

Article

# New Approximation Methods Based on Fuzzy Transform for Solving SODEs: I

Hussein ALKasasbeh <sup>1,\*</sup>, Irina Perfilieva <sup>2</sup> , Muhammad Zaini Ahmad <sup>1</sup> and Zainor Ridzuan Yahya <sup>1</sup>

<sup>1</sup> Institute of Engineering Mathematics, Universiti Malaysia Perlis, Kampus Tetap Pauh Putra, Arau 02600, Perlis, Malaysia; mzaini@unimap.edu.my (M.Z.A.); zainoryahya@unimap.edu.my (Z.R.Y.)

<sup>2</sup> Institute for Research and Applications of Fuzzy Modelling, University of Ostrava, NSC IT4Innovations, 30. dubna 22, 701 03 Ostrava, Czech Republic; Irina.Perfilieva@osu.cz

\* Correspondence: hussein.ahmad.alka@gmail.com

Received: 17 June 2018; Accepted: 14 August 2018; Published: 23 August 2018



**Abstract:** In this paper, new approximation methods for solving systems of ordinary differential equations (SODEs) by fuzzy transform (FzT) are introduced and discussed. In particular, we propose two modified numerical schemes to solve SODEs where the technique of FzT is combined with one-stage and two-stage numerical methods. Moreover, the error analysis of the new approximation methods is discussed. Finally, numerical examples of the proposed approach are confirmed, and applications are presented.

**Keywords:** fuzzy partition; fuzzy transform; numerical methods; systems of ordinary differential equations

## 1. Introduction

Differential equations have great potential to model and understand real-world problems in science and engineering. In many cases, differential equations cannot be solved analytically, so that numerical methods are required. Therefore, numerical methods have been elaborated frequently in scientific research for solving differential equations, for example [1–4]. In this connection, fuzzy approaches successfully cope with solving differential equations. One of fuzzy approaches that has been proposed in the literature is fuzzy transform (FzT).

FzT is a general mathematical technique coined and developed by Perfilieva [5]. The study of FzT is rapidly expanding as a new branch of approximation method based on fuzzy sets. The main idea of FzT is usually forming a fuzzy partition of a universe into fuzzy subsets. Two shapes for the basic functions of fuzzy partition, triangular- and sinusoidal-shaped membership functions, were considered by [6]. Applications of the FzT can be used in the construction of approximate models, the approximation of functions and the solution of differential equations. FzT has been generalized from the case of constant components to the case of polynomial components [7]. Later, FzT was successfully used by [8] for a second order initial value problem. From this idea, FzT was proposed for numerical solutions of two point boundary value problems by the more efficient way in comparison with the similar ones obtained by the finite difference method [9].

Recently, in [10], a new numerical method based on FzT to solve a class of delay differential equations by means of the Picard-like numerical scheme was presented. The author demonstrated the stability of the method, and the obtained results have good agreement with existing methods. Furthermore, in some cases, a better approximation was achieved through sinusoidal-shaped basic functions, while the Bernstein basis polynomials allow better results in the other examples. On the other hand, a new approach to the fuzzy boundary value problem in the form of the fuzzy relation

was investigated by [11]. Another approach for the second order linear differential equation with constant coefficients and Dirichlet boundary conditions was introduced by [12], where the ability of FzT was demonstrated to deal with boundary value problems affected by noise, and the results were compared with the finite difference method. To confirm again the ability of FzT with respect to noisy and non-noisy source functions, FzT based on the shooting method was introduced by [13] for solving a nonlinear second-order ODE, and the obtained results were better than the classical method, namely the second order Runge–Kutta. Further, in [14], FzT to approximate the solution of boundary value problems by minimizing the integral squared error in the two-norm was considered. The trigonometric function based on higher degree FzT and the accuracy of the resulting approximation increase with the increase in the degree of FzT were presented by [15], while the weighted transform method was discussed by [16]. The conditions for quasi-consensus in a multi-agent system with sampled data based on FzT were proposed by [17]. In [18], the dynamical properties of a two-neuron system with respect to FzT and a single delay have been investigated. Multi-step FzT was first studied by [19] for solving ODEs. From this perspective, FzT was considered for solving a class of ODEs.

In this regard, many approximation methods have been studied for solving ODEs, for example using neural networks [20], embedded three and two pairs of nonlinear methods [21], electrical analogy [22], multi-general purpose graphical processing units [23], the differential transform method [24] and a Galerkin finite element method [25]. A numerical method based on the trapezoidal rule for the Cauchy–Smoluchowski problem was discussed by [26]. In [27,28], the authors studied ODEs with the initial value as the triangular fuzzy number. A new fuzzification of the classical Euler method and then incorporating an unconstrained optimization technique were proposed by [1]. Furthermore, most real-life problems involve systems of ODEs, for example the Lotka–Volterra prey predator model based on an autonomous model [29], a non-autonomous model [30,31] and fuzzy initial populations [4].

In this study, our aim is to extend the applicability of the FzT to general coupled Systems of ODEs (SODEs) where this method works better than its classical counterpart. The motivation of the present research stems from the fuzzy approach as follows. The first application of the FzT for solving ODEs had been proposed by [6] where a generalization of the Euler method for an ordinary Cauchy problem was developed and its potential in comparison with classical methods (Euler method) was demonstrated. The same approach has been successfully used by [32] to solve the Cauchy problem for a more accurate comparison with the classical method (the second-order Runge–Kutta method) and with the generalization of Euler method based on FzT, as proposed by [6]. Further, in [19], new fuzzy methods based on FzT for solving the Cauchy problem were presented, and the authors compared the results with existing numerical results in [6,32] and with classical methods, including one, two and three steps. All these fuzzy approximation methods performed better than the classical trapezoidal rule (one step) and the classical Adam–Moulton method (two and three steps) and outperformed the previous fuzzy methods in [6,32].

In this contribution, two new approximation methods are presented in detail to solve SODEs where the technique of FzT is combined with one-stage and two-stage numerical methods. The first approximation method improves the Euler method (one-stage), and the other approximation method improves trapezoidal rule (two-stage). The primary focus of this contribution is to demonstrate the applicability of the FzT for functions of two variables based on the uniform fuzzy partition. The error analysis is discussed in the context of the uniform fuzzy partition. Algorithms inspired by the FzT are shown for solving SODEs. Two new approximation methods are applied to the Lotka–Volterra prey-predator model. This contribution is an important modification relative to classical methods, the Euler method and the trapezoidal rule. Thus, these methods are compared with the Euler method and trapezoidal rule. Both approximation methods with the help of FzT provide better numerical solutions than the classical Euler method and the classical trapezoidal rule.

The paper is organized as follows. In Section 2, several related concepts and results associated with the FzT are reviewed. In Section 3, we construct procedures to obtain an approximate solution

for SODEs by using the FzT method. Applications are discussed in Section 4. Finally, conclusions are given in Section 5.

## 2. Basic Concepts

Throughout this section, we deal with an interval  $[a, b] \subset \mathbb{R}$  of real numbers. Let  $[a, b]$  be an interval on the real line  $\mathbb{R}$ . Fuzzy sets on  $[a, b]$  will be identified with their membership functions mapping from  $[a, b]$  into  $[0, 1]$ . We will assume an interval  $[a, b]$  as a real domain. In this section, we remind about the definitions and claims that were introduced and proven by [5].

**Definition 1.** (Fuzzy partition) Let  $x_1 < \dots < x_n$  be fixed nodes within  $[a, b]$  such that  $x_1 = a$ ,  $x_n = b$  and  $n \geq 2$ . The fuzzy sets  $A_1, \dots, A_n$  are often called basic functions. We say that fuzzy sets  $A_1, \dots, A_n \subset [a, b]$  establish a fuzzy partition of  $[a, b]$  if they fulfill the following conditions for  $k = 1, \dots, n$  (for the uniformity of notation, we set  $x_0 = a$  and  $x_{n+1} = b$ ):

1.  $A_k(x) : [a, b] \rightarrow [0, 1]$  is continuous with  $A_k(x_k) = 1$ ,  $A_k(x) > 0$  if  $x \in (x_{k-1}, x_{k+1})$  and  $A_k(x) = 0$  if  $x \notin (x_{k-1}, x_{k+1})$ ;
2.  $A_k(x)$ ,  $k = 2, \dots, n$ , strictly increases on  $[x_{k-1}, x_k]$ , and  $A_k(x)$ ,  $k = 1, \dots, n - 1$ , strictly decreases on  $[x_k, x_{k+1}]$ ;
3. For all  $x \in [a, b]$ ,  $\sum_{k=1}^n A_k(x) = 1$ . This is called the Ruspini condition.

We say that a fuzzy partition of  $[a, b]$  is  $h$ -uniform if its nodes  $x_1, \dots, x_n$ , where  $n \geq 2$  are equidistant. This means that  $x_k = a + h(k - 1)$ ,  $k = 1, \dots, n$ , where  $h = \frac{b-a}{n-1}$ ,  $n \geq 2$ , and the two additional properties are fulfilled:

- $A_k(x_k - x) = A_k(x_k + x)$ ,  $k = 2, \dots, n - 1$ , for all  $x \in [0, h]$  and;
- $A_k(x) = A_{k-1}(x - h)$  and  $A_{k+1}(x) = A_k(x - h)$ ,  $k = 2, \dots, n - 1$ , for all  $x \in [x_k, x_{k+1}]$ .

Two uniform fuzzy partitions with triangular- and sinusoidal-shaped basic functions can be found in [5,6]. Throughout this paper, we will write uniform fuzzy partition instead of  $h$ -uniform fuzzy partition.

**Definition 2.** Let  $f$  be a continuous function on  $[a, b]$  and  $A_k(x)$ ,  $k = 1, \dots, n$ , be a uniform fuzzy partition of  $[a, b]$ ,  $n \geq 2$ . A vector of real numbers  $F[f] = (F_1, F_2, \dots, F_n)$  given by (to complete this notation, we set  $x_1 = a$  and  $x_{n+1} = b$ ):

$$F_k[f] = \frac{\int_a^b f(x) A_k(x) dx}{\int_a^b A_k(x) dx} = \frac{\int_{x_{k-1}}^{x_{k+1}} f(x) A_k(x) dx}{\int_{x_{k-1}}^{x_{k+1}} A_k(x) dx}, \quad k = 1, \dots, n, \tag{1}$$

is called the direct FzT of  $f$  with respect to  $A_1, \dots, A_n$ .

**Remark 1.** The elements  $F_1[f], \dots, F_n[f]$  are called components of the FzT. If  $A_1, \dots, A_n$  forms a uniform fuzzy partition, then the expression (1) can be simplified as follows:

$$F_1[f] = \frac{2}{h} \int_{x_1}^{x_2} f(x) A_1(x) dx, \quad F_n[f] = \frac{2}{h} \int_{x_{n-1}}^{x_n} f(x) A_n(x) dx,$$

$$F_k[f] = \frac{1}{h} \int_{x_{k-1}}^{x_{k+1}} f(x) A_k(x) dx, \quad k = 2, \dots, n - 1. \tag{2}$$

**Definition 3.** Let  $F[f] = (F_1, F_2, \dots, F_n)$  be the direct FzT of  $f \in C[a, b]$  with respect to  $A_k(x)$ ,  $k = 1, \dots, n$ . Then, the inverse FzT of  $f$ ,  $\hat{f} : [a, b] \rightarrow \mathbb{R}$ , given by:

$$\hat{f}(x) = \sum_{k=1}^n F_k A_k(x). \tag{3}$$

**Lemma 1.** [6] Let  $f(x)$  be continuous on  $[a, b]$  and twice continuously differentiable in  $(a, b)$ , and let basic functions form a uniform fuzzy partition of  $[a, b]$ . Then, for each  $k = 1, \dots, n$ :

$$F_k = f(x_k) + \mathcal{O}(h^2). \tag{4}$$

**Remark 2.** An important property of the direct FzT, as well as inverse FzT is their linearity, namely, given  $f, g \in C[a, b]$  and  $\alpha, \beta \in \mathbb{R}$ , if  $h = \alpha f + \beta g$ , then  $F[h] = \alpha F[f] + \beta F[g]$  and  $\hat{h} = \alpha \hat{f} + \beta \hat{g}$ .

### 3. FzT for Solving SODEs

In this section, we present methodological remarks and numerical schemes for solving SODEs.

#### 3.1. Methodological Remarks to Applications of the FzT

Consider the Initial Value Problem (IVP) for the SODEs:

$$\begin{cases} x'(t) = f(t, x(t), y(t)), & x(t_1) = \alpha, t_1 \leq t \leq t_n, \\ y'(t) = g(t, x(t), y(t)), & y(t_1) = \beta, \end{cases} \tag{5}$$

where  $\alpha, \beta \in \mathbb{R}$ ,  $f$  and  $g$  are continuous functions on  $[t_1, t_n] \times \mathbb{R} \times \mathbb{R}$  and satisfy the Lipschitz condition. Unfortunately, the analytical solution  $(x(t), y(t))$  of Problem (5) is often difficult and sometimes impossible to obtain. Thus, approximate solutions by means of FzT are extremely important for solving (5). A numerical method for (5) is an algorithm that computes FzT components  $X_k \approx x(t_k)$  and  $Y_k \approx y(t_k)$ , for each  $k = 2, \dots, n$  (to complete this notation, we set  $\alpha = X_1 = x(t_1)$  and  $\beta = Y_1 = y(t_1)$ ).

Below, we extend the main principles of FzT detailed in Formulas (6) that are needed later.

**Definition 4.** Let  $f$  ( $g$ ) be a continuous function on  $[t_1, t_n]$  and  $A_1, \dots, A_n$  be the fuzzy partition of  $[t_1, t_n]$ . A vector of real numbers  $F_k[f] = (F_1[f], \dots, F_n[f])$  ( $G_k[g] = (G_1[g], \dots, G_n[g])$ ) given by:

$$F_k[f] = \frac{\int_{t_1}^{t_n} f(t, x(t), y(t)) A_k(t) dt}{\int_{t_1}^{t_n} A_k(t) dt} \left( G_k[g] = \frac{\int_{t_1}^{t_n} g(t, x(t), y(t)) A_k(t) dt}{\int_{t_1}^{t_n} A_k(t) dt} \right), \tag{6}$$

is called the direct FzT of  $f$  ( $g$ ) that is extended with independent variable  $t$  and two dependent variables  $x$  and  $y$ .

**Remark 3.** We need a way to approximate the direct FzT components 6. This is discussed in Corollary 1.

In the following, the necessary steps of the FzT are given.

1. Construction of the fuzzy partition:
  - (a) Specify the number  $n$  of components, and compute the step  $h = (t_n - t_1) / (n - 1)$ .
  - (b) Construct the nodes  $t_1 < \dots < t_n$ , where  $t_k = t_1 + h(k - 1)$ .
  - (c) Select the shape of basic functions. We mostly use triangular- or sinusoidal-shaped basic functions. Recall that the shape of the basic functions determines the course of  $\hat{f}$ , that is whether it is piecewise linear or nonlinear.
  - (d) Construct a uniform fuzzy partition of  $[t_1, t_n]$  by triangular- or sinusoidal-shaped basic functions [5].
2. Computation of FzT: We replace  $x'(t)$  and  $y'(t)$  by their approximations based on the Taylor expansion as new functions with respect to the fuzzy partition  $A_1, \dots, A_n$  by Step 1. In this way, similarly to [6], we transfer the original SODEs to the space of fuzzy units, solve them in the new

space and then transfer them back by the inverse FzT. Compute the approximation for  $x$  and  $y$  by the inverse FzT applied to  $[X_1, \dots, X_n]$  and  $[Y_1, \dots, Y_n]$ . In the next subsections, the schemes provide formulas for the computation of components of FzT.

### 3.2. Numerical Scheme I for SODEs

In this subsection, we present a modified scheme to solve SODEs using the FzT. Suppose that the functions  $f$  and  $g$  on  $[t_1, t_n]$  are sufficiently smooth in (5). For solving SODEs (5) on  $[t_1, t_n]$ , the interval is divided into  $n - 1$  subintervals. Let us choose some uniform fuzzy partition of interval  $[t_1, t_n]$  with parameter  $h = (t_n - t_1)/(n - 1)$ ,  $n \geq 2$ , and basic functions  $A_1, \dots, A_n$ . In view of the methodological remarks in Subsection 3.1, we describe the complete sequence of steps, which leads to the approximation solution of SODEs (5) (see [6] for technical details). Before we apply the direct FzT to both parts of the differential equation, we will use the Taylor expansion and replace the first derivatives of the left-hand sides in (5) by their approximations, i.e.,

$$\begin{cases} x(t+h) = x(t) + hx'(t) + \mathcal{O}(h^2), \\ y(t+h) = y(t) + hy'(t) + \mathcal{O}(h^2). \end{cases} \tag{7}$$

Denote  $\begin{cases} x^+(t) = x(t+h), \\ y^+(t) = y(t+h), \end{cases}$  as new functions and then apply the direct FzT components  $\begin{cases} F_n \\ G_n \end{cases}$  to both parts of Equation (7).

$$\begin{cases} F_n[x'(t)] = \frac{1}{h}(F_n[x^+] - F_n[x]) + \mathcal{O}(h^2), \\ G_n[y'(t)] = \frac{1}{h}(G_n[y^+] - G_n[y]) + \mathcal{O}(h^2), \end{cases}$$

where  $\begin{cases} F_n[x'] = [X'_1, \dots, X'_{n-1}], \\ G_n[y'] = [Y'_1, \dots, Y'_{n-1}], \end{cases} \quad \begin{cases} F_n[x] = [X_1, \dots, X_{n-1}], \\ G_n[y] = [Y_1, \dots, Y_{n-1}], \end{cases} \quad \text{and} \quad \begin{cases} F_n[x^+] = [X_1^+, \dots, X_{n-1}^+], \\ G_n[y^+] = [Y_1^+, \dots, Y_{n-1}^+]. \end{cases}$

Now, prove that:

$$\begin{cases} X_1^+ = X_2 + \mathcal{O}(h^2), \\ X_k^+ = X_{k+1}, \quad k = 2, \dots, n-2, \end{cases} \quad \text{and} \quad \begin{cases} Y_1^+ = Y_2 + \mathcal{O}(h^2), \\ Y_k^+ = Y_{k+1}, \quad k = 2, \dots, n-2. \end{cases}$$

For the values  $k = 1$  and  $k = n - 1$  by Lemma 1, we have  $\begin{cases} X_k^+ = X_{k+1} + \mathcal{O}(h^2), \\ Y_k^+ = Y_{k+1} + \mathcal{O}(h^2), \end{cases}$  , and for  $k = 2, \dots, n - 2$ , we have:

$$\begin{cases} X_k^+ = \frac{1}{h} \int_{t_{k-1}}^{t_k+1} x(t+h) \cdot A_k(t) dt = \frac{1}{h} \int_{t_k}^{t_k+2} x(s) \cdot A_{k+1}(s) ds = X_{k+1}, \\ Y_k^+ = \frac{1}{h} \int_{t_{k-1}}^{t_k+1} y(t+h) \cdot A_k(t) dt = \frac{1}{h} \int_{t_k}^{t_k+2} y(s) \cdot A_{k+1}(s) ds = Y_{k+1}. \end{cases}$$

Then, we get:

$$\begin{cases} hX'_k = (X_{k+1} - X_k) + \mathcal{O}(h^2), \\ hY'_k = (Y_{k+1} - Y_k) + \mathcal{O}(h^2). \end{cases}, \quad k = 1, \dots, n-1. \tag{8}$$

Therefore, we can introduce the  $(n - 1) \times n$  matrix:

$$D = \frac{1}{h} \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 1 \end{pmatrix}.$$

Thus, according to (5), the equality (8) can be rewritten (up to  $\mathcal{O}(h^2)$ ) as matrix equality:

$$\begin{cases} F_n[x'] = DF_n[x], \\ G_n[y'] = DG_n[y], \end{cases} \tag{9}$$

where  $\begin{cases} F_n[x'] = [X'_1, \dots, X'_{n-1}]^T, \\ G_n[y'] = [Y'_1, \dots, Y'_{n-1}]^T, \end{cases}$  and  $\begin{cases} F_n[x] = [X_1, \dots, X_{n-1}]^T, \\ G_n[y] = [Y_1, \dots, Y_{n-1}]^T. \end{cases}$

Now, let us return to the problem (5) and apply the FzT to both sides of the differential equation. Based on the linearity of FzT and Formula (9), we obtain the following system with respect to the unknown  $F_n[x]$  and  $G_n[y]$ :

$$\begin{cases} DF_n[x] = F_n[f], \\ DG_n[y] = G_n[g], \end{cases} \tag{10}$$

where  $\begin{cases} F_n[f] = [F_1, \dots, F_{n-1}]^T, \\ G_n[g] = [G_1, \dots, G_{n-1}]^T, \end{cases}$  are the FzT of  $f(t, x(t), y(t))$  ( $g(t, x(t), y(t))$ ) as a function of  $t$  w.r.t. the chosen basic functions  $A_1, \dots, A_n$ . Note that the last components  $F_n$  and  $G_n$  are not presented in  $F_n[f]$  and  $G_n[g]$ , respectively, due to the dimensionality limitation, and (10) does not include two initial conditions of (5). Thus, we complete the matrix  $D$  by adding the first row as the initial value as follows:

$$D^* = \frac{1}{h} \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & \dots & -1 & 1 \end{pmatrix},$$

so that  $D^*$  is an  $n \times n$  nonsingular matrix. Based on the initial conditions and the matrix  $D^*$ , we also expand  $F_n[f]$  and  $G_n[g]$  by adding the first element, as follows:

$$\begin{cases} F_n^*[f] = [\frac{x_1}{h}, F_1, \dots, F_{n-1}]^T, \\ G_n^*[g] = [\frac{y_1}{h}, G_1, \dots, G_{n-1}]^T. \end{cases}$$

Then, the problem (5) can be completely represented by the following expression with respect to the unknown  $F_n[x]$  and  $G_n[y]$ :

$$\begin{cases} D^*.F_n[x] = F_n^*[f], \\ D^*.G_n[y] = G_n^*[g]. \end{cases} \tag{11}$$

The solution of (11) can be obtained by the following formula:

$$\begin{cases} F_n[x] = (D^*)^{-1}.F_n^*[f], \\ G_n[y] = (D^*)^{-1}.G_n^*[g]. \end{cases} \tag{12}$$

In fact, to obtain the solution of (12), we should compute the inverse matrix of  $D^*$ . Therefore, we have:

$$(D^*)^{-1} = h \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 1 & 1 & \cdots & 1 & 1 \end{pmatrix},$$

and by (12), we get:

$$X_{k+1} = X_k + hF_k, \quad X_1 = \alpha, \quad | \quad Y_{k+1} = Y_k + hG_k, \quad Y_1 = \beta, \quad k = 1, \dots, n - 1, \quad (13)$$

where  $F_k$  ( $G_k$ ) is given by Formula (6). Formula (13) can be applied to the computation of  $X_2, \dots, X_n$  and  $Y_2, \dots, Y_n$ . However, it cannot be applied directly by using the function  $f(t, x, y)$  or  $g(t, x, y)$ , because it uses unknown functions  $x$  and  $y$ . Therefore, we will use the same trick as in [6] and replace functions by their FzT components:

$$\hat{F}_k[f] = \frac{\int_{t_1}^{t_n} f(t, X_k, Y_k) A_k(t) dt}{\int_{t_1}^{t_n} A_k(t) dt}, \quad \hat{G}_k[g] = \frac{\int_{t_1}^{t_n} g(t, X_k, Y_k) A_k(t) dt}{\int_{t_1}^{t_n} A_k(t) dt}, \quad k = 1, \dots, n - 1. \quad (14)$$

Thus, the components of FzT of  $x$  and  $y$  can be approximated from the following Scheme I:

$$X_{k+1} = X_k + h\hat{F}_k, \quad X_1 = \alpha, \quad | \quad Y_{k+1} = Y_k + h\hat{G}_k, \quad Y_1 = \beta, \quad k = 1, \dots, n - 1. \quad (15)$$

Finally, the approximate solution of (5) can be obtained using the inverse FzT as follows:

$$x_n(t) = \sum_{k=1}^n X_k A_k(t), \quad y_n(t) = \sum_{k=1}^n Y_k A_k(t), \quad (16)$$

where  $A_k(t)$ ,  $k = 1, 2, \dots, n$  are given basic functions. The proposed method is similar to the well-known Euler method and under similar assumptions. It has the same degree of accuracy. In the next theorem, we obtain an error estimate in the context of a fuzzy partition and error analysis of numerical Scheme I.

**Theorem 1.** Let  $f, g : [t_1, t_n] \rightarrow \mathbb{R}$  be twice continuously differentiable on  $[t_1, t_n]$ . Let, moreover,  $f, g : [t_1, t_n] \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  be Lipschitz continuous with respect to  $x$  and  $y$ , i.e. there exists a constant  $L \in \mathbb{R}$ , such that for all  $t \in [t_1, t_n]$  and  $x, x', y, y' \in \mathbb{R}$ ,

$$\begin{aligned} |f(t, x, y) - f(t, x', y')| &\leq L(|x - x'| + |y - y'|), \\ |g(t, x, y) - g(t, x', y')| &\leq L(|x - x'| + |y - y'|). \end{aligned} \quad (17)$$

Assume that  $\{A_k \mid k = 1, \dots, n\}$ ,  $n \geq 2$ , is a uniform fuzzy partition of  $[t_1, t_n]$ . Then, the local (global) error of Scheme I (14)–(15) is of the order  $h^2$  ( $h$ ).

**Proof.** Let us choose and fix some  $k$ , where  $2 \leq k \leq n$ , and assume that  $X_k$  ( $Y_k$ ) is the  $k$ -th FzT component of  $x$  ( $y$ ). We consider the SODEs (5) and their FzT representation by the system of equations (11). We start with the following easy consequences from the Taylor expansions:

$$\begin{aligned} x(t_{k+1}) &= x(t_k) + hf(t_k, x_k, y_k) + \mathcal{O}(h^2), \\ y(t_{k+1}) &= y(t_k) + hg(t_k, x_k, y_k) + \mathcal{O}(h^2). \end{aligned}$$

Let  $e_k = x_k - X_k$ ,  $d_k = y_k - Y_k$ ,  $x(t_k) = x_k$  and  $y(t_k) = y_k$ , then according to (14)–(15), we get:

$$e_{k+1} = e_k + h (f(t_k, x_k, y_k) - \hat{F}_k) + \mathcal{O}(h^2), \tag{18}$$

$$d_{k+1} = d_k + h (g(t_k, x_k, y_k) - \hat{G}_k) + \mathcal{O}(h^2). \tag{19}$$

By the assumption that  $f$  and  $g$  have continuous second order derivatives on  $[t_1, t_n]$  and are Lipschitz continuous with respect to  $x$  and  $y$ , therefore, using the trapezoid rule and Remark 1, we get:

$$\begin{aligned} |f(t_k, x_k, y_k) - \hat{F}_k| &= |f(t_k, x_k, y_k) - \frac{1}{h} \int_{t_{k-1}}^{t_{k+1}} f(t, X_k, Y_k) A_k(t) dt|, \\ &= |f(t_k, x_k, y_k) - \frac{1}{h} \frac{h}{2} 2 f(t_k, X_k, Y_k) + \mathcal{O}(h^2)|, \\ &= |f(t_k, x_k, y_k) - f(t_k, X_k, Y_k) + \mathcal{O}(h^2)|, \\ &\leq L(|e_k| + |d_k|) + \mathcal{O}(h^2). \end{aligned} \tag{20}$$

Similarly,

$$|g(t_k, x_k, y_k) - \hat{G}_k| \leq L(|e_k| + |d_k|) + \mathcal{O}(h^2).$$

Therefore,

$$\begin{aligned} |e_{k+1}| &\leq |e_k| + hL(|e_k| + |d_k|) + \mathcal{O}(h^2), \\ |d_{k+1}| &\leq |d_k| + hL(|e_k| + |d_k|) + \mathcal{O}(h^2). \end{aligned}$$

Denoting  $\delta_k = \max(|e_k|, |d_k|)$  and using the obvious equality  $\max(a + b, c + b) = \max(a, c) + b$ , we obtain from the above :

$$\delta_{k+1} \leq \delta_k + 2hL(|e_k| + |d_k|) + \mathcal{O}(h^2), \tag{21}$$

and finally,

$$\delta_{k+1} \leq \delta_k(1 + 2hL) + \mathcal{O}(h^2).$$

By iteration, we come to:

$$\delta_{k+1} \leq \delta_1(1 + C)^k + \mathcal{O}(h^2)(1 + (1 + C) + \dots + (1 + C)^{k-1}) = \delta_1(1 + C)^k + \mathcal{O}(h^2) \frac{(1 + C)^k - 1}{C},$$

where  $C = 2hL$ . For  $k = n - 1$ , we have:

$$\delta_n \lesssim e^{2L} \delta_1 + \mathcal{O}(h), \tag{22}$$

where we made use of the following fact:  $h = (t_n - t_1)/(n - 1)$ , and the following asymptotic equalities:  $(1 + 1/n)^n \sim e$ ,  $(1 + a)^n \sim 1 + na$ .

By (21) and (22), we conclude that Scheme I has the local order  $h^2$  and the global order  $h$ .  $\square$

**Remark 4.** Theorem 1 extends Theorem 9.1 of [6] to the SODEs.

**Corollary 1.** Let the assumptions of Theorem 1 be fulfilled. Then, for each  $k = 1, \dots, n$ :

$$F_k - \hat{F}_k = \mathcal{O}(h^2), \text{ and } G_k - \hat{G}_k = \mathcal{O}(h^2),$$

where  $F_k, G_k$  are defined by (6) and  $\hat{F}_k, \hat{G}_k$  are defined by (14).

**Proof.** By Lemma 1,  $x_k - X_k = \mathcal{O}(h^2)$  and  $y_k - Y_k = \mathcal{O}(h^2)$  and (20), we get:

$$\hat{F}_k - f(t_k, x_k, y_k) = \mathcal{O}(h^2),$$

and using the trapezoid rule and Remark 1, we obtain:

$$\begin{aligned} F_k &= \frac{1}{h} \int_{t_{k-1}}^{t_{k+1}} f(t, x_k, y_k) A_k(t) dt, \\ &= \frac{1}{h} \frac{h}{2} 2 f(t_k, x_k, y_k) + \mathcal{O}(h^2), \\ &= f(t_k, x_k, y_k) + \mathcal{O}(h^2). \end{aligned}$$

which together with the previous estimation proves that:

$$F_k - \hat{F}_k = \mathcal{O}(h^2).$$

Similarly,

$$g(t_k, x_k, y_k) - \hat{G}_k = \mathcal{O}(h^2) \text{ and } G_k - \hat{G}_k = \mathcal{O}(h^2). \quad \square$$

**Corollary 2.** The approximation method for (5) is given by Scheme I (14)–(15) with the local error  $\mathcal{O}(h^2)$ . The approximate solution to (5) can be found by taking the inverse FzT (16) where  $A_1, \dots, A_n$  are fixed basic functions.

**Theorem 2.** Let  $f, g \in C^2[t_1, t_n]$  and bounded on  $I = [t_1, t_n]$ . Let, moreover, basic functions  $\{A_k | k = 1, \dots, n\}$ ,  $n \geq 2$ , form a uniform fuzzy partition of  $I$ . Assume that  $F_k (G_k)$ ,  $k = 1, \dots, n$ , and  $F'_k (G'_k)$ ,  $k = 1, \dots, n$ , are the FzT components of  $f (g)$  and  $f' (g')$  with respect to  $\{A_k | k = 1, \dots, n\}$ , respectively. Then, for  $k = 1, \dots, n - 1$ :

$$|F_{k+1} - F_k| \leq h |F'_k| + \frac{h^2}{2} M_f, \quad |G_{k+1} - G_k| \leq h |G'_k| + \frac{h^2}{2} M_g, \tag{23}$$

where  $M_f = \max_{t \in I} |f''(t, x(t), y(t))|$  and  $M_g = \max_{t \in I} |g''(t, x(t), y(t))|$ .

**Proof.** Let us write the following result from Taylor’s theorem:

$$\begin{aligned} f(t + h, x(t), y(t)) &= f(t, x(t), y(t)) + hf'(t, x(t), y(t)) + \frac{h^2}{2} f''(\varepsilon, x(\varepsilon), y(\varepsilon)), \\ g(t + h, x(t), y(t)) &= g(t, x(t), y(t)) + hg'(t, x(t), y(t)) + \frac{h^2}{2} g''(\xi, x(\xi), y(\xi)). \end{aligned}$$

where  $t_k < \varepsilon$ ,  $\xi < t_{k+1}$ . Using the linearity of the FzT by Remark 2 and the properties of the uniform fuzzy partition by Definition 1, we get:

$$\begin{aligned} F_k [f(t + h, x, y)] &= \frac{1}{h} \int_{t_{k-1}}^{t_{k+1}} f(t + h, x, y) A_k(t) dt, \\ &= \frac{1}{h} \int_{t_k}^{t_{k+2}} f(t, x, y) A_{k+1}(t) dt, \\ &= F_{k+1} [f(t, x, y)], \quad k = 2, 3, \dots, n - 1, \end{aligned}$$

and:

$$F_{k+1} = F_k + hF'_k + \mathcal{O}(h^2), \quad G_{k+1} = G_k + hG'_k + \mathcal{O}(h^2),$$

$$F_{k+1} \leq F_k + hF'_k + \frac{h^2}{2}M_f, \quad G_{k+1} \leq G_k + hG'_k + \frac{h^2}{2}M_g,$$

which easily leads to (23).  $\square$

**Lemma 2.** Consider Scheme I (14)–(15). If the set of fuzzy sets  $\{A_k \mid k = 1, \dots, n - 1\}$ ,  $n \geq 2$ , is a uniform fuzzy partition of  $[t_1, t_n]$ , then we have for fixed  $k = 1, \dots, n - 1$ :

$$X_{k+1} - X_k = \begin{cases} 2 \int_{t_1}^{t_n} f(t, X_k, Y_k) A_1(t) dt & \text{if } k = 1, \\ \int_{t_1}^{t_n} f(t, X_k, Y_k) A_k(t) dt & \text{if } k = 2, \dots, n - 1, \end{cases} \tag{24}$$

$$Y_{k+1} - Y_k = \begin{cases} 2 \int_{t_1}^{t_n} g(t, X_k, Y_k) A_1(t) dt & \text{if } k = 1, \\ \int_{t_1}^{t_n} g(t, X_k, Y_k) A_k(t) dt & \text{if } k = 2, \dots, n - 1, \end{cases}$$

**Proof.** The proof can be obtained from Remark 1; in particular, by the properties of the uniform fuzzy partition  $\int_{t_1}^{t_n} A_k(t) dt = h/2$  for  $k = 1$  and  $\int_{t_1}^{t_n} A_k(t) dt = h$  for  $k = 2, \dots, n - 1$  and after substituting this into (14)–(15).  $\square$

**Lemma 3.** Suppose that  $f, g$  are continuous and bounded on  $I = [t_1, t_n]$ , and consider that Scheme I (14)–(15) with respect to the basic functions forms a uniform fuzzy partition of  $I$ . Then, we have for fixed  $k = 1, \dots, n - 1$ :

$$|X_{k+1} - X_k| \leq M_1 h, \quad |Y_{k+1} - Y_k| \leq M_2 h,$$

where  $M_1 = \max_{t \in [t_1, t_n]} |f(t, x, y)|$  and  $M_2 = \max_{t \in [t_1, t_n]} |g(t, x, y)|$ .

**Proof.** Let us choose a value of  $k$  in the range  $1 \leq k \leq n - 1$ . By using Lemma 2, we get:

$$|X_2 - X_1| = \left| 2 \int_{t_1}^{t_2} f(t, X_k, Y_k) A_1(t) dt \right|,$$

$$\leq 2 \int_{t_1}^{t_2} |f(t, X_k, Y_k) A_1(t)| dt \leq 2M_1 \int_{t_{k-1}}^{t_{k+1}} A_1(t) dt = M_1 h,$$

$$|X_{k+1} - X_k| = \left| \int_{t_{k-1}}^{t_{k+1}} f(t, X_k, Y_k) A_k(t) dt \right|,$$

$$\leq \int_{t_{k-1}}^{t_{k+1}} |f(t, X_k, Y_k) A_k(t)| dt \leq M_1 \int_{t_{k-1}}^{t_{k+1}} A_k(t) dt = M_1 h.$$

Similarly,  $|Y_2 - Y_1| \leq M_2 h$  and  $|Y_{k+1} - Y_k| \leq M_2 h$ .  $\square$

Throughout the assumptions of Theorem 3, we consider  $(t, x_1, x_2)$  instead  $(t, x, y)$ .

**Theorem 3.** Suppose that  $f(t, x_1, x_2), g(t, x_1, x_2) \in C^2[t_1, t_2]$ . Let  $\left| \frac{\partial f}{\partial x_i} \right| \leq L_f$  ( $\left| \frac{\partial g}{\partial x_i} \right| \leq L_g$ ),  $i = 1, 2$ , and  $|f(t, x_1, x_2)| \leq M_1$  ( $|g(t, x_1, x_2)| \leq M_2$ ). Consider Scheme I (14)–(15) for some positive integer  $k$ , and  $\{A_k \mid k = 1, \dots, n - 1\}$ ,  $n \geq 2$ , is a uniform fuzzy partition of  $[t_1, t_n]$ , then the following hold true:

- for a value of  $k$  in the range  $1 \leq k \leq n - 1$ :

$$|\hat{F}_k - \hat{F}_{k-1}| \leq L_f h U_{k,k-1}, \quad |\hat{G}_k - \hat{G}_{k-1}| \leq L_g h U_{k,k-1},$$

where  $U_{k,k-1} = |X_k - X_{k-1}| + |Y_k - Y_{k-1}|$ .

- for all  $k = 1, \dots, n - 1$ :

$$|X_{n+1} - X_n| \leq \frac{Mh}{2} e^{n(2Lh^2)}, \quad |Y_{n+1} - Y_n| \leq \frac{Mh}{2} e^{n(2Lh^2)},$$

where  $L = L_f + L_g$  and  $M = \sum_i^2 M_i$ .

**Proof.** 1. Using (6), we can get for each  $k = 2, \dots, n - 1$  and  $t \in I \cap [t_k, t_{k+1}]$ :

$$|\hat{F}_k - \hat{F}_{k-1}| = \left| \frac{\int_{t_{k-1}}^{t_{k+1}} f(t, X_k, Y_k) A_k(t) dt}{\int_{t_{k-1}}^{t_{k+1}} A_k(t) dt} - \frac{\int_{t_{k-2}}^{t_k} f(t, X_{k-1}, Y_{k-1}) A_{k-1}(t) dt}{\int_{t_{k-2}}^{t_k} A_{k-1}(t) dt} \right|.$$

Based on Remark 1 and Definition 1, the properties of the uniform fuzzy partition, we replace  $t$  by  $t - h$  and then  $A_{k-1}(t - h)$  by  $A_k(t)$ . Thus,

$$\begin{aligned} |\hat{F}_k - \hat{F}_{k-1}| &= \left| \frac{\int_{t_{k-1}}^{t_{k+1}} f(t, X_k, Y_k) A_k(t) dt}{\int_{t_{k-1}}^{t_{k+1}} A_k(t) dt} - \frac{\int_{t_{k-1}}^{t_{k+1}} f(t - h, X_{k-1}, Y_{k-1}) A_k(t) dt}{\int_{t_{k-1}}^{t_{k+1}} A_k(t) dt} \right|, \\ &\leq L_f (|X_k - X_{k-1}| + |Y_k - Y_{k-1}|) \int_{t_{k-1}}^{t_{k+1}} A_k(t) dt, \\ &= L_f h U_{k,k-1}. \end{aligned} \tag{25}$$

In a similar way,  $|\hat{G}_k - \hat{G}_{k-1}| \leq L_g h U_{k,k-1}$ .

- We first prove the estimate for  $k = 1$ . Then, we show that for all  $k = 2, \dots, n - 1$ , by using Lemma 2, for  $k = 1$ ,

$$\begin{aligned} |X_2 - X_1| &= \left| \int_{t_1}^{t_2} f(t, X_k, Y_k) A_k(t) dt \right|, \\ &\leq \int_{t_1}^{t_2} |f(t, X_k, Y_k) A_k(t)| dt \leq M_1 \int_{t_1}^{t_2} A_k(t) dt \leq \frac{Mh}{2}, \end{aligned}$$

where  $M = \sum_i^2 M_i$ . By (15), we get:

$$X_{k+1} - X_k = X_k - X_{k-1} + h (\hat{F}_k - \hat{F}_{k-1}), \quad Y_{k+1} - Y_k = Y_k - Y_{k-1} + h (\hat{G}_k - \hat{G}_{k-1}).$$

By using (25), we get:

$$\begin{aligned} |X_{k+1} - X_k| &\leq |X_k - X_{k-1}| + Lh^2 (|X_k - X_{k-1}| + |Y_k - Y_{k-1}|), \\ &\leq |X_k - X_{k-1}| + |Y_k - Y_{k-1}| + 2Lh^2 (|X_k - X_{k-1}| + |Y_k - Y_{k-1}|), \\ &\leq [1 + 2Lh^2] (|X_k - X_{k-1}| + |Y_k - Y_{k-1}|), \\ &\leq (1 + 2Lh^2)^k U_{2,1} \leq \frac{Mh}{2} (1 + 2Lh^2)^k, \end{aligned}$$

where  $U_{2,1} = |X_2 - X_1| + |Y_2 - Y_1|$ ,  $L = L_f + L_g$  and  $M = \sum_i^2 M_i$ . In particular,

$$|X_{n+1} - X_n| \leq \frac{Mh}{2} (1 + 2Lh^2)^n \leq e^{n(2Lh^2)} \frac{Mh}{2},$$

where we have used inequality  $(1 + 2h^2L)^n \leq e^{n(2h^2L)}$ ,  $n \geq 0$ . Analogously,  $|Y_{n+1} - Y_n| \leq e^{n(2Lh^2)} \frac{Mh}{2}$ . which concludes the proof.  $\square$

**Remark 5.** The following estimations are used in Theorem 4 for  $k = 1, \dots, n - 1$ . Let  $f (g)$  satisfy the Lipschitz condition in the second and third arguments; we get:

$$\begin{aligned} |f(t_k, x_k, y_k) - f(t_k, X_k, Y_k)| &\leq L_1 |x_k - X_k| + L_2 |y_k - Y_k|, \\ |g(t_k, x_k, y_k) - g(t_k, X_k, Y_k)| &\leq L_3 |x_k - X_k| + L_4 |y_k - Y_k|. \end{aligned}$$

From (20), we get:

$$\begin{aligned} f(t_k, x_k, y_k) - \hat{F}_k &\leq f(t_k, x_k, y_k) - f(t_k, X_k, Y_k) + \frac{h^2}{12} 2M_2, \\ |f(t_k, x_k, y_k) - \hat{F}_k| &\leq L_1 |x_k - X_k| + L_2 |y_k - Y_k| + \frac{h^2}{6} M_2, \\ |g(t_k, x_k, y_k) - \hat{G}_k| &\leq L_3 |x_k - X_k| + L_4 |y_k - Y_k| + \frac{h^2}{6} M_2. \end{aligned}$$

Thus,

$$|f(t_k, X_k, Y_k) - \hat{F}_k| \leq \frac{h^2}{6} M_2, \text{ and } |g(t_k, X_k, Y_k) - \hat{G}_k| \leq \frac{h^2}{6} M_2,$$

where  $M_2 = M_{2f} + M_{2g}$ ,  $M_{2f} = \max_{t \in [t_1, t_n]} |f''(t, x, y)|$  and  $M_{2g} = \max_{t \in [t_1, t_n]} |g''(t, x, y)|$ .

Now, we show that the proposed Scheme I is convergent.

**Theorem 4.** Let the assumptions of Theorem (3) be fulfilled, and further assume that  $f (g)$  satisfies a Lipschitz condition in the second and third arguments. Consider Scheme I (14)–(15) for some positive integer  $k$ , and  $\{A_k \mid k = 1, \dots, n - 1\}$ ,  $n \geq 2$ , is a uniform fuzzy partition of  $[t_1, t_n]$ . Thus, if a sequence of  $h = \{h_1, \dots, h_m\} \rightarrow 0$ ,  $m > 0$ , and with each  $h$ , we compute the  $X_{k,h}, Y_{k,h}$  component, then  $|x(t_k) - X_{k,h}|, |y(t_k) - Y_{k,h}|$  converges to zero for each  $k = 1, \dots, n - 1$ .

**Proof.** Let us drop the  $h$  subscript in the errors, writing  $|x(t_k) - X_k|$  and  $|y(t_k) - Y_k|$ . Now, when  $k = 1$ , the result is clearly true, since  $x(t_1) = X_1 = x_1, y(t_1) = Y_1 = y_1$ . By Taylor’s theorem, we have:

$$\begin{aligned} x(t_{k+1}) &= x(t_k) + hf(t_k, x_k, y_k) + \frac{h^2}{2} f'(\varepsilon_k, x(\varepsilon_k), y(\varepsilon_k)), \\ y(t_{k+1}) &= y(t_k) + hg(t_k, x_k, y_k) + \frac{h^2}{2} g'(\zeta_k, x(\zeta_k), y(\zeta_k)), \end{aligned}$$

where  $t_k < \varepsilon_k, \zeta_k < t_{k+1}$ . Denote  $e1_k = x_k - X_k, e2_k = y_k - Y_k, x_k = x(t_k)$  and  $y_k = y(t_k)$ . Then, by (15), we get:

$$\begin{aligned} e1_{k+1} &= e1_k + h (f(t_k, x_k, y_k) - f(t_k, X_k, Y_k) + f(t_k, X_k, Y_k) - \hat{F}_k) + \frac{h^2}{2} x''(\varepsilon_k), \\ e2_{k+1} &= e2_k + h (g(t_k, x_k, y_k) - g(t_k, X_k, Y_k) + g(t_k, X_k, Y_k) - \hat{G}_k) + \frac{h^2}{2} y''(\zeta_k). \end{aligned}$$

By virtue of Remark 5, we get:

$$|e1_{k+1}| \leq |e1_k| + hL_1 |e1_k| + hL_2 |e2_k| + \frac{h^2}{2} \left( M_{1f} + \frac{h}{3} M_2 \right),$$

$$|e2_{k+1}| \leq |e2_k| + hL_4 |e2_k| + hL_3 |e1_k| + \frac{h^2}{2} \left( M_{1g} + \frac{h}{3} M_2 \right),$$

where  $M_{1f} = \max_{t \in [t_1, t_n]} |x''(t)|$  and  $M_{1g} = \max_{t \in [t_1, t_n]} |y''(t)|$ . Therefore,

Case 1. If  $c = M_{1f} + M_{1g} + \frac{2h}{3} M_2$ ,  $L = \sum_{i=1}^4 L_i$ , we get:

$$|e1_{k+1}| \leq |e1_{k+1}| + |e2_{k+1}| \leq (1 + 2hL) (|e1_k| + |e2_k|) + h^2 c / 2,$$

$$\leq (1 + 2hL) U_k + \frac{h^2}{2} c \leq (1 + 2hL)^k U_1 + \left( \sum_{j=0}^{k-1} (1 + 2hL)^j \right) \frac{h^2}{2} c,$$

$$= (1 + 2hL)^k U_1 + \left( \frac{(1 + 2hL)^k - 1}{2hL} \right) \frac{h^2}{2} c \leq e^{k(2hL)} \left( U_1 + \frac{hc}{4L} \right) - \frac{hc}{4L},$$

where  $U_k = |e1_k| + |e2_k|$ . Indeed, we have used inequality  $(1 + 2hL)^k \leq e^{k(2hL)}$ ,  $k \geq 0$ , and the quantity  $\sum_{j=0}^{k-1} (1 + 2hL)^j$  is a finite geometric series; these can be calculated by:

$$2Lh \left( \sum_{j=0}^{k-1} (1 + 2hL)^j \right) = (1 + 2Lh) \left( \sum_{j=0}^{k-1} (1 + 2hL)^j \right) - \left( \sum_{j=0}^{k-1} (1 + 2hL)^j \right) = (1 + 2hL)^k - 1.$$

In particular, when  $U_1 = 0$ , this implies that:

$$|x_n - X_n| \leq \frac{hc}{4L} \left( e^{2L(t_n-t_1)} - 1 \right), |y_n - Y_n| \leq \frac{hc}{4L} \left( e^{2L(t_n-t_1)} - 1 \right). \tag{26}$$

Case 2. In view of Remark 5, let  $L = L_i, i = 1, \dots, 4$

$$|f(t_k, x_k, y_k) - \hat{F}_k| \leq \frac{h^2}{6} M_2 + 2L \max \{|x_k - X_k|, |y_k - Y_k|\},$$

$$|g(t_k, x_k, y_k) - \hat{G}_k| \leq \frac{h^2}{6} M_2 + 2L \max \{|x_k - X_k|, |y_k - Y_k|\}.$$

Thus,  $e1_{k+1} = x_k - X_k$ ,  $e2_k = y_k - Y_k$ ,

$$|e1_{k+1}| \leq |e1_k| + 2Lh \max \{|e1_k|, |e2_k|\} + \frac{h^2}{2} c_1,$$

$$|e2_{k+1}| \leq |e2_k| + 2Lh \max \{|e1_k|, |e2_k|\} + \frac{h^2}{2} c_2,$$

where  $c_1 = M_{1f} + \frac{h}{3} M_2$  and  $c_2 = M_{1g} + \frac{h}{3} M_2$ . Consequently,

$$|e1_k| \leq (1 + 4Lh)^k |U_1| + h^2 c_1 \frac{(1 + 4Lh)^k - 1}{4Lh},$$

$$|e2_k| \leq (1 + 4Lh)^k |U_1| + h^2 c_2 \frac{(1 + 4Lh)^k - 1}{4Lh},$$

where  $U_1 = |e1_1| + |e2_1|$ . In particular, when  $U_1 = 0$ , this implies that:

$$|x_n - X_n| \leq hc_1 \frac{e^{4L(t_n-t_1)} - 1}{4L}, \quad |y_n - Y_n| \leq hc_2 \frac{e^{4L(t_n-t_1)} - 1}{4L}, \tag{27}$$

and if a sequence of  $h \rightarrow 0$ , we get  $|e1_{n,h}| \rightarrow 0, |e2_{n,h}| \rightarrow 0$ , which concludes the proof.  $\square$

### 3.3. Numerical Scheme II for SODEs

In this subsection, we will construct numerical Scheme II, a more advanced method than that of Scheme I. The components of FzT of  $x$  and  $y$  can be approximated by the average of the two methods, Scheme I (14)–(15) and the backward Scheme I (or implicit Scheme I), then the FzT is given by:

$$\left. \begin{aligned} X_p &= X_k + h\hat{F}_k, & X_1 &= \alpha, & Y_p &= Y_k + h\hat{G}_k, & Y_1 &= \beta, \\ X_c &= X_k + h\hat{F}_{k+1}, & X_1 &= \alpha, & Y_c &= Y_k + h\hat{G}_{k+1}, & Y_1 &= \beta, \\ X_{k+1} &= \frac{1}{2}(X_p + X_c), & & & Y_{k+1} &= \frac{1}{2}(Y_p + Y_c), & & \end{aligned} \right\} \tag{28}$$

for  $k = 1, \dots, n - 1$ .

The problem with the previous scheme (28) is that the unknown quantities  $\hat{F}_{k+1}$  and  $\hat{G}_{k+1}$  appear on both sides (an implicit method). Therefore, one solution to this problem would be to use an explicit method such as another fuzzy approach. The following Scheme II for  $k = 1, \dots, n - 1, X_1 = \alpha, Y_1 = \beta$ :

$$\left. \begin{aligned} X_{k+1}^* &= X_k + h\hat{F}_k, & Y_{k+1}^* &= Y_k + h\hat{G}_k, \\ X_{k+1} &= X_k + \frac{h}{2}(\hat{F}_k + \hat{F}_{k+1}^*), & Y_{k+1} &= Y_k + \frac{h}{2}(\hat{G}_k + \hat{G}_{k+1}^*), \end{aligned} \right\} \tag{29}$$

where:

$$\left. \begin{aligned} \hat{F}_k[f] &= \frac{\int_{t_1}^{t_n} f(t, X_k, Y_k) A_k(t) dt}{\int_{t_1}^{t_n} A_k(x) dx}, & \hat{G}_k[g] &= \frac{\int_{t_1}^{t_n} g(t, X_k, Y_k) A_k(t) dt}{\int_{t_1}^{t_n} A_k(x) dx}, \\ \hat{F}_{k+1}^*[f] &= \frac{\int_{t_1}^{t_n} f(t, X_{k+1}^*, Y_{k+1}^*) A_{k+1}(t) dt}{\int_{t_1}^{t_n} A_{k+1}(x) dx}, & \hat{G}_{k+1}^*[g] &= \frac{\int_{t_1}^{t_n} g(t, X_{k+1}^*, Y_{k+1}^*) A_{k+1}(t) dt}{\int_{t_1}^{t_n} A_{k+1}(x) dx}. \end{aligned} \right\} \tag{30}$$

This method computes the approximate coordinates  $[X_1, \dots, X_n]$  and  $[Y_1, \dots, Y_n]$  of the direct FzT of the functions  $x(t)$  and  $y(t)$ , respectively. In the sequel, the inverse FzT (16) approximates the solution  $x(t)$  ( $y(t)$ ) of the SODEs (5).

In the next theorem, we obtain an error estimate in the context of a fuzzy partition and error analysis of approximation Scheme II.

**Theorem 5.** Suppose that  $f(t, x_1, x_2), g(t, x_1, x_2) \in C^2[t_1, t_2]$ . Let  $\left| \frac{\partial f}{\partial x_i} \right| \leq L_f$  ( $\left| \frac{\partial g}{\partial x_i} \right| \leq L_g$ ),  $i = 1, 2$ , and  $|f(t, x_1, x_2)| \leq M_1$  ( $|g(t, x_1, x_2)| \leq M_2$ ). Consider Scheme II (29)–(30) for some positive integer  $k$  and  $\{A_k \mid k = 1, \dots, n - 1\}, n \geq 2$ , to be a uniform fuzzy partition of  $[t_1, t_n]$ , then the following hold true:

1. for a value of  $k$  in the range  $1 \leq k \leq n - 1$ :

$$|\hat{F}_{k+1}^* - \hat{F}_k^*| \leq Lh (1 + 2Lh^2) U_{k,k-1}, \quad |\hat{G}_{k+1}^* - \hat{G}_k^*| \leq Lh (1 + 2Lh^2) U_{k,k-1},$$

where  $U_{k,k-1} = |X_k - X_{k-1}| + |Y_k - Y_{k-1}|$ .

2. for all  $k = 1, \dots, n - 1$

$$|X_{n+1} - X_n| \leq \frac{Mh}{2} e^{n(2Lh^2)}, \quad |Y_{n+1} - Y_n| \leq \frac{Mh}{2} e^{n(2Lh^2)},$$

where  $L = L_f + L_g$  and  $M = \sum_i^2 M_i$ .

**Proof.** The proof is similar to the proof of Theorem 3, so we just write out the procedure. The proofs of Part 1 is as follows.

$$\begin{aligned} |X_{k+1}^* - X_k^*| + |Y_{k+1}^* - Y_k^*| &\leq |X_k - X_{k-1}| + h |\hat{F}_k - \hat{F}_{k-1}| + |Y_k - Y_{k-1}| + h |\hat{G}_k - \hat{G}_{k-1}|, \\ |\hat{F}_{k+1}^* - \hat{F}_k^*| &\leq L_f h (|X_{k+1}^* - X_k^*| + |Y_{k+1}^* - Y_k^*|), \\ |\hat{F}_k - \hat{F}_{k-1}| &\leq L_f h U_{k,k-1}, \text{ and } |\hat{G}_k - \hat{G}_{k-1}| \leq L_g h U_{k,k-1}. \end{aligned}$$

The proof of Part 2, using (29), gives:

$$\begin{aligned} |X_{k+1} - X_k| &\leq |X_k - X_{k-1}| + Lh^2 (1 + Lh^2) U_{k,k-1}, \\ |X_{k+1} - X_k| &\leq |X_{k+1} - X_k| + |Y_{k+1} - Y_k| \leq (1 + 2Lh^2 (1 + Lh^2))^k U_{2,1}, \\ &\leq \frac{Mh}{2} (1 + 2Lh^2 (1 + Lh^2))^k. \end{aligned}$$

where  $U_{k,k-1} = |X_k - X_{k-1}| + |Y_k - Y_{k-1}|$  and  $M = M_1 + M_2$ . In particular:

$$|X_{n+1} - X_n| \leq \exp(n(2Lh^2)(1 + Lh^2)) \frac{Mh}{2}, \quad |Y_{n+1} - Y_n| \leq \exp(n(2Lh^2)(1 + Lh^2)) \frac{Mh}{2},$$

which concludes the proof.  $\square$

**Lemma 4.** Let  $f$  and  $g$  have continuous second order derivatives on  $t \in [t_1, t_n]$ , and  $f$  ( $g$ ) satisfies a Lipschitz condition in the second and third arguments. Then, a local error of Scheme II (29)–(30) is of the order  $h^3$ .

**Proof.** We consider the SODEs (5). We start with the Taylor expansion and the forward divided difference approximation of the second derivative (please see Appendix A for more details), i.e.,

$$\begin{aligned} x(t_{k+1}) &= x(t_k) + hx'(t_k) + \frac{h^2}{2} \left( \frac{x'(t_{k+1}) - x'(t_k)}{h} - \frac{h}{2} x'''(\varepsilon_{2k}) \right) + \frac{h^3}{6} x'''(\varepsilon_{1k}), \\ x(t_{k+1}) &= x(t_k) + \frac{h}{2} x'(t_k) + \frac{h}{2} x'(t_{k+1}) + h^3 \left[ \frac{1}{6} x'''(\varepsilon_{1k}) - \frac{1}{4} x'''(\varepsilon_{2k}) \right], \end{aligned}$$

where  $\varepsilon_{ik} \in (t_k, t_{k+1})$ ,  $i = 1, 2$ . The first derivative can be replaced by the right-hand side of the differential Equation (5). The Taylor expansion becomes:

$$x(t_{k+1}) = x(t_k) + \frac{h}{2} (f(t_k, x_k, y_k) + f(t_{k+1}, x_k + hf(t_k, x_k, y_k), y_k + hg(t_k, x_k, y_k))) + e_f h^3,$$

where  $e_f = \left[ \frac{1}{6} x'''(\varepsilon_{1k}) - \frac{1}{4} x'''(\varepsilon_{2k}) + \frac{1}{4} f_2 \right]$  and  $f_2 = \frac{\partial}{\partial x} f(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k})) + \frac{\partial}{\partial y} f(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k}))$ . We can write this as:

$$\begin{aligned} x_{k+1} &= x_k + \frac{h}{2} (K_0 + K_f) + e_f h^3, \\ y_{k+1} &= y_k + \frac{h}{2} (K_1 + K_g) + e_g h^3, \end{aligned}$$

where:

$$K_0 = f(t_k, x_k, y_k), K_1 = g(t_k, x_k, y_k), K_f = f(t_{k+1}, x_k + hK_0, y_k + hK_1), K_g = g(t_{k+1}, x_k + hK_0, y_k + hK_1),$$

$$e_g = \left[ \frac{1}{6} y'''(\varepsilon_{1k}) - \frac{1}{4} y'''(\varepsilon_{2k}) + \frac{1}{4} g_2 y''(\varepsilon_{3k}) \right] \text{ and } g_2 = \frac{\partial}{\partial x} g(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k})) + \frac{\partial}{\partial y} g(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k})).$$

Now, let  $f (g)$  satisfy a Lipschitz condition in the second and third arguments. By Lemma 1 and Remark 5, we have:

$$|f(t_k, x_k, y_k) - f(t, X_k, Y_k)| \leq L(|x_k - X_k| + |y_k - Y_k|) \leq \alpha_f h^2 \leq \alpha h^2, |g(t_k, x_k, y_k) - g(t, X_k, Y_k)| \leq \alpha_g \leq \alpha h^2,$$

where  $\alpha = \alpha_f + \alpha_g$  is a positive constant. Once again, using Remark 5 and according to (29)–(30), we obtain for fixed  $k = 1, \dots, n - 1$ :

$$\begin{aligned} x_{k+1} - X_{k+1} &= x_k - X_k + \frac{h}{2} (K_0 - \hat{F}_k) + \frac{h}{2} (K_f - \hat{F}_{k+1}^*) + e_f h^3, \\ K_0 - \hat{F}_k &\leq |f(t_k, x_k, y_k) - f(t_k, X_k, Y_k)| + |f(t_k, X_k, Y_k) - \hat{F}_k| \leq \alpha h^2 + \frac{1}{6} M_2 h^2, \\ K_1 - \hat{G}_k &= |g(t_k, x_k, y_k) - g(t_k, X_k, Y_k)| + |g(t_k, X_k, Y_k) - \hat{F}_k| \leq \alpha h^2 + \frac{1}{6} M_2 h^2. \end{aligned}$$

By the trapezium formula, we have:

$$f(t_{k+1}, X_{k+1}^*, Y_{k+1}^*) - \hat{F}_{k+1}^* \leq f(t_{k+1}, X_{k+1}^*, Y_{k+1}^*) - f(t_{k+1}, X_{k+1}^*, Y_{k+1}^*) + \frac{1}{6} M_{2f} h^2 = \frac{1}{6} M_{2f} h^2 \leq \frac{1}{6} M_2 h^2,$$

Note that:

$$f(t_{k+1}, x_k + hK_0, y_k + hK_1) - f(t_{k+1}, X_{k+1}^*, Y_{k+1}^*) \leq L[|x_k - X_k| + h|K_0 - \hat{F}_k| + |y_k - Y_k| + h|K_1 - \hat{G}_k|].$$

Next,

$$\begin{aligned} K_f - \hat{F}_{k+1}^* &= f(t_{k+1}, x_k + hK_0, y_k + hK_1) - f(t_{k+1}, X_{k+1}^*, Y_{k+1}^*) + f(t_{k+1}, X_{k+1}^*, Y_{k+1}^*) - \hat{F}_{k+1}^*, \\ &\leq \left[ \alpha h^2 + \frac{1}{6} M_2 h^2 \right] (2Lh + 1). \end{aligned}$$

This leads to:

$$|x_{k+1} - X_{k+1}| \leq |x_k - X_k| + \frac{h}{2} \left( \alpha h^2 + \frac{1}{6} M_2 h^2 \right) + \frac{h}{2} \left( \left[ \alpha h^2 + \frac{1}{6} M_2 h^2 \right] (2Lh + 1) \right) + e_f h^3 = |x_k - X_k| + E_f h^3. \tag{31}$$

Similarly,  $|y_{k+1} - Y_{k+1}| \leq |y_k - Y_k| + E_g h^3$ , where  $E_f$  and  $E_g$  are appropriate constants that depend on  $f$  and  $g$ , respectively. Therefore, the error of this method is  $O(h^3)$ . □

To see that Scheme II is globally a second-order method, we need to establish its convergence.

**Theorem 6.** *Let the assumptions of Lemma 4 be fulfilled. Consider Scheme II (29)–(30) for some positive integer  $k$ , and  $\{A_k \mid k = 1, \dots, n - 1\}$ ,  $n \geq 2$ , is a uniform fuzzy partition of  $[t_1, t_n]$ . Thus, if a sequence of  $h \rightarrow 0$ , and with each  $h$ , we compute the  $X_{k,h}, Y_{k,h}$  component, then  $|x(t_k) - X_{k,h}|, |y(t_k) - Y_{k,h}|$  converges to zero for each  $k = 1, \dots, n - 1$ .*

**Proof.** The proof is similar to the proof of Theorem 4, so we just write out the procedure. According to Remark 5 and (31), we get:

$$\begin{aligned} x_{k+1} - X_{k+1} &= x_k - X_k + \frac{h}{2} (K_0 - \hat{F}_k) + \frac{h}{2} (K_f - \hat{F}_{k+1}^*) + e_f h^3, \\ K_0 - \hat{F}_k &\leq L_1 |x_k - X_k| + L_2 |y_k - Y_k| + \frac{1}{6} M_2 h^2, \\ K_1 - \hat{G}_k &\leq L_3 |x_k - X_k| + L_4 |y_k - Y_k| + \frac{1}{6} M_2 h^2, \\ f(t_{k+1}, X_{k+1}^*, Y_{k+1}^*) - \hat{F}_{k+1}^* &= \frac{1}{6} M_{2f} h^2 \leq \frac{1}{6} M_2 h^2, \end{aligned}$$

$$f(t_{k+1}, x_k + hK_0, y_k + hK_1) - f(t_{k+1}, X_{k+1}^*, Y_{k+1}^*) \leq L_1 (|x_k - X_k| + h |K_0 - \hat{F}_k|) + L_2 (|y_k - Y_k| + h |K_1 - \hat{G}_k|),$$

$$K_f - \hat{F}_{k+1}^* = f(t_{k+1}, x_k + hK_0, y_k + hK_1) - f(t_{k+1}, X_{k+1}^*, Y_{k+1}^*) + f(t_{k+1}, X_{k+1}^*, Y_{k+1}^*) - \hat{F}_{k+1}^*, \\ \leq L_1 |x_k - X_k| + hL_1 |K_0 - \hat{F}_k| + L_2 |y_k - Y_k| + hL_2 |K_1 - \hat{G}_k| + \frac{1}{6}M_2h^2.$$

Once again, using Remark 5 gives:

$$|x_{k+1} - X_{k+1}| \leq |x_k - X_k| + \frac{h}{2} \left( L_1 |x_k - X_k| + L_2 |y_k - Y_k| + \frac{1}{6}M_2h^2 \right) + \\ \frac{h}{2} \left( L_1 |x_k - X_k| + hL_1 |K_0 - \hat{F}_k| + L_2 |y_k - Y_k| + hL_2 |K_1 - \hat{G}_k| + \frac{1}{6}M_2h^2 \right) + \\ \left( \frac{1}{4}SM_1 - \frac{1}{12}M_2 \right) h^3,$$

where

$$e_f \leq \frac{1}{6}M_{2f} - \frac{1}{4}M_{2f} + \frac{1}{4}SM_{1f} = \frac{1}{4}SM_{1f} - \frac{1}{12}M_{2f},$$

$$M_1 = M_{1f} + M_{1g}, M_{1f} = \max_{t \in [t_1, t_n]} |f'(t, x, y)|, M_{1g} = \max_{t \in [t_1, t_n]} |g'(t, x, y)|,$$

$$M_2 = M_{2f} + M_{2g}, M_{2f} = \max_{t \in [t_1, t_n]} |f''(t, x, y)|, M_{2g} = \max_{t \in [t_1, t_n]} |g''(t, x, y)|,$$

$S = S_f + S_g$  is the upper bound of  $\frac{\partial f}{\partial x_i} \left( \frac{\partial g}{\partial x_i} \right), i = 1, 2, x = x_1$  and  $y = x_2$ .

Simplifying, then:

$$|x_{k+1} - X_{k+1}| \leq |x_k - X_k| + \\ + \frac{h}{2} (2L_1 + hL_1L_1 + hL_2L_3) |x_k - X_k| + \frac{h}{2} (2L_2 + hL_1L_2 + hL_2L_4) |y_k - Y_k| \\ + \left( \frac{1}{4}SM_1 + \frac{1}{12} [hL_1 + hL_2 + 1] M_2 \right) h^3.$$

Similarly,

$$|y_{k+1} - Y_{k+1}| \leq |y_k - Y_k| \\ + \frac{h}{2} (2L_4 + hL_3L_2 + hL_4L_4) |y_k - Y_k| + \frac{h}{2} (2L_3 + hL_3L_1 + hL_4L_3) |x_k - X_k| \\ + \left( \frac{1}{4}SM_1 + \frac{1}{12} [hL_3 + hL_4 + 1] M_2 \right) h^3.$$

Therefore,

Case 1. If  $L = \sum_{i=1}^4 L_i$  and  $c = \frac{1}{2}SM_1 + \frac{1}{6} [2hL + 1] M_2$ , we get

$$|x_{k+1} - X_{k+1}| \leq |x_{k+1} - X_{k+1}| + |y_{k+1} - Y_{k+1}| \leq [1 + 2hL (1 + hL)] (|x_k - X_k| + |y_k - Y_k|) + \frac{h^3}{2}c, \\ |y_{k+1} - Y_{k+1}| \leq |x_{k+1} - X_{k+1}| + |y_{k+1} - Y_{k+1}| \leq [1 + 2hL (1 + hL)] (|x_k - X_k| + |y_k - Y_k|) + \frac{h^3}{2}c.$$

By using the proof of Theorem 4, when  $U_1 = 0$ , this implies that:

$$|x_n - X_n| \leq \frac{h^2 c}{4L \left(1 + \frac{L(t_n - t_1)}{n}\right)} \left[ \exp \left( 2L (t_n - t_1) \left( 1 + \frac{L(t_n - t_1)}{n} \right) \right) - 1 \right]. \quad (32)$$

In a similar way,

$$|y_n - Y_n| \leq \frac{h^2 c}{4L \left(1 + \frac{L(t_n - t_1)}{n}\right)} \left[ \exp \left( 2L (t_n - t_1) \left( 1 + \frac{L(t_n - t_1)}{n} \right) \right) - 1 \right]. \quad (33)$$

Case 2. If  $L = L_i, i = 1, \dots, 4$  and:

$$\begin{cases} |f(t_k, x_k, y_k) - \hat{F}_k| & \leq \frac{h^2}{6} M_2 + 2L \max \{|x_k - X_k|, |y_k - Y_k|\}, \\ |g(t_k, x_k, y_k) - \hat{G}_k| & \leq \frac{h^2}{6} M_2 + 2L \max \{|x_k - X_k|, |y_k - Y_k|\}. \end{cases}$$

Thus,

$$\begin{cases} |x_{k+1} - X_{k+1}| \leq |x_k - X_k| + 2Lh(1+hL) \max \{|x_k - X_k|, |y_k - Y_k|\} + \frac{h^3}{2} c, \\ |y_{k+1} - Y_{k+1}| \leq |y_k - Y_k| + 2Lh(1+hL) \max \{|x_k - X_k|, |y_k - Y_k|\} + \frac{h^3}{2} c, \end{cases}$$

where  $c = \frac{1}{2}SM_1 + \frac{1}{6}[2hL + 1]M_2$ . Consequently,

$$\begin{cases} |x_k - X_k| \leq (1 + 4Lh(1+hL))^k |U_1| + h^3 c \frac{(1+4Lh(1+hL))^k - 1}{4Lh(1+hL)}, \\ |y_k - Y_k| \leq (1 + 4Lh(1+hL))^k |U_1| + h^3 c \frac{(1+4Lh(1+hL))^k - 1}{4Lh(1+hL)}, \end{cases}$$

where  $U_k = |x_k - X_k| + |y_k - Y_k|$ . In particular, when  $U_1 = 0$ , we get:

$$\begin{cases} |x_n - X_n| \leq h^2 c \frac{\left[ \exp \left( 4L(t_n - t_1) \left( 1 + \frac{L(t_n - t_1)}{n} \right) \right) - 1 \right]}{4L \left( 1 + \frac{L(t_n - t_1)}{n} \right)}, \\ |y_n - Y_n| \leq h^2 c \frac{\left[ \exp \left( 4L(t_n - t_1) \left( 1 + \frac{L(t_n - t_1)}{n} \right) \right) - 1 \right]}{4L \left( 1 + \frac{L(t_n - t_1)}{n} \right)}, \end{cases} \quad (34)$$

and if  $h = \{h_1, \dots, h_m\} \rightarrow 0, m > 0$  in (32), (33) and (34), we get  $|x_n - X_{n,h}| \rightarrow 0, |y_n - Y_{n,h}| \rightarrow 0$ , which concludes the proof.  $\square$

#### 4. Applications

One of the main problems of mathematics appears with variable coefficients when  $\alpha(t), \beta(t), \delta(t), \gamma(t)$  are analytic functions and added to the model. The new differential equations are represented by non-autonomous SODEs. In this model, time varying values for the growth rate of the prey, the efficiency of the predator, being the ability to capture prey, the death rate of the predator and the growth rate of the predator are considered. It is important to remark that since in this problem, the coefficients are time varying, careful attention must be paid in order to obtain the correct recurrence equation system of the model. The model, incorporating the above functions, is as follows [30,33,34]:

$$\begin{aligned} \frac{dx}{dt} &= \alpha(t)x(t) - \beta(t)x(t)y(t), & x(0) &= x_1 \\ \frac{dy}{dt} &= \delta(t)x(t)y(t) - \gamma(t)y(t), & y(0) &= y_1 \end{aligned} \quad (35)$$

Three examples are discussed in order to prove the results obtained by Scheme I (14)–(15) and Scheme II (29)–(30), two examples for the numerical solution of the model (35) and one example for the linear case.

**Example 1.** Consider the problem of the Lotka–Volterra prey–predator model (35). We take

$$\alpha(t) = 4 + \tan(t), \beta(t) = \exp(2t), \gamma(t) = -2, \delta(t) = \cos(t), x(0) = -4 \text{ and } y(0) = 4.$$

The exact solution for these coefficients is  $x(t) = \frac{-4}{\cos(t)}$ ,  $y(t) = 4 \exp(-2t)$ , as proposed by [30,33,34].

**Example 2.** Consider the problem of the Lotka–Volterra prey–predator model (35) with

$$\alpha(t) = -t, \beta(t) = -t, \gamma(t) = t, \delta(t) = t, x(0) = 2 \text{ and } y(0) = 2.$$

The exact solution for these coefficients is  $x(t) = \frac{2}{2 - \exp(t^2/2)}$ ,  $y(t) = \frac{2}{2 - \exp(t^2/2)}$ , as proposed by [30,33].

**Example 3.** Consider the following non-autonomous SODEs with initial values (5):

$$\begin{cases} x'(t) = x(t) - y(t) + 2t - t^2 - t^3, & x(0) = 1, \quad t \in [0, 1] \\ y'(t) = x(t) + y(t) - 4t^2 + t^3, & y(0) = 0. \end{cases} \quad (36)$$

The exact solution of (36) is given by  $x(t) = e^t \cos(t) + t^2$  and  $y(t) = e^t \sin(t) - t^3$ .

The results are listed in Tables 1–7 by the proposed fuzzy approximation methods with respect to the raised cosine generating function and Table 8 by the proposed fuzzy approximation methods with respect to the triangular generating function and raised cosine generating function. The proposed fuzzy approximation methods are generated by Algorithms A1 and A2 (please see Appendix B). The mean square error (MSE) is defined as  $MSE = \frac{1}{n} (\|Y_k - y(t_k)\|_2)^2$ . This is an easily computable quantity for a particular sample. From the numerical tests, the results are summarized as follows:

1. In view of Tables 2–7, a comparison is made between the two new proposed schemes (15), (29), the Euler method and the trapezoidal rule based on the Euler method for Examples 1–3.
2. Moreover, a comparison of MSE for Examples 1–3 is shown in Table 1. It is observed that the new fuzzy approximation methods yield more accurate results in comparison with the classical Euler and classical trapezoidal rule (one-step). The best result (in comparison with the Schemes I and II) is obtained by Scheme II.
3. In Table 8, a comparison is given between the errors for two proposed schemes based on the FzT with respect to fuzzy partitions determined by [6,19].

The better results (in comparison with the non-linear case) are obtained by the linear case and non-autonomous SODEs in Example 3. Further, the results obtained using proposed fuzzy approximation methods for Examples 1–3 are shown in Figure 1 by using the raised cosine generating function. In view of Figure 1, the graphical results of Examples 1–3 show a comparison between numerical schemes (I and II) and the exact solution. Furthermore, in view of Figure 1, a comparison is given between the numerical results of Examples 1 and 2 and exact solutions for  $h = 0.01$ , while a comparison is given between the numerical results of Example 3 and exact solutions for  $h = 0.1$ . All the graphs are plotted using MATLAB software. This constitutes an important improvement to the previous fuzzy approach, which did not provide such information for SODEs. Thus, this study will be particularly important.

**Remark 6.** We compare new results based on FzT with the conventional numerical methods. For a discussion of the conventional numerical methods, the Euler method and trapezoidal rule to solve SODEs, see for example [35,36].

**Table 1.** The values of MSE for Example 1–3.

Method	Example 1		Example 2		Example 3	
	$x(t)$	$y(t)$	$x(t)$	$y(t)$	$x(t)$	$y(t)$
Scheme I	$2.91443 \times 10^{-1}$	$6.67431 \times 10^{-3}$	$1.12399 \times 10^{-1}$	$1.12399 \times 10^{-1}$	$2.92534 \times 10^{-4}$	$1.58256 \times 10^{-3}$
Scheme II	$2.24139 \times 10^{-2}$	$3.77476 \times 10^{-4}$	$3.04846 \times 10^{-4}$	$3.04846 \times 10^{-4}$	$1.72082 \times 10^{-5}$	$4.10161 \times 10^{-5}$
Euler	$6.99731 \times 10^{-1}$	$1.19826 \times 10^{-2}$	$1.14890 \times 10^{-1}$	$1.14890 \times 10^{-1}$	$5.43867 \times 10^{-4}$	$1.68059 \times 10^{-3}$
Trapezoidal	$5.99915 \times 10^{-1}$	$1.40165 \times 10^{-3}$	$2.75574 \times 10^{-2}$	$2.75574 \times 10^{-2}$	$3.19103 \times 10^{-5}$	$5.21595 \times 10^{-4}$

**Table 2.** Comparison of numerical results of  $x(t)$  for Example 3.

$t_i$	Solution $x(t)$	Proposed Scheme I	Proposed Scheme II	Euler	Trapezoidal
0.00	1.00000	1.00000	1.00000	1.00000	1.00000
0.10	1.10965	1.10581	1.11259	1.10000	1.10945
0.20	1.23706	1.22517	1.24000	1.21890	1.23671
0.30	1.37957	1.36112	1.38257	1.35438	1.37924
0.40	1.53406	1.51087	1.53717	1.50368	1.53402
0.50	1.69689	1.67119	1.70015	1.66355	1.69755
0.60	1.86386	1.83827	1.86733	1.83022	1.86575
0.70	2.03020	2.00777	2.03392	1.99935	2.03396
0.80	2.19055	2.17473	2.19456	2.16601	2.19692
0.90	2.33891	2.33356	2.34322	2.32461	2.34870
1.00	2.46869	2.47798	2.47776	2.46891	2.48270

**Table 3.** Comparison of numerical results of  $y(t)$  for Example 3.

$t_i$	Solution $y(t)$	Proposed Scheme I	Proposed Scheme II	Euler	Trapezoidal
0.00	0.00000	0.00000	0.00000	0.00000	0.00000
0.10	0.10933	0.09948	0.10781	0.10000	0.10805
0.20	0.23466	0.21563	0.23168	0.21610	0.23113
0.30	0.37191	0.34407	0.36755	0.34440	0.36521
0.40	0.51694	0.48088	0.51126	0.48098	0.50621
0.50	0.66544	0.62209	0.65855	0.62184	0.64994
0.60	0.81285	0.76359	0.80488	0.76288	0.79202
0.70	0.95430	0.90109	0.94543	0.89979	0.92782
0.80	1.08451	1.03002	1.07494	1.02801	1.05236
0.90	1.19767	1.14549	1.18766	1.14261	1.16023
1.00	1.28736	1.24213	1.28463	1.23823	1.24549

**Table 4.** Comparison of numerical results of  $x(t)$  for Example 1.

$t_i$	Solution $x(t)$	Proposed Scheme I	Proposed Scheme II	Euler	Trapezoidal
0.00	-4.00000	-4.00000	-4.00000	-4.00000	-4.00000
0.05	-4.00501	-3.97865	-3.99616	-4.00000	-4.00714
0.10	-4.02008	-3.99232	-4.01193	-4.01429	-4.02627
0.15	-4.04543	-4.01914	-4.03732	-4.04276	-4.05752
0.20	-4.08136	-4.05937	-4.07264	-4.08574	-4.10134
0.25	-4.12834	-4.11358	-4.11833	-4.14389	-4.15850
0.30	-4.18701	-4.18268	-4.17490	-4.21830	-4.23015
0.35	-4.25816	-4.26794	-4.24305	-4.31052	-4.31783
0.40	-4.34282	-4.37105	-4.32359	-4.42260	-4.42361
0.45	-4.44224	-4.49423	-4.41752	-4.55722	-4.55014
0.50	-4.55798	-4.64028	-4.52605	-4.71784	-4.70086
0.55	-4.69195	-4.81278	-4.65062	-4.90891	-4.88023
0.60	-4.84651	-5.01628	-4.79297	-5.13613	-5.09399
0.65	-5.02460	-5.25662	-4.95520	-5.40685	-5.34964
0.70	-5.22984	-5.54127	-5.13983	-5.73061	-5.65700
0.75	-5.46680	-5.87990	-5.34992	-6.11990	-6.02912
0.80	-5.74130	-6.28518	-5.58925	-6.59122	-6.48349
0.85	-6.06076	-6.77381	-5.86249	-7.16663	-7.04390
0.90	-6.43490	-7.36820	-6.17551	-7.87602	-7.74310
0.95	-6.87660	-8.09874	-6.53582	-8.76042	-8.62699
1.00	-7.40326	-9.00740	-6.96630	-9.87710	-9.76103

**Table 5.** Comparison of numerical results of  $y(t)$  for Example 1.

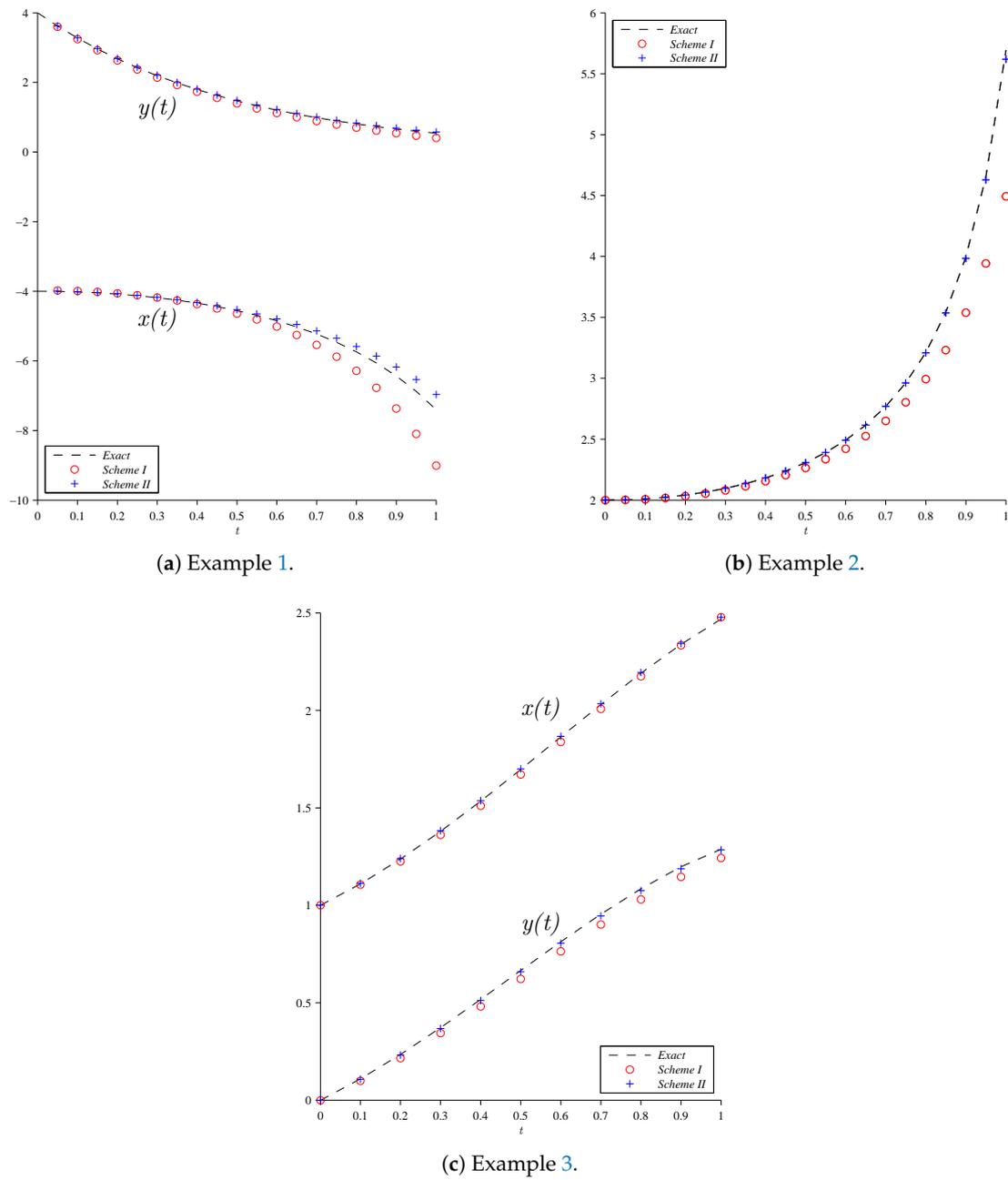
$t_i$	Solution $y(t)$	Proposed Scheme I	Proposed Scheme II	Euler	Trapezoidal
0.00	4.00000	4.00000	4.00000	4.00000	4.00000
0.05	3.61935	3.60013	3.62249	3.60000	3.62045
0.10	3.27492	3.24497	3.28040	3.24090	3.27766
0.15	2.96327	2.92506	2.97057	2.91774	2.96757
0.20	2.68128	2.63645	2.69004	2.62635	2.68671
0.25	2.42612	2.37573	2.43612	2.36315	2.43201
0.30	2.19525	2.13994	2.20634	2.12506	2.20079
0.35	1.98634	1.92646	1.99847	1.90938	1.99067
0.40	1.79732	1.73299	1.81047	1.71374	1.79951
0.45	1.62628	1.55749	1.64050	1.53607	1.62541
0.50	1.47152	1.39815	1.48689	1.37451	1.46664
0.55	1.33148	1.25333	1.34811	1.22742	1.32166
0.60	1.20478	1.12159	1.22279	1.09333	1.18908
0.65	1.09013	1.00161	1.10969	0.97093	1.06762
0.70	0.98639	0.89223	1.00768	0.85906	0.95615
0.75	0.89252	0.79241	0.91576	0.75670	0.85366
0.80	0.80759	0.70122	0.83301	0.66295	0.75923
0.85	0.73073	0.61784	0.75862	0.57703	0.67208
0.90	0.66120	0.54154	0.69184	0.49827	0.59150
0.95	0.59827	0.47170	0.63202	0.42612	0.51692
1.00	0.54134	0.40778	0.57734	0.36016	0.44787

**Table 6.** Comparison of numerical results of  $x(t)$  for Example 2.

$t_i$	Solution $x(t)$	Proposed Scheme I	Proposed Scheme II	Euler	Trapezoidal
0.00	2.00000	2.00000	2.00000	2.00000	2.00000
0.05	2.00250	2.00149	2.00325	2.00000	2.00250
0.10	2.01008	2.00650	2.01082	2.00500	2.01005
0.15	2.02289	2.01660	2.02364	2.01508	2.02279
0.20	2.04124	2.03197	2.04201	2.03042	2.04101
0.25	2.06557	2.05294	2.06636	2.05134	2.06511
0.30	2.09650	2.07996	2.09731	2.07830	2.09567
0.35	2.13485	2.11365	2.13568	2.11191	2.13344
0.40	2.18171	2.15485	2.18256	2.15301	2.17942
0.45	2.23852	2.20462	2.23939	2.20265	2.23493
0.50	2.30720	2.26437	2.30808	2.26226	2.30167
0.55	2.39031	2.33595	2.39116	2.33365	2.38192
0.60	2.49133	2.42177	2.49211	2.41923	2.47868
0.65	2.61513	2.52506	2.61574	2.52224	2.59605
0.70	2.76863	2.65022	2.76888	2.64702	2.73967
0.75	2.96202	2.80329	2.96157	2.79961	2.91754
0.80	3.21093	2.99285	3.20907	2.98854	3.14130
0.85	3.54059	3.23143	3.53586	3.22625	3.42845
0.90	3.99443	3.53788	3.98346	3.53151	3.80653
0.95	4.65413	3.94192	4.62841	3.93381	4.32100
1.00	5.69348	4.49277	5.61875	4.48201	5.05197

**Table 7.** Comparison of numerical results of  $y(t)$  for Example 2.

$t_i$	Solution $y(t)$	Proposed Scheme I	Proposed Scheme II	Euler	Trapezoidal
0.00	2.00000	2.00000	2.00000	2.00000	2.00000
0.05	2.00250	2.00149	2.00325	2.00000	2.00250
0.10	2.01008	2.00650	2.01082	2.00500	2.01005
0.15	2.02289	2.01660	2.02364	2.01508	2.02279
0.20	2.04124	2.03197	2.04201	2.03042	2.04101
0.25	2.06557	2.05294	2.06636	2.05134	2.06511
0.30	2.09650	2.07996	2.09731	2.07830	2.09567
0.35	2.13485	2.11365	2.13568	2.11191	2.13344
0.40	2.18171	2.15485	2.18256	2.15301	2.17942
0.45	2.23852	2.20462	2.23939	2.20265	2.23493
0.50	2.30720	2.26437	2.30808	2.26226	2.30167
0.55	2.39031	2.33595	2.39116	2.33365	2.38192
0.60	2.49133	2.42177	2.49211	2.41923	2.47868
0.65	2.61513	2.52506	2.61574	2.52224	2.59605
0.70	2.76863	2.65022	2.76888	2.64702	2.73967
0.75	2.96202	2.80329	2.96157	2.79961	2.91754
0.80	3.21093	2.99285	3.20907	2.98854	3.14130
0.85	3.54059	3.23143	3.53586	3.22625	3.42845
0.90	3.99443	3.53788	3.98346	3.53151	3.80653
0.95	4.65413	3.94192	4.62841	3.93381	4.32100
1.00	5.69348	4.49277	5.61875	4.48201	5.05197



**Figure 1.** A comparison between three fuzzy numerical methods and the exact solution for three examples.

**Table 8.** The values of MSE for Examples 1–3 by the different types of fuzzy partitions.

Case		Proposed Scheme for $x(t)$		Proposed Scheme for $y(t)$	
		I	II	I	II
Ex.1	T <sup>1</sup>	$2.48353 \times 10^{-1}$	$3.37890 \times 10^{-2}$	$6.03282 \times 10^{-3}$	$5.38734 \times 10^{-4}$
	C <sup>2</sup>	$2.91443 \times 10^{-1}$	$2.24139 \times 10^{-2}$	$6.67431 \times 10^{-3}$	$3.77476 \times 10^{-4}$
Ex.2	T	$1.12099 \times 10^{-1}$	$3.01900 \times 10^{-4}$	$1.12099 \times 10^{-1}$	$3.01900 \times 10^{-4}$
	C	$1.12399 \times 10^{-1}$	$3.04846 \times 10^{-4}$	$1.12399 \times 10^{-1}$	$3.04846 \times 10^{-4}$
Ex.3	T	$2.71905 \times 10^{-4}$	$2.08807 \times 10^{-5}$	$1.61509 \times 10^{-3}$	$4.84176 \times 10^{-5}$
	C	$2.92534 \times 10^{-4}$	$1.72082 \times 10^{-5}$	$1.58256 \times 10^{-3}$	$4.10161 \times 10^{-5}$

<sup>1</sup> Triangular generating function; <sup>2</sup> Raised cosine generating function.

### 5. Conclusions

We extended the applicability of fuzzy-based numerical methods to the problems of conventional mathematics. In particular, we contributed to approximation methods of the SODEs. Two approximation methods based on the FzT were proposed and their error estimate analyzed. Moreover, we proved that two approximation methods, namely Schemes I and II, determine an approximate solution, which converges to the exact solution, and the local truncation error of the Scheme I (Scheme II) is  $\mathcal{O}(h^2)$  ( $\mathcal{O}(h^3)$ ). As an application, a system of nonlinear differential equations is solved by using Schemes I and II. From the numerical results, it is observed that the new fuzzy approximation methods yield more accurate results in comparison with the classical Euler method (one-stage) and classical trapezoidal rule (two-stage). Hence, the new fuzzy approximation methods provided alternative techniques for solving differential equations with better results, and the objective of this research was achieved and tested.

As a consequence, it should be noted that the numerical solutions depend on the types of uniform fuzzy partitions. For cases  $\left(1 - \left|\frac{x-x_k}{h}\right|\right)$  and  $\frac{1}{2} \left(1 + \cos\left(\pi \left(\frac{x-x_k}{h}\right)\right)\right)$ , the shape of the basic functions determines the form of representation (linear or non-linear) of the numerical solution. This agrees with the results proposed by [5,6] using uniform fuzzy partitions. It is also worth pointing out that the results in this research are better in comparison with the classical numerical methods using uniform fuzzy partitions for linear and nonlinear cases. Thus, the proposed method is very much suitable for solving SODEs (5) in a linear or nonlinear case under the assumption of  $f$  and  $g$  satisfying the Lipschitz condition. If we want to obtain the best approximation of  $f$  and  $g$  as possible, then the number  $n$  of components should be large. It should be stressed that the application of the FzT can be used for removing noise from the given data. This is especially important for various practical applications of FzT. The proposed methods can also be applied to the  $n$ -dimensional system of first-order coupled differential equations in the case of a non-noisy or noisy right-hand side. The discussion will continue in [37] to give more details about the fuzzy partition and the modification of multiple steps.

**Author Contributions:** Conceptualization and performed the numerical experiments, H.A. ALKasasbeh; Evaluated the results and supported this work, I. Perfilieva; Project administration and designed the numerical methods, M. Z. Ahmad; Software and data curation, Z. R. Yahyaa.

**Funding:** The work of Irina Perfilieva has been supported by the project "IT4Innovations excellence in science, LQ1602" and by the Grant Agency of the Czech Republic (Project No. 16-09541S).

**Acknowledgments:** The authors would like to express their thanks to the editors and the anonymous referees for their valuable comments and suggestions that contributed to the paper. Many thanks are given to Universiti Malaysia Perlis for providing all facilities until this work was completed successfully.

**Conflicts of Interest:** The authors declare no conflicts of interest.

### Appendix A. Taylor Series

A Taylor series is given that:

$$\begin{aligned}
 x(t_{k+1}) &= x(t_k) + hx'(t_k) + \frac{h^2}{2}x''(t_k) + \frac{h^3}{6}x'''(\varepsilon_{1k}), \\
 &= x(t_k) + hx'(t_k) + \frac{h^2}{2}\left(\frac{x'(t_{k+1}) - x'(t_k)}{h} - \frac{h}{2}x'''(\varepsilon_{2k})\right) + \frac{h^3}{6}x'''(\varepsilon_{1k}), \\
 &= x(t_k) + \frac{h}{2}x'(t_k) + \frac{h}{2}x'(t_{k+1}) + h^3\left[\frac{1}{6}x'''(\varepsilon_{1k}) - \frac{1}{4}x'''(\varepsilon_{2k})\right], \tag{A1}
 \end{aligned}$$

where  $x''(t) = \frac{x'(t_{k+1}) - x'(t_k)}{h} - \frac{h}{2}x'''(\varepsilon_{2k})$ . Calculus can be used to derive that:

$$\begin{aligned}
 x'(t_{k+1}) &= f(t_{k+1}, x(t_{k+1}), y(t_{k+1})) \\
 &= f(t_{k+1}, x(t_k) + hf(t_k, x(t_k), y(t_k)), y(t_k) + hg(t_k, x(t_k), y(t_k))) \\
 &\quad + \frac{h^2}{2}f_2(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k}))x''(\varepsilon_{3k}),
 \end{aligned}$$

where  $f_2(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k})) = \frac{\partial}{\partial x}f(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k})) + \frac{\partial}{\partial y}f(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k}))$ .

Substituting Equation (A1), it is given that:

$$\begin{aligned}
 x(t_{k+1}) &= x(t_k) + \frac{h}{2}x'(t_k) \\
 &\quad + \frac{h}{2}f(t_{k+1}, x(t_k) + hf(t_k, x(t_k), y(t_k)), y(t_k) + hg(t_k, x(t_k), y(t_k))) \\
 &\quad + \frac{h}{2}\frac{h^2}{2}f_2(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k}))x''(\varepsilon_{3k}) \\
 &\quad + h^3\left[\frac{1}{6}x'''(\varepsilon_{1k}) - \frac{1}{4}x'''(\varepsilon_{2k})\right], \\
 x(t_{k+1}) &= x(t_k) + \\
 &\quad \frac{h}{2}(x'(t_k) + f(t_{k+1}, x(t_k) + hf(t_k, x(t_k), y(t_k)), y(t_k) + hg(t_k, x(t_k), y(t_k)))) \\
 &\quad + h^3\left[\frac{1}{6}x'''(\varepsilon_{1k}) - \frac{1}{4}x'''(\varepsilon_{2k}) + \frac{1}{4}f_2(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k}))x''(\varepsilon_{3k})\right].
 \end{aligned}$$

It can be rewritten as:

$$x(t_{k+1}) = x(t_k) + \frac{h}{2}(K_0 + K_f) + e_f h^3,$$

where:

$$\begin{aligned}
 K_0 &= x'(t) = f(t_k, x(t_k), y(t_k)), K_1 = y'(t) = g(t_k, x(t_k), y(t_k)), K_f = f(t_{k+1}, x(t_k) + hK_0, y(t_k) + hK_1), \\
 e_f &= \frac{1}{6}x'''(\varepsilon_{1k}) - \frac{1}{4}x'''(\varepsilon_{2k}) + \frac{1}{4}f_2(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k}))x''(\varepsilon_{3k}), \text{ and } t_k < \varepsilon_{1k}, \varepsilon_{2k}, \varepsilon_{3k} < t_{k+1}.
 \end{aligned}$$

Similarly,

$$y(t_{k+1}) = y(t_k) + \frac{h}{2}(K_1 + K_g) + e_g h^3,$$

where:

$$K_g = g(t_{k+1}, x(t_k) + hK_0, y(t_k) + hK_1), e_g = \frac{1}{6}y'''(\xi_{1k}) - \frac{1}{4}y'''(\xi_{2k}) + \frac{1}{4}g_2(\xi_{3k}, x(\xi_{3k}), y(\xi_{3k}))y''(\xi_{3k}),$$

$$g_2(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k})) = \frac{\partial}{\partial x}g(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k})) + \frac{\partial}{\partial y}g(\varepsilon_{3k}, x(\varepsilon_{3k}), y(\varepsilon_{3k})),$$

and  $t_k < \xi_{1k}, \xi_{2k}, \xi_{3k} < t_{k+1}$ .

### Appendix B. Algorithms

In this Appendix, the algorithms of the approximation methods based on FzT for Sections 3.2 and 3.3 are explained in detail. Pseudocode is used to describe the algorithms and a simplified code that is easy to read. This pseudocode specifies the form of the input to be supplied and the form of the desired output. As a consequence, a stopping technique independent of the numerical technique is incorporated into each algorithm to avoid infinite loops. Two punctuation symbols are used in the algorithms: a period (.) indicates the termination of a step, and a semicolon (;) separates tasks within a step. The integral symbol (integral(function,upper limits,lower limits)) is used to denote a definite integral. The steps in the algorithms follow the rules of structured program construction. They have been arranged so that there should be minimal difficulty translating pseudocode into any programming language suitable for scientific applications. We approximate the solution of SODEs (5) at  $(N + 1)$  equally-spaced numbers in the interval  $[a, b]$  as follows.

#### Algorithm A1. One-stage (modified Euler) algorithm for the system of ODEs.

INPUT:  $f(t, x, y)$  and  $g(t, x, y)$  in Equation (5); endpoints  $a, b$ ; integer  $N$ ; initial condition  $y_1$ .

Step 1 Set  $h = (b - a)/N$ ;  $X_1 = x_1$ ;  $Y_1 = y_1$ ;  $t_1 = a$ ;  $k = 1, \dots, N + 1$ ;  $t_k = a + (k - 1)h$ .

Step 2 Define the generalized uniform fuzzy partitions as  $A_k(t) = \frac{1}{2} \left( 1 + \cos \left( \pi \left( \frac{t-t(k)}{h} \right) \right) \right)$ .

Step 3 For  $k = 1$  to  $N$ , do Steps 4–7.

Step 4  $F(k) = \text{integral}(f(t, X(k), Y(k))A_k(t), t(k - 1), t(k + 1)) / \text{integral}(A_k(t), t(k - 1), t(k + 1))$ .

Step 5  $G(k) = \text{integral}(g(t, X(k), Y(k))A_k(t), t(k - 1), t(k + 1)) / \text{integral}(A_k(t), t(k - 1), t(k + 1))$ .

Step 6  $X(k + 1) = X(k) + hF(k)$ .

Step 7  $Y(k + 1) = Y(k) + hG(k)$ .

end.

OUTPUT: Approximation  $X$  and  $Y$  to  $x$  and  $y$ , respectively, at the  $(N + 1)$  values of  $t$ .

#### Algorithm A2. Two-stage (modified trapezoidal rule) algorithm for the system of ODEs.

INPUT:  $f(t, x, y)$ ;  $g(t, x, y)$ ; endpoints  $a, b$ ; integer  $N$ ; initial condition  $y_1$ .

Step 1 Set  $h = (b - a)/N$ ;  $X_1 = x_1$ ;  $Y_1 = y_1$ ;  $t_1 = a$ ;  $k = 1, \dots, N + 1$ ;  $t_k = a + (k - 1)h$ .

Step 2 Define the generalized uniform fuzzy partitions as  $A_k(t) = \frac{1}{2} \left( 1 + \cos \left( \pi \left( \frac{t-t(k)}{h} \right) \right) \right)$ .

Step 3 For  $k = 1$  to  $N$ , do Steps 4–11.

Step 04  $F(k) = \text{integral}(f(t, X(k), Y(k))A_k(t), t(k - 1), t(k + 1)) / \text{integral}(A_k(t), t(k - 1), t(k + 1))$ .

Step 05  $G(k) = \text{integral}(g(t, X(k), Y(k))A_k(t), t(k - 1), t(k + 1)) / \text{integral}(A_k(t), t(k - 1), t(k + 1))$ .

Step 06  $Xstar(k + 1) = X(k) + hF(k)$ .

Step 07  $Ystar(k + 1) = Y(k) + hG(k)$ .

Step 08  $Fstar(k + 1) = \text{integral}(f(t, Xstar(k + 1), Ystar(k + 1))A_{k+1}(t), t(k), t(k + 2)) / \text{integral}(A_{k+1}(t), t(k), t(k + 2))$ .

Step 09  $Gstar(k + 1) = \text{integral}(g(t, Xstar(k + 1), Ystar(k + 1))A_{k+1}(t), t(k), t(k + 2)) / \text{integral}(A_{k+1}(t), t(k), t(k + 2))$ .

Step 10  $X(k + 1) = X(k) + h(F(k) + Fstar(k + 1)) / 2$ .

Step 11  $Y(k + 1) = Y(k) + h(G(k) + Gstar(k + 1)) / 2$ .

end.

OUTPUT: Approximation  $X$  and  $Y$  to  $x$  and  $y$ , respectively, at the  $(N + 1)$  values of  $t$ .

## References

1. Ahmad, M.Z.; Hasan, M.K.; Baets, B.D. Analytical and numerical solutions of fuzzy differential equations. *Inf. Sci.* **2013**, *236*, 156–167. [[CrossRef](#)]
2. Shawagfeh, N.; Kaya, D. Comparing numerical methods for the solutions of systems of ordinary differential equations. *Appl. Math. Lett.* **2004**, *17*, 323–328. [[CrossRef](#)]
3. Atkinson, K.; Han, W.; Stewart, D. *Numerical Solution of Ordinary Differential Equations*; Wiley: New York, NY, USA, 2009.
4. Ahmad, M.Z.; De Baets, B. A Predator-Prey Model with Fuzzy Initial Populations. In Proceeding of the Joint 13th IFSA World Congress and 6th EUSFLAT Conference, Lisbon, Portugal, 20–24 July 2009; pp. 1311–1314.
5. Perfilieva, I. Fuzzy transforms: Theory and applications. *Fuzzy Sets Syst.* **2006**, *157*, 993–1023. [[CrossRef](#)]
6. Perfilieva, I. Fuzzy transform: Application to the Reef growth problem. In *Fuzzy Logic in Geology*; Demicco, R.V., Klir, G.J., Eds.; Academic Press: Amsterdam, The Netherlands, 2003; Chapter 9, pp. 275–300.
7. Perfilieva, I.; Daňková, M.; Bede, B. Towards a higher degree F-transform. *Fuzzy Sets Syst.* **2011**, *180*, 3–19. [[CrossRef](#)]
8. Chen, W.; Shen, Y. Approximate solution for a class of second-order ordinary differential equations by the fuzzy transform. *J. Intell. Fuzzy Syst.* **2014**, *27*, 73–82.
9. Alireza, K.; Zahra, A.; Irina, P. Fuzzy transform to approximate solution of two-point boundary value problems. *Math. Meth. Appl. Sci.* **2017**, *40*, 6147–6154.
10. Tomasiello, S. An alternative use of fuzzy transform with application to a class of delay differential equations. *Int. J. Comput. Math.* **2017**, *94*, 1719–1726. [[CrossRef](#)]
11. Hodakova, P.; Perfilieva, I.; Valasek, R. A new approach to fuzzy boundary value problem. In *Uncertainty Modelling in Knowledge Engineering and Decision Making*; World Scientific Proceedings Series on Computer Engineering and Information Science; World Scientific Publishing Company: Singapore, 2016; Volume 10, pp. 276–281; ISBN: 978-981-3146-96-9.
12. Perfilieva, I.; Števíliáková, P.; Valášek, R. F-transform for numerical solution of two-point boundary value problem. *Iran. J. Fuzzy Syst.* **2017**, *14*, 1–13.
13. Perfilieva, I.; Števíliáková, P.; Valášek, R. F-transform-based shooting method for nonlinear boundary value problems. *Soft Comput.* **2017**, *21*, 3493–3502. [[CrossRef](#)]
14. Alijani, Z.; Khastan, A.; Khattri, S.K.; Tomasiello, S. Fuzzy Transform to Approximate Solution of Boundary Value Problems via Optimal Coefficients. In Proceedings of the 2017 International Conference on High Performance Computing Simulation (HPCS), Genoa, Italy, 17–21 July 2017; pp. 466–471.
15. Alikhani, R.; Zeinali, M.; Bahrami, F.; Shahmorad, S.; Perfilieva, I. Trigonometric  $F^m$ -transform and its approximative properties. *Soft Comput.* **2017**, *21*, 3567–3577. [[CrossRef](#)]
16. Jahedi, S.; Javadi, F.; Mehdipour, M.J. Weighted transform and approximation of some functions on unbounded sets. *Soft Comput.* **2017**, *21*, 3579–3585. [[CrossRef](#)]
17. Tomasiello, S.; Gaeta, M.; Loia, V. Quasi-consensus in Second-Order Multi-agent Systems with Sampled Data Through Fuzzy Transform. *J. Uncertain Syst.* **2016**, *10*, 243–250.
18. Tomasiello, S. A First Investigation on the Dynamics of Two Delayed Neurons through Fuzzy Transform Approximation. In Proceedings of the 2017 International Conference on High Performance Computing Simulation (HPCS), Genoa, Italy, 17–21 July 2017; pp. 460–465.
19. Alkasasbeh, H.A.; Perfilieva, I.; Ahmad, M.Z.; Yahya, Z.R. New fuzzy numerical methods for solving Cauchy problems. *Appl. Syst. Innov.* **2018**, *1*, 15. [[CrossRef](#)]
20. Parapari, H.F.; Menhaj, M.B. Solving nonlinear ordinary differential equations using neural networks. In Proceedings of the 2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA), Qazvin, Iran, 27–28 January 2016; pp. 351–355.
21. Ramos, H.; Singh, G.; Kanwar, V.; Bhatia, S. An embedded 3 (2) pair of nonlinear methods for solving first order initial-value ordinary differential systems. *Numer. Algorithms* **2017**, *75*, 509–529. [[CrossRef](#)]
22. Perez, J.F.S.; Conesa, M.; Alhama, I. Solving ordinary differential equations by electrical analogy: A multidisciplinary teaching tool. *Eur. J. Phys.* **2016**, *37*, 065703. [[CrossRef](#)]
23. Al-Omari, A.; Arnold, J.; Taha, T.; Schüttler, H.B. Solving Large Nonlinear Systems of First-Order Ordinary Differential Equations with Hierarchical Structure Using Multi-GPGPUs and an Adaptive Runge Kutta ODE Solver. *IEEE Access* **2013**, *1*, 770–777. [[CrossRef](#)]

24. Opanuga, A.; Edeki, S.; Okagbue, H.; Akinlabi, G.; Osheku, A.; Ajayi, B. On numerical solutions of systems of ordinary differential equations by numerical-analytical method. *Appl. Math. Sci.* **2014**, *8*, 8199–8207. [[CrossRef](#)]
25. Al-Omari, A.; Schuttler, H.B.; Arnold, J.; Taha, T. Solving Nonlinear Systems of First Order Ordinary Differential Equations Using a Galerkin Finite Element Method. *IEEE Access* **2013**, *1*, 408–417. [[CrossRef](#)]
26. Matveev, S.A.; Smirnov, A.P.; Tyrtysnikov, E. A fast numerical method for the Cauchy problem for the Smoluchowski equation. *J. Comput. Phys.* **2015**, *282*, 23–32. [[CrossRef](#)]
27. Mondal, S.P.; Roy, T.K. First order homogeneous ordinary differential equation with initial value as triangular intuitionistic fuzzy number. *J. Uncertain. Math. Sci.* **2014**, *2014*, 1–17. [[CrossRef](#)]
28. Mondal, S.P.; Roy, T.K. System of Differential Equation with Initial Value as Triangular Intuitionistic Fuzzy Number and its Application. *Int. J. Appl. Comput. Math.* **2015**, *1*, 449–474. [[CrossRef](#)]
29. Paul, S.; Mondal, S.P.; Bhattacharya, P. Numerical solution of Lotka Volterra prey predator model by using Runge-Kutta-Fehlberg method and Laplace Adomian decomposition method. *Alex. Eng. J.* **2016**, *55*, 613–617. [[CrossRef](#)]
30. Yusufoglu, E.; Erbas, B. He's variational iteration method applied to the solution of the prey and predator problem with variable coefficients. *Phys. Lett. A* **2008**, *372*, 3829–3835. [[CrossRef](#)]
31. Li, J.; Zhao, A. Stability analysis of a non-autonomous Lotka-Volterra competition model with seasonal succession. *Appl. Math. Modell.* **2016**, *40*, 763–781. [[CrossRef](#)]
32. Khastan, A.; Perfilieva, I.; Alijani, Z. A new fuzzy approximation method to Cauchy problems by fuzzy transform. *Fuzzy Sets Syst.* **2016**, *288*, 75–95. [[CrossRef](#)]
33. Bougoffa, L. Solvability of the predator and prey system with variable coefficients and comparison of the results with modified decomposition. *Appl. Math. Comput.* **2006**, *182*, 383–387. [[CrossRef](#)]
34. González-Parra, G.C.; Arenas, A.J.; Cogollo, M.R. Numerical-analytical solutions of predator-prey models. *WSEAS Trans. Biol. Biomed.* **2013**, *10*, 79–87.
35. Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*, 3rd ed.; John Wiley & Sons, Ltd.: Chichester, UK, 2016.
36. Burden, R.L.; Faires, J.D. *Numerical Analysis*, 9th ed.; Brooks/Cole Cengage Learning: Boston, MA, USA, 2010; ISBN: 978-0-538-73351-9.
37. Alkasasbeh, H.A.; Perfilieva, I.; Ahmad, M.Z.; Yahya, Z.R. New approximation methods based on fuzzy transform for solving SODEs: II. *Appl. Syst. Innov.* **2018**, *1*, 30.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).