is approximated by the following formula (inverse B-spline based fuzzy transform)
$y(x) \approx B_{-1,\ 3}(x) \ +1.1\times B_{0,\ 3}(x) \ +1.4\times B_{1,\ 3}(x) \ +1.9\times B_{2,\ 3}(x) \ +2.38\times B_{3,\ 3}(x) \ + \ 3.131\times B_{4,\ 3}(x) \ +5.51\times B_{5,\ 3}(x)$.
The graph of our approximated solution $y(x)$ and the exact solution is given below:

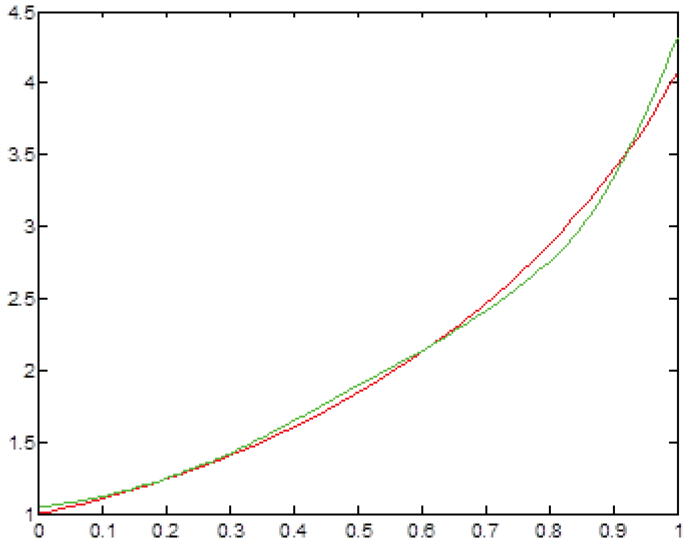


Figure 4.2: Graph of the exact solution of the differential equation and its approximated solution by using B-spline based fuzzy transform

Conclusion: In this paper he have introduced a new method for solving first order differential equation by using B-spline based fuzzy transform and particular example is also given.

References:

1. B. Kosko, "Fuzzy systems as universal approximators", *IEEE Internet. Conf. On Fuzzy System, San Diego, CA, 1992, pp.1143-116.*
2. Barnabas Bede, Imre J. Rudas, "Approximation properties of fuzzy transform", *Fuzzy Sets and Systems, 180(2011), 20-40.*
3. I. Perfilieva, E.Chaldeeva, "Fuzzy transformation", *in: Proc. of IFSA' 2001 WorldCongress, Vancouver, Canada, 2001.*
4. I. Perfilieva, "Logical approximation", *soft Comput. 7(2002) pp.73-78.*
5. I. Perfilieva, "Fuzzy transforms and their applications to data Compression", *in: Proc. Int. Conf.on Fuzz – IEEE 2005, Reno, Nevada,USA, May 22-25, 2005, pp.294-299.*
6. I. Perfilieva, "Fuzzy transforms, in: J.F.Peters, et al.(Eds)",*Transactions on Rough set II, Lecture Notes in Computer Science, vol.3135, 2004, pp.63-81.*
7. I. Perfilieva, "Fuzzy transforms: Theory and Applications", *Fuzzy Sets and Systems, 157(2006) 993-1023.*
8. I. Perfilieva, "Fuzzy Transforms and their applications to data Compression. In I. Bloch et al. (Eds.)", *Fuzzy logic and Applications, LNAI 3849, Springer, 2006, 19-31.*
9. I. Perfilieva, Fuzzy transforms: a challenge to conventional transforms, Advances in Imaging and Electron Physics 47 (2007) 137–196.
10. I. Perfilieva, M. Daˇnková, Image fusion on the basis of fuzzy transforms, in: Computational Intelligence in Decision and Control, World Scientific, New Jersey, 2008, pp. 471–476.

11. I. Perfilieva, V. Novák, A. Dvoˇrák, Fuzzy transform in the analysis of data, International Journal of Approximate Reasoning 48 (1) (2008) 36–46.
12. I. Perfilieva, V. Novák, V. Pavliska, A. Dvorák, M. Stepnicka, Analysis and prediction of time series using fuzzy transform, in: IJCNN, IEEE, 2008, pp. 3876–3880.
13. I. Perfilieva, V. Pavliska, M. Vajgl, B. De Baets, Advanced image compression on the basis of fuzzy transforms, in: Proceedings of the Conference on IPMU'2008, Torremolinos (Malaga), Spain, 2008.

***