axioms

MDPI

Article

On the Numerical Solution of Ordinary, Interval and Fuzzy Differential Equations by Use of F-Transform

Davide Radi [1,2], Laerte Sorini [3] and Luciano Stefanini [3,*]

[1] Department of Economics and Management, University of Pisa, Via C. Ridolfi, 10, 56124 Pisa (PI), Italy; davide.radi@unipi.it or davide.radi@vsb.cz

[2] Department of Finance, Faculty of Economics, VŠB—Technical University of Ostrava, Sokolská tr. 33, 70121 Ostrava, Czech Republic

[3] DESP, Department of Economics, Society, Politics, University of Urbino Carlo Bo, Via A. Saffi 42, 61029 Urbino, Italy; laerte.sorini@uniurb.it

* Correspondence: luciano.stefanini@uniurb.it

Received: 24 November 2019; Accepted: 28 January 2020; Published: 5 February 2020

Abstract: An interesting property of the inverse F-transform $\hat{f}$ of a continuous function f on a given interval $[a,b]$ says that the integrals of $\hat{f}$ and f on $[a,b]$ coincide. Furthermore, the same property can be established for the restrictions of the functions to all subintervals $[a,p_k]$ of the fuzzy partition of $[a,b]$ used to define the F-transform. Based on this fact, we propose a new method for the numerical solution of ordinary differential equations (initial-value ordinary differential equation (ODE)) obtained by approximating the derivative $\dot{x}(t)$ via F-transform, then computing (an approximation of) the solution $x(t)$ by exact integration. For an ODE, a global second-order approximation is obtained. A similar construction is then applied to interval-valued and (level-wise) fuzzy differential equations in the setting of generalized differentiability (gH-derivative). Properties of the new method are analyzed and a computational section illustrates the performance of the obtained procedures, in comparison with well-known efficient algorithms.

Keywords: F-transform; initial-value ODE; numerical ODE solver; interval differential equations; gH-Derivative; fuzzy differential equations

1. Introduction

The fuzzy transform (F-transform) of a continuous function $f:[a,b]\longrightarrow\mathbb{R}$ was introduced by Perfilieva in [1,2]. This special fuzzy method is particularly appealing and useful to handle many real-world problems and an extensive research activity has both analyzed its properties and its fields of applications; for the literature related to this paper we refer to, e.g., [3–11] and the references therein.

In recent research, attention has been paid to the numerically approximated solutions of ordinary differential equations (ODEs) $\dot{x}(t)=F(t,x)$ of various types. In particular, it is shown that by using the inverse F-transform, it is possible to obtain good approximations of the solution $x(t)$. The methods that use the F-transform are (computationally) superior with respect to other ones such as the second-order Runge–Kutta algorithm or basic multi-step algorithms (see [12–16]). In the final section of this paper, we will present some comments and a preliminary comparative valuation of the proposed methods.

In this paper, we propose a numerical method where the inverse F-transform is used to approximate the derivative $\dot{x}(t)$; the solution $x(t)$ is then obtained by exact integration of the approximated derivative: this is allowed by an interesting property which says that the integral of the inverse F-transform of f on $[a,b]$ coincides (exactly) with the integral of the function f itself; this idea was presented in a preliminary form in [17].

For an ordinary differential equation, including the case of a system, a global second-order approximation is obtained. Properties of the new method are analyzed and a computational section illustrates the performance of the obtained procedures, in comparison with well-known efficient algorithms.

A similar construction is then applied to interval-valued (IDE) and (level-wise) fuzzy differential equations (FDE) in the setting of generalized differentiability (gH-derivative, see [18–20]). IDEs and FDEs are designed to model uncertainty and its propagation in a dynamical setting and it is well known that the fuzzy case can be expressed in terms of a family of IDEs by adopting the level-wise representation of fuzzy numbers and fuzzy-valued functions. For a modern introduction to FDEs under Hukuhara and generalized derivative, we refer to chapter 9 of Bede's book [21]. The interested reader is also referred to the recent book [22], in particular to chapter 4, and the references therein.

This paper is organized into five sections. In Section 2 we recall some of the basic definitions and properties of the F-transform as contained in [1,2,23]. Then, in Section 3, we describe our approach to the numerical solution of ordinary (and systems of) differential equations with initial conditions, usually referred as Cauchy problem. The F-transform is used to approximate the derivative of the solution to be founded and the unknown function is determined by point-wise by exact integration of the derivative; this section also contains the main properties of the method and a proof of its (global) convergence. Numerical examples and a computational comparison with other well-performing algorithms is presented in Section 4. Section 5 considers the case of interval differential equations and extends the use of F-transform to the numerical solution of IDEs and FDEs under (level-wise) generalized Hukuhara differentiability; in particular, the switching phenomenon is analyzed and rule to manage the switching is proposed and implemented in the proposed procedure. Several computational examples of interval and fuzzy differential equation are presented and discussed in Section 6. Section 7 presents some conclusions and present some ideas for further work.

2. Preliminaries

We briefly recall the basic definitions and properties of the F-transform setting. For all the details, we refer to the papers [1,2,23].

A fuzzy partition $(\mathbb{P}, \mathbb{A})$ of the compact interval $[a, b]$ is defined by a finite decomposition $\mathbb{P} = \{a = p_1 < p_2 < ... < p_n = b\}$ of $[a, b]$ with n points and by a family $\mathbb{A} = \{A_1, A_2, ..., A_n\}$ of n continuous basic functions $A_k : [a, b] \longrightarrow [0, 1]$ with the following properties (the decomposition $\mathbb{P}$ is not required to be uniform):

1. $A_k(t) = 0$ for $t \notin]p_{k-1}, p_{k+1}[$ $(k = 2, ..., n-1)$, $A_1(t) = 0$ for $t \notin [p_1, p_2[$, $A_n(t) = 0$ for $t \notin]p_{n-1}, p_n]$;
2. $A_k(p_k) = 1$ for all $k = 1, 2, ..., n$ and $A_1(t) + A_2(t) + ... + A_n(t) = 1$ for all $t \in [a, b]$;
3. for $k = 2, ..., n-1$, A_k is increasing on $[p_{k-1}, p_k]$ and decreasing on $[p_k, p_{k+1}]$, A_1 is decreasing on $[p_1, p_2]$, A_n is increasing of $[p_{n-1}, p_n]$.

Let us define the following integrals

$$\begin{cases} I_1^- = 0 & , \quad I_1^+ = \int_{p_1}^{p_2} A_1(t)\, dt \\ I_k^- = \int_{p_{k-1}}^{p_k} A_k(t)\, dt & , \quad I_k^+ = \int_{p_k}^{p_{k+1}} A_k(t)\, dt \\ I_n^- = \int_{p_{n-1}}^{p_n} A_1(t)\, dt & , \quad I_n^+ = 0 \end{cases} \tag{1}$$

and

$$I_k = I_k^- + I_k^+ \text{ for } k = 1, ..., n. \tag{2}$$

The direct F-transform of f with respect to $(\mathbb{P},\mathbb{A})$ is the n-tuple of real numbers $\mathbf{F}_{(\mathbb{P},\mathbb{A})} = (F_1, F_2, ..., F_n)$ defined as

$$F_1 = \frac{1}{I_1}\int_{p_1}^{p_2} f(t)A_k(t)dt, \quad F_n = \frac{1}{I_n}\int_{p_1}^{p_n} f(t)A_k(t)dt, \tag{3}$$
$$F_k = \frac{1}{I_k}\int_{p_{k-1}}^{p_{k+1}} f(t)A_k(t)dt, \quad k = 2, ..., n-1$$

and, obtained from the direct fuzzy transform $\mathbf{F}_{(\mathbb{P},\mathbb{A})}$, the inverse F-transform (iF-transform) of f is the function $f_{(\mathbb{P},\mathbb{A})} : [a,b] \longrightarrow R$ given by

$$f_{(\mathbb{P},\mathbb{A})}(t) = \sum_{k=1}^{n} F_k A_k(t) \text{ for } t \in [a,b]. \tag{4}$$

The following properties (see [1]) are the fundamentals of the F-transform setting.

Proposition 1. *(from [1]) If $f : [a,b] \longrightarrow \mathbb{R}$ is a continuous function then;*

1. *for any positive real ε, there exists a fuzzy partition $(\mathbb{P}_\varepsilon, \mathbb{A}_\varepsilon)$ such that the corresponding iF-transform $f_{(\mathbb{P}_\varepsilon,\mathbb{A}_\varepsilon)} : [a,b] \longrightarrow \mathbb{R}$ satisfy*

$$\left| f(t) - f_{(\mathbb{P}_\varepsilon,\mathbb{A}_\varepsilon)}(t) \right| < \varepsilon \textit{ for all } t \in [a,b].$$

2. *for all $k = 1, ..., n$,*

$$F_k = f_{(\mathbb{P},\mathbb{A})}(p_k) \tag{5}$$
$$f(p_k) = F_k + O(h^2) \textit{ as } h \longrightarrow 0 \tag{6}$$

where $h = max\{p_{k+1} - p_k; k = 1, ..., n-1\}$.

3. *the direct and inverse F-transforms are linear, i.e., for any $\lambda \in \mathbb{R}$ and for any two continuous functions $f, g : [a,b] \longrightarrow \mathbb{R}$, with direct F-transforms $\mathbf{F}_{(\mathbb{P},\mathbb{A})}$ and $\mathbf{G}_{(\mathbb{P},\mathbb{A})}$ with respect to the same partition $(\mathbb{P},\mathbb{A})$; then*

   3.1. *the direct F-transforms of λf and $f+g$ are, respectively, $\lambda\mathbf{F}_{(\mathbb{P},\mathbb{A})}$ and $\mathbf{F}_{(\mathbb{P},\mathbb{A})} + \mathbf{G}_{(\mathbb{P},\mathbb{A})}$,*
   3.2. *the inverse F-transform of λf and $f+g$ are, respectively,$\lambda f_{(\mathbb{P},\mathbb{A})}$ and $f_{(\mathbb{P},\mathbb{A})} + g_{(\mathbb{P},\mathbb{A})}$.*

In [10], the following property has been established:

Proposition 2. *Let $f, g : [a,b] \longrightarrow \mathbb{R}$ be continuous functions and let $(\mathbb{P},\mathbb{A})$ be a fuzzy partition of $[a,b]$. Then*

(i) *the iF-transform satisfies*

$$\int_a^b f_{(\mathbb{P},\mathbb{A})}(t)dt = \int_a^b f(t)dt \tag{7}$$

(ii) *if f ang g have the same direct F-transform components $\mathbf{F}_{(\mathbb{P},\mathbb{A})} = \mathbf{G}_{(\mathbb{P},\mathbb{A})}$, then*

$$\int_a^b f(t)\,dt = \int_a^b g(t)\,dt.$$

It is interesting to observe that a fuzzy partition has a “nesting” property (its proof is immediate):

Proposition 3. *Let $(\mathbb{P}, \mathbb{A})$ be a fuzzy partition of interval $[a, b]$ with $\mathbb{P} = \{a = p_1 < p_2 < ... < p_n = b\}$ and consider any subinterval $[a, p_k]$ with $k = 2, ..., n$; let $(\mathbb{P}_k, \mathbb{A}_k)$ be the fuzzy partition of $[a, p_k]$ defined by*

$$\begin{aligned} \mathbb{P}_k &= \{a, p_2, ..., p_k\} \\ \mathbb{A}_k &= \{A_1, ..., \widetilde{A}_k\} \end{aligned} \tag{8}$$

where the last basic function $\widetilde{A}_k$ is given by the restriction of the basic function A_k to the subinterval $[p_{k-1}, p_k]$. The fuzzy partition $(\mathbb{P}_k, \mathbb{A}_k)$, $k = 1, 2, ..., n$ is called the k-th nested partition associated with $(\mathbb{P}, \mathbb{A})$ and clearly $(\mathbb{P}_n, \mathbb{A}_n) = (\mathbb{P}, \mathbb{A})$.

A consequences of the properties above is the following proposition:

Proposition 4. *Let $f : [a, b] \longrightarrow \mathbb{R}$ be a continuous function and let $\mathbf{F}_{(\mathbb{P},\mathbb{A})} = (F_1, F_2, ..., F_n)^T$ be its F-transform with respect to $(\mathbb{P}, \mathbb{A})$. Then, for any $k = 2, ..., n - 1$, the F-transform of the restriction of f to the subinterval $[a, p_k]$ (with respect to the k-th nested partition $(\mathbb{P}_k, \mathbb{A}_k)$), is given by $\mathbf{F}^{(k)} = (F_1, ..., F_{k-1}, \widetilde{F}_k)^T$ where only the last component $\widetilde{F}_k$ is changed with respect to the components in $\mathbf{F}_{(\mathbb{P},\mathbb{A})}$ and is given by*

$$\widetilde{F}_k = \frac{\int_a^{p_k} f(t) A_k(t) dt}{\int_a^{p_k} A_k\,(t)\,dt}. \tag{9}$$

We have, for all $k = 2, ..., n - 1$ (the central summation is assumed to be zero if $k = 2$),

$$f_{(\mathbb{P}_k,\mathbb{A}_k)}(t) = F_1 A_1(t) + \sum_{j=2}^{k-1} F_j A_j(t) + \widetilde{F}_k \widetilde{A}_k(t)$$

and

$$\int_a^{p_k} f(t)dt = F_1 \int_a^{p_k} A_1\,(t)\,dt + \sum_{j=2}^{k-1} F_j \int_a^{p_{j+1}} A_j\,(t)\,dt + \widetilde{F}_k \int_a^{p_k} A_k\,(t)\,dt. \tag{10}$$

With the notation in (1), equality (10) is written (for $k > 1$) as

$$\int_a^{p_k} f(t)dt = F_1 I_1^+ + \sum_{j=2}^{k-1} F_j I_j + \widetilde{F}_k I_k^- \tag{11}$$

and we have:

Proposition 5. *Let $f : [a, b] \longrightarrow \mathbb{R}$ be a continuous function, let $(\mathbb{P}, \mathbb{A})$ be a uniform fuzzy partition of $[a, b]$ with $h = \frac{b-a}{n-1}$ and let $\mathbf{F}_{(\mathbb{P},\mathbb{A})} = (F_1, F_2, ..., F_n)^T$ be the F-transform of f; let also $\mathbf{F}^{(k)} = (F_1, ..., F_{k-1}, \widetilde{F}_k)^T$ be as in (9) for any $k = 2, ..., n - 1$. Then*

$$\begin{aligned} I_k^- \widetilde{F}_k &= \frac{h}{2} f(p_k) + O(h^3) \text{ as } h \longrightarrow 0 \\ I_k F_k &= h f(p_k) + O(h^3) \text{ as } h \longrightarrow 0. \end{aligned}$$

Proof. Consider $2 \le k \le n - 1$. We have $I_k^- \widetilde{F}_k = \int_{p_{k-1}}^{p_k} f(t) A_k(t) dt$ and, by the trapezoidal integration,

$$I_k^- \widetilde{F}_k = \frac{h}{2} [f(p_{k-1}) A_k(p_{k-1}) + f(p_k) A_k(p_k)] + O(h^3);$$

on the other hand, $A_k(p_k) = 1$ and $A_k(p_{k-1}) = 0$, so $I_k^- \widetilde{F}_k = \frac{h}{2} f(p_k) + O(h^3)$. For the second equality (consider that $A_k(p_{k+1}) = 0$) we have

$$\begin{aligned} I_k F_k &= I_k^- \widetilde{F}_k + \frac{h}{2} f(p_k) + O(h^3) \\ &= \frac{h}{2} f(p_k) + \frac{h}{2} f(p_k) + O(h^3) = h f(p_k) + O(h^3). \end{aligned}$$

□

3. Numerical Solution of Initial-Value ODE by F-Transform

Let us consider the following initial-value ordinary differential equation (ODE):

$$\begin{cases} \dot{x}(t) = f(t, x(t)), \ t \in [t_0, t_1] \\ x(t_0) = x_0 \end{cases} \tag{12}$$

We assume the usual requirements on function $f(t,x)$ that ensure existence and unicity of the solution $x(t)$, $t \in [t_0, t_1]$.

We are interested to find an approximation of the final value $x(t_1)$ of the solution $x(t)$. Let $(\mathbb{P}, \mathbb{A})$ be a fixed (but arbitrary) uniform fuzzy partition of $[t_0, t_1]$ with $p_1 = t_0$, $p_k = p_{k-1} + h$, $k = 2, ..., n$ and $h = \frac{t_1 - t_0}{n-1}$; let $\mathbb{A} = \{A_1, .., A_n\}$ be the basic functions. Let $\dot{x}_{(\mathbb{P},\mathbb{A})}$ denote the iF-transform of $\dot{x}(t)$ with (exact) direct F-transform components $(F_k)_{k=1,...,n}$.

In the rest of the paper, we will make use of the following functions and notation: for any basic function $A_k \in \mathbb{A}$, $k = 1, 2, ..., n$, we will denote by B_k the integral function defined by

$$B_k(t) = \int_{t_0}^{t} A_k(s)\, ds.$$

The components of the direct F-transform of $\dot{x}(t)$ will be denoted by $F_1, F_2, ..., F_n$ and the inverse F-transform of $\dot{x}_{(\mathbb{P},\mathbb{A})}$ of is the function

$$\dot{x}_{(\mathbb{P},\mathbb{A})}(t) = \sum_{k=1}^{n} F_k A_k(t) \text{ for } t \in [a, b]. \tag{13}$$

The following proposition follows immediately from (10).

Proposition 6. *If the exact values $(F_k)_{k=1,...,n}$ of the direct F-transform components of the function $t \longrightarrow f(t, x(t))$ on $[t_0, t_1]$ with respect to the fuzzy partition $(\mathbb{P}, \mathbb{A})$ are known, then the final value $x(t_1)$ of the solution of (12) is exactly given by*

$$x(t_1) = x_0 + \sum_{k=1}^{n} F_k I_k.$$

Furthermore, at the intermediate points p_k, $k = 2, ..., n-1$ of the decomposition $\mathbb{P}$, we have that the solution $x(p_k)$ is exactly given by

$$x(p_k) = x_0 + \sum_{j=1}^{k-1} F_j I_j + \widetilde{F}_k I_k^- \tag{14}$$

Proof. By definition, we have

$$\dot{x}_{(\mathbb{P},\mathbb{A})}(t) = \sum_{k=1}^{n} F_k A_k(t)$$

and, from property (ii) in Proposition 1, also

$$\int_{t_0}^{t_1} \dot{x}_{(\mathbb{P},\mathbb{A})}(t)\,dt = \int_{t_0}^{t_1} \dot{x}(t)\,dt;$$

we then obtain

$$x(t_1) = x_0 + \int_{t_0}^{t_1} \dot{x}_{(\mathbb{P},\mathbb{A})}(s)\,ds = x_0 + \sum_{k=1}^{n} F_k B_k(t_1).$$

By Proposition 2 applied to the derivative function $\dot{x}(t) = f(t, x(t))$, with $(\mathbb{P},\mathbb{A})$ on $[t_0, t_1]$, and using (1)–(2), we have $B_j(p_k) = I_j$ and $B_k(p_k) = I_k^-$ and the conclusion follows. □

Remark 1. *Consider that in general we have* $x(t) \neq x_0 + \int_{t_0}^{t} \dot{x}_{(\mathbb{P},\mathbb{A})}(s)ds$, *for* $t \notin \{p_1, ..., p_n\}$.

From the properties of F-transform as in Proposition 1 (point 3.) we finally have the following

Proposition 7. *If the exact values* $(F_k)_{k=1,...,n}$ *of the direct F-transform components of the function* $t \longrightarrow f(t, x(t))$ *on* $[t_0, t_1]$ *with respect to a fuzzy partition* $(\mathbb{P},\mathbb{A})$ *are known, then the solution* $x(t)$ *of (12), for all* $k = 1, ..., n-1$, *satisfies*

$$x(t) = x_{(\mathbb{P},\mathbb{A},x_0)}(p_k) + \int_{p_k}^{t} f(s, x(s))\,ds, \text{for all } t \in [p_k, p_{k+1}[,$$

where

$$x_{(\mathbb{P},\mathbb{A},x_0)}(t) = x_0 + \int_{t_0}^{t} \dot{x}_{(\mathbb{P},\mathbb{A})}(s)ds \tag{15}$$

Proof. Apply the identity $x(t) = x_0 + \int_{t_0}^{p_k} f(s, x(s))\,ds + \int_{p_k}^{t} f(s, x(s))\,ds$ and (14). □

3.1. An F-Transform Algorithm for ODE

In view of Propositions 6 and 7, we then need a way to compute or to approximate the direct F-transform components $(F_k)_{k=1,...,n}$ of $\dot{x}(t)$.

By definition, we have that (here $p_0 = p_1$ and $p_{n+1} = p_n$), for $k = 1, 2, ..., n$,

$$I_k F_k = \int_{p_{k-1}}^{p_{k+1}} \dot{x}(t)\,A_k(t)\,dt = \int_{p_{k-1}}^{p_{k+1}} f(t, x(t))A_k(t)\,dt$$
$$I_k^- \widetilde{F}_k = \int_{p_{k-1}}^{p_k} \dot{x}(t)\,A_k(t)\,dt = \int_{p_{k-1}}^{p_k} f(t, x(t))A_k(t)\,dt$$

and, from Proposition 5,

$$I_k F_k = hf(p_k, x(p_k)) + O(h^3)$$
$$I_k^- \widetilde{F}_k = \frac{h}{2} f(p_k, x(p_k)) + O(h^3).$$

As a final step, substitute $x(p_k) = x_0 + \sum_{j=1}^{k-1} F_j I_j + \widetilde{F}_k I_k^-$ and obtain, for $h \longrightarrow 0$,

$$I_k F_k = hf\left(p_k, x_0 + \sum_{j=1}^{k-1} F_j I_j + \widetilde{F}_k I_k^-\right) + O(h^3) \tag{16}$$

$$I_k^- \widetilde{F}_k = \frac{h}{2} f\left(p_k, x_0 + \sum_{j=1}^{k-1} F_j I_j + \widetilde{F}_k I_k^-\right) + O(h^3). \tag{17}$$

Approximated values G_k and $\widetilde{G}_k$ of F_k and $\widetilde{F}_k$, respectively, can be computed by solving the following equations (we will assume that the sums below will be zero if $k = 1$)

$$I_k G_k = hf\left(p_k, x_0 + \sum_{j=1}^{k-1} G_j I_j + \widetilde{G}_k I_k^-\right) \tag{18}$$

$$I_k^- \widetilde{G}_k = \frac{h}{2} f\left(p_k, x_0 + \sum_{j=1}^{k-1} G_j I_j + \widetilde{G}_k I_k^-\right). \tag{19}$$

To solve Equations (18) and (19) let us write them for different values of $k = 1, ..., n$. The first component G_1 can be determined from the initial condition $\dot{x}(t_0) = f(t_0, x(t_0)) = f(p_1, x_0)$, obtained from $I_1 F_1 = hf(p_1, x(p_1)) + O(h^3)$ by ignoring the term $O(h^3)$:

$$G_1 = \frac{hf(t_0, x_0)}{I_1}. \tag{20}$$

When computing G_2 and $\widetilde{G}_2$, the value of G_1 is known, and Equations (18) and (19) become

$$I_2^- \widetilde{G}_2 = \frac{h}{2} f\left(p_2, x_0 + G_1 I_1 + \widetilde{G}_2 I_2^-\right) \tag{21}$$

$$I_2 G_2 = hf\left(p_2, x_0 + G_1 I_1 + \widetilde{G}_2 I_2^-\right). \tag{22}$$

From the first equation we determine $\widetilde{G}_2$ and, by substituting into the second, we compute $G_2 = 2I_2^- \widetilde{G}_2 / I_2$.

For a general k, the values $G_1, ..., G_{k-1}$ are known and we need to solve Equation (19) only for $\widetilde{G}_k$

$$I_k^- \widetilde{G}_k = \frac{h}{2} f\left(p_k, x_0 + \sum_{j=1}^{k-1} G_j I_j + \widetilde{G}_k I_k^-\right); \tag{23}$$

then we set $G_k = 2I_k^- \widetilde{G}_k / I_k$.

Summarizing, we determine the values $\widetilde{G}_k$ and G_k iteratively from (23) starting with (20). Each Equation (23) has the form of a fixed-point problem for $\widetilde{G}_k$

$$\widetilde{G}_k = \frac{h}{2I_k^-} f\left(p_k, x_0 + \sum_{j=1}^{k-1} G_j I_j + \widetilde{G}_k I_k^-\right)$$

and can be solved by any *zero finder* routine or, considering that the system is in (nonlinear) triangular form, by any iterative procedure.

We have the following approximation property for the solution of (12).

Theorem 1. *Let $\mathbb{G}_k = (G_1, ..., G_{k-1}, \widetilde{G}_k)$ be solutions of (20)–(23) and define*

$$x_{\mathbb{G}_k}(p_k) = x_0 + \sum_{j=1}^{k-1} G_j I_j + \widetilde{G}_k I_k^- \textit{ for } k = 1, 2, ..., n.$$

Then, for the solution $x(p_k)$ of (12) at the points p_k, $k = 1, 2, ..., n$, we have

$$x(p_k) = x_{\mathbb{G}_k}(p_k) + O(kh^3) \textit{ as } h \longrightarrow 0,$$
$$x(t_1) = x_{\mathbb{G}_n}(t_1) + (t_1 - t_0) O(h^2) \textit{ as } h \longrightarrow 0.$$

Proof. From (14), (24) and (25) we have

$$\begin{aligned}|x(p_k) - x_{\mathbb{G}_k}(p_k)| &\leq \sum_{j=1}^{k-1} |F_j - G_j| I_j + |\widetilde{F}_k - \widetilde{G}_k| I_k^- \\ &= \sum_{j=1}^{k} O(h^3) = kO(h^3) \text{ as } h \longrightarrow 0.\end{aligned}$$

From $h = \frac{t_1 - t_0}{n-1}$ we get $n = (t_1 - t_0)O(\frac{1}{h})$ and $nO(h^3) = (t_1 - t_0)O(\frac{1}{h})O(h^3) = (t_1 - t_0)O(h^2)$ as $h \longrightarrow 0$ and the conclusion follows. □

The last theorem allows development of the following algorithm for the numerical solution of ODE based on F-transform.

Algorithm ODE-FT: Find an approximated final value $x_{(\mathbb{P},\mathbb{A})}(t_1)$ of the ODE $\dot{x}(t) = f(t, x(t))$, $t \in [t_0, t_1]$ with initial condition $x(t_0) = x_0$.

Step 1. Choose a uniform fuzzy partition $(\mathbb{P}, \mathbb{A})$ of $[t_0, t_1]$ with n points $p_1 = t_0$, $p_k = p_{k-1} + h$, $k = 2, ..., n$ and $h = \frac{t_1 - t_0}{n-1}$; let I_k^-, I_k^+ and I_k as in (1)-(2).

Step 2. For $k = 1, 2, ..., n$, compute the solutions $G_1, ..., G_{k-1}, \widetilde{G}_k$ of Equation (23) and define

$$x_{(\mathbb{P},\mathbb{A})}(p_k) = x_0 + \sum_{j=1}^{k-1} G_j I_j + \widetilde{G}_k I_k^- \text{ for } k = 1, 2, ..., n.$$

Step 3. The final value $x_{(\mathbb{P},\mathbb{A})}(t_1)$ (corresponding to $p_n = t_1$) is the desired approximation of $x(t_1)$ with $x(t_1) - x_{(\mathbb{P},\mathbb{A})}(t_1) = (t_1 - t_0)O(h^2)$ as $h \longrightarrow 0$.

Theorem 1 ensures that the algorithm ODE-FT is (globally) convergent. Clearly, the convergence of the algorithm ODE-FT assumes that the solutions of Equation (23) are solved with high precision, independent on the number n of points p_k of the fuzzy partition; in practice, if the exact solutions $G_1, ..., G_{k-1}, \widetilde{G}_k$ are approximated and substituted in the algorithm by quantities $G_1^*, ..., G_{k-1}^*, \widetilde{G}_k^*$, it is required that they are such that $\left|G_j - G_j^*\right| < tol_j$ for small positive tolerances $tol_j < \varepsilon$ and fixed small $\varepsilon > 0$; in this case, taking into account that $I_j \leq 2h$, $I_j^- \leq h$ and consequently $\sum_{j=1}^{k-1} |G_j - G_j^*| I_j + |\widetilde{G}_k - \widetilde{G}_k^*| I_k^- \leq (2k-1)\varepsilon h = \dfrac{2k-1}{n-1}(t_1 - t_0)\varepsilon$, the precision of ODE-FT is such that

$$\begin{aligned}|x(p_k) - x_{\mathbb{G}_k^*}(p_k)| &\leq \sum_{j=1}^{k-1} |F_j - G_j| I_j + \sum_{j=1}^{k-1} |G_j - G_j^*| I_j + |\widetilde{F}_k - \widetilde{G}_k| I_k^- + |\widetilde{G}_k - \widetilde{G}_k^*| I_k^- \\ &= \sum_{j=1}^{k} O(h^3) + \sum_{j=1}^{k-1} |G_j - G_j^*| I_j + |\widetilde{G}_k - \widetilde{G}_k^*| I_k^- \text{ as } h \longrightarrow 0 \\ &\leq kO(h^3) + \frac{2k-1}{n-1}(t_1 - t_0)\varepsilon \text{ as } h \longrightarrow 0\end{aligned}$$

and, for $k = n$, $|x(p_n) - x_{\mathbb{G}_n^*}(p_n)| \leq (t_1 - t_0)O(h^2 + 2\varepsilon)$ as $h \longrightarrow 0$ and $n \longrightarrow \infty$. In the computations reported in this paper, we have solved the fixed-point problems either exactly (in the case of linear differential equations) or, in the nonlinear cases, with a tolerance $tol \leq 10^{-12}$ in the absolute difference between two successive iterates of the used equation solver.

3.2. Extension to Systems of ODEs

The extension of the described procedure to solve systems of ordinary differential equations with initial conditions in the form

$$\begin{cases} \dot{x}_i(t) = f_i(t, x_1(t), ..., x_n(t)), i = 1, 2, ..., d \\ x_i(t_0) = x_{i,0}, i = 1, 2, ..., d \\ t \in [t_0, t_1] \end{cases}$$

We can find an approximation of the final value of each function $x_i(t_1)$, $i = 1, 2, ..., d$, in terms of a fixed fuzzy partition $(\mathbb{P}, \mathbb{A})$ of $[t_0, t_1]$, e.g., $p_1 = t_0$, $p_k = p_{k-1} + h$, $k = 2, ..., n$ and $h = \frac{t_1 - t_0}{n-1}$ and basic functions $\mathbb{A} = \{A_1, .., A_n\}$. Let $\dot{x}_{i,(\mathbb{P},\mathbb{A})}$ denote the iF-transform of $\dot{x}_i\,(t)$ with direct F-transform components $(F_{i,k})_{k=1,...,n}$, then, the final value $x_i\,(t_1)$ can be obtained, in terms of the components $(F_{i,k})_{k=1,...,n}$ and the initial condition $x_{i,0}$:

$$x_{i,(\mathbb{P},\mathbb{A})}(t) = x_0 + \int_{t_0}^{t} \dot{x}_{i,(\mathbb{P},\mathbb{A})}(s)ds = x_0 + \sum_{k=1}^{n} F_{i,k} B_k\,(t)$$

where $B_k\,(t) = \int_{t_0}^{t} A_k(s)ds$, $k = 1, ..., n$.

The direct F-transform components of the functions $t \longrightarrow f_i\,(t, x_1\,(t)\,, ..., x_d\,(t))$ on $[t_0, t_1]$ with respect to a fuzzy partition $(\mathbb{P}, \mathbb{A})$ are given, in this case, by d simultaneous systems of equalities (here $p_0 = p_1$ and $p_{n+1} = p_n$), for $k = 1, 2, ..., n$,

$$I_k F_{i,k} = \int_{p_{k-1}}^{p_{k+1}} \dot{x}_i\,(t)\,A_k\,(t)\,dt = \int_{p_{k-1}}^{p_{k+1}} f_i(t, x_1\,(t)\,, ..., x_d\,(t)) A_k\,(t)\,dt$$
$$I_k^- \widetilde{F}_{i,k} = \int_{p_{k-1}}^{p_k} \dot{x}_i\,(t)\,A_k\,(t)\,dt = \int_{p_{k-1}}^{p_k} f_i(t, x_1\,(t)\,, ..., x_d\,(t)) A_k\,(t)\,dt$$

and, for all $i = 1, ..., d$ and $k = 1, ..., n$, we obtain, by substituting $x_i(p_k) = x_{i,0} + \sum_{j=1}^{k-1} F_{i,j} I_j + \widetilde{F}_{i,k} I_k^-$, for $h \longrightarrow 0$,

$$I_k F_{i,k} = hf\left(p_k, x_{1,0} + \sum_{j=1}^{k-1} F_{1,j} I_j + \widetilde{F}_{1,k} I_k^-, ..., x_{d,0} + \sum_{j=1}^{k-1} F_{d,j} I_j + \widetilde{F}_{d,k} I_k^-\right) + O(h^3) \tag{24}$$

$$I_k^- \widetilde{F}_{i,k} = \frac{h}{2} f\left(p_k, x_{1,0} + \sum_{j=1}^{k-1} F_{1,j} I_j + \widetilde{F}_{1,k} I_k^-, ..., x_{d,0} + \sum_{j=1}^{k-1} F_{d,j} I_j + \widetilde{F}_{d,k} I_k^-\right) + O(h^3). \tag{25}$$

Approximated values $G_{i,k}$ and $\widetilde{G}_{i,k}$ of $F_{i,k}$ and $\widetilde{F}_{i,k}$, respectively, are computed by solving the systems of d equations

$$I_k G_{i,k} = hf_i\left(p_k, x_{1,0} + \sum_{j=1}^{k-1} G_{1,j} I_j + \widetilde{G}_{1,k} I_k^-, ..., x_{d,0} + \sum_{j=1}^{k-1} G_{d,j} I_j + \widetilde{G}_{d,k} I_k^-\right) \tag{26}$$

$$I_k^- \widetilde{G}_{i,k} = \frac{h}{2} f_i\left(p_k, x_{1,0} + \sum_{j=1}^{k-1} G_{1,j} I_j + \widetilde{G}_{1,k} I_k^-, ..., x_{d,0} + \sum_{j=1}^{k-1} G_{d,j} I_j + \widetilde{G}_{d,k} I_k^-\right). \tag{27}$$

The first components $G_{i,1}$, from the initial condition $\dot{x}_i\,(t_0) = f_i(p_1, x_{1,0}, ..., x_{d,0})$, are

$$G_{i,1} = \frac{hf_i(t_0, x_{1,0}, ..., x_{d,0})}{I_1} \tag{28}$$

and, for $k \geq 2$, with the known values $G_{i,1}, ..., G_{i,k-1}$, we need to solve the system of d equations for $\widetilde{G}_{i,k}$

$$I_k^- \widetilde{G}_{i,k} = \frac{h}{2} f_i \left(p_k, x_{1,k} + \widetilde{G}_{1,k} I_k^-, ..., x_{d,k} + \widetilde{G}_{d,k} I_k^- \right), i = 1, ..., d \tag{29}$$

where we have denoted $x_{i,k} = x_{i,0} + \sum\limits_{j=1}^{k-1} G_{i,j} I_j$; finally, we set $G_{i,k} = 2 I_k^- \widetilde{G}_{i,k} / I_k$.

Each system of Equation (29) has the form of a fixed-point problem for $\widetilde{G}_{i,k}$ and can be solved by any *zero finder* routine or by an iterative procedure similar to the one already described, adapted for the simultaneous case:

$$\widetilde{G}_{i,k}^{(l+1)} = \frac{h}{2 I_k^-} f_i \left(p_k, x_{1,k} + \widetilde{G}_{1,k}^{(l)} I_k^-, ..., x_{d,k} + \widetilde{G}_{d,k}^{(l)} I_k^- \right), i = 1, ..., d.$$

If $G_{i,k}$, $i = 1, ..., d$ and $k = 1, ..., n$, are the solutions of (28) and (29) and we define

$$x_{i,\mathbb{G}_{i,k}} (p_k) = x_{i,0} + \sum_{j=1}^{k-1} G_{i,j} I_j + \widetilde{G}_{i,k} I_k^- \text{ for } k = 1, 2, ..., n, i = 1, ..., d,$$

then, for the solution $x_i(p_k)$ of the system of ODEs at the points p_k, $k = 1, 2, ..., n$, we have

$$x_i(p_k) = x_{i,\mathbb{G}_{i,k}} (p_k) + O(kh^3) \text{ as } h \longrightarrow 0,$$
$$x_i(t_1) = x_{i,\mathbb{G}_{i,n}} (t_1) + (t_1 - t_0) O(h^2) \text{ as } h \longrightarrow 0.$$

3.3. The Case of Linear ODE

When the ODE is linear, with $f(t, x(t)) = a(t)x(t) + b(t)$, then the algorithm ODE-FT can be simplified considerably, because Equation (23) for $\widetilde{G}_k$ is

$$I_k^- \widetilde{G}_k = \frac{h}{2} \left(a(p_k) \left(x_0 + \sum_{j=1}^{k-1} G_j I_j + \widetilde{G}_k I_k^- \right) + b(p_k) \right) \tag{30}$$

and one obtains G_1 from $I_1 G_1 = h\left(a(p_1)x_0 + b(p_1)\right)$ and, for $k > 1$,

$$\widetilde{G}_k = \frac{a(p_k) \left(x_0 + \sum\limits_{j=1}^{k-1} G_j I_j \right) + b(p_k)}{I_k^- \left(\frac{2}{h} - a(p_k) \right)}; \tag{31}$$

remark that the denominator can be controlled to be non-zero by choosing an appropriate (small) value of h; in particular when $a(t)$ has positive values, in order to have $\left(\frac{2}{h} - a(p_k) \right) > 0$ it is sufficient to choose h (or, equivalently, n) such that $a(p_k) < \frac{2}{h}$ for all $k = 1, 2, ..., n$.

4. Computational Results for ODEs

The algorithm ODE-FT described in Section 2 has been implemented in MATLAB and used to solve a series of d-dimensional ODEs with $d = 1, 2, 3$ and 4. For some of them the exact solution $x_j(t)$, $j = 1, ..., d$, is available and the performance of our algorithm is evaluated by comparing the found solutions $x_{j,(\mathbb{P},\mathbb{A})}(t)$ and $x_j(t)$ at a number M of points $t \in \{t_1, t_2, ..., t_M\}$; for the problems without exact solutions, the comparison is performed with the solution $x_{j,RK}(t)$ found using the well-known Runge–Kutta–Fehlberg (RK) combination of fourth- and fifth-order approximation algorithm available in MATLAB function *ode45*, one of the possibly best implementations of the explicit RK method with variable step-size adaptation.

The first example proposes a simple (non-trivial) calculation of the *erf* function (see [24], Problem P6-1)

$$x(t) = erf(t) = \frac{2}{\sqrt{\pi}} \int_0^t e^{-s^2} ds, t \geq 0$$

at points $t_i = (i-1)/100, i = 1, ..., 101$ by solving the ODE (anti derivative problem)

$$\begin{aligned} \dot{x}(t) &= \frac{2}{\sqrt{\pi}} e^{-t^2}, t \in [0,1] \\ x(0) &= 0. \end{aligned} \tag{32}$$

Our solution $\widehat{x}_i$ at points t_i, that approximates $x(t_i)$, is found by applying algorithm ODE-FT successively on each subinterval $[t_i, t_{i+1}]$, $i = 1, ..., 100$, starting with the first interval $[t_1, t_2] = [0, \frac{1}{100}]$ (initial condition $\widehat{x}_1 = x(t_1) = 0$ to find $\widehat{x}_2 \simeq x(t_2)$) and continuing by solving (32) on subsequent subintervals $[t_i, t_{i+1}]$, $i = 2, ..., 100$, with the initial condition $x(t_i) = \widehat{x}_i$. On each subinterval, algorithm ODE-FT is applied with a uniform fuzzy partition $(\mathbb{P}, \mathbb{A})$, with $n = 101$ nodes and triangular basic functions A_k. We remark that with the decomposition of interval $[0,1]$ into 101 points and a fuzzy partition of each subinterval $[t_i, t_{i+1}]$, $i = 1, ..., 100$, with $n = 101$ nodes, the resulting (fixed) step size for solving (32) on $[0,1]$ is $h = 1.0 \times 10^{-4}$ (and $h^2 = 1.0 \times 10^{-8}$).

To compare the found solution with the values obtained using routine erf() in MATLAB, we found the average absolute error $\frac{1}{101} \sum_{i=1}^{101} |\widehat{x}_i - \text{erf}(t_i)| = 5.9191 \times 10^{-10} < h^2$; at the final point $t = 1$, we have $\text{erf}(1) = 8.427007929497149 \times 10^{-1}$ and $\widehat{x}_{101} = 8.427007922578713 \times 10^{-1}$ with error $\widehat{x}_{101} - \text{erf}(1) = -6.9184 \times 10^{-10}$.

To test the validity of the ODE-FT algorithm, we have performed two numerical experiments on two series of ODEs.

For the first experiment, we have chosen five non-trivial problems with known exact solution; this allows comparison of the one obtained by ODE-FT with the exact one.

In the second experiment we solve other five typical well-known ODEs and we compare our solutions with the ones obtained by well-established routines in MATLAB, like **ode45** (based on explicit single step Runge–Kutta method of orders 4 and 5 with step-size control), **ode113** (a variable-step and variable-order method based on Adams–Bashforth–Moulton discretizations of orders 1 to 13) or **ode15i** (an implicit variable-step and variable-order solver of orders 1 to 5, particularly efficient for stiff problems).

4.1. ODEs with Known Exact Solutions

The five ODEs for the first experiment are denoted **Ex1** (single equation, $d = 1$) and **Ex2**, **Ex3**, **Ex4**, **Ex5** (systems of two equations, $d = 2$).

For all the examples, we provide the following MATLAB output:

- MFT = ... number M of points t_i where the solution is computed;
- N = Number of nodes in the partition $(\mathbb{P}, \mathbb{A})$;
- hstep = ... step size h used on each subinterval $[t_i, t_{i+1}]$, $i = 1, ..., MFT-1$;
- FunctionEvaluations = total number of function evaluations required by algorithm ODE-FT.
- Final FT value $x(j)$ = ...(final value found for $\widehat{x}_j(t)$ at $t = t_{MFT}$), $j = 1, ..., d$;
- AveErrFT = Average absolute error $\frac{1}{MFT} \sum_{i=1}^{MFT} |\widehat{x}_{j,i} - x_j(t_i)|$, $j = 1, ..., d$;
- MaxErrFT = Maximum absolute error $max_{i=1,...,MFT} |\widehat{x}_{j,i} - x_j(t_i)|$, $j = 1, ..., d$;
- Final Exact value $x(j)$ = ...(final value of $x_j(t)$ at $t = t_{MFT}$, $j = 1, ..., d$;
- Error in Final value $Errx(j)$ = $(\widehat{x}_j(t_{MFT}) - x_j(t_{MFT}), j : 1, ..., d$)

Problem Ex1 (Logistic equation with unitary carrying capacity): Solution interval is $t \in [0,1]$;

$$\begin{cases} \dot{x} = rx(1-x) \\ x(0) = x_0. \end{cases}$$

The exact solution is $x_1(t) = \frac{1}{1+\left(\frac{1}{x_0}-1\right)e^{-rt}}$ $(x_0 = 0.1, r = 3)$.

The first run with MFT = 101 and N = 2 (corresponding step size $h = 0.01$) produced the following results:

AveErrFT = 3.131431926296436 $\times 10^{-6}$, MaxErrFT = 8.708952653480040 $\times 10^{-6}$
Final FT: $x(1)$ = 6.905591487503622 $\times 10^{-1}$,
Final Exact: $x(1)$ = 6.905678577030157 $\times 10^{-1}$,
Final Error: Errx(1) = -8.708952653480040 $\times 10^{-6}$.

The second run uses MFT = 101 and N = 41, corresponding to a step size $h = 0.0025$, produces the following better result, with a final error of the order 5 $\times 10^{-9}$:

MFT = 101, N = 41, hstep = 2.5 $\times 10^{-4}$, FunctionEvaluations = 14,798.
AveErrFT = 1.957156456839409 $\times 10^{-9}$, MaxErrFT = 5.443002715210810 $\times 10^{-9}$.
Final FT: $x(1)$ = 6.905678522600129 $\times 10^{-1}$,
Final Exact: $x(1)$ = 6.905678577030157 $\times 10^{-1}$ and
Final Error: Errx(1) = -5.443002715210810 $\times 10^{-9}$.

Problem Ex2: (non-autonomous nonlinear system): Solution interval is $t \in [0, 1]$;

$$\begin{cases} \dot{x}_1 = x_1 - t\cos(t)(x_2+1) + \sin(t) \\ \dot{x}_2 = x_1^2 + \sec^2(t) - t^2\sin^2(t) \\ x_1(0) = 0 \\ x_2(0) = 0. \end{cases}$$

The exact solution is $\begin{cases} x_1(t) = t\sin(t) \\ x_2(t) = \tan(t) \end{cases}$.

The solution is first computed using MFT = 101, N = 41, hstep = 2.50 $\times 10^{-4}$, FunctionEvaluations = 15,988, with the following results:

AveErrFT = 1.0 $\times 10^{-7}$ * (0.057812917960167, 0.117536307911936),
MaxErrFT = 1.0 $\times 10^{-7}$ * (0.208599681972288, 0.491491372045516),
Final FT: $x(1)$ = 8.414709639479283 $\times 10^{-1}$, $x(2)$ = 1.557407773804039 $\times 10^{0}$,
Final Exact: $x(1)$ = 8.414709848078965 $\times 10^{-1}$, $x(2)$ = 1.557407724654902 $\times 10^{0}$,
Final Error: Errx(1) = $-$2.085996819722880 $\times 10^{-8}$, Errx(2) = 4.914913720455161 $\times 10^{-8}$

If the step size is reduced to $h = 0.00001$ the error decreases significantly,

MFT = 1001, N = 101, hstep = 1.0 $\times 10^{-5}$, FunctionEvaluations = 301,000.
AveErrFT = 1.0$\times 10^{-10}$ * (0.091807074367543, 0.186150663977414),
MaxErrFT = 1.0 $\times 10^{-10}$ * (0.333676419828066, 0.786386511464343),
Final FT: $x(1)$ = 8.414709847745289 $\times 10^{-1}$, $x(2)$ = 1.557407724733541 $\times 10^{0}$.

Problem Ex3 (stiff, linear, see [24], Section 6.5): Solution interval is $t \in [0, \frac{1}{2}]$;

$$\begin{cases} \dot{x}_1 = 998x_1 + 1998x_2 \\ \dot{x}_2 = -999x_1 - 1999x_2 \\ x_1(0) = 1 \\ x_2(0) = 1. \end{cases}$$

The exact solution is $\begin{cases} x_1(t) = 4e^{-t} - 3e^{-1000t} \\ x_2(t) = -2e^{-t} + 3e^{-1000t} \end{cases}$.

The first run uses MFT = 51, N = 11, hstep = 0.001 and FunctionEvaluations = 550, with:

AveErrFT = 1.0 $\times 10^{-5}$ * (0.173411281358034, 0.170428606844124),
MaxErrFT = 1.0 $\times 10^{-4}$ * (0.853912256957301, 0.853928757786893),
Final FT: $x(1)$ = 2.426122537762081 $\times 10^{-1}$, $x(2)$ = -1.213061268881040 $\times 10^{0}$,
Final Exact: $x(1)$ = 2.426122638850534 $\times 10^{0}$, $x(2)$ = -1.213061319425267 $\times 10^{0}$ and
Final Error: Errx(1) = -1.010884531638112 $\times 10^{-7}$, Errx(2) = 5.054422635986100 $\times 10^{-8}$.

By reducing the step size, i.e., MFT = 501, N = 101, hstep = 0.00001, FunctionEvaluations = 50,500, we obtain:

AveErrFT = 1.0 $\times 10^{-7}$ * (0.459480472441389, 0.459450580577288),
MaxErrFT = 1.0 $\times 10^{-5}$ * (0.919708563218436, 0.919708564739441),
Final FT: $x(1) = 2.426122638840430 \times 10^{-0}$, $x(2) = -1.213061319420215 \times 10^{0}$,
Final Exact: $x(1) = 2.426122638850534 \times 10^{0}$, $x(2) = -1.213061319425267 \times 10^{0}$ and
Final Error: Errx(1) = $-1.010391770250862 \times 10^{-11}$, Errx(2) = $5.051958851254312 \times 10^{-12}$.

Problem Ex4: (Lotka–Volterra system, see, e.g., [14]) Solution interval is $t \in [0, 1]$;

$$\begin{cases} \dot{x}_1 = (4 + \tan(t))x_1 - e^{2t}x_1x_2 \\ \dot{x}_2 = \cos(t)x_1x_2 - 2x_2 \\ x_1(0) = -4 \\ x_2(0) = 4. \end{cases}$$

The exact solution is $\begin{cases} x_1(t) = \frac{-4}{\cos(t)} \\ x_2(t) = 4e^{-2t} \end{cases}$.

In this case, the first run with MFT = 21, N = 6, hstep = 0.01, FunctionEvaluations = 820, produced:

AveErrFT = 1.0 $\times 10^{-3} * (0.788752989084725, 0.115131629788532)$,
MaxErrFT = (0.004132636807097, 0.000295217766130),
Final FT: $x(1) = -7.407395507530799 \times 10^{0}, x(2) = 5.410459151803206 \times 10^{-1}$,
Final Exact: $x(1) = -7.403262870723702 \times 10^{0}, x(2) = 5.413411329464508 \times 10^{-1}$,
Final Error: Errx(1) = $-4.132636807097079 \times 10^{-3}$, Errx(2) = $-2.952177661301736 \times 10^{-4}$.

while in the second run with MFT = 201, N = 41, hstep = 1.25 $\times 10^{-4}$, FunctionEvaluations = 32,200, the results are significantly better:

AveErrFT = 1.0 $\times 10^{-6} * (0.113447339005566, 0.017745086026901)$,
MaxErrFT = 1.0 $\times 10^{-6} * (0.645177737013114, 0.046113228457934)$,
Final FT: $x(1) = -7.403263515901439 \times 10^{0}, x(2) = 5.413410868332224 \times 10^{-1}$,
Final Exact: $x(1) = -7.403262870723702e \times 10^{0}, x(2) = 5.413411329464508 \times 10^{-1}$,
Final Error: Errx(1) = $-6.451777370131140 \times 10^{-7}$, Errx(2) = $-4.611322845793353 \times 10^{-8}$.

Problem Ex5: (linear, see [14]) Solution interval is $t \in [0, 1]$;

$$\begin{cases} \dot{x}_1 = x_1 - x_2 + 2t - t^2 - t^3 \\ \dot{x}_2 = x_1 + x_2 - 4t^2 + t^3 \\ x_1(0) = 1 \\ x_2(0) = 0. \end{cases}$$

The exact solution is $\begin{cases} x_1(t) = e^t \cos(t) + t^2 \\ x_2(t) = e^t \sin(t) - t^3 \end{cases}$.

The first run is executed with MFT = 11, N = 2, hstep = 0.1, FunctionEvaluations = 20 and produced a solution with

AveErrFT = (0.000949321158793, 0.003176265306333),
MaxErrFT = (0.001690735853275, 0.008258267736959),
Final FT: $x(1) = 2.467003204062610 \times 10^0, x(2) = 1.279097019441884 \times 10^0$,
Final Exact: $x(1) = 2.468693939915886 \times 10^0, x(2) = 1.287355287178843 \times 10^0$,
Final Error: Errx(1) = $-1.690735853275172 \times 10^{-3}$, Errx(2) = $-8.258267736958924 \times 10^{-3}$.

By reducing the step size to 0.01 (MFT = 11, N = 11, FunctionEvaluations = 110), the errors reduce to AveErrFT = $1.0 \times 10^{-4} * (0.095673985461337, 0.317351098208354)$, MaxErrFT = $1.0 \times 10^{-4} * (0.171320181454604, 0.825459662427974)$.

Finally, using MFT = 101, N = 201, hstep = 5.0000×10^{-5}, FunctionEvaluations = 20,100, the result is

Final FT: $x(1) = 2.468693939487529 \times 10^0, x(2) = 1.287355285115205 \times 10^0$,
Final Exact: $x(1) = 2.468693939915886 \times 10^0$, $x(2) = 1.287355287178843 \times 10^0$, with
AveErrFT = $1.0 \times 10^{-9} * (0.241829418976129, 0.769468268633189)$,
MaxErrFT = $1.0 \times 10^{-8} * (0.042918868459196, 0.206363814925226)$.

4.2. *ODEs without Known Closed-Form Solution*

The five ODEs for the second experiment are nonlinear d-dimensional systems denoted **No1** ($d = 3$), **No2** ($d = 2$), **No3** ($d = 3$), **No4** ($d = 3$) and **No5** ($d = 4$).

These ODEs are generally considered to be numerically difficult to solve and are frequently used to test ODE solvers, in comparison with existing well-performing procedures. We have executed algorithm ODE-FT using MATLAB in parallel with well-established routines in MATLAB, in particular with routine *ode45* (based on explicit single step Runge–Kutta method of orders 4 and 5 with step-size control) and, in one case, with routines *ode113* (a variable-step and variable-order method based on Adams–Bashforth–Moulton discretization of orders 1 to 13) and *ode15i* (an implicit variable-step and variable-order solver of orders 1 to 5, particularly efficient for stiff problems).

Now, as we have explained in Section 2, the performance of algorithm ODE-FT depends essentially on the step size h resulting from the two parameters MFT and N; on the other hand, our simple implementation of ODE-FT does not consider variable step size, designed to control adaptively the local discretization and/or approximation errors. For this reason, we have first executed routine *ode45* to compute the solution at an appropriate number M of points $t_i, i = 1, 2, ..., M$ and, from the output of *ode45*, we have computed the number N of nodes for the fuzzy partition in such a way that h_{FT} is of the same order as the optimal step size h_{ode45} determined by *ode45*. In this way, the two routines produce solutions by using a step size of the same magnitude and the comparison does not depend on this element; for example, suppose that in solving an ODE on time domain $[a, b]$ at M uniform points t_i, $i = 1, 2, ..., M$, routine *ode45* returns, e.g., an optimal step size h_{ode45}, we then execute algorithm ODE-FT with MFT = M and with N$\sim 1 + \dfrac{b-a}{(M-1)h_{ode45}}$ so that $h_{ode45} \sim h_{FT} = \dfrac{b-a}{(M-1)(N-1)}$.

We remark that to determine the step size h_{FT} for ODE-FT, we can freely play with the two parameters MFT and N; in our experiment, in order to balance the number of Equation (29) to solve and the memory requirements, we have limited values of N between 11 and 501, so determining M to obtain the required step size, eventually by increasing M to take N in the range above.

Problem No1: Solution interval is $t \in [0, 5\pi]$;

$$\begin{cases} \dot{x}_1 = x_2 x_3 & x_1(0) = 0 \\ \dot{x}_2 = -x_1 x_3 & x_2(0) = 1 \\ \dot{x}_3 = -0.51 x_1 x_2 & x_3(0) = 1 \end{cases}$$

Routine *ode45* solves this ODE with optimal step size $h_{ode45} = 3.93 \times 10^{-5}$ and final solution vector $x(1) = 6.946876684 \times 10^{-1}, x(2) = 7.193115065 \times 10^{-1}, x(3) = 8.682618339 \times 10^{-1}$.

We execute ODE-FT with MFT = 10,001, N = 21 so that hstep = 7.853981633974483 $\times 10^{-5}$, with final solution $x(1) = 6.946876670 \times 10^{-1}, x(2) = 7.193115081 \times 10^{-1}, x(3) = 8.682618346 \times 10^{-1}$.

At the MFT points t_i where the solution is approximated, the average and the maximum absolute differences between the FT solution and the RK solution for each $x(j)$ are:

AveDiffFTRK = 1.0 $\times$ $10^{-9} * (0.670804140961247, 0.565236806318909, 0.252472631608888)$,
MaxDiffFTRK = 1.0 $\times$ $10^{-8} * (0.208394865253148, 0.161572011325717, 0.078973372286129)$,

See Figure 1 for the graphical representation of the found solutions. For this example, Figure 2 pictures the differences $FT(t) - RK(t)$ between the FT solution and the RK solution for the three components $x_j(t), j = 1, 2, 3$ at the MFT points. It is interesting to observe that the differences oscillate in sign, showing that in absolute value the differences are not systematically diverging.

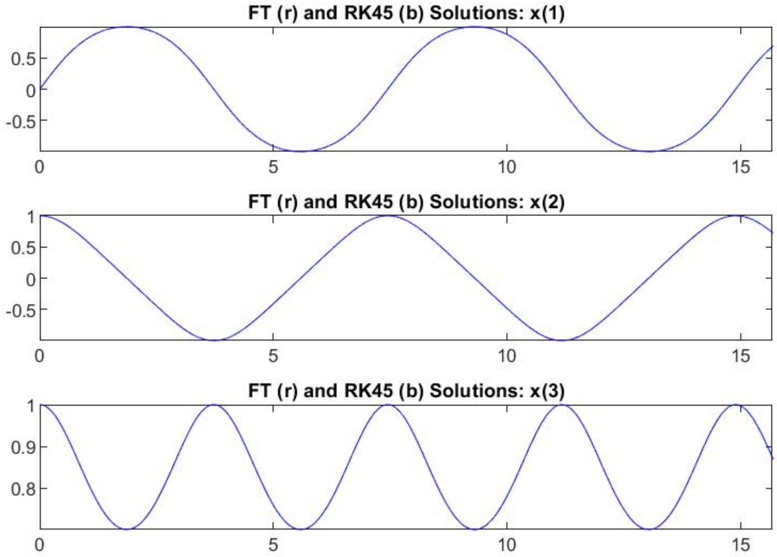

Figure 1. Problem No1: The three components $x_j(t), j = 1, 2, 3$ are displayed in the order from top to bottom; FT solution is red-colored, RK solution is blue-colored. The two solutions are highly coincident at all points.

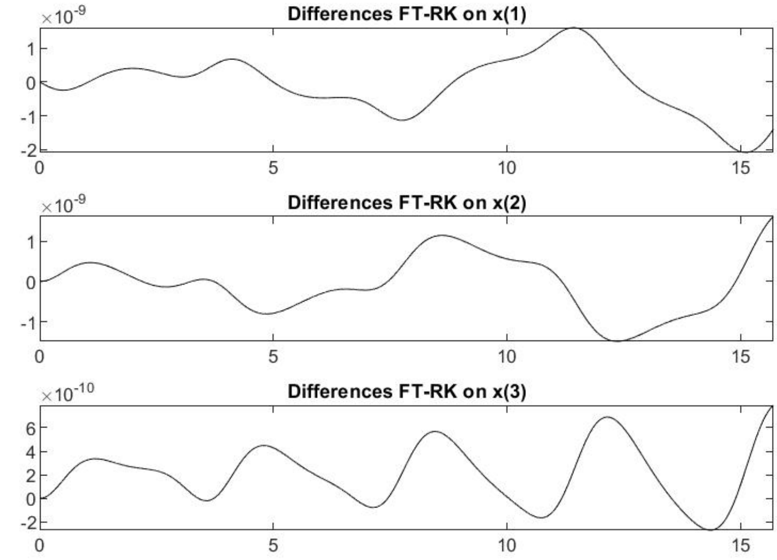

Figure 2. Problem No1: Differences $FT(t) - RK(t)$ between the FT solution and the RK solution for the three components $x_j(t), j = 1, 2, 3$. The two solutions coincide up to precision 10^{-9}.

Problem No2: (*Van der Pol* equations) Solution interval is $t \in [0, 300]$;

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \mu(1 - x_1^2)x_2 - x_1 + a\sin(\omega t) \\ x_1(0) = 2 \\ x_2(0) = 0. \end{cases}$$

The parameters are $\mu = 50, a = 3, \omega = \frac{\pi}{5}$.

This system is considered to be a hard ODE and requires very small step size to determine points where the solution changes suddenly (see Figure 3). To capture this phenomenon, we use M = 30,001. The solution by *ode45* has optimal step size $h_{ode45} = 2.196 \times 10^{-5}$ and final solution vector $x(1) = 1.700147566 \times 10^0, x(2) = -1.819270275 \times 10^{-2}$.

With MFT = 30,001, we choose N = 501 to have hstep = 2.0 $\times 10^{-5}$ and ODE-FT finds the final value $x(1) = 1.700150689 \times 10^0, x(2) = -1.819263248 \times 10^{-2}$. The comparison gives AveDiffFTRK = 1.398228278980211 $\times 10^{-4}$ and MaxDiffFTRK = 0.397462807473403.

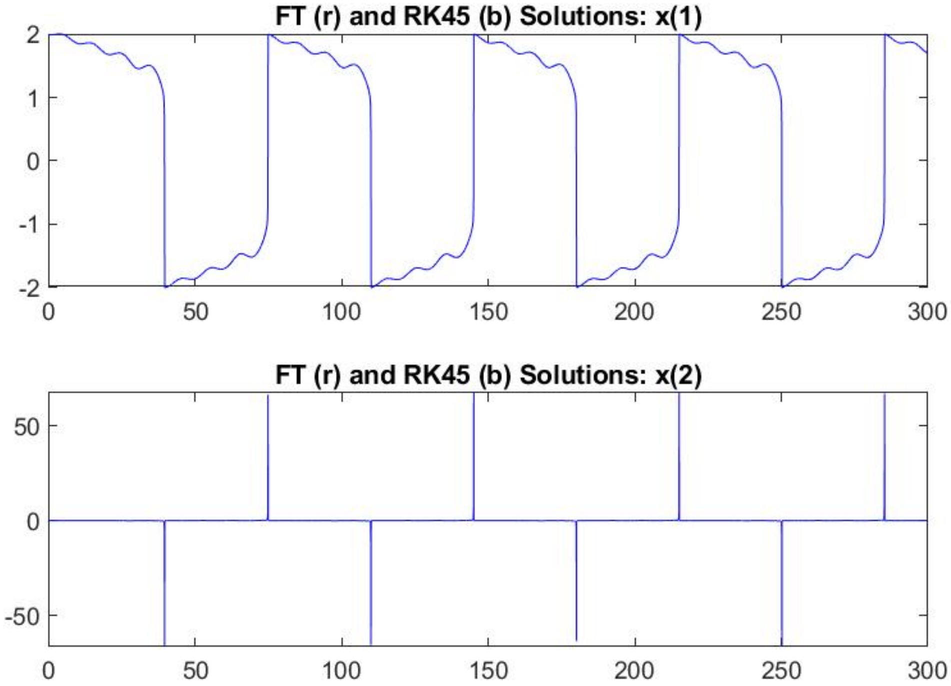

Figure 3. Problem No2 (Van der Pol): The two components $x_j(t), j = 1, 2$ are displayed in the order from top to bottom; FT solution is red-colored, RK solution is blue-colored. Remark the eight jumps of the solution from positive to negative values or vice versa.

Problem No3: (*Rössler's* equations) Solution interval is $t \in [0, 20]$;

$$\begin{cases} \dot{x}_1 = -x_1 - x_3 & x_1(0) = 1 \\ \dot{x}_2 = x_1 + \alpha x_2 & x_2(0) = 1 \\ \dot{x}_3 = \beta + x_3(x_1 - \gamma) & x_3(0) = 1 \end{cases}$$

The parameters are $\alpha = 0.2$, $\beta = 0.2$, $\gamma = 5.0$.

The third equation is nonlinear, but this problem is considered to be not numerically hard. The solution by *ode45* has optimal step size $h_{ode45} = 0.00144$ and final solution vector $x(1) = -3.722813228 \times 10^{-1}, x(2) = 7.646204266 \times 10^0, x(3) = 3.722813233 \times 10^{-1}$.

With MFT = 101, we choose N = 41 to have hstep = 0.005 and ODE-FT finds the final value $x(1) = -3.722813228 \times 10^{-1}, x(2) = 7.646166172 \times 10^{0}, x(3) = 3.722813233 \times 10^{-1}$. The comparison (see Figure 4) gives

AveDiffFTRK = 1.0 × $10^{-5} * (0.025127112474346, 0.947056407667963, 0.015362751349660)$,
MaxDiffFTRK = 1.0 × $10^{-4} * (0.030880110357123, 0.380942339370804, 0.039961137723310)$.

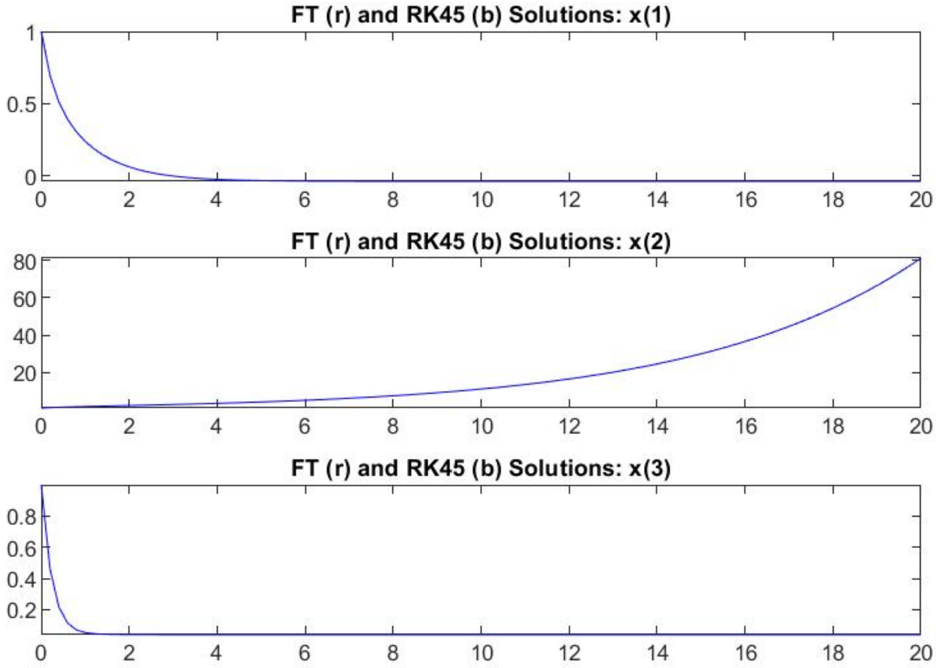

Figure 4. Problem No3 (Rössler system):The three components $x_j(t)$, $j = 1, 2, 3$ are displayed in the order from top to bottom; FT solution is red-colored, RK solution is blue-colored.

Problem No4: (*Lorenz's* system) Solution interval is $t \in [0, 100]$;

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) & x_1(0) = 10 \\ \dot{x}_2 = x_1(b - x_3) - x_1 & x_2(0) = 20 \\ \dot{x}_3 = x_1 x_2 - c x_3 & x_3(0) = 10 \end{cases}$$

The parameters are $a = 10$, $b = 28.0$, $c = \frac{8}{3}$.

It is well known that this nonlinear system is hard to solve numerically as it exhibits chaotic trajectories. Routine *ode45* solves the Lorenz system by optimal step size $h_{ode45} = 9.37 \times 10^{-7}$ and computes the final value $x(1) = -2.480991301578959 \times 10^{0}, x(2) = 4.324784533758371 \times 10^{-1}, x(3) = 2.470283374011438 \times 10^{1}$. Taking M = 30,001 the step size hstep = 1.0 ×10^{-6} is obtained with N = 1001; the found final value using ODE-FT is $x(1) = -2.453228212514948 \times 10^{0}, x(2) = 3.798179329138090 \times 10^{-1}, x(3) = 2.457087763696305 \times 10^{1}$ with AveDiffFTRK = (0.004740929429043, 0.006537238822312, 0.008150973471168) and MaxDiffFTRK = (0.149405722393993, 0.285676182819206, 0.328764820957794). See Figure 5 for the trajectories of components $x_j(t)$, $j = 1, 2, 3$ and Figure 6 for the 3D representations of the solutions.

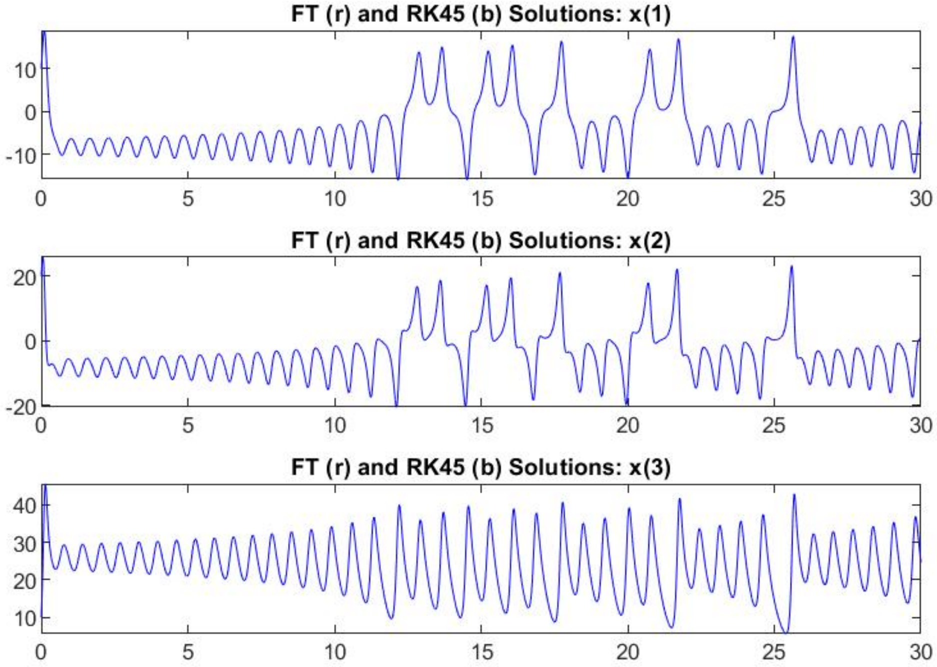

Figure 5. Problem No4 (Lorenz system): The three components $x_j(t)$, $j = 1, 2, 3$ are displayed in the order from top to bottom; FT solution is red-colored, RK solution is blue-colored. Remark that this system, with the given values of parameters, exhibits a strong sensitivity to changes in initial conditions and the numerical solutions are a famous example of chaotic trajectory.

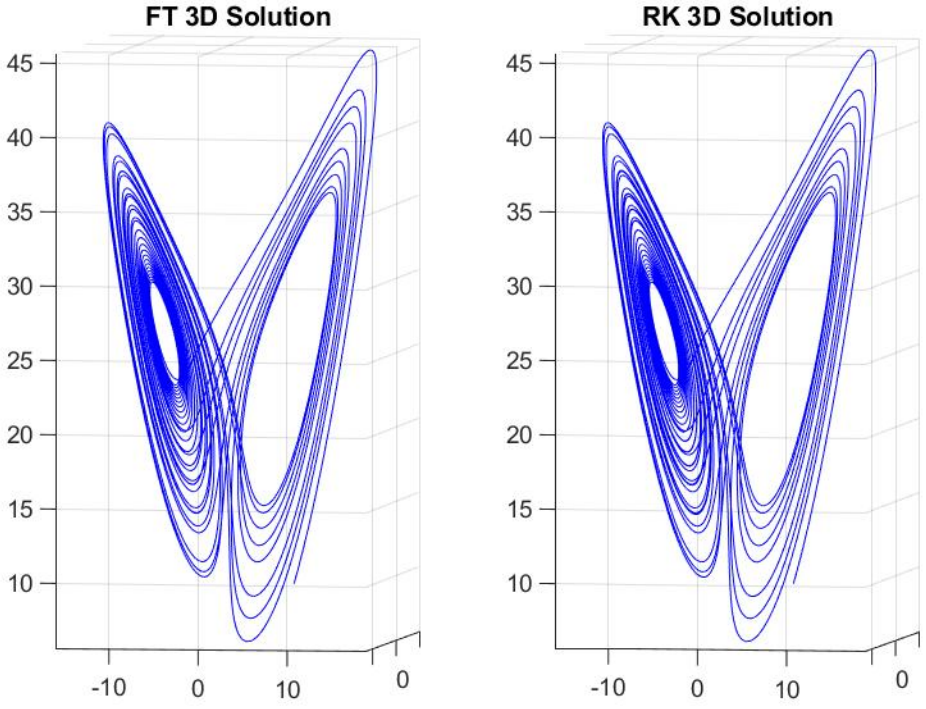

Figure 6. Problem No4 (Lorenz system): 3D representations of the solutions obtained by ODE-FT (**left**) and ode45 (**right**); they appear to be coincident to graphical precision.

We have also solved this system by a run of ODE-FT with MFT = 30,001, N = 101 and by running the two MATLAB routines *ode113* and *ode15i*; the resulting solutions are visualized in Figure 7 and we see that all the routines tend to generate trajectories with very different behavior for large values of time t.

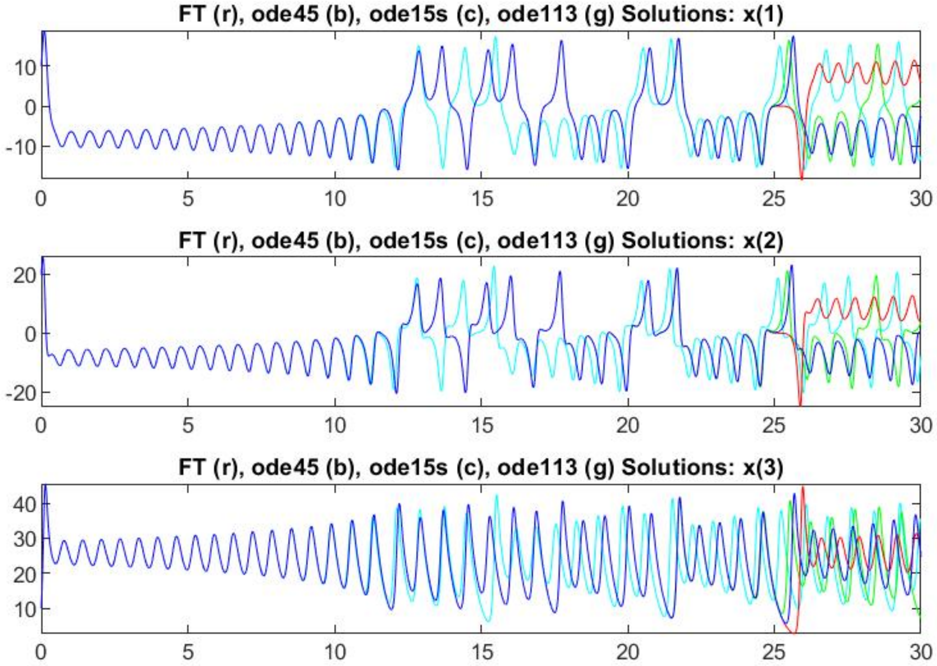

Figure 7. Problem No4 (Lorenz system): The three components $x_j(t)$, $j = 1,2,3$ are obtained also by routines ode15i (cyan color) and ode113 (green color); FT solution is red-colored, RK solution is blue-colored. Remark that differences between the solutions of this system are essentially due to very small differences in the solutions found by the different methods.

Problem No5: (Periodic system with period $T = 8$, see [24], Section 6.8): Solution interval is $t \in [0,8]$;

$$\begin{cases} \dot{x}_1 = x_3 & x_1(0) = 1-k \\ \dot{x}_2 = x_4 & x_2(0) = 0 \\ \dot{x}_3 = \frac{-\alpha^2 x_1}{\left(x_1^2+x_2^2\right)^{\frac{3}{2}}} & x_3(0) = 0 \\ \dot{x}_4 = \frac{-\alpha^2 x_2}{\left(x_1^2+x_2^2\right)^{\frac{3}{2}}} & x_4(0) = \alpha\left(\frac{1+k}{1-k}\right) \end{cases}$$

The parameters are $k = 0.25$, $\alpha = \frac{\pi}{4}\sqrt{\frac{1+k}{1-k}}$; we remark that from periodicity, the exact solution has $x_i(8) = x_i(0)$, $i = 1,...,4$.

The initial condition is $x(1) = 7.50 \times 10^{-1}, x(2) = 0.0, x(3) = 0.0, x(4) = 1.013944668993403$.

With MFT = 801, N = 201 and the step size hstep = 5.0 $\times 10^{-5}$, the final solution found by ODE-FT is $x(1) = 7.500000000000394 \times 10^{-1}, x(2) = -1.890855 \times 10^{-8}, x(3) = 1.923939 \times 10^{-8}, x(4) = 1.013944668993378$, and the one computed by *ode45* is $x(1) = 7.499999999999980 \times 10^{-1}, x(2) = -5.566185 \times 10^{-14}, x(3) = 5.123376 \times 10^{-14}, x(4) = 1.013944668993403$; comparatively (see Figure 8), we get

AveDiffFTRK = $1.0 \times 10^{-8} * (0.4273611, 0.3695634, 0.3303905, 0.3234898)$,
MaxDiffFTRK = $1.0 \times 10^{-7} * (0.1278607, 0.18908499, 0.19239342, 0.13127243)$.

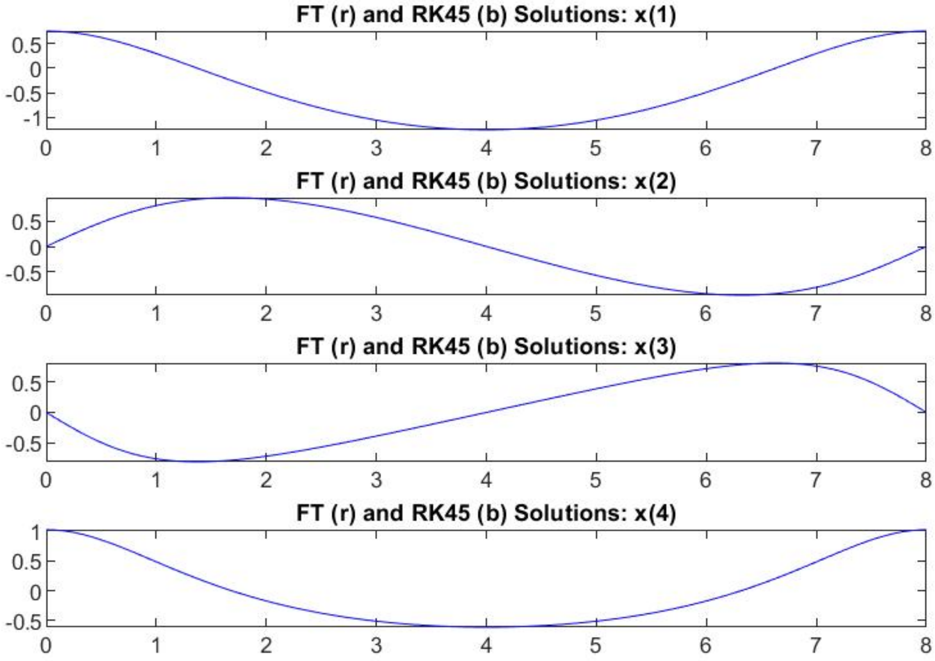

Figure 8. Problem No5 (Periodic system): The four components $x_j(t)$, $j = 1, 2, 3, 4$ are displayed in the order from top to bottom; FT solution is red-colored, RK solution is blue-colored and the two coincide at graphical precision.

5. Interval (Fuzzy) Differential Equations and F-Transform

In this section, we consider interval and fuzzy differential equations in the setting of generalized Hukuhara differentiability as described in [18,19,25]. A preliminary version of this section has been presented as a conference paper in [17].

Compact intervals of real numbers will be denoted by the usual endpoints notation $A = [a^-, a^+]$, $B = [b^-, b^+]$ or by the midpoint notation $A = (\widehat{a}; \widetilde{a})$, $B = (\widehat{b}; \widetilde{b})$ where $\widehat{a} = \frac{1}{2}(a^- + a^+)$ is the midpoint and $\widetilde{a} = \frac{1}{2}(a^+ - a^-)$ is the radius (half-length). The set of all compact real intervals will be denoted by $\mathcal{K}_C$.

The use of midpoint representation for intervals has been recently adopted to study several topics in the analysis of interval-valued functions, the single variable case (see [26,27] and the references therein), to which the interested reader is referred for a complete description of interval-valued generalized Hukuhara derivative (gH-derivative for short).

Given two intervals $A, B \in \mathcal{K}_C$, the gH-difference is the interval $C \in \mathcal{K}_C$ (it always exists and is unique) such that

$$A \ominus_{gH} B = C \Longleftrightarrow \begin{cases} & (i) \quad A = B + C \\ or & (ii) \quad B = A - C \end{cases} . \tag{33}$$

Using midpoint notation, we have $A \ominus_{gH} B = (\widehat{a} - \widehat{b}; |\widetilde{a} - \widetilde{b}|)$ and *(i)* is verified if $\widetilde{a} \geq \widetilde{b}$ and $C = (\widehat{a} - \widehat{b}; \widetilde{a} - \widetilde{b})$, (ii) is verified when $\widetilde{a} \leq \widetilde{b}$ and $C = (\widehat{a} - \widehat{b}; \widetilde{b} - \widetilde{a})$. If $\widetilde{a} = \widetilde{b}$ then clearly $A \ominus_{gH} B = (\widehat{a} - \widehat{b}; 0) = \{\widehat{a} - \widehat{b}\}$ is a singleton (real number).

Please note that if $A_i \ominus_{gH} B_i$ are gH-differences of the same type for all $i = 1, ..., n$, i.e., all satisfy either *(i)* or *(ii)* above, then (see [25])

$$\sum_{i=1}^{n} (A_i \ominus_{gH} B_i) = \left(\sum_{i=1}^{n} A_i \right) \ominus_{gH} \left(\sum_{i=1}^{n} B_i \right) . \tag{34}$$

An interval-valued function $F : [a,b] \to \mathcal{K}_C$ will be denoted by $F(t) = [f^-(t), f^+(t)]$ or, in equivalent midpoint notation, by $F(t) = (\widehat{f}(t); \widetilde{f}(t))$, for $t \in [a,b]$.

We will denote by $\mathbb{R}_\mathcal{F}$ the set of fuzzy numbers, i.e. normal, fuzzy convex, upper semi-continuous and compactly supported fuzzy sets defined over the real line $\mathbb{R}$. The α-level set of u (or simply its α-cut) is defined by $[u]_\alpha = \{x | x \in \mathbb{R}, u(x) \geq \alpha\}$ and, for $\alpha = 0$, it is the closure of the support $[u]_0 = cl\{x | x \in \mathbb{R}, u(x) > 0\}$.

A well-known result (see [21]) allows us to represent a fuzzy number as a pair $u = (u^-, u^+)$ of functions $u^-, u^+ : [0,1] \longrightarrow \mathbb{R}$, defining the endpoints of the α-cuts as $[u]_\alpha = [u_\alpha^-, u_\alpha^+] = (\widehat{u}_\alpha; \widetilde{u}_\alpha)$

We refer to functions $u_{(.)}^-$ and $u_{(.)}^+$ as the lower and upper branches of u, respectively; the two functions $\widehat{u}_{(.)}$, $\widetilde{u}_{(.)}$ are the (level-wise) midpoint and radius functions.

Given fuzzy numbers $u, v \in \mathbb{R}_\mathcal{F}$, the *level-wise* generalized Hukuhara difference (LgH-difference for short) is the family of intervals

$$u \ominus_{LgH} v = \left\{ [u]_\alpha \ominus_{gH} [v]_\alpha | \alpha \in [0,1] \right\}.$$

A fuzzy-valued function $F : [a,b] \to \mathbb{R}_\mathcal{F}$ will have α-cuts denoted by $[F(t)]_\alpha = [f_\alpha^-(t), f_\alpha^+(t)]$ or, equivalently, by $(\widehat{f}_\alpha(t); \widetilde{f}_\alpha(t))$, for $t \in [a,b]$. Remark that for each $\alpha \in [0,1]$, the functions $[F]_\alpha : [a,b] \to \mathcal{K}_C$ defined by $[F]_\alpha(t) = [F(t)]_\alpha$ are properly interval-valued functions and their family (sometimes called their bunch) $\{[F]_\alpha | \alpha \in [0,1]\}$ gives a unique and equivalent representation of F.

The following definition of generalized derivative can be applied both to interval-valued function or to the level-cuts of a fuzzy-valued function.

Definition 1 ([18]). *Let $t_0 \in]a,b[$ and h be such that $t_0 + h \in]a,b[$.*

- *If $F : [a,b] \to \mathcal{K}_C$ is an interval-valued function, its gH-derivative at t_0 is defined to be the limit, if it exists,*

$$F'_{gH}(t_0) = \lim_{h \to 0} \left(\frac{F(t_0 + h) \ominus_{gH} F(t_0)}{h} \right). \tag{35}$$

- *If $F :]a,b[\to \mathbb{R}_\mathcal{F}$ is a fuzzy-valued function, its level-wise gH-derivative (LgH-derivative for short) at t_0 is defined to be the family of the gH-derivatives of $[F]_\alpha$, if they exist for all $\alpha \in [0,1]$, i.e.,*

$$F'_{LgH}(t_0) = \left\{ ([F]_\alpha)'_{gH}(t_0) | \alpha \in [0,1] \right\} \text{ where} \tag{36}$$

$$([F]_\alpha)'_{gH}(t_0) = \lim_{h \to 0} \frac{1}{h} \left([F(t_0 + h)]_\alpha \ominus_{gH} [F(t_0)]_\alpha \right). \tag{37}$$

For an interval-valued function $F : [a,b] \to \mathcal{K}_C$, $F(t) = [f^-(t), f^+(t)]$, when $f^-(t)$ and $f^+(t)$ are both differentiable, we can distinguish two cases, corresponding to *(i)* and *(ii)* of (33) (see [18])

Definition 2. *Let $F : [a,b] \longrightarrow \mathcal{K}_C$ and $t_0 \in]a,b[$ with $f_\alpha(t)$ and $f_\alpha(t)$ both differentiable at t_0. We say that*

- *F is (i)-gH-differentiable at t_0 if*

$$(i.) \qquad F'_{gH}(t_0) = \left[\left(f^-\right)'(t_0), \left(f^+\right)'(t_0) \right] \tag{38}$$

- *F is (ii)-gH-differentiable at t_0 if*

$$(ii.) \qquad F'_{gH}(t_0) = [\left(f^+\right)'(t_0), \left(f^-\right)'(t_0)]. \tag{39}$$

If $F :]a,b[\to \mathbb{R}_\mathcal{F}$ is a fuzzy-valued function, we define analogously (i)-LgH and (ii)-LgH differentiability of F, with the additional requirement that (38) (or (39), respectively) are valid for all $[F]_\alpha$.

As in [18], we say that a point $t_0 \in]a, b[$ is an l-critical point of F if it is a critical point for the length function $len([F(t)]) = f^+(t) - f^-(t)$. A point $t_0 \in]a, b[$ is a switching point for the gH-differentiability of F, if in any neighborhood V of t_0 there exist points $t_1 < t_0 < t_2$ such that

type-I switch point): at t_1 (38) holds while (39) does not hold and at t_2 (39) holds and (38) does not hold, or

type-II switch point): at t_1 (39) holds while (38) does not hold and at t_2 (38) holds and (39) does not hold.

Analogous definitions can be given level-wise for a fuzzy-valued function.

5.1. Numerical Interval ODE by F-Transform

An interval differential equation (IDE) with initial condition can be written in the form

$$\begin{cases} \dot{x}_{gH}(t) = F(t, x(t)), \; t \in [t_0, t_1] \\ x(t_0) = x_0 \end{cases} \tag{40}$$

where $F(t, x(t)) = [F^-(t, x(t)), F^+(t, x(t))]$ and $x(t) = [x^-(t), x^+(t)]$ are intervals for all $t \in [t_0, t_1]$ and $x_0 = [x_0^-, x_0^+]$ is an interval initial value.

The gH-derivative of $x(t)$ is denoted by

$$\dot{x}_{gH}(t) = [\dot{x}^-_{gH}(t), \dot{x}^+_{gH}(t)].$$

We will approximate $\dot{x}_{gH}(t)$ by the iF-transforms of the two functions $\dot{x}^-_{gH}(t)$ and $\dot{x}^+_{gH}(t)$ on the same fuzzy partition $(\mathbb{P},\mathbb{A})$, i.e.,

$$(\dot{x}^-_{gH})_{(\mathbb{P},\mathbb{A})}(t) = \sum_{j=1}^{n} F_j^- A_j(t) \tag{41}$$

$$(\dot{x}^+_{gH})_{(\mathbb{P},\mathbb{A})}(t) = \sum_{j=1}^{n} F_j^+ A_j(t) \tag{42}$$

where for $j = 1, 2, ..., n$, using the same notation as in Section 2 with $[a, b] = [t_0, t_1]$, F_j^- and F_j^+ are the direct F-transforms of $(\dot{x}^-_{gH})$ and $(\dot{x}^+_{gH})$.

From the monotonicity properties of F-transform (see [4,10]), we know that $F_j^- \leq F_j^+$ (because $\dot{x}^-_{gH}(t) \leq \dot{x}^+_{gH}(t)$ for all t) and we can define the intervals $F_j = [F_j^-, F_j^+]$. Consequently, in terms of interval arithmetic operations, we also have the following approximation of the (interval-valued) inverse F-transform $(\dot{x}_{gH})_{(\mathbb{P},\mathbb{A})}(t)$ of $\dot{x}_{gH}$:

$$(\dot{x}_{gH})_{(\mathbb{P},\mathbb{A})}(t) = \sum_{j=1}^{n} F_j A_j(t).$$

On the other hand, the following function is well defined:

$$H(t) = \int_{t_0}^{t} \left(\sum_{j=1}^{n} F_j A_j(s)\, ds \right) = \sum_{j=1}^{n} F_j B_j(t), \, t \in [t_0, t_1] \tag{43}$$

and $H(t_0) = 0$ (here 0 stands for the interval $[0, 0]$). The interval-valued function $H(t)$ will play a central role in our method to solve the IDE (40).

For simplicity, define the following interval-valued function

$$\varphi(t) = \sum_{j=1}^{n} F_j A_j(t), t \in [t_0, t_1]; \tag{44}$$

Clearly, φ is a continuous interval-valued function and, from Theorem 43(i) in [18], the integral function $H(t) = \int\limits_{t_0}^{t} \varphi(s)ds$ is gH-differentiable with

$$H'_{gH}(t) = \varphi(t), H(t_0) = 0.$$

The following property is proved in [17].

Proposition 8. *Consider the function $H(t)$ in (43) and define the two interval-valued functions $\Phi(t)$ and $\Psi(t)$ on $[t_0, t_1]$ by*

$$\Phi(t) = H(t) \ominus_{gH} (-x_0) \tag{45}$$

$$\Psi(t) = H(t) + x_0. \tag{46}$$

Then

(1) $\Psi(t)$ is gH-differentiable with $\Psi'_{gH}(t) = \sum_{j=1}^{n} F_j A_j(t)$ and $\Psi(t_0) = x_0$.

(2) If all the gH-differences $H(t) \ominus_{gH} (-x_0)$ are of the same type for all t, then $\Phi(t)$ is gH-differentiable with $\Phi'_{gH}(t) = \sum_{j=1}^{n} F_j A_j(t)$ and $\Phi(t_0) = x_0$.

Consequently, if function $\varphi : [t_0, t_1] \longrightarrow \mathcal{K}_C$ is generated by a fixed fuzzy partition $(\mathbb{P},\mathbb{A})$ as in (44) and $H(t) = \sum_{j=1}^{n} F_j B_j(t)$, then we have the following two cases:

(1) If the intervals $F_j, j = 1, ..., n$ are such that

$$\varphi(t) = F(t, H(t) + x_0) \text{ for } t \in [t_0, t_1]$$

then $\Psi(t)$ defined on $[t_0, t_1]$ as in (46) is a solution of (40).

(2) If the intervals $F_j, j = 1, ..., n$ are such that

$$\varphi(t) = f\left(t, H(t) \ominus_{gH} (-x_0)\right) \text{ for } t \in [t_0, t_1]$$

then $\Phi(t)$ defined on $[t_0, t_1]$ as in (45) is a solution of (40), provided that all the gH-differences $H(t) \ominus_{gH} (-x_0)$ are of the same type for all $t \in [t_0, t_1]$.

Remark 2. *From the properties of gH-difference, we have that*

$$\Psi(t) \ominus_{gH} x_0 = H(t)$$

$$\Phi(t) \ominus_{gH} x_0 = (H(t) \ominus_{gH} (-x_0)) \ominus_{gH} x_0.$$

It is interesting to observe that function $\Psi(t)$ is (i)-gH-differentiable at all points, while function $\Phi(t)$ is (i)-gH-differentiable if the differences $H(t) \ominus_{gH} (-x_0)$ are of type (i), i.e., if $H(t) = \Phi(t) - x_0$, and is (ii)-gH-differentiable if the differences $H(t) \ominus_{gH} (-x_0)$ are of type (ii), i.e., if $x_0 = \Phi(t) - H(t)$. Consequently, solution

$$\Psi(t) = [\Psi^-(t), \Psi^+(t)] = (\widehat{\Psi}(t); \widetilde{\Psi}(t))$$

has always increasing length, but the same is not true for solution

$$\Phi(t) = [\Phi^-(t), \Phi^+(t)] = (\widehat{\Phi}(t); \widetilde{\Phi}(t))$$

in the case where $\Phi(t)$ *is (ii)-gH-differentiable.*

Finally, let G_j^- and G_j^+ be the $O(h^3)$ approximations, respectively, of F_j^- and F_j^+ similar to (18)–(19); we can see that the interval-valued iF-transform approximation $(\dot{x}_{gH})_{(\mathbb{P},\mathbb{A})}(t) = \sum_{j=1}^{n} G_j A_j(t)$ of $\dot{x}_{gH}$ is useful in determining conditions for a switching point.

Indeed, if $p_{k-1} < p_k < p_{k+1}$ are three adjacent points of $\mathbb{P}$ ($2 \leq k \leq n-1$) and we suppose that no switching point exists internally to the two subintervals $[p_{k-1}, p_k]$ and $[p_k, p_{k+1}]$ (i.e., possibly, the switching is exactly at p_k) then the solution $x(t)$ must satisfy

$$x(p_k) \ominus_{gH} x(p_{k-1}) = H_k^L \text{ and } x(p_{k+1}) \ominus_{gH} x(p_k) = H_k^R \tag{47}$$

$$\text{where } H_k^L = \int\limits_{p_{k-1}}^{p_k} \dot{x}_{gH}(t)\,dt \text{ and } H_k^R = \int\limits_{p_k}^{p_{k+1}} \dot{x}_{gH}(t)\,dt. \tag{48}$$

On the other hand, we have $H_k^L = F_{k-1} I_{k-1}^+ + F_k I_k^-$ and $H_k^R = F_k I_k^+ + F_{k+1} I_{k+1}^-$.

Suppose now that p_k is a switching point for the gH-differentiability of $x(t)$. We have two cases:

(I): (i)-to-(ii) switch: $x(t)$ is (i)-gH-differentiable on $]p_{k-1}, p_k[$ and is (ii)-gH-differentiable on $]p_k, p_{k+1}[$, i.e., $x(p_k) = x(p_{k-1}) + H_k^L$ and $x(p_k) = x(p_{k+1}) - H_k^R$ so that $x(p_{k+1}) = (x(p_{k-1}) + H_k^L) \ominus_{gH} (-H_k^R)$, where the difference is of type *(i)*;

(II): (ii)-to-(i) switch: $x(t)$ is (ii)-gH-differentiable on $]p_{k-1}, p_k[$ and is (i)-gH-differentiable on $]p_k, p_{k+1}[$, i.e., $x(p_{k-1}) = x(p_k) - H_k^L$ and $x(p_{k+1}) = x(p_k) + H_k^R$ so that $x(p_{k+1}) = (x(p_{k-1}) \ominus_{gH} (-H_k^L)) + H_k^R$, where the difference is of type *(i)*.

Instead, if p_k is not a switching point and $x(p_{k+1}) = x(p_{k-1}) + H_k^L + H_k^R$ we have a type (i) solution on $]p_{k-1}, p_{k+1}[$; or, if $x(p_{k-1}) = x(p_{k+1}) - H_k^L + H_k^R$ we have a type (ii) solution on $]p_{k-1}, p_{k+1}[$.

In terms of midpoint notation for intervals, we can summarize the discussion above as follows:

Types of switching points: *Let* $(\mathbb{P},\mathbb{A})$ *be a fuzzy partition of* $[t_0, t_1]$ *and* $p_{k-1} < p_k < p_{k+1}$ $(2 \leq k \leq n-1)$*; let* $x(t) = (\widehat{x}(t); \widetilde{x}(t))$ *be a solution to (40) and* $H_k^L = (\widehat{H}_k^L, \widetilde{H}_k^L)$ *and* $H_k^R = (\widehat{H}_k^R, \widetilde{H}_k^R)$ *be given as in (48). Then, the midpoint values satisfy*

$$\begin{aligned} \widehat{x}(p_k) &= \widehat{x}(p_{k-1}) + \widehat{H}_k^L \\ \widehat{x}(p_{k+1}) &= \widehat{x}(p_{k-1}) + \widehat{H}_k^L + \widehat{H}_k^R. \end{aligned}$$

Assuming that only p_k *is eventually a switching point, we have the following four cases*

(a) *if* p_k *is a (i)-to-(ii) switch, then*

$$\begin{aligned} \widetilde{x}(p_k) &= \widetilde{x}(p_{k-1}) + \widetilde{H}_k^L \\ \widetilde{x}(p_{k+1}) &= \widetilde{x}(p_{k-1}) + \widetilde{H}_k^L - \widetilde{H}_k^R \geq 0 \end{aligned}$$

(b) *if* p_k *is a (ii)-to-(i) switch, then*

$$\begin{aligned} \widetilde{x}(p_k) &= \widetilde{x}(p_{k-1}) - \widetilde{H}_k^L \geq 0 \\ \widetilde{x}(p_{k+1}) &= \widetilde{x}(p_{k-1}) - \widetilde{H}_k^L + \widetilde{H}_k^R \geq 0 \end{aligned}$$

(c) *if* $x(t)$ *is (i)-gH- differentiable on* $[p_{k-1}, p_{k+1}]$*, then*

$$\begin{aligned} \widetilde{x}(p_k) &= \widetilde{x}(p_{k-1}) + \widetilde{H}_k^L \\ \widetilde{x}(p_{k+1}) &= \widetilde{x}(p_{k-1}) + \widetilde{H}_k^L + \widetilde{H}_k^R \end{aligned}$$

(d) *if $x(t)$ is (ii)-gH- differentiable on $[p_{k-1}, p_{k+1}]$, then*

$$\widetilde{x}(p_k) = \widetilde{x}(p_{k-1}) - \widetilde{H}_k^L \geq 0$$
$$\widetilde{x}(p_{k+1}) = \widetilde{x}(p_{k-1}) - (\widetilde{H}_k^L + \widetilde{H}_k^R) \geq 0.$$

There are no general rules to locate a switching point. Denote by $x(t) = [x^-(t), x^+(t)] = (\widehat{x}(t); \widetilde{x}(t))$ a solution of the IDE (40); if $x(t)$ is (i)-gH-differentiable, its length $\widetilde{x}(t)$ will not decrease, while $\widetilde{x}(t)$ will not increase if $x(t)$ is (ii)-gH-differentiable. Some authors have noticed that possibly, the sequence of switching points can be pre-defined a priori, by positioning them in the time domain of the interval differential equation; this is true, at least in principle, provided that the found solution is guaranteed to have exactly them and not other switching points, but such purely *exogenous* proposal is not fully convincing.

An *endogenous* way may be preferred, similar to control strategies, to connect the type of gH-differentiability to the evolution of the trajectory. For example, it seems reasonable to locate the switching points depending on how the solution is evolving, by fixing a lower $l(t)$ and an upper threshold $u(t)$, say $0 \leq L \leq l(t) \leq u(t) \leq U$ and requiring that $l(t) \leq \widetilde{x}(t) \leq u(t)$ for all t; then,

(a) a (i)-to-(ii) switch is decided at $t = p_k$ if the following condition is reached

$$\begin{cases} \widetilde{x}(p_k) = \widetilde{x}(p_{k-1}) + \widetilde{H}_k^L \leq u(p_k) \\ \widetilde{x}(p_{k+1}) = \widetilde{x}(p_{k-1}) + \widetilde{H}_k^L + \widetilde{H}_k^R > u(p_{k+1}) \end{cases}$$

(b) a (ii)-to-(i) switch is decided at $t = p_k$ if the following condition is reached

$$\begin{cases} \widetilde{x}(p_k) = \widetilde{x}(p_{k-1}) - \widetilde{H}_k^L \geq l(p_k) \\ \widetilde{x}(p_{k+1}) = \widetilde{x}(p_{k-1}) - \widetilde{H}_k^L - \widetilde{H}_k^R < l(p_{k+1}). \end{cases}$$

In some sense, the rule above will control the increasing and decreasing of "uncertainty" in an endogenous way, without any reference to the interval initial condition x_0 or to the interval-valued function $F(t, x)$.

Furthermore, we are essentially free to decide the type of differentiability at the initial point; if the initial length $\widetilde{x}_0$ is such that $L < \widetilde{x}_0 < U$, we can start either with a type *c)* or a type *d)* and a unique solution is then found by applying the decided switching rule.

A different purely endogenous rule can be obtained by following the increase or decrease of $\widetilde{x}(t)$ and trying to intercept points t^* where the function $\widetilde{x}(t)$ has a local maximum (for a (i)-to-(ii) switch) or a local minimum (for a (ii)-to-(i) switch). A necessary condition can be obtained according to the following simple result:

Endogenous criteria for a switching point: *Assume that $x(t) = [x^-(t), x^+(t)] = (\widehat{x}(t); \widetilde{x}(t))$ is such that $x^-(t)$ and $x^+(t)$ are differentiable so that its gH-derivative $\dot{x}_{gH}(t)$ can be expressed in terms of the derivatives $(x^-)'(t)$ and $(x^+)'(t)$. Let $(\mathbb{P}, \mathbb{A})$ be a fuzzy partition of $[t_0, t_1]$ and let $(\dot{x}_{gH})_{(\mathbb{P},\mathbb{A})}(t) = \sum_{k=1}^n F_k A_k(t)$ be the iF-transform of $\dot{x}_{gH}(t)$ with interval-valued components $F_k = (\widehat{F}_k; \widetilde{F}_k)$. If p_k is a local minimum or maximum of $\widetilde{x}(t)$, then $\widetilde{F}_k = 0 + O(h^2)$.*

Proof. From the properties of F-transform, we have $F_k^- = (\dot{x}_{gH})^-(p_k) + O(h^2)$, $F_k^+ = (\dot{x}_{gH})^+(p_k) + O(h^2)$ and, in particular, $\widetilde{F}_k = \widetilde{\dot{x}_{gH}(p_k)} + O(h^2)$. On the other hand, from the differentiability of $x^-(t)$ and $x^+(t)$, we have $0 = \frac{d}{dt}\widetilde{x}(t))$ for $t = p_k$, i.e., $(x^+)'(p_k) - (x^-)'(p_k) = 0$. It follows that $\widetilde{\dot{x}_{gH}(p_k)} = \left| \frac{(x^+)'(p_k) - (x^-)'(p_k)}{2} \right| = 0$, i.e., $\widetilde{F}_k = \widetilde{\dot{x}_{gH}(p_k)} + O(h^2) = 0 + O(h^2)$. □

We then suggest the following (purely endogenous) switching rule.

Switching rule for IDE: Let $(\mathbb{P}, \mathbb{A})$ be a fuzzy partition of $[t_0, t_1]$ and suppose we have computed the approximate solution $x(p_i)$ of IDE (40) at the points $t_0 = p_1 < p_2 < ... < p_k$ with $k < n$. Assume that $h = \frac{t_1 - t_0}{n-1}$ is sufficiently small and $p_i = p_{i-1} + h$.

In any case, the midpoint value is computed as $\widehat{x}(p_k) = \widehat{x}(p_{k-1}) + \widehat{F}_{k-1} I^+_{k-1} + \widehat{F}_k I^-_k$.

(1) If $x(p_k)$ is computed according to (i)-gH-differentiability on $[p_{k-1}, p_k]$, i.e., $\widetilde{x}(p_k) = \widetilde{x}(p_{k-1}) + \widetilde{F}_{k-1} I^+_{k-1} + \widetilde{F}_k I^-_k$ and if $\widetilde{F}_k = O(h^2)$, then p_k is assumed to be a (i)-to-(ii) switch and the next iteration is performed according to (ii)-gH-differentiability.

(2) If $x(p_k)$ is computed according to (ii)-gH-differentiability on $[p_{k-1}, p_k]$, i.e., $\widetilde{x}(p_k) = \widetilde{x}(p_{k-1}) - \widetilde{F}_{k-1} I^+_{k-1} - \widetilde{F}_k I^-_k$ and if $\widetilde{F}_k = O(h^2)$, then p_k is assumed to be a (ii)-to-(i) switch and the next iteration is performed according to (i)-gH-differentiability.

In our computations, the test $\widetilde{F}_k = O(h^2)$ is performed by the condition $\widetilde{F}_k < \varepsilon$ for a small tolerance $\varepsilon > 0$ (say $\varepsilon \in [10^{-3}, 10^{-5}]$).

5.2. Numerical Procedure for IDE

We are now ready to summarize the proposed procedure to solve the IDE (40) on $[t_0, t_1]$, based on F-transform of the gH-derivative $\dot{x}_{gH}(t)$.

Chose a uniform fuzzy partition $(\mathbb{P},\mathbb{A})$ with $n > 2$ points $\mathbb{P}= \{t_0 = p_1 < ... < p_n = t_1\}$ and basic functions $\mathbb{A}= \{A_1, ..., A_n\}$.

From the properties of the F-transforms on $(\mathbb{P}_k,\mathbb{A}_k)$, we have

$$\int_{p_1}^{p_k} \dot{x}_{gH}(t)\, dt = \sum_{j=1}^{k-1} \int_{p_1}^{p_k} F_j A_j(t)\, dt + \int_{p_1}^{p_k} \widetilde{F}_k \widetilde{A}_k(t)\, dt$$

for all $k = 2, ..., n$, where

$$\widetilde{F}_k = \frac{\int_{p_{k-1}}^{p_k} \dot{x}_{gH}(t)\, A_k(t)\, dt}{\int_{p_{k-1}}^{p_k} A_k(t)\, dt}.$$

With the introduced notation, we solve (40) by the following procedure, assuming that no switching point exists on $[p_{k-1}, p_k]$ and x_0 is the given initial condition.

Algorithm IDE-FT: Find an approximated final value $x_{(\mathbb{P},\mathbb{A})}(t_1)$ of the IDE $\dot{x}_{gH}(t) = F(t, x(t))$, $t \in [t_0, t_1]$ with initial condition $x(t_0) = x_0$.

Step 1. Chose one of the two models (45) or (46) to start with, say $Meth = 1$ for (i)-gH-differentiability (increasing length) and $Meth = 2$ for (ii)-gH-differentiability (decreasing length); chose a positive tolerance $\varepsilon > 0$ to test for switching; chose $n > 2$, set $h = \frac{t_1 - t_0}{n-1}$ so that $p_j = t_0 + (j-1)h$ for $j = 1, .., n$ and chose a family $\mathbb{A}$ of basic functions.

Step 2. Compute $F_1 = f(t_0, x_0)$. Set $H = F_1 I^+_1$, $y = x_0$.

Step 3. For $k = 2, 3, ..., n$, the interval-valued components $\widetilde{F}_k$, F_k are first approximated iteratively by finding $\widetilde{F}_k$ as a fixed point

$$\widetilde{F}_k = \frac{h}{I^-_k} f\left(p_k, y + H + \widetilde{F}_k I^-_k\right) \text{ if } Meth = 1$$

or

$$\widetilde{F}_k = \frac{h}{I^-_k} f\left(p_k, \left(y + H + \widetilde{F}_k I^-_k\right) \ominus_{gH} (-x_0)\right) \text{ if } Meth = 2$$

and $F_k = 2 I^-_k \widetilde{F}_k / I_k$. Set the new solution interval value for $t = p_k$ as

$$x(p_k) = \begin{cases} y + H + F_k I^-_k & \text{if} \quad Meth = 1 \\ (H + F_k I^-_k) \ominus_{gH} (-y) & \text{if} \quad Meth = 2 \end{cases}$$

Step 4. (Test for a switching at p_k) If $\widetilde{F}_k < \varepsilon$, then p_k is declared to be a switch of gH-differentiability: if so, exchange the value of *Meth* from 1 to 2 or from 2 to 1; set $H = f(p_k, x(p_k))I_k^+, y = x(p_k)$. If instead $\widetilde{F}_k > \varepsilon$, then p_k is not a switch: in this case, update H to $H + F_k I_k^-$, without changing *Meth*.

Step 5. Continue from step 3 with next k, until the final point t_1 is reached.

5.3. Application to Fuzzy Differential Equations

If the fuzzy-valued function $\varphi : [a,b] \longrightarrow \mathbb{R}_{\mathcal{F}}$ is gH-differentiable with respect to t, or if it is level-wise gH-differentiable, we can approximate its gH-derivative φ'_{gH} or its LgH-derivative φ'_{LgH} by a family of interval-valued iF-transforms, parametrized with $\alpha \in [0,1]$, in terms of its α-cuts (consider that $A_j(x) \geq 0$ for all x and all j)

$$\left([\varphi'_{gH}(x)]_\alpha\right)_{(P,A)} = \sum_{j=1}^{n} F_{j,\alpha} A_j(x) = \left[\sum_{j=1}^{n} F^-_{j,\alpha} A_j(x), \sum_{j=1}^{n} F^+_{j,\alpha} A_j(x)\right].$$

The F-transform of φ'_{gH} is then obtained by the family of F-transforms of the lower and upper functions $(\varphi'_{gH})^-_{j,\alpha}(x)$ and $(\varphi'_{gH})^+_{j,\alpha}(x)$:

$$F^-_{j,\alpha} = \frac{\int_a^b (\varphi'_{gH})^-_{j,\alpha}(x) A_j(x)dx}{\int_a^b A_j(x)dx}, \quad j = 1,2,...,n$$

$$F^+_{j,\alpha} = \frac{\int_a^b (\varphi'_{gH})^+_{j,\alpha}(x) A_j(x)dx}{\int_a^b A_j(x)dx}, \quad j = 1,2,...,n.$$

and, considering that each A_j is a continuous non-negative function, we can obtain an approximation of $\varphi(x)$ by

$$\varphi(x) \ominus_{gH} \varphi(a) = \int_a^x \varphi'_{gH}(t)dt = \sum_{j=1}^{n} \left(\varphi'_{gH}(x_j) \int_a^x A_j(t)dt\right).$$

Consider now an FDE in gH-derivative form

$$\begin{aligned} \varphi'_{gH}(x) &= f(x, \varphi(x)) \text{ for } x \in [x_1, x_q] \\ \varphi(x_1) &= \varphi_1; \end{aligned}$$

we approximate $\varphi'_{gH}(x) = \sum_{j=1}^{q} \varphi'_{gH}(x_j) B_j(x)$ and substituting the condition $\varphi'_{gH}(x) = f(x, \varphi(x))$ into the approximation, we obtain

$$\varphi'_{gH}(x) = \sum_{j=1}^{q} f(x, \varphi(x)) A_j(x);$$

Integrating with given initial condition $\varphi(x_1) = \varphi_1$ we obtain

$$\varphi(x) \ominus_{gH} \varphi_1 = \int_{x_1}^{x} f(t, \varphi(t))dt$$

and, at the points $x_j \in \mathbb{P}$, defining $\varphi_j = \varphi(x_j)$, $j = 2, ..., q$,

$$\begin{array}{l} \varphi_2 \ominus_{gH} \varphi_1 = f(x_1, \varphi_1) I_1^+ + f(x_2, \varphi_2) I_2^- \\ \text{ and, for } j = 3, ..., n \\ \varphi_j \ominus_{gH} \varphi_1 = \sum\limits_{l=1}^{j-1} f(x_l, \varphi_l)(I_l^- + I_l^+) + f(x_j, \varphi(x_j)) I_j^- . \end{array}$$

These are (generally nonlinear) non-differential fuzzy equations of the form

$$\varphi_j \ominus_{gH} \varphi_1 = c_{j-1} + f(x_j, \varphi_j) I_j^-, \, j = 2, 3, ..., n$$

where when solving for φ_j, the fuzzy numbers $\varphi_1, \varphi_2, ..., \varphi_{j-1}$ are determined in previous steps and the fuzzy quantity

$$c_{j-1} = \sum_{l=1}^{j-1} f(x_l, \varphi_l)(I_l^- + I_l^+)$$

is known.

If we search for a (i)-gH-differentiable solution, the fuzzy equation for φ_j is

$$\varphi_j = \varphi_1 + c_{j-1} + f(x_j, \varphi_j) I_j^-;$$

if we search for a (ii)-gH-differentiable solution, the fuzzy equation for φ_j is

$$\varphi_1 = \varphi_j + (-c_{j-1} - f(x_j, \varphi_j) I_j^-).$$

In terms of α-cuts, they can be written (for $\alpha \in [0, 1]$ or $\alpha \in \{\alpha_1, \alpha_2, ..., \alpha_p\}$)

$$\begin{array}{l} \varphi_{j,\alpha}^- = \varphi_{1,\alpha}^- + c_{j-1,\alpha}^- + f_\alpha^-(x_j, \varphi_{j,\alpha}^-, \varphi_{j,\alpha}^+) I_j^- \\ \varphi_{j,\alpha}^+ = \varphi_{1,\alpha}^+ + c_{j-1,\alpha}^+ + f_\alpha^+(x_j, \varphi_{j,\alpha}^-, \varphi_{j,\alpha}^+) I_j^- \end{array}$$

and

$$\begin{array}{l} \varphi_{1,\alpha}^- = \varphi_{j,\alpha}^- + (-c_{j-1,\alpha}^+ - f_\alpha^+(x_j, \varphi_{j,\alpha}^-, \varphi_{j,\alpha}^+) I_j^-) \\ \varphi_{1,\alpha}^+ = \varphi_{j,\alpha}^+ + (-c_{j-1,\alpha}^- - f_\alpha^-(x_j, \varphi_{j,\alpha}^-, \varphi_{j,\alpha}^+) I_j^-) \end{array}$$

respectively.

6. Computational Results for Interval and Fuzzy Differential Equations

For all problems we use $M = 10,001$ as the number of points t_i where the solution is computed and $n = 11$ as the number of nodes in the partition $(\mathbb{P}, \mathbb{A})$; it results that the step size of the method to solve the differential equation on $[a, b]]$ is $h = \dfrac{b-a}{(M-1)*(n-1)} = \dfrac{b-a}{100{,}000}$.

The number of α-cuts is $N_\alpha = 11$ so that the LgH-differentiable solutions are computed for all $\alpha \in \left\{ \dfrac{i-1}{N_\alpha - 1} | i = 1, 2, ..., N_\alpha \right\}$; remark that the interval-valued solutions corresponding to the α-cuts are independent each other. In particular, the family of interval-valued solutions does not always form a fuzzy-valued function.

Problem IDE1: Solution interval is $t \in [0, 2]$; functions a and b are $a(t) = 2\sin(2\pi t)\cos(2\pi t)$ and $b(t) = \frac{t|2-t|}{2}$ and the equations,

$$\begin{cases} [\dot{x}_{LgH}(t)]_\alpha = a(t)[x(t)]_\alpha + b(t) \\ x(0) = x_0. \end{cases}$$

The fuzzy initial condition is given in terms of the α-cuts of x_0, $[x_0]_\alpha = [1 + 0.5\alpha, 3 - \alpha]$, $\alpha \in [0, 1]$, with support $[1, 3]$ and core $[1.5, 2]$.

In this case, the interval-valued solutions for the different values of α form a fuzzy-valued function, i.e., the family of IDEs (parametrized with $\alpha \in [0, 1]$) solve the corresponding FDE.

For the two runs with Meth = 1 and Meth = 2, the step size is h = 2.000 $\times 10^{-5}$. The solution intervals (initial and final) and the solution intervals at the internal switching points are inserted in the following tables, for three values of $\alpha = 0, 0.5, 1$.

Starting with Meth = 1, i.e., by an increasing-length solution (see Figure 9), the algorithm finds seven switching points at $tw \in Sw = \{.25, .5, .75, 1, 1.25, 1.5, 1.75\}$ (see Table 1).

Table 1. (Interval-valued problem IDE1-a): For three values of $\alpha \in \{0, 0.5, 1\}$, the seven switching points tw corresponding to Meth=1 are given in the first column; the second column contains the interval-valued solution $X(tw) = [x^-_\alpha(tw), x^+_\alpha(tw)]$.

α = 0.00	Initial value = [1.0, 3.0]
tw = 0.25	X(tw) = [1.2026308981, 3.5476701109]
tw = 0.50	X(tw) = [1.0961169068, 3.0961169068]
tw = 0.75	X(tw) = [1.4006060055, 3.7456452183]
tw = 1.00	X(tw) = [1.3078549504, 3.3078549504]
tw = 1.25	X(tw) = [1.6663651377, 4.0114043505]
tw = 1.50	X(tw) = [1.5195929939, 3.5195929939]
tw = 1.75	X(tw) = [1.8643402451, 4.2093794579]
	Final value = [1.61570990, 3.61570990]
α = 0.50	Initial value: [1.25, 2.50]
tw = 0.25	X(tw) = [1.4957607997, 2.9614103077]
tw = 0.50	X(tw) = [1.3461169068, 2.5961169068]
tw = 0.75	X(tw) = [1.6937359071, 3.1593854151]
tw = 1.00	X(tw) = [1.5578549504, 2.8078549504]
tw = 1.25	X(tw) = [1.9594950393, 3.4251445473]
tw = 1.50	X(tw) = [1.7695929939, 3.0195929939]
tw = 1.75	X(tw) = [2.1574701467, 3.6231196547]
	Final value = [1.86570990, 3.11570990]
α = 1.00	Initial value: [1.50, 2.00]
tw = 0.25	X(tw) = [1.7888907013, 2.3751505045]
tw = 0.50	X(tw) = [1.5961169068, 2.0961169068]
tw = 0.75	X(tw) = [1.9868658087, 2.5731256119]
tw = 1.00	X(tw) = [1.8078549504, 2.3078549504]
tw = 1.25	X(tw) = [2.2526249409, 2.8388847441]
tw = 1.50	X(tw) = [2.0195929939, 2.5195929939]
tw = 1.75	X(tw) = [2.4506000483, 3.0368598515]
	Final value: [2.11570990, 2.61570990]

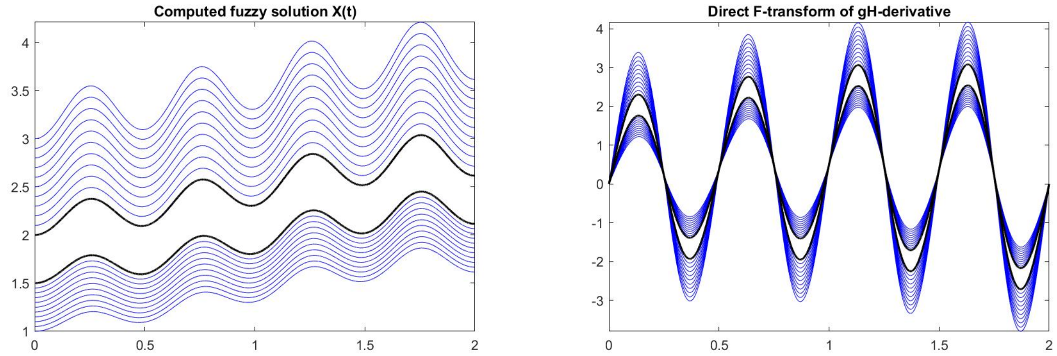

Figure 9. Problem IDE1, Meth = 1: Level-wise LgH-differentiable solution (**left**) and its LgH-derivative (**right**). There are seven switching points which are in the same position for all level α and correspond to the values of t where the gH-derivative is 0.

Starting with Meth = 2, i.e., by a decreasing-length solution (see Figure 10), we find the same seven switching points (but with different interval values) as reported in Table 2.

Table 2. (Interval-valued problem IDE1-b): For three values of $\alpha \in \{0, 0.5, 1\}$, the seven switching points tw corresponding to Meth=2 are given in the first column; the second column contains the interval-valued solution $X(tw) = [x_\alpha^-(tw), x_\alpha^+(tw)]$.

α = 0.00	Initial value = [1.0, 3.0]
tw = 0.25	X(tw) = [1.5222863012, 3.2280147078]
tw = 0.50	X(tw) = [1.0961169068, 3.0961169068]
tw = 0.75	X(tw) = [1.7202614085, 3.4259898152]
tw = 1.00	X(tw) = [1.3078549504, 3.3078549504]
tw = 1.25	X(tw) = [1.9860205408, 3.6917489475]
tw = 1.50	X(tw) = [1.5195929939, 3.5195929939]
tw = 1.75	X(tw) = [2.1839956481, 3.8897240548]
	Final value: [1.61570990, 3.61570990]
α = 0.50	Initial value: [1.25, 2.50]
tw = 0.25	X(tw) = [1.6955454266, 2.7616256808]
tw = 0.50	X(tw) = [1.3461169068, 2.5961169068]
tw = 0.75	X(tw) = [1.8935205340, 2.9596007882]
tw = 1.00	X(tw) = [1.5578549504, 2.8078549504]
tw = 1.25	X(tw) = [2.1592796662, 3.2253599204]
tw = 1.50	X(tw) = [1.7695929939, 3.0195929939]
tw = 1.75	X(tw) = [2.3572547736, 3.4233350278]
	Final value: [1.86570990, 3.11570990]
α = 1.00	Initial value: [1.50, 2.00]
tw = 0.25	X(tw) = [1.8688045521, 2.2952366537]
tw = 0.50	X(tw) = [1.5961169068, 2.0961169068]
tw = 0.75	X(tw) = [2.0667796594, 2.4932117611]
tw = 1.00	X(tw) = [1.8078549504, 2.3078549504]
tw = 1.25	X(tw) = [2.3325387917, 2.7589708934]
tw = 1.50	X(tw) = [2.0195929939, 2.5195929939]
tw = 1.75	X(tw) = [2.5305138991, 2.9569460007]
	Final value: [2.11570990, 2.61570990]

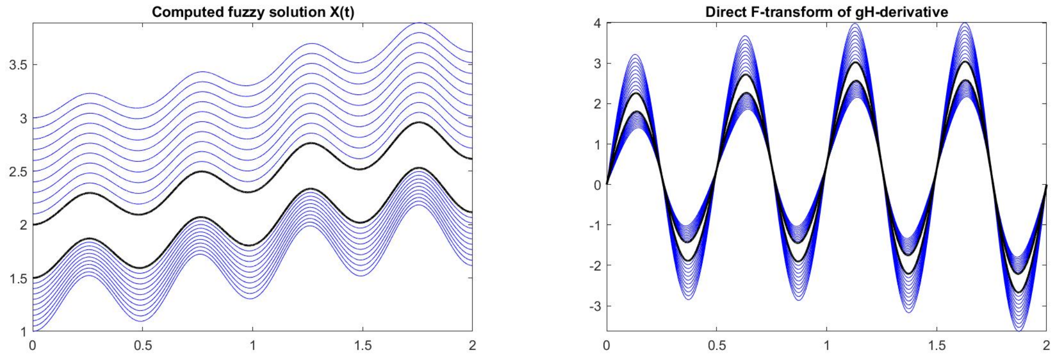

Figure 10. Problem IDE1, Meth = 2: Level-wise LgH-differentiable solution (**left**) and its LgH-derivative (**right**). There are seven switching points which are in the same position for all level α and correspond to the values of t where the gH-derivative is 0.

It seems remarkable that in this example, the two solutions are different, but share the same value at some of the switching points; indeed, denoting x_1 and x_2 the solution found with Meth = 1 and Meth = 2, we have that $x_1(.5) = x_2(.5)$, $x_1(1) = x_2(1)$, $x_1(1.5) = x_2(1.5)$ while the solutions are not equal at all other points.

Problem IDE2: Solution interval is $t \in [0, \frac{3}{2}\pi]$; function $c(t) = -cos(t)(1 - sin(t))$ and two (symmetric) fuzzy numbers A, B are given with α-cuts $[A]_\alpha = [-0.4 + 0.3\alpha, 0.4 - 0.3\alpha]$, $[B]_\alpha = [-0.5 + 0.5\alpha, 0.5 - 0.5\alpha]$, $\alpha \in [0, 1]$; the fuzzy differential equation is, level-wise,

$$\begin{cases} \dot{x}_{LgH}(t) = Ax(t) \ominus_{gH} c(t)B \\ x(0) = x_0 \end{cases}$$

where $Ax(t)$ and $c(t)B$ are obtained by standard interval operations and $\ominus_{gH}$ is gH-difference; the interval initial condition in terms of α-cuts is $[x_0]_\alpha = [-1 + \alpha, 1 - \alpha]$, $\alpha \in [0, 1]$, with support $[-1, 1]$ and core $\{0\}$.

In this case, the step size is h = 4.712 $\times 10^{-5}$. The computations are performed with 11 α-cuts; in the next two tables we insert the results obtained with Meth = 1 and Meth = 2 for a subset of the computed α-cuts.

Remark the important fact that in this example, the switching points change in position for different values of α (see Figures 11 and 12 and Tables 3 and 4. Viewing the solution function as fuzzy-valued, it is not gH-differentiable and there exist three switching regions.

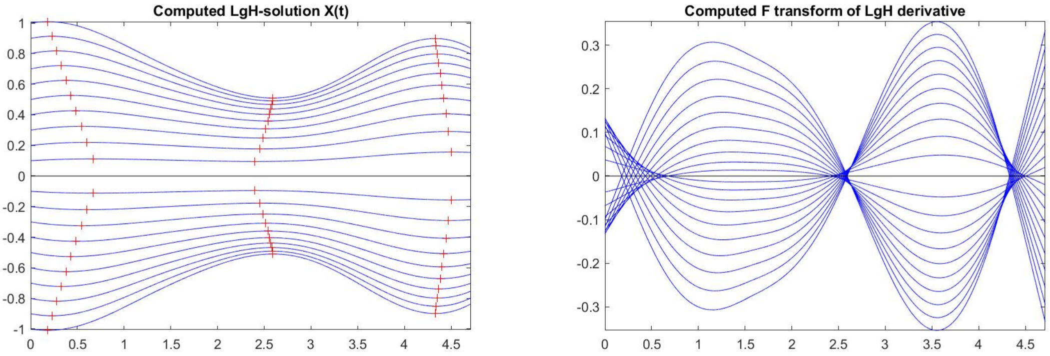

Figure 11. Problem IDE2, Meth = 1: Level-wise LgH-differentiable solution (**left**) and its LgH-derivative (**right**). For all α-cuts, there are three switching points which are not in the same position for the levels α (**left**); on **right**, the LgH-derivatives are represented.

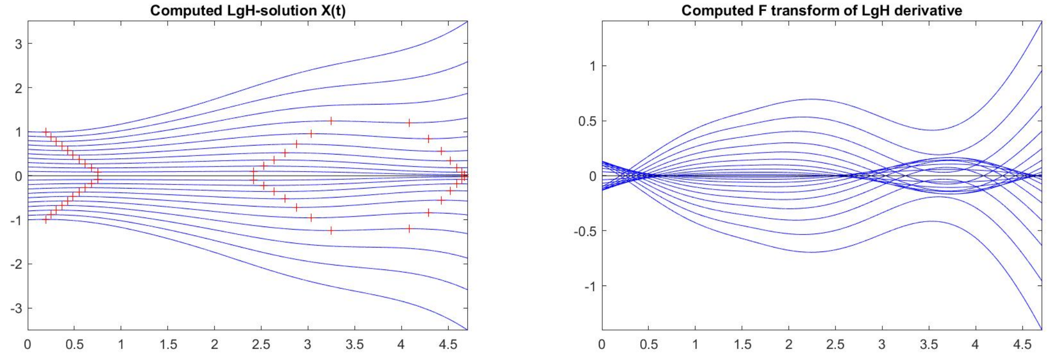

Figure 12. Problem IDE2, Meth = 2: Level-wise LgH-differentiable solution (**left**) and its LgH-derivative (**right**). For all α-cuts, there are one or three switching points which are not in the same position for the levels α (**left**); the LgH-derivative is represented on the **right** picture.

Table 3. (Interval-valued problem IDE2-a): For five values of $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8\}$, the three switching points tw corresponding to Meth=1 are given in the first column; the second column contains the interval-valued solution $X(tw) = [x_\alpha^-(tw), x_\alpha^+(tw)]$. If $\alpha = 1$, the solution is single-valued and no switching point exists.

α = 0.00	Initial value = $[-1.0, 1.0]$
tw = 1.7954×10^{-1}	X(tw) = $[-1.0090673275\times10^{-1}, 1.0090673275\times10^{-1}]$
tw = 2.5918×10^{0}	X(tw) = $[-5.0989773431\times10^{-1}, 5.0989773431\times10^{-1}]$
tw = 4.3297×10^{0}	X(tw) = $[-8.9894395242\times10^{-1}, 8.9894395242\times10^{-1}]$
	Final value: $[-8.35571783\times10^{-1}, 8.35571783\times10^{-1}]$
α = 0.20	Initial value: $[-8.00\times10^{-1}, 8.00\times10^{-1}]$
tw = 2.7944×10^{-1}	X(tw) = $[-8.1796697089\times10^{-1}, 8.1796697089\times10^{-1}]$
tw = 2.5810×10^{0}	X(tw) = $[-4.6779933026\times10^{-1}, 4.6779933026\times10^{-1}]$
tw = 4.3542×10^{0}	X(tw) = $[-7.9775755945\times10^{-1}, 7.9775755945\times10^{-1}]$
	Final value: $[-7.52396593\times10^{-1}, 7.52396593\times10^{-1}]$
α = 0.40	Initial value: $[-6.00\times10^{-1}, 6.00\times10^{-1}]$
tw = 3.7935×10^{-1}	X(tw) = $[-6.2502008124\times10^{-1}, 6.2502008124\times10^{-1}]$
tw = 2.5579×10^{0}	X(tw) = $[-4.0293011585\times10^{-1}, 4.0293011585\times10^{-1}]$
tw = 4.3844×10^{0}	X(tw) = $[-6.7059314969\times10^{-1}, 6.7059314969\times10^{-1}]$
	Final value: $[-6.41450486\times10^{-1}, 6.41450486\times10^{-1}]$
α = 0.60	Initial value: $[-4.00\times10^{-1}, 4.00\times10^{-1}]$
tw = 4.8538×10^{-1}	X(tw) = $[-4.2690473012\times10^{-1}, 4.2690473012\times10^{-1}]$
tw = 2.5174×10^{0}	X(tw) = $[-3.0871431900\times10^{-1}, 3.0871431900\times10^{-1}]$
tw = 4.4226×10^{0}	X(tw) = $[-5.0673707018\times10^{-1}, 5.0673707018\times10^{-1}]$
	Final value: $[-4.91260393\times10^{-1}, 4.91260393\times10^{-1}]$
α = 0.80	Initial value: $[-2.00\times10^{-1}, 2.00\times10^{-1}]$
tw = 6.0271×10^{-1}	X(tw) = $[-2.2003436269\times10^{-1}, 2.2003436269\times10^{-1}]$
tw = 2.4500×10^{0}	X(tw) = $[-1.7731361249\times10^{-1}, 1.7731361249\times10^{-1}]$
tw = 4.4716×10^{0}	X(tw) = $[-2.9073091262\times10^{-1}, 2.9073091262\times10^{-1}]$
	Final value: $[-2.85325600\times10^{-1}, 2.85325600\times10^{-1}]$
α = 1.00	Initial value: $[\ 0.00, 0.00]$
	Final value: $[0.00000000\times10^{0}, 0.00000000\times10^{0}]$

Table 4. (Interval-valued problem IDE2-b): For five values of $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8\}$, the switching points tw corresponding to Meth=2 are given in the first column; the second column contains the interval-valued solution $X(tw) = [x_\alpha^-(tw), x_\alpha^+(tw)]$. The number of switching points changes with α and if $\alpha = 1$, the solution is single-valued and no switching point exists.

α = 0.00	Initial value = $[-1.0, 1.0]$
tw = 1.9321×10^{-1}	X(tw) = $[-9.8998479170\times10^{-1}, 9.8998479170\times10^{-1}]$
	Final value: $[-3.50656260\times10^{0}, 3.50656260\times10^{0}]$
α = 0.20	Initial value: $[-8.00\times10^{-1}, 8.00\times10^{-1}]$
tw = 3.0866×10^{-1}	X(tw) = $[-7.7947575655\times10^{-1}, 7.7947575655\times10^{-1}]$
	Final value: $[-1.87166877\times10^{0}, 1.87166877\times10^{0}]$
α = 0.40	Initial value: $[-6.00\times10^{-1}, 6.00\times10^{-1}]$
tw = 4.2600×10^{-1}	X(tw) = $[-5.7091483288\times10^{-1}, 5.7091483288\times10^{-1}]$
tw = 3.0348×10^{0}	X(tw) = $[-9.5345951680\times10^{-1}, 9.5345951680\times10^{-1}]$
tw = 4.2887×10^{0}	X(tw) = $[-8.4078653821\times10^{-1}, 8.4078653821\times10^{-1}]$
	Final value: $[-8.90550965\times10^{-1}, 8.90550965\times10^{-1}]$
α = 0.60	Initial value: $[-4.00\times10^{-1}, 4.00\times10^{-1}]$
tw = 5.4899×10^{-1}	X(tw) = $[-3.6865624377\times10^{-1}, 3.6865624377\times10^{-1}]$
tw = 2.7506×10^{0}	X(tw) = $[-5.2242232939\times10^{-1}, 5.2242232939\times10^{-1}]$
tw = 4.5230×10^{0}	X(tw) = $[-3.3741623443\times10^{-1}, 3.3741623443\times10^{-1}]$
	Final value: $[-3.44451472\times10^{-1}, 3.44451472\times10^{-1}]$
α = 0.80	Initial value: $[-2.00\times10^{-1}, 2.00\times10^{-1}]$
tw = 6.8094×10^{-1}	X(tw) = $[-1.7696972113\times10^{-1}, 1.7696972113\times10^{-1}]$
tw = 2.5277×10^{0}	X(tw) = $[-2.1957335579\times10^{-1}, 2.1957335579\times10^{-1}]$
tw = 4.6445×10^{0}	X(tw) = $[-8.1718701678\mathrm{e}{-02}, 8.1718701678\mathrm{e}{-02}]$
	Final value: $[-8.21485320\mathrm{e}{-02}, 8.21485320\mathrm{e}{-02}]$
α = 1.00	Initial value: $[0.00, 0.00]$
	Final value: $[0.00000000\times10^{0}, 0.00000000\times10^{0}]$

Problem FDE1: Solution interval is $t \in [0, \frac{1}{2}]$;

$$\begin{cases} \dot{x}_{gH}(t) = -\frac{1}{2}x(t) + 2\sin(3t) \\ x(0) = x_0 \end{cases}$$

where $[x_0]_\alpha = [-1+\alpha, 1-\alpha]$, $\alpha \in [0,1]$.

The step size in this case is h = 4.000 $\times 10^{-5}$.

The solution found with both Meth = 1 and Meth = 2 are fuzzy-valued with lengths of the α-cuts all increasing (Meth = 1, Figure 13) or all decreasing (Meth = 2, Figure 14) and there are no switching points.

The α-cuts of the initial condition x_0 and the final fuzzy solution for Meth = 1 and Meth = 2 are inserted in Table 5.

Table 5. (Fuzzy-valued problem FDE1): For eleven values of $\alpha \in \left\{\frac{i-1}{10} | i = 1, ..., 10\right\}$, the table contains the interval level-wise initial condition (column 2), the final interval value with Meth = 1 (column 3) and the final interval value with Meth = 2.

α	Initial Condition	Final solution (Meth = 1)	Final solution (Meth = 2)
0.0	$[-1.0000\times10^{0}, 1.0000\times10^{0}]$	$[-7.9066\times10^{0}, 6.8715\times10^{0}]$	$[-6.5292\times10^{-1}, -3.8225\times10^{-1}]$
0.1	$[-9.0000\times10^{-1}, 9.0000\times10^{-1}]$	$[-7.1677\times10^{0}, 6.1326\times10^{0}]$	$[-6.3939\times10^{-1}, -3.9579\times10^{-1}]$
0.2	$[-8.0000\times10^{-1}, 8.0000\times10^{-1}]$	$[-6.4288\times10^{0}, 5.3937\times10^{0}]$	$[-6.2586\times10^{-1}, -4.0932\times10^{-1}]$
0.3	$[-7.0000\times10^{-1}, 7.0000\times10^{-1}]$	$[-5.6899\times10^{0}, 4.6548\times10^{0}]$	$[-6.1232\times10^{-1}, -4.2285\times10^{-1}]$
0.4	$[-6.0000\times10^{-1}, 6.0000\times10^{-1}]$	$[-4.9510\times10^{0}, 3.9158\times10^{0}]$	$[-5.9879\times10^{-1}, -4.3639\times10^{-1}]$
0.5	$[-5.0000\times10^{-1}, 5.0000\times10^{-1}]$	$[-4.2121\times10^{0}, 3.1769\times10^{0}]$	$[-5.8526\times10^{-1}, -4.4992\times10^{-1}]$
0.6	$[-4.0000\times10^{-1}, 4.0000\times10^{-1}]$	$[-3.4732\times10^{0}, 2.4380\times10^{0}]$	$[-5.7172\times10^{-1}, -4.6345\times10^{-1}]$
0.7	$[-3.0000\times10^{-1}, 3.0000\times10^{-1}]$	$[-2.7343\times10^{0}, 1.6991\times10^{0}]$	$[-5.5819\times10^{-1}, -4.7699\times10^{-1}]$
0.8	$[-2.0000\times10^{-1}, 2.0000\times10^{-1}]$	$[-1.9954\times10^{0}, 9.6022\times10^{-1}]$	$[-5.4465\times10^{-1}, -4.9052\times10^{-1}]$
0.9	$[-1.0000\times10^{-1}, 1.0000\times10^{-1}]$	$[-1.2565\times10^{0}, 2.2132\times10^{-1}]$	$[-5.3112\times10^{-1}, -5.0405\times10^{-1}]$
1.0	$[0.0000\times10^{0}, 0.0000\times10^{0}]$	$[-5.1759\times10^{-1}, -5.1759\times10^{-1}]$	$[-5.1759\times10^{-1}, -5.1759\times10^{-1}]$

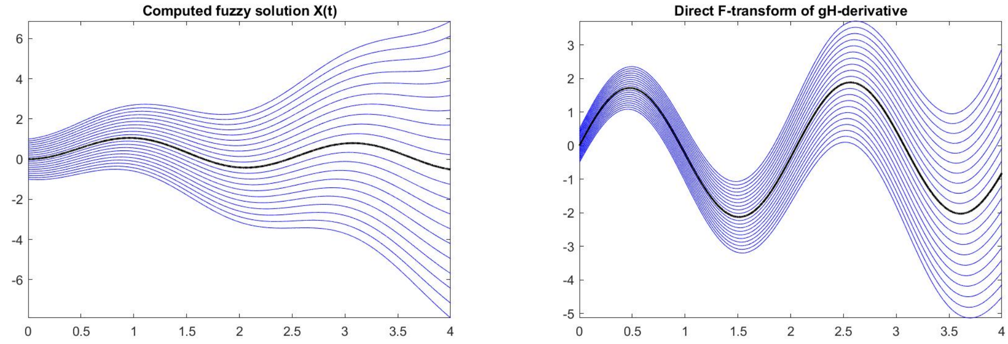

Figure 13. Problem FDE1, Meth = 1: Fuzzy-valued gH-differentiable solution (**left**) and its gH-derivative (**right**). There are no switching points.

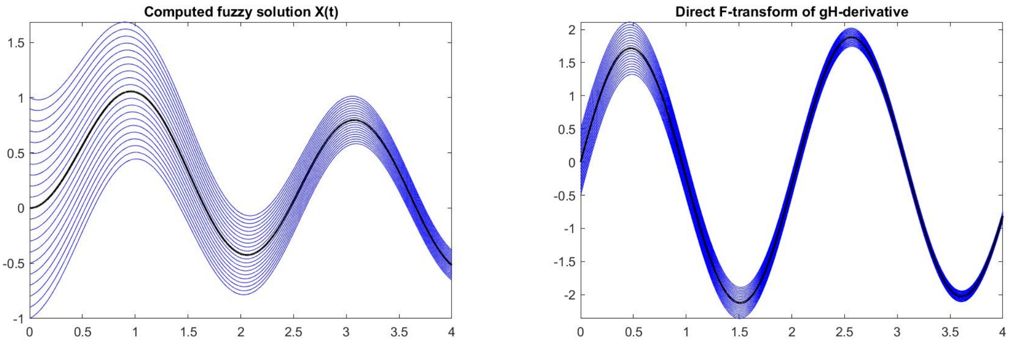

Figure 14. Problem FDE1, Meth = 2: Fuzzy-valued gH-differentiable solution (**left**) and its gH-derivative (**right**). For all α-cuts, there are no switching points.

Problem FDE2: Solution interval is $t \in [0, 4\pi]$;

$$\begin{cases} \dot{x}_{gH}(t) = \sin(t)x(t) \\ x(0) = x_0 \end{cases}$$

with $[x_0]_\alpha = [-1+\alpha, 1-\alpha]$, $\alpha \in [0,1]$. In this case, the step size is h = 1.257 $\times 10^{-4}$.

The fuzzy solution is periodic with period $T = 2\pi$. For both Meth = 1 and Meth = 2, there are three internal switching points at $tw \in Sw = \{\pi, 2\pi, 3\pi\}$, where the length of $\dot{x}_{gH}$ is zero and the length of the solution $x(t)$ is maximal (at $tw = \pi, 3\pi$) or minimal (at $tw = 2\pi$.) At points $t = 2\pi$ and $t = 4\pi$ the solution coincides with the initial condition (see Figures 15 and 16).

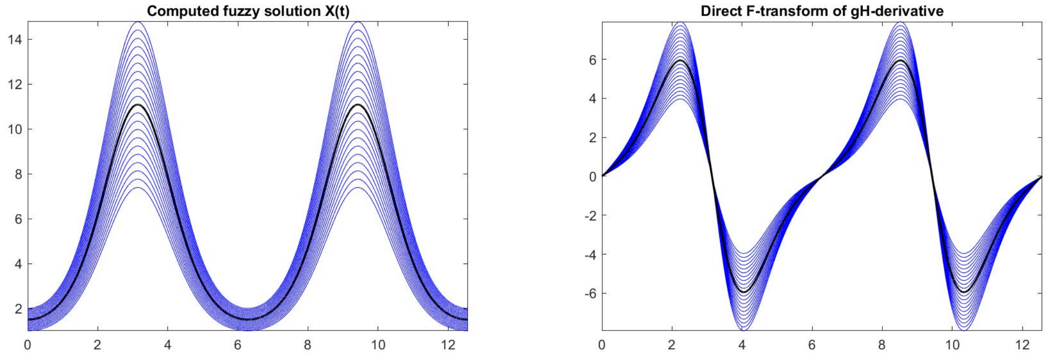

Figure 15. Problem FDE2, Meth = 1: gH-differentiable solution (**left**) and its gH-derivative (**right**). There are three switching points, in the same position for all α-cuts.

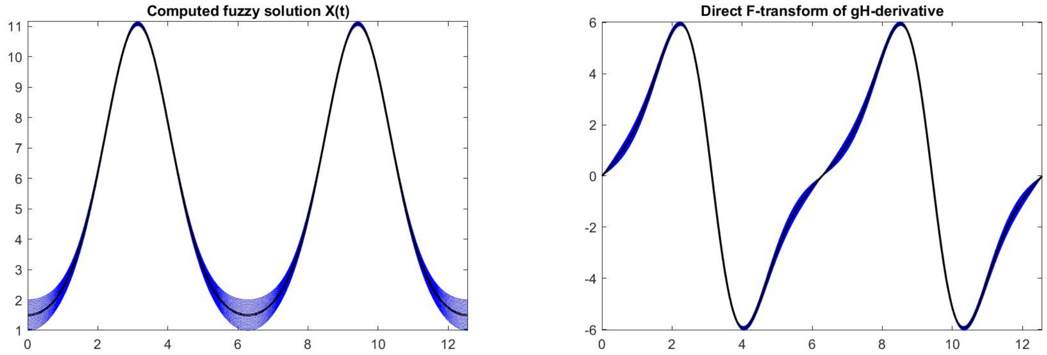

Figure 16. Problem FDE2, Meth = 2: gH-differentiable solution (**left**) and its gH-derivative (**right**). There are three switching points, in the same position for all α-cuts.

7. Concluding Comments and Further Work

In this paper, we see that the F-transform approximation setting allows good numerical procedures to solve ordinary differential equations (ODEs) and to approach the numerical solution of interval and fuzzy differential equations. The computational comparison of the proposed ODE-FT method with other well-known and well behaving numerical routines available in MATLAB, such as ode45, ode15i or ode113, positions F-transform among the most promising mathematical tools for the approximation of functions.

One of our conclusions is then that developing numerical procedures based on F-transform is a promising area of research, anticipated by some successes in recent research such as [13–16]; a complete comparison of (and between) the various F-transform-based proposed methods and our approach was not a scope of our study, where we have chosen standard well-performing routines as benchmarks and we have evaluated algorithm ODE-FT with the same and sufficiently small step size h on typical (including hard) ODEs. Two of the examples in Section 4 are also presented in [14], where the quantity MSE (mean squared error of approximate solution and the exact one) is computed. Ex4 is Example 1 in [14] and Ex5 is Example 3 in [14]. The best MSE quantities obtained by [14] and by ODE-FT are reported in Table 6:

Table 6. For the two ODE problems in example Ex4 and Ex5, using different step-sizes, the table contains the computed MSE for the two solution-variables $x_1(t)$ and $x_2(t)$, obtained by ODE-FT and Scheme II in [14].

Example/(Algorithm, Step Size)	$MSE(x_1)$	$MSE(x_2)$
Ex4/(Scheme II in [14], h = 0.01)	2.241×10^{-2}	3.775×10^{-4}
Ex4/(ODE-FT, h = 0.01)	1.865×10^{-6}	2.027×10^{-6}
Ex5/(Scheme II in [14], h = 0.1)	1.721×10^{-5}	4.102×10^{-5}
Ex5/(ODE-FT, h = 0.1)	1.242×10^{-6}	1.685×10^{-5}
Ex5/(ODE-FT, h = 0.01)	1.264×10^{-10}	2.693×10^{-9}

It seems that in general, the different algorithms behave similarly, at least for the chosen step size $h = 0.1$ and $h = 0.01$ (consider that such h is a big one and values around $h = 0.00001$ or less are more adequate for a comparison, as suggested, e.g., by the values used in routine **ode45** that chooses h dynamically). Possibly, more efficient and elaborated implementations of the proposed algorithms will require some additional analysis of F-transform properties to allow variable-order and/or step-size control.

As a tool for numerical solution of interval (IDEs) and fuzzy (FDEs) differential equations in terms of gH-derivative, the F-transform allows an immediate approach to handle the switching phenomenon, a still open problem in this area. This approach is obtained by the application of Equation (43) and proposition 8, which is possible because the interval-valued function $H(t)$ offers an approximation of the interval solution $x(t)$ at all points $t \in [t_0, t_1]$ and not only at the discretized points p_k, as usual in the (explicit) single- or multi-step ODE solvers.

Further research can be planned in the design and experimentation of efficient numerical procedures to solve real-world applications. In this direction, a possible improvement in the approximation can be obtained by higher-order F^d-transform (see [11] for recent results of its properties), by introducing local polynomials or, more generally, local parametric functions $F_k(t;\theta)$ in place of constant direct components F_k (coefficients of the polynomials or parameters $\theta \in \mathbb{R}^d$ are then estimated by least squares). In these cases, the inverse F-transform of a function $f(t)$ on $[a, b]$ has the form $f^d_{(\mathbb{P},\mathbb{A})}(t) = \sum\limits_{k=1}^{n} F_k(t;\theta^{(k)})A_k(t)$, with estimated parameters $\theta^{(k)}$ for the k-th direct component. It is worth to remark that the same integral property used in this paper for the standard F-transform $f_{(\mathbb{P},\mathbb{A})}(t)$ is also valid for $f^d_{(\mathbb{P},\mathbb{A})}(t)$, i.e.,

$$\int\limits_a^b f(t)dt = \int\limits_a^b f^d_{(\mathbb{P},\mathbb{A})}(t)dt = \sum_{k=1}^{n} \int\limits_a^b F_k(t;\theta^{(k)})A_k(t)dt. \tag{49}$$

It should be interesting to see if higher-order F-transform approximations will be able to generate high orders $O(h^q), q > 2$ of convergence, and to obtain possibly increasing orders q by increasing d (two numeric schemes of order $q = 2$ based on the F^2-transform are obtained in [16]).

Similar results can be obtained by considering the discrete F-transform $f_{(\mathbb{P},\mathbb{A})}(t_j)$ on a data set of points $\mathbb{S} = \{(t_j, f_j), j = 1, 2, ..., m\}$; the integrals are substituted by summations and we have (see [10])

$$\sum_{j=1}^{m} f_{(\mathbb{P},\mathbb{A})}(t_j) = \sum_{j=1}^{m} f_j. \tag{50}$$

Finally, it is worth mentioning the possibility of applying the ideas presented in this paper to the numerical solution of other kinds of differential and integral equations, such as delay differential equations (e.g., [28]), differential equations on time scales, fractional differential equations ([29]) and *implicit* differential algebraic equations (DAE, see, e.g., [30,31]); a general DAE with additional constraints, on an interval $[t_0, t_1]$ is expressed in the form

$$\begin{cases} F\left(t, x(t), \dot{x}\,(t)\right) = 0 \\ G\left(t, x(t)\right) = 0 \\ H\left(t, x(t)\right) \leq 0 \\ x\left(t_0\right) = x_0. \end{cases} \tag{51}$$

In this cases, the discretization of $[a, b]$ by a fuzzy partition $(\mathbb{P}, \mathbb{A})$ and the substitution of $\dot{x}\,(t)$ and $x(t)$ with the functions $\dot{x}_{(\mathbb{P},\mathbb{A})}(t)$ and $x_{(\mathbb{P},\mathbb{A},x_0)}(t)$ at points $p_k \in \mathbb{P}$ will transform the DAE into a standard feasibility problem, consisting of finding feasible solutions for the transformed system at points p_k, $k = 1, ..., n$.

Author Contributions: All authors contributed equally to the final version of this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: Davide Radi gratefully acknowledges the support of the Czech Science Foundation (GACR) under project [20-16701S] and the VŠB-TU Ostrava under the SGS project SP2020/11.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Perfilieva, I. Fuzzy Transforms: Theory and Applications. *Fuzzy Sets Syst.* **2006**, *157*, 993–1023. [CrossRef]
2. Perfilieva, I. Fuzzy Transforms: A challenge to conventional transform. In *Advances in Images and Electron Physics*; Hawkes, P.W., Ed.; Elsevier Academic Press: Cambridge, CA, USA, 2007; Volume 147, pp. 137–196.
3. Coroianu, L.; Stefanini, L. General approximation of fuzzy numbers by F-transform. *Fuzzy Sets Syst.* **2016**, *288*, 46–74. [CrossRef]
4. Coroianu, L.; Stefanini, L. Properties of fuzzy transform obtained from L_p minimization and a connection with Zadeh's extension principle. *Inf. Sci.* **2019**, *478*, 331–354. [CrossRef]
5. Guerra, M.L.; Stefanini, L. Quantile and expectile smoothing based on L_1-norm and L_2-norm fuzzy transforms. *Int. J. Approx. Reason.* **2019**, *107*, 17–43. [CrossRef]
6. Perfilieva, I.; de Baets, B. Fuzzy transforms of monotone functions with applications to image compression. *Inf. Sci.* **2010**, *180*, 3304–3315. [CrossRef]
7. Perfilieva, I.; Novak, V.; Dvorak, A. Fuzzy transform in the analysis of data. *Int. J. Approx. Reason.* **2008**, *48*, 36–46. [CrossRef]
8. Stefanini, L. Fuzzy Transform with Parametric LU-Fuzzy Partitions. In *Computational Intelligence in Decision and Control*; Ruan et Al, D., Ed.; World Scientific: Singapore, 2008; pp. 399–404.
9. Stefanini, L. Fuzzy Transform and Smooth Functions. In Proceedings of the IFSA-EUSFLAT 2009 Conference, Lisbon, Portugal, 20–24 July 2009; pp. 579–584.
10. Stefanini, L. F-Transform with Parametric Generalized Fuzzy Partitions. *Fuzzy Sets Syst.* **2011**, *180*, 98–120. [CrossRef]
11. Zeinali, M.; Alikhani, R.; Shahmorad, S.; Bahrani, F.; Perfilieva, I. On the structural properties of F^m-transform with applications. *Fuzzy Sets Syst.* **2018**, *342*, 32–52. [CrossRef]
12. Ahmad, M.Z.; Hasan, M.K.; de Baets, B. Analytical and numerical solutions of fuzzy differential equations. *Inf. Sci.* **2013**, *236*, 156–167. [CrossRef]
13. Kasasbeh, H.A.L.; Perfilieva, I.; Ahmad, M.Z.; Yahya, Z.R. New Fuzzy Numerical Methods for Solving Cauchy Problems. *Appl. Syst. Innov.* **2018**, *1*, 15. [CrossRef]
14. Kasasbeh, H.A.L.; Perfilieva, I.; Ahmad, M.Z.; Yahya, Z.R. New Approximation Methods Based on Fuzzy Transform for solving SODEs: I. *Appl. Syst. Innov.* **2018**, *1*, 29. [CrossRef]
15. Kasasbeh, H.A.L.; 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. [CrossRef]

16. 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]
17. Radi, D.; Stefanini, L. Fuzzy Differential Equations by F-Transform. In Proceedings of the NAFIPS 2015 Annual Meeting, Redmond, WA, USA, 17–19 August 2015; pp. 384–389.
18. Bede, B.; Stefanini, L. Generalized differentiability of fuzzy-valued functions. *Fuzzy Sets Syst.* **2013**, *230*, 119–141; doi:10.1016/j.fss.2012.10.003. [CrossRef]
19. Stefanini, L.; Bede, B. Generalized Hukuhara differentiability of interval-valued functions and interval differential equations. *Nonlinear Anal.* **2009**, *71*, 1311–1328. [CrossRef]
20. Stefanini, L.; Bede, B. Generalized fuzzy differentiability with LU-parametric representation. *Fuzzy Sets Syst.* **2014**, *257*, 184–203. [CrossRef]
21. Bede, B. *Mathematics of Fuzzy Sets and Fuzzy Logic*; Springer: Berlin, Germany, 2013.
22. Gomes, L.T.; de Barros, L.C.; Bede, B. *Fuzzy Differential Equations in Various Approaches*; Springer: Berlin, Germany, 2015.
23. Stefanini, L. A generalization of Hukuhara difference and division for interval and fuzzy arithmetic. *Fuzzy Sets Syst.* **2010**, *161*, 1564–1584. [CrossRef]
24. Forsythe, G.E.; Malcolm, M.A.; Moler, C.B. *Computer Methods for Mathematical Computations*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1977.
25. Stefanini, L.; Arana-Jimenez, M. Karush-Kuhn-Tucker conditions for interval and fuzzy optimization in several variables under total and directional generalized differentiability. *Fuzzy Sets Syst.* **2019**, *362*, 1–34. [CrossRef]
26. Stefanini, L.; Guerra, M.L.; Amicizia, B. Interval Analysis and Calculus for Interval-Valued Functions of a Single Variable. Part I: Partial Orders, gH-derivative, Momotonicity. *Axioms* **2019**, *8*, 113. [CrossRef]
27. Stefanini, L.; Sorini, L.; Amicizia, B. Interval Analysis and Calculus for Interval-Valued Functions of a Single Variable. Part II: Extremal Points, Convexity, Periodicity. *Axioms* **2019**, *8*, 114. [CrossRef]
28. 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]
29. Stamova, I.M.; Stamov, G.T. *Functional and Impulsive Differential Equations of Fractional Order—Qualitative Analysis and Applications*; CRC Press: Boca Raton, FL, USA, 2017.
30. Ascher, U.M.; Petzold, L.R. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*; SIAM: Philadelphia, PA, USA, 1998.
31. Kunkel, P.; Mehrmann, V. *Differential-Algebraic Equations—Analysis and Numerical Solution*; European Mathematical Society: Zürich, Switzerland, 2006.