Journal of Engineering Science and Technology
Vol. 13, No. 11 (2018) 3795 - 3807

FUZZY DELAY DIFFERENTIAL EQUATIONS WITH HYBRID SECOND AND THIRD ORDERS RUNGE-KUTTA METHOD

RUI SIH LIM, SU HOE YEAK*, ROHANIN AHMAD

Department of Mathematical Sciences, Faculty of Science Universiti Teknologi Malaysia
81310 UTM Johor Bahru, Johor, Malaysia
*Corresponding Author: s.h.yeak@utm.my

Abstract

This paper considers fuzzy delay differential equations with known state-delays. A dynamic problem is formulated by time-delay differential equations and an efficient scheme using a hybrid second and third orders Runge-Kutta method is developed and applied. Runge-Kutta is well-established methods and can be easily modified to overcome the discontinuities, which occur in delay differential equations. Our objective is to develop a scheme for solving fuzzy delay differential equations. A numerical example was run, and the solutions were validated with the exact solution. The numerical results from *C* program will show that the hybrid Runge-Kutta scheme able to calculate the fuzzy solutions successfully.

Keywords: Fuzzy delay differential equations, Hybrid runge-kutta method.

1. Introduction

The research of FDDEs has been rapidly growing in recent years. FDDEs by using hybrid second and third orders RK methods, which are to investigate the problem of finding a numerical approximation of solutions were presented in this paper. DDEs plays an important role in an increasing number of modelling problems in science and engineering; and a multitude of real-world phenomena in various fields of applications, which involves vagueness or variations in some parameters. The fuzzy theory has contributed to it, which leads to the deliberation of FDEs.

Zadeh [1] was first who introduced the concept of the fuzzy set via membership function. Recently, there have been many researchers working on the numerical solution of FDEs by use of RK methods [2-5]. All these studies involve the process of introducing RK methods as well as modifying the general methods in solving FDEs. Kanagarajan and Indrakumar [6] presented the numerical solution for FDDEs by using Euler's method under generalized differentiability concept. According to Kim and Sakthivel [7], Jayakumar and Kanagarajan [8], Pederson and Sambantham [9] and Pederson and Sambantham [10], the numerical solution of hybrid FDEs has been studied. The modifications of problem-solving tools influence completely how we handle the world around us as efficiently as possible. The emphases of this paper are to find fuzzy solutions with constant delays.

A program based on a coupling of explicit RK2 and implicit RK3 with the usage of the parametric form of an alpha-cut representation of the symmetric triangular fuzzy numbers that can generate acceptable output was developed. An example of fuzzy delay differential equations with two state delays was applied to the scheme. The results are compared with the exact solutions, which are derived in a stepwise approach by using Maple; the relative errors are calculated for the purpose of accuracy checking of our scheme. A small value of stopping criterion e is used. The iterations will stop when the error is less than the stopping criterion; the value of the stopping criterion was set as small as possible for accuracy. The results show that this scheme can successfully calculate fuzzy solutions accurately.

2. Preliminaries for Fuzzy Number

As stated by Zadeh [1], Barzinji et al. [11] and Gao et al. [12], this section gives some basic definitions and introduces the necessary notation, which will be used throughout this paper.

2.1. Definition of fuzzy number

A fuzzy set u is a fuzzy subset of $\mathbb{R}$, i.e., $u\colon \mathbb{R} \to [0,1]$, satisfying the following conditions:

- u is normal, i.e., $\exists x_0 \in \mathbb{R}$ with $u(x_0) = 1$;
- u is a convex fuzzy set, i.e., $u(\lambda x + (1-\lambda)y) \geq \min\{u(x), u(y)\}, \forall x, y \in \mathbb{R}, \forall \lambda \in [0,1]$;
- u is upper semi continuous on $\mathbb{R}$;
- $\overline{\{x \in \mathbb{R} : u(x) > 0\}}$ is compact where $\bar{A}$ denotes the closure of A.

2.2. Definition of triangular fuzzy number

The triangular fuzzy number is a fuzzy interval represented by two end points a_1 and a_3, and a peak point a_2 as $A = (a_1, a_2, a_3)$, as shown in Fig.1. The α -cut interval for a fuzzy number A as A_α, the obtained interval A_α is defined as $A_\alpha = [a_1^{(\alpha)}, a_3^{(\alpha)}]$ and we have:

$$\mu_{(A)}(x) = \begin{cases} 0, & x < a_1 \\ \dfrac{x - a_1}{a_2 - a_1}, & a_1 \le x \le a_2 \\ \dfrac{a_3 - x}{a_3 - a_2}, & a_2 \le x \le a_3 \\ 0, & x > a_3 \end{cases} \tag{1}$$

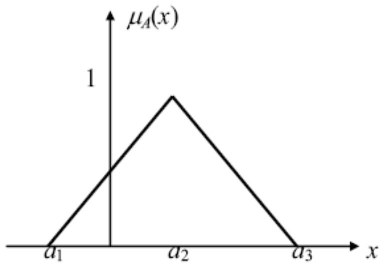

Fig. 1. Concept of fuzzy number.

The crisp interval is obtained as follows $\forall\alpha \in [0,1]$ by α-cut operation. From $\frac{a_1^{(\alpha)} - a_1}{a_2 - a_1} = \alpha$, $\frac{a_3 - a_3^{(\alpha)}}{a_3 - a_2} = \alpha$, we get $a_1^{(\alpha)} = (a_2 - a_1)\alpha + a_1$, and $a_3^{(\alpha)} = -(a_3 - a_2)\alpha + a_3$. Thus, $A_\alpha = \left[a_1^{(\alpha)}, a_3^{(\alpha)}\right]$. Then, $A_\alpha = [(a_2 - a_1)\alpha + a_1, -(a_3 - a_2)\alpha + a_3]$.

2.3. Definition of operations fuzzy number

The operations of fuzzy number can be generalized from that crisp interval. Assuming $A = [a_1, a_3]$, and $B = [b_1, b_3]$, $\forall a_1, a_3, b_1, b_3 \in \mathbb{R}$ are numbers expressed as intervals, the main operations of the interval are as follows:

- **Addition of fuzzy number:**

  To calculate the addition of fuzzy numbers A and B , we have:

  $[a_1, a_3](+)[b_1, b_3] = [a_1 + b_1,\ a_3 + b_3]$.

- **Subtraction of fuzzy number**

  To calculate subtraction of fuzzy numbers A and B , we get:

$[a_1, a_3](-)[b_1, b_3] = [a_1 - b_3,\ a_3 - b_1]$.

- **Multiplication of fuzzy number**

  To calculate multiplication of fuzzy numbers A and B , we give:

$$[a_1, a_3](\cdot)[b_1, b_3] = \begin{bmatrix} a_1 \cdot b_1 \wedge a_1 \cdot b_3 \wedge a_3 \cdot b_1 \wedge a_3 \cdot b_3, \\ a_1 \cdot b_1 \vee a_1 \cdot b_3 \vee a_3 \cdot b_1 \vee a_3 \cdot b_3 \end{bmatrix}.$$

- **Division of fuzzy number**

To calculate the division of fuzzy numbers A and B , we have:

$$[a_1, a_3](/)[b_1, b_3] = \begin{bmatrix} a_1/b_1 \wedge a_1/b_3 \wedge a_3/b_1 \wedge a_3/b_3, \\ a_1/b_1 \vee a_1/b_3 \vee a_3/b_1 \vee a_3/b_3 \end{bmatrix}$$

excluding the case $b_1 = 0$ or $b_3 = 0$.

- **Inverse interval of fuzzy number**

To calculate the inverse interval of fuzzy numbers A and B , we get:

$$[a_1, a_3]^{-1} = [1/a_1 \wedge 1/a_3, 1/a_1 \vee 1/a_3],$$

excluding the case $a_1 = 0$ or $a_3 = 0$.

All the operations will be used for all the calculation, which include the fuzzy number.

3. Numerical Methods for FDDEs

Consider the general form of DDE with time lag $t > 0$ as follows:

$$\begin{aligned} & y'(t) = f(t, y(t), y(t - \tau_1), \cdots, y(t - \tau_i), \cdots), t \in [-\tau, T], \\ & y(t) = S(t), t \in [-\tau, 0], \end{aligned} \tag{2}$$

where $y(t)$ and $y(t - \tau_i)$, $i = 1, 2, \cdots, m$ are n-dimensional fuzzy functions of t.

The function $y'(t)$ is a fuzzy derivative of $y(t)$at $t \in [-\tau, T]$; and $S(t)$ is a fuzzy number corresponding to the given initial function when $t < 0$; it may contribute to discontinuities in derivatives and affect the numerical solution. Here only constant delays are considered. Discontinuities are easily handled by using mappings and interpolation in RK calculation.

The hybrid explicit RK2 and implicit RK3 methods with a fuzzy number will be used to calculate the node points. The explicit RK2 will solve the problem with step-size, $h = \frac{H}{2}$; while implicit RK3 will solve the problem with step-size, H.

The RK2 also called Heun's method is as follows:

$$\begin{aligned} & y(t + h) \approx y(t) + \tfrac{1}{2}(k_1 + k_2), \\ & \underline{k}_1 = h\underline{f}(t, y), \\ & \overline{k}_1 = h\overline{f}(t, y), \\ & \underline{k}_2 = h\underline{f}(t + h, y + k_1), \\ & \overline{k}_2 = h\overline{f}(t + h, y + k_1). \end{aligned} \tag{3}$$

The implicit RK3 is given as below:

$$y(t + H) \approx y(t) + \tfrac{1}{6}K_1 + \tfrac{2}{3}K_2 + \tfrac{1}{6}K_3) , \tag{4}$$

$$\underline{K}_1^{j+1} = H\underline{f}\left(t, y + \frac{1}{6}K_1^j - \frac{1}{6}K_2^j\right),$$
$$\overline{K}_1^{j+1} = H\overline{f}\left(t, y + \frac{1}{6}K_1^j - \frac{1}{6}K_2^j\right),$$
$$\underline{K}_2^{j+1} = H\underline{f}\left(t + \frac{H}{2}, y + \frac{1}{6}K_1^{j+1} + \frac{1}{3}K_2^j\right),$$
$$\overline{K}_2^{j+1} = H\overline{f}\left(t + \frac{H}{2}, y + \frac{1}{6}K_1^{j+1} + \frac{1}{3}K_2^j\right),$$
$$\underline{K}_3^{j+1} = H\underline{f}\left(t + H, y + \frac{1}{6}K_1^{j+1} + \frac{5}{6}K_2^{j+1}\right),$$
$$\overline{K}_3^{j+1} = H\overline{f}\left(t + H, y + \frac{1}{6}K_1^{j+1} + \frac{5}{6}K_2^{j+1}\right),$$
$$y(t+H) - y(t) \approx error\ . \tag{5}$$

where j is a number of iteration index, y , k and K are fuzzy numbers with $y(t) = [\underline{y}(t), \overline{y}(t)]$, $k(t) = [\underline{k}(t), \overline{k}(t)]$ and $K(t) = [\underline{K}(t), \overline{K}(t)]$. The algorithm for FDDEs as below:

RK for FDDEs Algorithm

The RK algorithm can be described briefly as below:

Step 0: Set the history/past values and initial value which are given.

Step 1: Identify the time delays τ_i by using the Greatest Common Divisor (GCD). The biggest time delay will be set as the "Step".

Step 2: Solve the discretized state system in Eq. (2) by using Eqs. (3) and (4), the iterations will stop when the error is less than stopping criterion, ε = 0.0000001.

Step 3: Set $t = t + H$ and go to Step 2.

The RK methods compose of explicit and implicit recipes for computing $y(t + H)$ given $y(t)$ with ability to evaluate DDEs. For the purpose of algorithm coding, the larger delay τ_i will be set as "Step", and the smaller delay, τ_i will be set as H as "node", hence this will yield a specific accurate result. Besides that, the value of H can also be set as small as users' need, the smaller value of H will give a more accurate result. It will show that our scheme has convergence. In taking a step $t + H$ as step-size the approximation of the solution will be stored automatically; this can easily be retrieved when they are needed. For reasons of effectiveness, the user tries to use the larger "node" size, H that will yield the specified accuracy, but what if the value is larger than the smallest delay, τ_i?

In this case, if the needed value of the solution is between two points, then the Newton Forward Interpolation will take part. The result of "node" size, H which is larger than the smallest delay, τ_i will be discussed in Section 5. Our scheme successfully avoids the non-uniform time step solutions. The formula used for relative error is:

$$\textbf{relative error} = \frac{|\textbf{exact}-\textbf{approximate}|}{\textbf{exact}}. \tag{6}$$

4. Convergence of Algorithm

As reported by Ghanaie and Moghadam [14], the convergence of RK3 is as shown in this section.

The α-cut intervals of $y(t)$and $y(t-\tau_i)$ when $t \in [-\tau, T]$ are given as follows:

$$\begin{aligned} y_\alpha^p(t) &= \left[\underline{y}_\alpha^p(t), \overline{y}_\alpha^p(t)\right], \\ y_\alpha^p(t-\tau_i) &= \left[\underline{y}_\alpha^p(t-\tau_i), \overline{y}_\alpha^p(t-\tau_i)\right], \qquad p = 1, 2, \cdots, n \end{aligned} \tag{7}$$

Let exact solution is $[Y(t)]_\alpha = [\underline{Y}(t,\alpha), \overline{Y}(t,\alpha)]$ approximated by some $[y(t)]_\alpha = [\underline{y}(t,\alpha), \overline{y}(t,\alpha)]$ at t_n, $0 \le n \le N$, respectively. We define,

$$\begin{aligned} \underline{K}_1^{j+1}(t, y(t;\ \alpha)) &= H\underline{f}\left(t_n, y(t_n;\ \alpha) + \tfrac{1}{6}K_1^j - \tfrac{1}{6}K_2^j\right), \\ \overline{K}_1^{j+1}(t, y(t;\ \alpha)) &= H\overline{f}\left(t_n, y(t_n;\ \alpha) + \tfrac{1}{6}K_1^j - \tfrac{1}{6}K_2^j\right), \\ \underline{K}_2^{j+1}(t, y(t;\ \alpha)) &= H\underline{f}\left(t_n + \tfrac{H}{2}, y(t_n;\ \alpha) + \tfrac{1}{6}K_1^{j+1} + \tfrac{1}{3}K_2^j\right), \\ \overline{K}_2^{j+1}(t, y(t;\ \alpha)) &= H\overline{f}\left(t_n + \tfrac{H}{2}, y(t_n;\ \alpha) + \tfrac{1}{6}K_1^{j+1} + \tfrac{1}{3}K_2^j\right), \\ \underline{K}_3^{j+1}(t, y(t;\ \alpha)) &= H\underline{f}\left(t_n + H, y(t_n;\ \alpha) + \tfrac{1}{6}K_1^{j+1} + \tfrac{5}{6}K_2^{j+1}\right), \\ \overline{K}_3^{j+1}(t, y(t;\ \alpha)) &= H\overline{f}\left(t_n + H, y(t_n;\ \alpha) + \tfrac{1}{6}K_1^{j+1} + \tfrac{5}{6}K_2^{j+1}\right), \end{aligned} \tag{8}$$

$$\begin{aligned} \underline{F}(t, y(t;\ \alpha)) &= \tfrac{1}{6}\underline{K_1}(t, y(t;\ \alpha)) + \tfrac{2}{3}\underline{K_2}(t, y(t;\ \alpha)) + \tfrac{1}{6}\underline{K_3}(t, y(t;\ \alpha)), \\ \overline{F}(t, y(t;\ \alpha)) &= \tfrac{1}{6}\overline{K}_1(t, y(t;\ \alpha)) + \tfrac{2}{3}\overline{K}_2(t, y(t;\ \alpha)) + \tfrac{1}{6}\overline{K}_3(t, y(t;\ \alpha)). \end{aligned} \tag{9}$$

The solution calculated by grid points at $c = t_0 \le t_1 \le t_2 \le \cdots \le t_N = d$ and $H = \frac{d-c}{N} = t_{n+1} - t_n$. Therefore, we have

$$\begin{aligned} \underline{y}(t_{n+1};\ \alpha) &\approx \underline{y}(t_n;\ \alpha) + \underline{F}(t_n, y(t;\ \alpha)), \\ \overline{y}(t_{n+1};\ \alpha) &\approx \overline{y}(t_n;\ \alpha) + \overline{F}(t_n, y(t;\ \alpha)), \\ \underline{Y}(t_{n+1};\ \alpha) &\approx \underline{Y}(t_n;\ \alpha) + \underline{F}(t_n, Y(t;\ \alpha)), \\ \overline{Y}(t_{n+1};\ \alpha) &\approx \overline{Y}(t_n;\ \alpha) + \overline{F}(t_n, Y(t;\ \alpha)), \end{aligned} \tag{10}$$

Hence, we show the convergence of these approximations by using the following lemmas:

$$\lim_{h\to 0} \underline{y}(t;\ \alpha) = \underline{Y}(t;\ \alpha), \text{ and } \lim_{h\to 0} \overline{y}(t;\ \alpha) = \overline{Y}(t;\ \alpha). \tag{11}$$

4.1. Lemma A

Let the sequence of the number $\{W\}_{n=0}^N$ satisfy

$|W_{n+1}| \leq A|W_n| + B, 0 \leq n \leq N-1,$

for some given positive constants A and B . Then,

$$|W_n| \leq A^n|W_0| + B\frac{A^n - 1}{A - 1}, 0 \leq n \leq N$$

4.2. Lemma B

Let the sequence of the number $\{W_n\}_{n=0}^{N}$ and $\{V_n\}_{n=0}^{N}$ satisfy

$$|W_{n+1}| \leq |W_n| + A\, max\{|W_n|, |V_n|\} + B\ ,$$
$$|V_{n+1}| \leq |V_n| + A\, max\{|W_n|, |V_n|\} + B.$$

For some given positive constants A and B , and we get

$$|U_n| \leq \overline{A}^n|W_0| + \tilde{B}\frac{\overline{A}^n - 1}{\overline{A} - 1}, 0 \leq n \leq N,$$

Thus,

$$|U_n| \leq \tilde{A}^n |W_0| + \tilde{B}\frac{\tilde{A}^n - 1}{\tilde{A} - 1},\ 0 \leq n \leq N\ ,$$

where $\overline{A} = 1 + 2A$ and $\tilde{B} = 2B$.

Let $\underline{F}(t,u,v)$and $\overline{F}(t,u,v)$ are obtained by substituting $[y(t)]_\alpha = [u,v]$ in Eq. (9),

$$\underline{F}(t,u,v) = \tfrac{1}{6}\underline{K_1}(t,u,v) + \tfrac{2}{3}\underline{K_2}(t,u,v) + \tfrac{1}{6}\underline{K_3}(t,u,v),$$
$$\overline{F}(t,u,v) = \tfrac{1}{6}\overline{K}_1(t,u,v) + \tfrac{2}{3}\overline{K}_2(t,u,v) + \tfrac{1}{6}\overline{K}_3(t,u,v).$$

The domain where $\underline{F}$ and $\overline{F}$ are defined is, therefore,

$$G = \{(t,u,v)|0 \leq t \leq T, -\infty < v < \infty, -\infty < u < v\}.$$

4.3. Theorem C

Let $\underline{F}(t,u,v)$and $\overline{F}(t,u,v)$ belong to $C^3(G)$ and the partial derivatives of $\underline{F}$ and $\overline{F}$ be bounded over G. Then, for arbitrary fixed α, $0 \leq \alpha \leq 1$, the approximate solutions Eq.(10) converge to the exact solutions $\underline{Y}(t;\ \alpha)$ and $\overline{Y}(t;\ \alpha)$ uniformly in t.

Proof

It is sufficient to show:

$$\lim_{h\to 0}\underline{y}(t_N;\ \alpha) = \underline{Y}(t_N;\ \alpha), \text{ and } \lim_{h\to 0}\overline{y}(t_N;\ \alpha) = \overline{Y}(t_N;\ \alpha),$$

where $t_N = T$. For $n = 0, 1, \cdots, N-1$, by using Taylor's theorem we have

$$\underline{Y}(t_{n+1};\ \alpha) = \underline{Y}(t_n;\ \alpha) + \underline{F}\big(t_n, Y(t_n;\ \alpha)\big) + \tfrac{h^3}{3!}\underline{Y}'''\left(\underline{\xi}_n\right),$$
$$\overline{Y}(t_{n+1};\ \alpha) \approx \overline{Y}(t_n;\ \alpha) + \overline{F}\big(t_n,\ Y(t;\ \alpha)\big) + \tfrac{h^3}{3!}\overline{Y}'''\big(\overline{\xi}_n\big). \tag{12}$$

where $t_n < \overline{\xi}_n$, and $\underline{\xi}_n < t_{n+1}$. Consequently,

$$\underline{Y}(t_{n+1};\ \alpha) - \underline{y}(t_{n+1};\ \alpha)$$
$$=\underline{Y}(t_n;\ \alpha) - \underline{y}(t_n;\ \alpha) + \left\{\underline{F}\big(t_n, y(t_n;\ \alpha)\big) - \underline{F}\big(t_n, y(t_n;\ \alpha)\big)\right\} + \tfrac{h^3}{3!}\underline{Y}'''\left(\underline{\xi}_n\right),$$
$$\overline{Y}(t_{n+1};\ \alpha) - \overline{y}(t_{n+1};\ \alpha)$$
$$= \overline{Y}(t_n;\ \alpha) - \overline{y}(t_n;\ \alpha) + \left\{\overline{F}\big(t_n, Y(t_n;\ \alpha)\big) - \overline{F}\big(t_n, y(t_n;\ \alpha)\big)\right\} + \frac{h^3}{3!}\overline{Y}'''\big(\overline{\xi}_n\big).$$

Denote $W_n = \underline{Y}(t_n;\ \alpha) - \underline{y}(t_n;\ \alpha)$ and $V_n = \overline{Y}(t_n;\ \alpha) - \overline{y}(t_n;\ \alpha)$. Then

$$|W_{n+1}| \le |W_n| + 2Lh.max\{|W_n|, |V_n|\} + \tfrac{h^3}{3!}\underline{M},$$
$$|V_{n+1}| \le |V_n| + 2Lh.max\{|W_n|, |V_n|\} + \tfrac{h^3}{3!}\overline{M}.$$

where $\underline{M} = \max\limits_{t_0 \le t \le T} \underline{Y}'''(t;\ \alpha)$, and $\overline{M} = \max\limits_{t_0 \le t \le T} \overline{Y}'''(t;\ \alpha)$; and $L > 0$ is a bound for partial derivatives of of $\underline{F}$ and $\overline{F}$. Thus, by Lemma B

$$|W_n| \le (1 + 4Lh)^n|U_0| + \tfrac{h^3}{3}\underline{M}\tfrac{(1+4Lh)^n - 1}{4Lh},$$
$$|V_n| \le (1 + 4Lh)^n|U_0| + \tfrac{h^3}{3}\overline{M}\tfrac{(1+4Lh)^n - 1}{4Lh}.$$

where $|U_n| = |W_n| + |V_n|$.

In particular,

$$|W_N| \le (1 + 4Lh)^N|U_0| + \tfrac{h^3}{3}\underline{M}\tfrac{(1+4Lh)^{\frac{T-t_0}{h}} - 1}{4Lh},$$
$$|V_N| \le (1 + 4Lh)^N|U_0| + \tfrac{h^3}{3}\overline{M}\tfrac{(1+4Lh)^{\frac{T-t_0}{h}} - 1}{4Lh}.$$

Since $W_0 = V_0 = 0$, we obtain

$$|W_N| \le \underline{M}\tfrac{e^{4L(T-t_0)} - 1}{12L}h^2,$$
$$|V_N| \le \overline{M}\tfrac{e^{4L(T-t_0)} - 1}{12L}h^2,$$

and if we get $h \to 0$, $W_N \to 0$and $V_N \to 0$ which concludes the proof.

5. Numerical Examples of FDDEs

In this section, we use a problem from the literature [15] of DDEs and solved with our current hybrid fuzzy explicit RK 2 and implicit RK 3.

Numerical example with two time-delays

As stated by Wille and Baker [15], as Example 3 the system is as follow:

$$y_1'(t) = y_1(t-1)$$

$$y_2'(t) = y_1(t-1) + y_2(t-0.2)$$

$$y_3'(t) = y_2(t)$$

with history $y_1(0) = [0.9, 1.1], y_2(0) = [0.9, 1.1], y_3(0) = [0.9, 1.1]$ for $t \leq 0$.

In the numerical computations, the following parameters were used: the number of "Step", $Step = 0, 1, 2,$ where $Step = 0$refers to history solution; the number of nodes, *node* = 1, 2, 3, …Here, we will test the convergence of our scheme by using different step-size, *H* = 0.005, *H* = 0.05, and *H* = 0.25.

The first column indicates the time, the second, fourth, and sixth columns are outputs of minimum $y, \underline{y},$; and third, fifth and last columns are outputs of maximum of $y, \overline{y}$. Table 1 indicants the exact solution by using stepwise from Maple and it will be used to calculated the relative errors with all the numerical solutions, which are Tables 2 and 4 in Example 5.1. Table 3 and Table 5 represent the relative errors between exact solutions and numerical solutions when *H* = 0.005 and *H* = 0.05; we found that the maximum relative error was 0.000035 in Table 3; while the maximum relative error was 0.000601 in Table 5, this means that the smaller *H* give more accurate results. Thus, our scheme is convergent as smaller step-size the smaller relative error.

Table 7 represents the relative errors between exact and numerical solutions when *H* = 0.25, which is Table 6; we found that the maximum relative error was 0.092255 in Table 7. Here, we can conclude that the proposed scheme can manage different step-sizes.

Table 1. Exact solutions when *H* = 0.005.

Time	$\underline{y}_1$	$\overline{y}_1$	$\underline{y}_2$	$\overline{y}_2$	$\underline{y}_3$	$\overline{y}_3$
0.000000	0.900000	1.100000	0.900000	1.100000	0.900000	1.100000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1.985000	3.123101	3.817124	9.362151	11.442629	8.654074	10.577201
1.990000	3.132045	3.828055	9.409400	11.500378	8.701003	10.634559
1.995000	3.141011	3.839014	9.456869	11.558395	8.748168	10.692205
2.000000	3.150000	3.850000	9.504557	11.616681	8.795572	10.750143

Table 2. Numerical solutions when *H* = 0.005.

Time	$\underline{y}_1$	$\overline{y}_1$	$\underline{y}_2$	$\overline{y}_2$	$\underline{y}_3$	$\overline{y}_3$
0.000000	0.900000	1.100000	0.900000	1.100000	0.900000	1.100000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1.985000	3.123101	3.817124	9.362073	11.442533	8.653776	10.577418
1.990000	3.132045	3.828055	9.409322	11.500283	8.700703	10.634776
1.995000	3.141011	3.839014	9.456790	11.558299	8.747867	10.692424
2.000000	3.150000	3.850000	9.504478	11.616585	8.795268	10.750362

Table 3. Relative errors when *H* = 0.005.

Time	$\underline{y}_1$	$\overline{y}_1$	$\underline{y}_2$	$\overline{y}_2$	$\underline{y}_3$	$\overline{y}_3$
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2.000000	0.000000	0.000000	0.000008	0.000008	0.000035	0.000020

Table 4. Numerical solutions when *H* = 0.05.

Time	$\underline{y}_1$	$\overline{y}_1$	$\underline{y}_2$	$\overline{y}_2$	$\underline{y}_3$	$\overline{y}_3$
0.000000	0.900000	1.100000	0.900000	1.100000	0.900000	1.100000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1.850000	2.890125	3.532375	8.166014	9.980684	7.468244	9.137261
1.900000	2.974500	3.635500	8.591575	10.500813	7.886743	9.649901
1.950000	3.061125	3.741375	9.037574	11.045924	8.327213	10.188253
2.000000	3.150000	3.850000	9.504604	11.616738	8.790283	10.755479

Table 5. Relative errors when *H* = 0.05.

Time	$\underline{y}_1$	$\overline{y}_1$	$\underline{y}_2$	$\overline{y}_2$	$\underline{y}_3$	$\overline{y}_3$
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2.000000	0.000000	0.000000	0.000005	0.000005	0.000601	0.000496

Table 6. Numerical solutions when *H* = 0.25.

Time	$\underline{y}_1$	$\overline{y}_1$	$\underline{y}_2$	$\overline{y}_2$	$\underline{y}_3$	$\overline{y}_3$
0.000000	0.900000	1.100000	0.900000	1.100000	0.900000	1.100000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1.37500	2.20078	2.68984	4.52846	5.53370	4.34001	5.31542
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1.62500	2.53828	3.10234	6.29893	7.69725	5.79429	7.09699
1.75000	2.72813	3.33438	7.24942	8.85878	6.64368	8.14036
1.87500	2.93203	3.58359	8.11407	9.91541	7.60870	9.31962
2.00000	3.15000	3.85000	9.34379	11.41818	8.70486	10.66627
2.00000	0.000000	0.000000	0.016915	0.017087	0.010313	0.007802

Table 7. Relative errors when *H* = 0.25.

Time	$\underline{y}_1$	$\overline{y}_1$	$\underline{y}_2$	$\overline{y}_2$	$\underline{y}_3$	$\overline{y}_3$
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1.37500	0.000000	0.000000	0.092076	**0.092255**	0.014918	0.012883
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2.00000	0.000000	0.000000	0.016915	0.017087	0.010313	0.007802

Figure 2 indicates that it is a convex fuzzy set solution. Figure 3 shows that the relative error between exact which is Table 1 and numerical solutions when H = 0.005, 0.05 and 0.25 which are Tables 2, 4, and 6, we notice that they are overlapping when H = 0.005 and 0.05, in the solution range, but worse when H = 0.25.

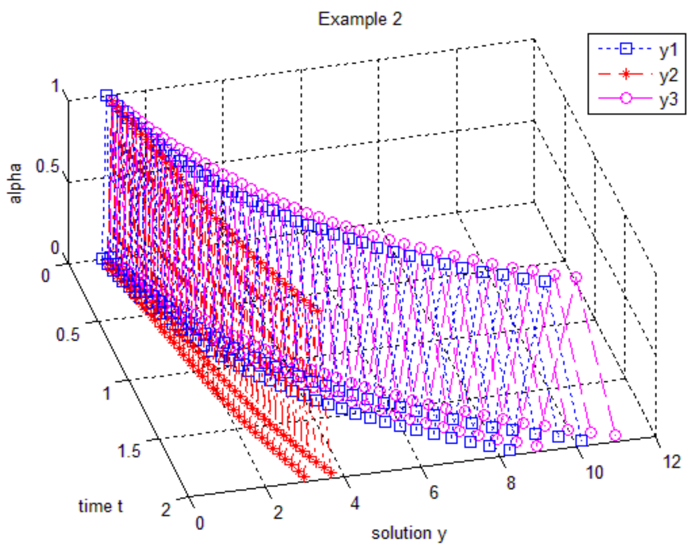

Fig. 2. Triangular fuzzy numbers.

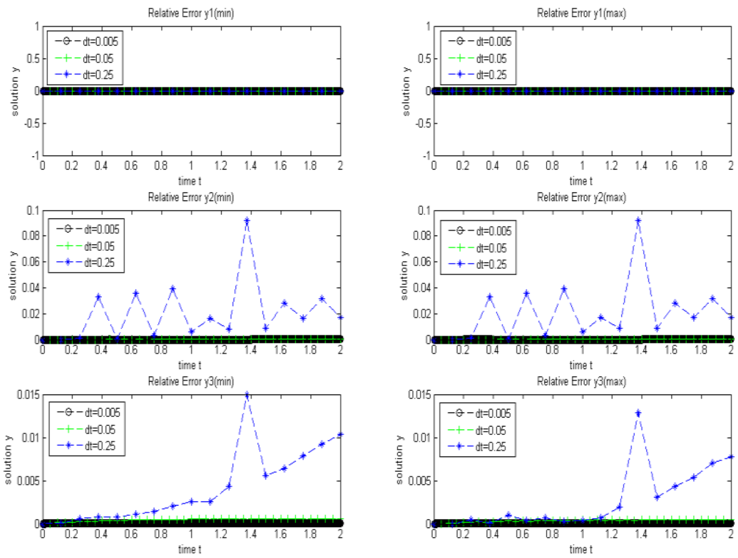

Fig. 3. Relative error.

6. Conclusions

We have successfully developed a numerical scheme composed of fuzzy technique with explicit second order and implicit third order Runge-Kutta for overcoming the discontinuities specifically at the joining point of the step size in delay differential equations with the fuzzy operation. In our numerical scheme, epsilon was set as small as possible. For illustration purposes, we presented two cases of fuzzy delay differential equation to show the computational accuracy and convergence based on the stable result on above. The numerical results indicated that our hybrid scheme performed well and efficient in solving the problems.

Acknowledgements

We would like to acknowledge Kementerian Pendidikan Malaysia and Universiti Teknologi Malaysia for their support through MyBrain15 (MyPhD).

Nomenclatures

h, H	Step size
j	Iteration index
k, K	Fuzzy number
$S(t)$	Initial function
$\underline{y}$	Lower bound/minimum of fuzzy functions
$\overline{y}$	Upper bound/maximum of fuzzy functions
$y(t)$	Fuzzy functions of time
$y(t - t_i)$	Fuzzy functions of time delay with time delays, t_i

Abbreviations

DDEs	Delay Differential Equations
FDDEs	Fuzzy Delay Differential Equations
FDEs	Fuzzy Differential Equations
GCD	Greatest Common Divisor
RK	Runge-Kutta
RK2	Second Order Runge-Kutta
RK3	Third Order Runge-Kutta

References

1. Zadeh, L.A. (1965) Fuzzy set. *Information and Control*, 8(3), 338-353.
2. Jayakumar, T.; Maheskumar, D.; and Kanagarajan, K. (2012). Numerical solution of fuzzy differential equations by Runge-Kutta method of order five. *Applied Mathematical Sciences*, 6(60), 2989-3002.
3. Kanagarajan, K.; and Suresh, R. (2014). Numerical solution of fuzzy differential equations under generalized differentiability by modified Euler method. *International Journal of Mathematical Engineering and Science*, 2(11), 5-15.
4. Rubanraj, S.; and Rajkumar, P. (2015). Numerical solution of fuzzy differential equation by sixth order Runge-Kutta method. *International Journal Fuzzy Mathematics Archive*, 7(1), 35-42.
5. Jayakumar, T.; Muthukumar, T.; and Kanagarajan, K. (2015). Numerical solution of fuzzy delay differential equations by Runge-Kutta Verner method. *Communications in Numerical Analysis*, 1, 1-15.
6. Kanagarajan, K.; and Indrakumar, S. (2015). Numerical solution of fuzzy delay differential equations under generalized differentiability by Euler's method. *Journal of Advances in Mathematics*, 10(7): 3674-3687.

7. Kim, H.; and Sakthivel, K. (2012). Numerical solution of hybrid fuzzy differential equations using improved predictor-corrector method. *Communication in Nonlinear Science and Numerical Simulat*ion, 17(10), 3788-3794.
8. Jayakumar, T.; and Kanagarajan, K. (2014). Numerical solution of hybrid fuzzy differential equations by Adams fifth order predictor-corrector method. *International Journal of Mathematical Trends and Technology*, 9(1), 70-83.
9. Pederson, S.; and Sambandham, M. (2007). Numerical solution of hybrid fuzzy systems. *Mathematical and Computer Modelling*, 45(9-10), 1133-1144.
10. Pederson, S.; and Sambandham, M. (2009). Numerical solution of hybrid fuzzy differential equation IVPs by characterization theorem. *Information Sciences - Informatics and Computer Science*, 179(3), 319-328.
11. Barzinji, K.; Maan, N.; and Aris, N. (2014). Linear fuzzy delay differential system: Analysis on stability of steady state. *Matematika*, 30(1a), 1-7.
12. Gao, S.; Zhang, Z.; and Cao, Z. (2009). Multiplication operation on fuzzy numbers. *Journal of Software*, 4(4), 331-338.
13. Wille, D.R.; and Baker, C.T.H.(1992). DELSOL-A numerical code for the solution of systems of delay-differential equations. *Applied Numerical Mathematics*, 9(3-5), 223-234.
14. Ghanaian, Z.A.; and Moghadam, M.M. (2011). Solving fuzzy differential equations by Runge-Katta method. *The Journal of Mathematics and Computer Science*, 2(2), 208-221.
15. Wille, D.R.; and Baker, C.T.H. (1992). DELSOL-A numerical code for the solution of systems of delay-differential equations. *Applied Numerical Mathematics*, 9(3-5), 223-234.