



# **Hybrid Fuzzy Differential Equations and Different Numerical Solutions**

**By**

**Marah Adnan Subuh**

**Supervisor**

**Prof. Dr. Saed Mallak**

**Cosupervisor**

**Prof. Dr. Basem Attili**

**This Thesis was submitted in partial fulfillment of the  
requirements for the Master's Degree of Science in  
Mathematical Modeling**

**Faculty of Graduate Studies  
Palestinian Technical University-Kadoorie**

**December 2020**



## المعادلات التفاضلية الضبابية الهجينية وحلول عددية مختلفة

إعداد

مرح عدنان صبح

المشرف

أ.د. سائد ملاك

المشرف المشارك

أ.د. باسم عتيلي

قدمت هذه الرسالة استكمالاً لمتطلبات الحصول على درجة الماجستير  
في النمذجة الرياضية

كلية الدراسات العليا

جامعة فلسطين التقنية – خضوري

كانون أول ٢٠٢٠

**Palestine Technical University –Kadoorie**

**PTUK**

**Authorization form**

I, Marah Adnan Mahmoud Subuh, authorize the PTUK to supply copies of my thesis / dissertation to libraries or establishments or individuals on request, according to the university of PTUK regulations.

Signature: .....

Date: .....

**جامعة فلسطين التقنية - خضوري**

**نموذج التفويض**

أنا مرح عدنان محمود صبح، أفوض جامعة فلسطين التقنية خضوري بتزويد نسخ من رسالتي / أطروحتي للمكتبات والمؤسسات أو الهيئات أو الأشخاص عند طلبهم حسب التعليمات النافذة في الجامعة.

التوقيع: .....

التاريخ: .....

## الإقرار

أنا الموقعة أدناه مقدمة الرسالة التي تحمل العنوان:

### Hybrid Fuzzy Differential Equations and Different Numerical Solutions

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل، أو أي جزء منها لم يقدم من قبل لنيل درجة علمية لدى مؤسسة تعليمية أو بحثية أخرى.

### Declaration

I hereby declare that this thesis is the product of my own efforts, except what has been referred to, and this thesis as a whole or any part of it has not been submitted as a requirement for attaining a scientific degree to any other educational or research institution.

**Marah Adnan Mahmoud Subuh**

**Signature:** .....

**Date:** .....

**مرح عدنان محمود صبح**

**التوقيع:** .....

**التاريخ:** .....

## **COMMITTEE DECISION**

This thesis/dissertation (Hybrid Fuzzy Differential Equations and Different Numerical Solutions), was successfully defended and approved on 10/1/2021

**Examination committee**

**Signature**

-Prof. Dr. Saed Mallak (Supervisor) .....

-Prof. Dr. Basem Attili (Cosupervisor) .....

-Asst. Prof. Dr. Hadi Hamad (External Examiner) .....

- Asst. Prof. Dr. Rania Wannan (Internal Examiner) .....

## **DEDICATION**

I dedicate this thesis to my wonderful parents, my sister and brothers,  
my husband and my friends for their love, unfailing support and  
continuous encouragement throughout my life.

To my little baby, whom I haven't seen yet.

To everyone who inspires me by his/her science.

## **AKNOWLEDGEMENT**

After thanking Allah, who granted me the ability to finish this work, I would like to express my sincere gratitude to my thesis supervisors: Prof. Dr. Saed Mallak for all his guidance, understanding, support and sound advice in all aspects of my research work and Prof. Dr. Basem Atilli for his scientific support, encouragement, guidance and constructive advice. I appreciate help and support from members of the Mathematical Department at Palestine Technical University – Kadoorie.

I would like to thank my family, especially my parents, for their encouragement, patience, and assistance over the years. I'm forever indebted to my parents, who have always kept me in their prayers, thanks to my sister and brothers for being a great source of support, love and encouragement which have always carried me through the way. My greatest gratitude to my husband for his care and unreserved support which gave me the courage and confidence that I need in my life, and my warm thanks to my best friend (Raghad Tweir) for tirelessly supporting, motivating and helping me.

Finally, I am very thankful to all who gave me advised, helped and encouraged me all time to get here.

## Contents

Subject	Page
Committee Decision	I
Dedication	II
Acknowledgment	III
Contents	IV
Abstract	VI
Abstract in Arabic	VII
<b>Chapter One</b>	<b>Introduction</b>
<b>Chapter Two</b>	<b>Basic Concepts</b>
2.1 Fuzzy Sets and Fuzzy Numbers	4
2.2 Hybrid Fuzzy Differential System (HFDS)	12
2.2.1 Hybrid System	12
2.2.2 Fuzzy Differential Equation	13
2.2.3 Hybrid Fuzzy Differential Equation	14
2.3 Numerical Methods	16
2.3.1 Picard's Method	16
2.3.2 Euler Method	16
2.3.3 Runge-Kutta Method	17
2.3.4 Predictor-Corrector Method	18
2.3.5 Linear Multistep Method	18
2.3.6 General Linear Methods (GLM)	21
2.3.7 Variational Iteration Method (VIM)	23
2.3.8 Adomian Decomposition Method (ADM)	24
<b>Chapter Three</b>	<b>Numerical Solutions for HFDEs with Triangular Fuzzy Number as Initial Conditions</b>
3.1 Numerical Example	26
3.2 Picard's Method	28
3.3 Runge-Kutta of Order Five Method	39
3.4 General Linear Methods (GLM)	54

---

3.5 Variational Iteration Method (VIM)	79	
3.6 Adomian Decomposition Method (ADM)	87	
3.7 Predictor-Corrector Method	109	
3.7.1 Milne's Four Step Method	109	
3.7.2 Adams-Bashforth Four Step Method	111	
3.8 Improved Predictor-Corrector Method	120	
3.9 Summery	129	
<b>Chapter Four</b>	Numerical Solutions for HFDEs with Selected Types of Fuzzy Numbers as Initial Conditions	130
4.1 Picard's Method	130	
4.2 Runge-Kutta of Order Five Method	137	
4.3 General Linear Methods (GLM)	144	
4.4 Variational Iteration Method (VIM)	157	
4.5 Adomian Decomposition Method (ADM)	160	
4.6 Summery	166	
<b>Chapter Five</b>	Conclusion and Comments	167
	References	169
	Appendix	174

---

# **Hybrid Fuzzy Differential Equations and Different Numerical Solutions**

By: Marah Subuh

Supervised by Prof. Dr. Saed Mallak and Prof. Dr. Basem Atilli

## **Abstract**

In this thesis, hybrid fuzzy differential equations (HFDEs) are considered and solved under Hukuhara derivative by several numerical methods. We study hybrid fuzzy differential equations with different fuzzy initial conditions using different types of fuzzy numbers (triangular, trapezoidal and triangular shaped fuzzy numbers). To the best of our knowledge, it is the first time to use trapezoidal and triangular shaped fuzzy numbers as initial conditions. We have solved the HFDE's under generalized Hukuhara using different numerical methods with a model to illustrate each methods. A Matlab code was built for each of the methods to find the exact solution and to approximate the solution numerically and represent it graphically. Finally, accurate results were obtained for most numerical methods used with different types of fuzzy numbers as initial conditions.

# **المعادلات التفاضلية الضبابية الهجينية وحلول عددية مختلفة**

إعداد: مرح صبح

بإشراف أ.د. سائد ملاك وأ.د. باسم عتيلي

## **الملخص**

في هذه الأطروحة، سيتم التعامل مع المعادلات التفاضلية الضبابية الهجينية وحلها باستخدام مشتقه (Hukuhara) بعدة طرق عددية. حيث يقوم بدراسة المعادلات التفاضلية الهجينية الضبابية بشروط أولية ضبابية مختلفة باستخدام عدة أنواع من الأرقام الضبابية (الأرقام الضبابية المثلثية والمنحرفة وشبه المثلثية). على حد علمنا، هذه هي المرة الأولى لاستخدام رقم ضبابي شبه منحرف أو رقم ضبابي شبه مثلثي كشرط أولي. يقوم بحل المعادلات التفاضلية الضبابية الهجينية تحت مشتقه (Hukuhara) المعتمدة بعدة طرق عددية. بالإضافة إلى ذلك، نوضح نموذج على كل طريقة. فقد قمنا بإنشاء برنامج بلغة ماتلاب لكل من الطرق العددية المستخدمة وإيجاد الحلول الدقيقة والتقريرية وتمثيلها بيانياً. وفي الختام، تم الحصول على نتائج دقيقة لمعظم الطرق العددية المستخدمة مع أنواع مختلفة من الأرقام الضبابية كشروط أولية.

# **Chapter One**

## **Introduction**

Hybrid systems are dedicated to modeling, design, and validation of interactive systems of computer programs and continuous systems. That is, complicated systems which have discrete event dynamics, as well as, continuous time dynamics, can be modeled by hybrid systems. The differential systems containing fuzzy valued functions and at the same time interacting with a discrete time controller, these systems are named hybrid fuzzy differential systems [27, 51].

In most situations of our life, uncertainty can appear with real world problems. To recognize these problems, Lotfi A. Zadeh in 1965 defined the fuzzy logic and the concept of fuzzy set by working on the problem of computer understanding of natural language which led to the definition of the fuzzy number [20, 55]. Fuzzy numbers have been used to obtain better results in problems where decision making and analysis are involved. A Fuzzy number, which is an extension of a real number, has its own properties which can be related to number theory [18].

Fuzzy logic is a very powerful tool in dealing with complex problems. Engineers and scientists generally encounter problems which are impossible to solve numerically using traditional mathematical rules. By making use of fuzzy logic, one can characterize and control a system whose model is not known or is ill-defined. Fuzzy logic is extremely useful for many people involved in research and development [6, 49, 53].

The fuzzy differential calculus was developed by different authors, like Dubois [21], Puri and Ralescu [48], Kaleva [32] and Seikkala [52]. While the concept of the fuzzy derivative was first introduced by Chang and Zadeh [17], it was followed up by many researchers see [3, 33]. Bede and Gal in [8] proposed a new concept of derivative based on the Hukuhara difference, later on, Puri and Ralescu [48] defined Hukuhara derivative.

The role of differential equations in the modelling of real world phenomena, is very vast. Though on a broader view, the initial conditions are considered to be exactly defined within a model, certain errors in observation, measurement, or experimented data may lead to fuzzy, incorrect or incomplete information[18].

Fuzzy differential equations (FDEs) can be used to over-rule such uncertainties or lack of precision. In 1987, Kandel and Byatt [38] presented the fuzzy differential equations and fuzzy Hukuhara derivative and generalization [4, 5, 8, 9, 16, 25, 36, 37, 50].

Of particular interest is the use of Hybrid Fuzzy Differential Equations (HFDEs) as a natural way to model control systems with embedded uncertainty. There are several numerical methods to solve HFDEs. The numerical solution of FDEs by Euler's method was studied by Ma et al. [40]. Abbasbandy and Allviranloo [1, 2] proposed the Taylor method and the Runge-kutta method for solving FDEs.

Consider the initial value problems for hybrid differential equation (HDE)[19] ,

$$\begin{cases} \frac{d}{dt}[y(t) - f(t, y(t))] = g(t, x(t)), & t \in [0, T] \\ y(t_0) = y_0 \in \mathbb{R} \end{cases}$$

where,  $f, g: [0, T] \times \mathbb{R} \rightarrow \mathbb{R}$  are continuous functions.

The hybrid fuzzy differential systems

$$\begin{cases} y'(t) = f(t, y, \lambda_k(y_k)), & t \in [t_k, t_{k+1}], \\ y(t_k) = y_k \end{cases}$$

Where  $0 \leq t_0 < t_1 < \dots < t_k < \dots, t_k \rightarrow \infty$ ,  $f \in C[\mathbb{R}^+ \times E \times E, E]$ ,  $\lambda_k \in C[E, E]$ , where  $E$  is the set of all fuzzy numbers.

Pederson and Sambandham [46,47] used the Euler and Rung-Kutta methods for solving the HFDEs. Saveetha and Pandian [45] solved hybrid fuzzy differential equations by applying Runge–Kutta Nystrom method of order three. Kanagarajan and Sambath [35] proposed an improved predictor-corrector (IPC) method to solve HFDEs. Jayakumar [28] studied numerical methods for HFDEs by applying the Runge-Kutta Method of Order Five. Kanagarajan et. al. [34] solved hybrid fuzzy differential equations by applying Runge-Kutta method of order five. K. Ivaz et. al. [27] solved HFDEs by applying Trapezoidal rule. Narayananamoorthy and Mathankumar [41] solved the HFDEs by Picard's Method. Mojtaba Ghanbari [24] solved first order linear FDEs by Variational iteration method (VIM) and Adomian decomposition method (ADM). Farzi and Moradi [22] solved FDEs by Fuzzy General Linear Methods (FGLM).

Narayananamoorthy et. al [42] solved HFDEs using Taylor series method. Some numerical methods can be found in [10, 29, 31, 43].

The reason to study hybrid systems are varied; some sources of motivation are, for example but not limited to, the design of technological systems, networked control systems, and physical processes exhibiting non-smooth behavior. Also, the HFDEs have a wide range of applications in applied sciences and engineering.

In this thesis, the HFDEs will be solved under generalized Hukuhara derivative by several numerical methods and different types of fuzzy numbers as initial conditions.

In both Chapters 1 and 2, an introduction and some basic definitions will be presented also properties of fuzzy sets and fuzzy numbers.

In Chapter 3, the HFDEs will be studied with triangular fuzzy numbers as initial conditions, then a HFDEs will be solved under the generalized Hukuhara derivative using Picard method, Runge-Kutta method of order five, general linear method (GLM), Variational iteration method (VIM), Adomian decomposition method (ADM), Predictor-Corrector method of order four (PCM) and Improved Predictor-Corrector method (IPCM). A Matlab code construct for each method to give a numerical and graphical approximation to the solution in addition to discussing the obtained results.

In Chapter 4, the HFDEs will be studied with trapezoidal and triangular shaped fuzzy numbers as initial conditions, then a HFDEs will be solved under Hukuhara derivative by Picard method, Runge-Kutta method of order five, General Linear method (GLM), Variational iteration method (VIM), Adomian decomposition method (ADM). A Matlab code construct for each method to give a numerical and graphical approximation to the solution in addition to discussing the obtained results.

Finally in Chapter 5, some conclusions and remarks are presented.

## Chapter Two

### Basic Concepts

This Chapter consists of three sections. In section 2.1 some basic definitions and properties will be presented about fuzzy sets and fuzzy numbers then in 2.2 the HFDS will be explained. Finally in 2.3 an overview of some numerical methods will be presented.

#### 2.1 Fuzzy Sets and Fuzzy Numbers

The fuzzy set theory is built to deal with uncertainty phenomenon's such as randomness, ambiguity and imprecision.

**Definition 1 [11, 49]** A classical (crisp) set  $A$  is normally defined as a collection of elements or objects  $x \in X$  that can be finite, countable, or overcountable. Each single element can either belong or not belong to a set  $A$ ,  $A \subseteq X$ . In the former case, the statement " $x$  belongs to  $A$ " is true, whereas in the latter case this statement is false, which can be represented by the membership function

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

**Definition 2 [7, 39, 56]** If  $X$  is a collection of objects denoted generically by  $x$ , then a fuzzy set  $A$  in  $X$  is a set of ordered pairs:

$$A = \{(x, \mu_A(x)) | x \in X\}$$

$\mu_A(x)$  is called the membership function or grade of membership (also degree of compatibility or degree of truth) of  $x$  in  $A$ . In other words, if  $X$  is a collection of objects, then a fuzzy subset  $A$  of  $X$  is defined by the membership function  $\mu_A: X \rightarrow [0,1]$ .

**Definition 3 [56]** The support of a fuzzy set  $A$ ,  $S(A)$ , is the crisp set of all  $x \in X$  such that  $\mu_A(x) > 0$ .

**Definition 4 [11, 39]** Let  $A$  be a fuzzy subset of  $X$  then the  $\alpha$ -cut set denoted by  $A_\alpha$  is made up of members whose membership is not less than  $\alpha$ .

$$A_\alpha = \{x \in X | \mu_A(x) \geq \alpha\}.$$

Note that  $\alpha$  can be chosen arbitrary between  $0 < \alpha \leq 1$ .  $\alpha$ -cut set is a crisp set, the  $\alpha = 0$  cut, or  $A_0$ , must be defined separately.

Alpha-cuts are slices through a fuzzy set producing regular (non-fuzzy) sets.

### Remarks 1 [39, 43]

1.  $\alpha$ -cut set is a closed and bounded interval denoted by  $A_\alpha = [\underline{a}(\alpha), \bar{a}(\alpha)]$ , and clearly,  $A_0 = \{x | u(x) > 0\}$  is compact.
2. For any fuzzy set  $A$ ,  $A_0$  is called the support, or base of  $A$  and  $A_1$  is called the core of  $A$ . See Figure 2.1.1
3. If  $\alpha' < \alpha$  then  $A_\alpha \subseteq A_{\alpha'}, 0 < \alpha' < \alpha < 1$ . See Figure 2.1.2

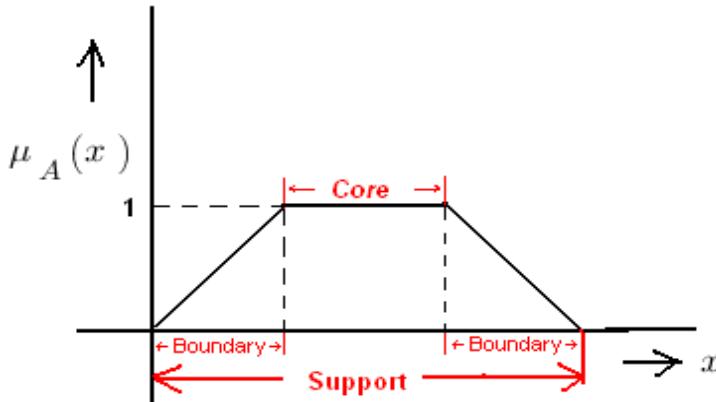


Figure 2.1.1: Support and Core of  $A$

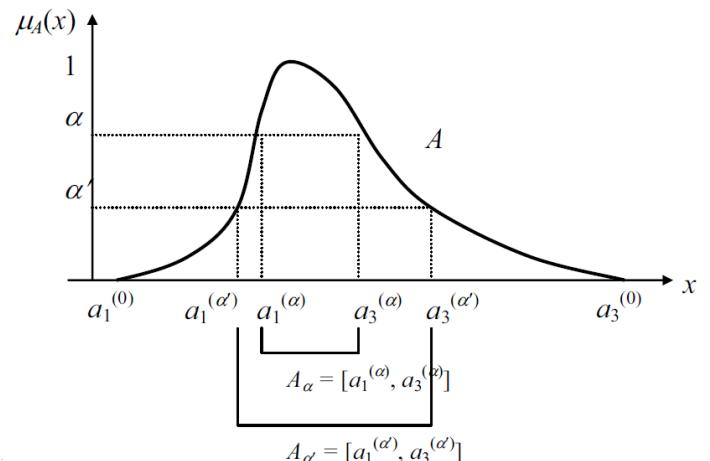


Figure 2.1.2:  $(\alpha' < \alpha) \Rightarrow A[\alpha] \subset A[\alpha']$

**Definition 5 (Convex set) [39]** The term convex is applicable to a set  $A$  in  $\mathbb{R}^n$  ( $n$ -dimensional Euclidian vector space) if the followings are satisfied.

1. Two arbitrary points  $s$  and  $r$  are defined in  $A$   
 $r = (r_i | i \in N_n)$ ,  $s = (s_i | i \in N_n)$ , ( $N_n$  is a set of positive integers)
2. For arbitrary real number  $\alpha$  between 0 and 1, point  $t$  is involved in  $A$  where  $t$  is

$$t = (\alpha r_i + (1 - \alpha)s_i | i \in N_n).$$

See Figure 2.1.3

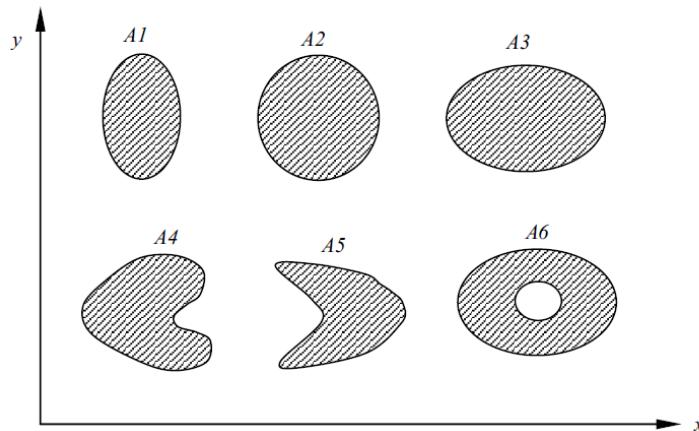


Figure 2.1.3: Convex sets  $A1, A2, A3$  and non-convex sets  $A4, A5, A6$  in  $\mathbb{R}^2$

**Definition 6 (Convex fuzzy set) [39]** Assuming universal set  $X$  is defined in  $n$  dimensional Euclidean Vector space  $\mathbb{R}^n$ . If all the  $\alpha$ - cut sets are convex, the fuzzy set with these  $\alpha$ - cut sets is convex. In other words, if the relation

$$\mu_A(t) \geq \min\{\mu_A(r), \mu_A(s)\}$$

where  $t = \alpha r + (1 - \alpha)s$  such that  $r, s \in \mathbb{R}^n, \alpha \in [0,1]$  holds, the fuzzy set  $A$  is convex. In another formula,  $A$  is convex set if for  $r, s \in \mathbb{R}$  and  $0 \leq \alpha \leq 1$  then  $\mu_A(\alpha r + (1 - \alpha)s) \geq \min\{\mu_A(r), \mu_A(s)\}$ .

See Figure 2.1.4

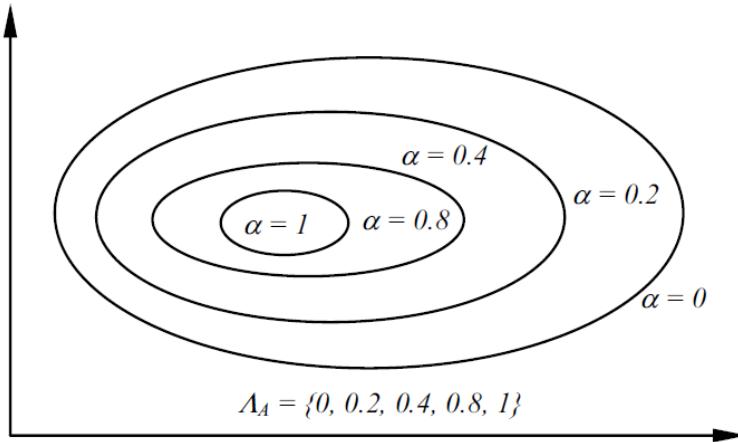


Figure 2.1.4: Convex Fuzzy Set

Figure 2.1.5 shows a convex fuzzy set and Figure 2.1.6 describes a non-convex set.

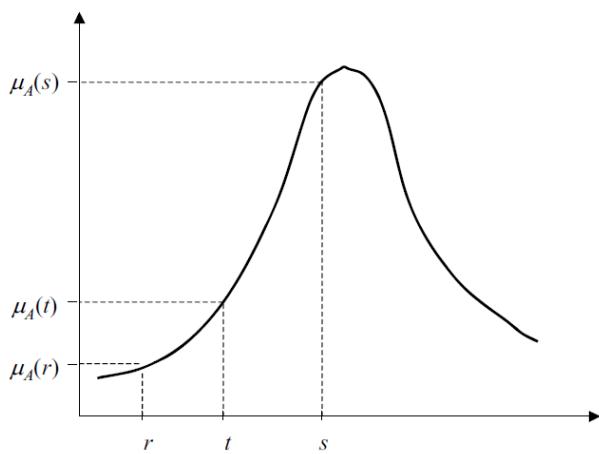


Figure 2.1.5: Convex Fuzzy Set  $\mu_A(t) \geq \mu_A(r)$

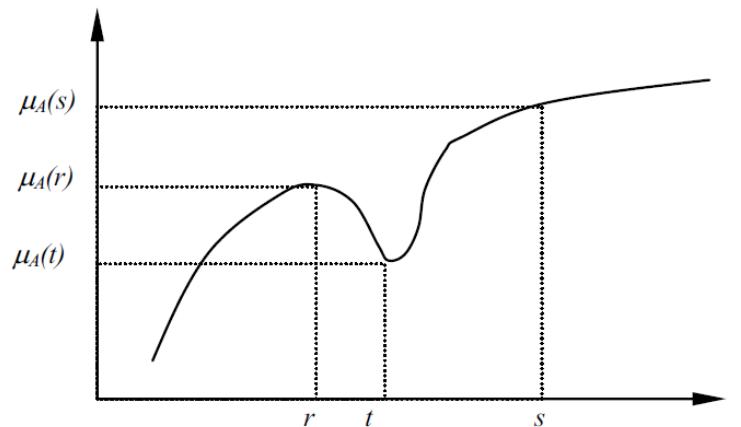


Figure 2.1.6: Non-Convex Fuzzy Set  $\mu_A(t) < \mu_A(r)$

Fuzzy number is expressed as a fuzzy set defining a fuzzy interval in the real number  $\mathbb{R}$ . Since the boundary of this interval is ambiguous, the interval is also a fuzzy set. Generally a fuzzy interval is represented by two end points  $a_1$  and  $a_3$  and a peak point  $a_2$  as  $(a_1, a_2, a_3)$  Figure 2.1.7. If we denote  $\alpha$ -cut interval for fuzzy number  $A$  as  $A_\alpha$ , the obtained interval  $A_\alpha$  is defined as

$$A_\alpha = [a_1(\alpha), a_3(\alpha)].$$

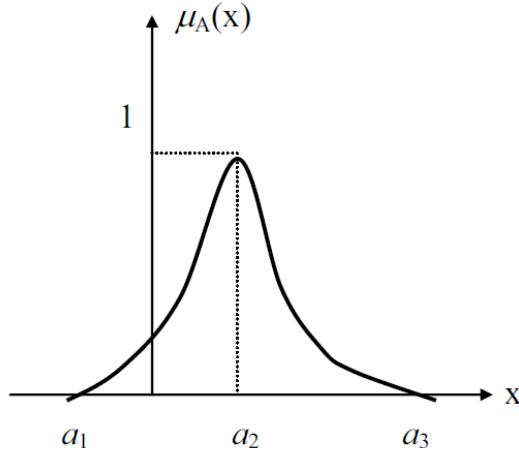


Figure 2.1.7: Fuzzy Number  $A = (a_1, a_2, a_3)$

**Definition 7 [39]** A fuzzy number  $A$  is considered as a fuzzy set if it satisfies the following conditions :

1. Convex fuzzy set.
2. Normalized fuzzy set ( $\exists x \in \mathbb{R}$  such that  $\mu_A(x) = 1$ ).
3. Its membership function is piecewise continuous (i.e.  $A$  is upper semi continuous).
4. Its membership function is defined on the real numbers line.
5.  $A_0 \equiv$  the closure of  $\{x \in \mathbb{R}: \mu_A(x) \geq 0\}$  is compact.

The family of all fuzzy numbers will be denoted by  $R_F$ .

**Remarks 2 [43]** An arbitrary fuzzy number is represented by an ordered pair of functions  $(\underline{r}(\alpha), \bar{r}(\alpha))$ ,  $0 \leq \alpha \leq 1$  that satisfies the following requirements:

- $\underline{r}(\alpha)$  is a bounded left continuous non decreasing function over  $[0,1]$ , with respect to any  $\alpha$ .
- $\bar{r}(\alpha)$  is a bounded right continuous non increasing function over  $[0,1]$ , with respect to any  $\alpha$ .
- $\underline{r}(\alpha) \leq \bar{r}(\alpha)$ ,  $0 \leq \alpha \leq 1$

**Definition 8 (Triangular fuzzy number) [39]** It is a fuzzy number represented with three points as follows :  $A = (a_1, a_2, a_3)$

$$\mu_{(A)}(x) = \begin{cases} 0, & x < a_1 \\ \frac{x - a_1}{a_2 - a_1}, & a_1 \leq x < a_2 \\ \frac{a_3 - x}{a_3 - a_2}, & a_2 \leq x < a_3 \\ 0, & a_3 \leq x. \end{cases}$$

See Figure 2.1.8

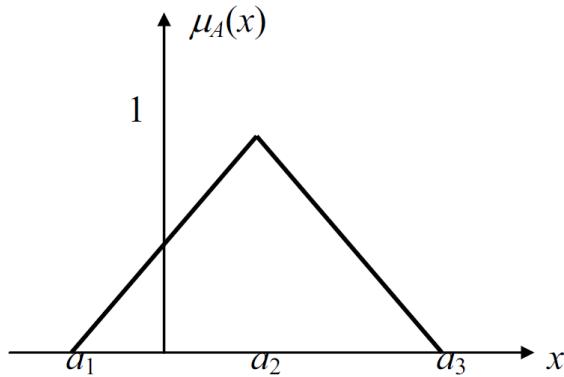


Figure 2.1.8: Triangular fuzzy number  $A = (a_1, a_2, a_3)$

$A_\alpha = [\underline{a}(\alpha), \bar{a}(\alpha)]$  where

$$\underline{a}(\alpha) = a_1 + (a_2 - a_1)\alpha, \quad \bar{a}(\alpha) = a_3 - (a_3 - a_2)\alpha, \quad \forall \alpha \in [0,1].$$

Or a triangular fuzzy number  $A$ , is defined by three numbers  $a_1 < a_2 < a_3$  where the graph of  $\mu_A(x)$ , the membership function of the fuzzy number  $A$ , is a triangle with base on the interval  $[a_1, a_3]$  and vertex at  $x = a_2$ .

If at least one of the graphs described above is not a straight line (curve), then  $A$  is called a triangular shaped fuzzy number and we write  $A \approx (a_1, a_2, a_3)$ .

**Definition 9 [39] (Trapezoidal fuzzy number)** A trapezoidal fuzzy number  $A$  can be defined as  $A = (a_1, a_2, a_3, a_4)$  where

$$\mu_A(x) = \begin{cases} 0, & x < a_1 \\ \frac{x - a_1}{a_2 - a_1}, & a_1 \leq x < a_2 \\ 1, & a_2 \leq x < a_3 \\ \frac{a_4 - x}{a_4 - a_3}, & a_3 \leq x < a_4 \\ 0, & a_4 \leq x. \end{cases}$$

See Figure 2.1.9

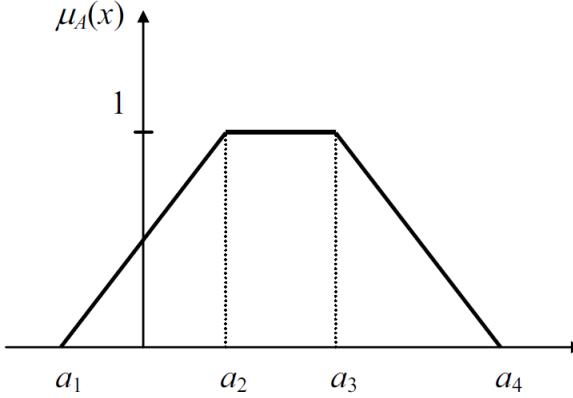


Figure 2.1.9: Trapezoidal fuzzy number  $A = (a_1, a_2, a_3, a_4)$

$\alpha$ - cut interval for this shape is written below:

$$A_\alpha = [(a_2 - a_1)\alpha + a_1, a_4 - (a_4 - a_3)\alpha], \forall \alpha \in [0, 1].$$

If at least one of the graphs described above is not a straight line (curve), then  $A$  is called trapezoidal shaped fuzzy number and we write  $A \approx (a_1, a_2, a_3, a_4)$ .

**Theorem 1 [39]** If  $u$  and  $v$  are two fuzzy numbers and  $\mu$  is real number, then for each  $\alpha \in [0,1]$ , we have:

- 1-  $[u + v]_\alpha = [u]_\alpha + [v]_\alpha = [\underline{u}(\alpha) + \underline{v}(\alpha), \bar{u}(\alpha) + \bar{v}(\alpha)]$
- 2-  $[\mu u]_\alpha = \mu[u]_\alpha = [\min\{\underline{\mu u}(\alpha), \bar{\mu u}(\alpha)\}, \max\{\underline{\mu u}(\alpha), \bar{\mu u}(\alpha)\}]$ .
- 3-  $[uv]_\alpha = [\min\{\underline{u}(\alpha)\underline{v}(\alpha), \underline{u}(\alpha)\bar{v}(\alpha), \bar{u}(\alpha)\underline{v}(\alpha), \bar{u}(\alpha)\bar{v}(\alpha)\}, \max\{\underline{u}(\alpha)\underline{v}(\alpha), \underline{u}(\alpha)\bar{v}(\alpha), \bar{u}(\alpha)\underline{v}(\alpha), \bar{u}(\alpha)\bar{v}(\alpha)\}]$ .

**Definition 10 [25]** Let  $u, v \in R_F$ . If there exists fuzzy number  $w$  such that  $u = v + w$  then  $w$  is called the Hukuhara difference ( $H$ -difference) of  $u$  and  $v$ , and it is denoted by  $u \ominus v$ .

**Definition 11 [25]** Given two fuzzy number  $u, v$ , the  $H$ -difference  $u \ominus v = w$  is a fuzzy number  $w$  such that  $u = v + w$ , if it exists and

$$\alpha-\text{cut } [u \ominus v]_\alpha = [\underline{u}(\alpha) - \underline{v}(\alpha), \bar{u}(\alpha) - \bar{v}(\alpha)], \forall \alpha \in [0,1].$$

### Remarks 3 [25]

- The Hukuhara difference has the property  $u \ominus u = \{0\}$ .
- This difference is not defined for a pair of fuzzy numbers such that the support of a fuzzy number has bigger diameter than the one that is subtracted.

**Definition 12 [25]** A function  $f: [a, b] \rightarrow R_F$  is said to be a fuzzy function. Then  $f$  is called Hukuhara differentiable ( $H$ - differentiable). The derivative  $f'(x_0)$  is defined by

$$f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0 + h) \ominus f(x_0)}{h}, \quad f'(x_0) = \lim_{h \rightarrow 0} \frac{f(x_0) \ominus f(x_0 - h)}{h}$$

if the limits exist and equal then the derivative  $f'(x_0)$  exists.

**Definition 13 [25]** Let  $f : [a, b] \rightarrow R_F$ . If

$$[\underline{f}'_\alpha(x_0), \bar{f}'_\alpha(x_0)]$$

exists for all  $\alpha \in [0, 1]$  and defines the  $\alpha$ -cuts of a fuzzy number  $f'_S(x_0)$ , then  $f$  is Seikkala differentiable at  $x_0$  and  $f'_S(x_0)$ , is the Seikkala derivative of  $f$  at  $x_0$ .

If  $f: [a, b] \rightarrow R_F$  is  $H$ -differentiable, then  $\underline{f}_\alpha(x)$  and  $\bar{f}_\alpha(x)$  are differentiable and  $[f'(x_0)]_\alpha = [\underline{f}'_\alpha(x_0), \bar{f}'_\alpha(x_0)]$ , that is, if  $f$  is  $H$ -differentiable, it is Seikkala differentiable and the derivatives are the same [52].

**Theorem 2 [25]** Let  $f : [a, b] \rightarrow R_F$  be a  $H$ -differentiable function. Then it is continuous.

**Theorem 3 [25]** Let  $f, g : [a, b] \rightarrow R_F$  be  $H$ -differentiable functions and  $\lambda \in \mathbb{R}$ . Then  $(f + g)'_H = f'_H + g'_H$  and  $(\lambda f)'_H = \lambda f'_H$ .

**Remark 4 [25]**

- The  $H$ -difference doesn't always exist, so the  $H$ -differentiable doesn't always exist.
- Let  $f(x) = c \odot g(x)$  where  $f: [a, b] \rightarrow R_F$ ,  $c \in \mathbb{R}_F$ , for all  $x \in [a, b]$ , and let  $g: [a, b] \rightarrow \mathbb{R}^+$  be differentiable at  $x_0 \in [a, b] \subset \mathbb{R}^+$ . If  $g'(x_0) > 0$  then  $f$  is  $H$ -differentiable at  $x_0$  with  $f'(x) = c \odot g'(x)$ . But if  $g'(x) < 0$  then  $f$  is not  $H$ -differentiable.

**Definition 14 [25]** Let  $f: [a, b] \rightarrow R_F$ .  $f$  is strongly generalized differentiable ( $GH$ -differentiable) at  $x_0$  if the limits of some pair of the following exist and equal:

$$\begin{aligned} 1 - & \lim_{h \rightarrow 0} \frac{f(x_0 + h) \ominus f(x_0)}{h} \text{ and } \lim_{h \rightarrow 0} \frac{f(x_0) \ominus f(x_0 - h)}{h}. \\ 2 - & \lim_{h \rightarrow 0} \frac{f(x_0) \ominus f(x_0 + h)}{-h} \text{ and } \lim_{h \rightarrow 0} \frac{f(x_0 - h) \ominus f(x_0)}{-h}. \\ 3 - & \lim_{h \rightarrow 0} \frac{f(x_0 + h) \ominus f(x_0)}{h} \text{ and } \lim_{h \rightarrow 0} \frac{f(x_0 - h) \ominus f(x_0)}{-h}. \\ 4 - & \lim_{h \rightarrow 0} \frac{f(x_0) \ominus f(x_0 + h)}{-h} \text{ and } \lim_{h \rightarrow 0} \frac{f(x_0) \ominus f(x_0 - h)}{-h} \end{aligned}$$

**Definition 15 [25]** Let  $f: [a, b] \rightarrow R_F$ .  $f$  is  $1 -$  differentiable on  $[a, b]$  if  $f$  is differentiable in the sense (1) of definition 14. Similarly,  $f$  is  $2 -$  differentiable on  $[a, b]$  if  $f$  is differentiable in the sense (2) of definition 14.

**Theorem 4 [25]** Let  $f: [a, b] \rightarrow R_F$ . Where  $[f(x)]_\alpha = [\underline{f}_\alpha(x), \bar{f}_\alpha(x)]$  for each  $\alpha \in [0, 1]$ ,

1. If  $f$  is  $1 -$  differentiable , then  $\underline{f}_\alpha$  and  $\bar{f}_\alpha$  are differentiable functions and  $[f'(x)]_\alpha = [\underline{f}'_\alpha(x), \bar{f}'_\alpha(x)]$ .
2. If  $f$  is  $2 -$  differentiable , then  $\underline{f}_\alpha$  and  $\bar{f}_\alpha$  are differentiable functions and  $[f'(x)]_\alpha = [\bar{f}'_\alpha(x), \underline{f}'_\alpha(x)]$ .

**Notation 1 [23]** : For fuzzy linear spline we use the following expression:

$$f_s(t) = \frac{(t_{i+1} - t)}{(t_{i+1} - t_i)} \mu_A(t_i) + \frac{(t - t_{i+1})}{(t_{i+1} - t_i)} \mu_A(t_{i+1}), \quad t \in [t_i, t_{i+1}].$$

## 2.2 Hybrid Fuzzy Differential System (HFDS)

### 2.2.1 Hybrid System

When continuous and discrete dynamics interact, hybrid systems arise. This is especially profound in many technological systems, in which logic decision making and embedded control actions are combined with continuous physical processes. To capture the evolution of these systems, Mathematical models are needed that combine in one way or another the dynamics of the continuous parts of the system with the dynamics of the logic and discrete parts. These mathematical models basically consist of some form of differential or difference equations on one hand and automata or other discrete-event models on the other hand [26].

#### Example [26] (simple hybrid system)

Consider the regulation of the temperature in a house. In a simplified description, the heating system is assumed either to work at its maximum power or to be turned off completely. This is a system that can operate in two modes: “on” and “off”. In each mode of operation (given by the discrete state  $q \in \{\text{on}, \text{off}\}$ ) the evolution of the temperature  $T$  can be described by a differential equation. As such, this system has a hybrid state  $(q, T)$  consisting of a discrete state  $q$  taking the discrete values “on” and “off” from one side, and a continuous state  $T$  taking real values. This is explained in Figure 2.1.1 in which each mode corresponds to a node of a directed graph, while the edges indicate the possible discrete state transitions.

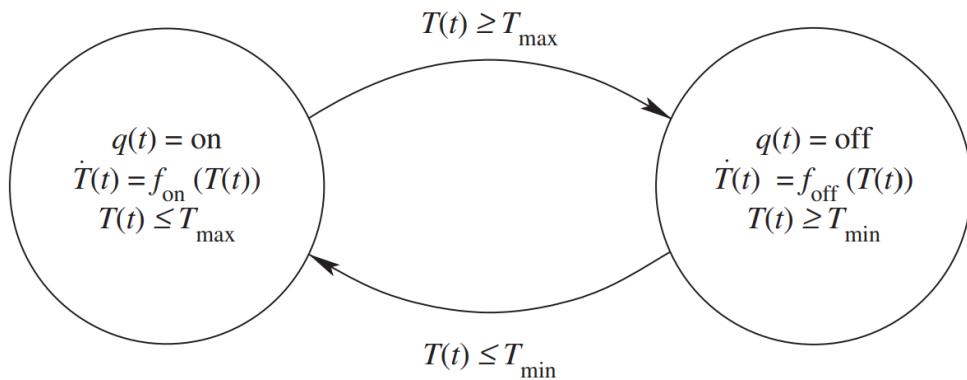


Figure 2.1.1: Model of a temperature control system.

## 2.2.2 Fuzzy Differential Equation

The first order differential equation will be explained when using fuzzy numbers and Seikkala derivative.

Let  $y: I \rightarrow R_F$  where  $I \subset \mathbb{R}$  is an interval. If  $y(t, \alpha) = [\underline{y}(t, \alpha), \bar{y}(t, \alpha)]$  for all  $t \in I$  and  $\alpha \in [0,1]$ , then  $y'(t, \alpha) = [\underline{y}'(t, \alpha), \bar{y}'(t, \alpha)]$  if  $y' \in R_F$ . Next consider the initial value problem (IVP)

$$u(y) = \begin{cases} y'(t) = f(t, y(t)), \\ y(0) = y_0, \end{cases} \quad (1)$$

where  $f: [0, \infty) \times \mathbb{R} \rightarrow \mathbb{R}$  is continuous. We would like to interpret (1) using the Seikkala derivative and  $y_0 \in R_F$ . Let  $y_0(\alpha) = [\underline{y}_0(\alpha), \bar{y}_0(\alpha)]$  and  $y(t, \alpha) = [\underline{y}(t, \alpha), \bar{y}(t, \alpha)]$ , we get  $f: [0, \infty) \times R_F \rightarrow R_F$  where

$$\begin{aligned} [f(t, y)]^\alpha &= [\min\{f(t, u): u \in [\underline{y}^\alpha(t), \bar{y}^\alpha(t)]\}, \max\{f(t, u): u \in [\underline{y}^\alpha(t), \bar{y}^\alpha(t)]\}], \end{aligned}$$

Then  $y: [0, \infty) \rightarrow R_F$  is a solution of (1), using Seikkala derivative and  $y_0 \in R_F$  if

$$(\underline{y}^\alpha)'(t) = \min\{f(t, u): u \in [\underline{y}^\alpha(t), \bar{y}^\alpha(t)]\}, \quad \underline{y}^\alpha(0) = \underline{y}_0^\alpha$$

$$(\bar{y}^\alpha)'(t) = \max\{f(t, u): u \in [\underline{y}^\alpha(t), \bar{y}^\alpha(t)]\}, \quad \bar{y}^\alpha(0) = \bar{y}_0^\alpha$$

for all  $t \in [0, \infty)$  and  $\alpha \in [0,1]$ .

Lastly consider an  $f: [0, \infty) \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  which is continuous and satisfying the IVP:

$$\begin{cases} y'(t) = f(t, y(t), k), \\ y(0) = y_0. \end{cases} \quad (2)$$

To interpret (2) using the seikkala derivative and  $x_0, k \in R_F$ , we use  $f: [0, \infty) \times R_F \times R_F \rightarrow R_F$  where

$$[f(t, y, k)]^\alpha = [\min\{f(t, u, u_k): u \in [\underline{y}^\alpha(t), \bar{y}^\alpha(t)], u_k \in [k^\alpha, \bar{k}^\alpha]\}],$$

$$\max \left\{ f(t, u, u_k) : u \in [\underline{y}^\alpha(t), \bar{y}^\alpha(t)], u_k \in [\underline{k}^\alpha, \bar{k}^\alpha] \right\}$$

Where  $k^\alpha = [\underline{k}^\alpha, \bar{k}^\alpha]$ . Then  $y: [0, \infty) \rightarrow R_F$  is a solution of (2) using the Seikkala derivative and  $y_0, k \in R_F$  if

$$\begin{aligned} (\underline{y}^\alpha)'(t) &= \min \left\{ f(t, u, u_k) : u \in [\underline{y}^\alpha(t), \bar{y}^\alpha(t)], u_k \in [\underline{k}^\alpha, \bar{k}^\alpha] \right\}, \\ \underline{y}^\alpha(0) &= y_0^\alpha \end{aligned}$$

$$\begin{aligned} (\bar{y}^\alpha)'(t) &= \max \left\{ f(t, u, u_k) : u \in [\underline{y}^\alpha(t), \bar{y}^\alpha(t)], u_k \in [\underline{k}^\alpha, \bar{k}^\alpha] \right\}, \\ \bar{y}^\alpha(0) &= \bar{y}_0^\alpha \end{aligned}$$

for all  $t \in [0, \infty)$  and  $\alpha \in [0, 1]$ .

### 2.2.3 Hybrid Fuzzy Differential Equations (HFDEs)

The fuzzy differential equation will be explained when interacting with a discrete time controller which is called HFDEs

Consider the HFDS:

$$\begin{cases} y'(t) = f(t, y(t), \lambda_k(y_k)), & t \in [t_k, t_{k+1}], \\ y(t_k) = y_k \end{cases} \quad (3)$$

Where ' denotes Seikkala differentiation,  $0 \leq t_0 < t_1 < \dots < t_k < \dots, t_k \rightarrow \infty$ ,  $f \in C[\mathbb{R}^+ \times R_F \times R_F, R_F], \lambda_k \in C[R_F, R_F]$ . To be more specific, the system looks like

$$y'(t) = \begin{cases} y'_0(t) = f(t, y_0(t), \lambda_0(y_0)), & y_0(t_0) = y_0, \quad t_0 \leq t < t_1, \\ y'_1(t) = f(t, y_1(t), \lambda_1(y_1)), & y_1(t_1) = y_1, \quad t_1 \leq t < t_2, \\ \dots \\ y'_k(t) = f(t, y_k(t), \lambda_k(y_k)), & y_k(t_k) = y_k, \quad t_k \leq t < t_{k+1}, \\ \dots \end{cases}$$

Assuming that the existence and uniqueness of solution of (1) holds for each  $[t_k, t_{k+1}]$ , by the solution of (1) we mean the following function:

$$y(t) = y(t, t_0, y_0) = \begin{cases} y_0(t), & t_0 \leq t \leq t_1 \\ y_1(t), & t_1 \leq t \leq t_2 \\ \dots \\ y_k(t), & t_k \leq t \leq t_{k+1} \\ \dots \end{cases}$$

We note that the solutions of (3) are piecewise differentiable in each interval for  $t \in [t_k, t_{k+1}]$  for a fixed  $y_k \in R_F$  and  $k = 0, 1, 2, \dots$

Using a representation of fuzzy numbers, we may represent  $y \in R_F$  by pair of functions  $(\underline{y}(\alpha), \bar{y}(\alpha))$ ,  $0 \leq \alpha \leq 1$ , such that

(i)  $\underline{y}(\alpha)$  is bounded, left continuous, and nondecreasing,

(ii)  $\bar{y}(\alpha)$  is bounded, left continuous, and non-increasing, and

(iii)  $\underline{y}(\alpha) \leq \bar{y}(\alpha)$ ,  $0 \leq \alpha \leq 1$ .

Therefore, we may replace (3) by an equivalent system

$$\begin{cases} \underline{y}'(t) = \underline{f}(t, y, \lambda_k(y_k)) \equiv F_k(t, \underline{y}, \bar{y}), & \underline{y}(t_k) = \underline{y}_k \\ \bar{y}'(t) = \bar{f}(t, y, \lambda_k(y_k)) \equiv G_k(t, \underline{y}, \bar{y}), & \bar{y}(t_k) = \bar{y}_k \end{cases} \quad (4)$$

which possesses a unique solution  $(\underline{y}, \bar{y})$  which is a fuzzy function. That is for each  $t$ , the pair  $[\underline{y}(t, \alpha), \bar{y}(t, \alpha)]$  is a fuzzy number, where  $\underline{y}(t, \alpha), \bar{y}(t, \alpha)$  are respectively the solutions of the parametric form given by

$$\begin{cases} \underline{y}'(t, \alpha) = F_k[t, \underline{y}(t, \alpha), \bar{y}(t, \alpha)], & \underline{y}(t_k, \alpha) = \underline{y}_k(\alpha) \\ \bar{y}'(t, \alpha) = G_k[t, \underline{y}(t, \alpha), \bar{y}(t, \alpha)], & \bar{y}(t_k, \alpha) = \bar{y}_k(\alpha) \end{cases} \quad (5)$$

for  $\alpha \in [0, 1]$

## 2.3 Numerical Methods

Our goal is to illustrate various numerical methods to approximate the solution to a first-order prime value problem:

$$y' = f(t, y(t)), \quad t \geq t_0, \quad y(t_0) = y_0 = \alpha \quad (6)$$

see [12, 13, 14, 15, 44, 54]

### 2.3.1 Picard's Method

This approximating method of solving a differential equation is one of the successive approximation methods; that is, it is an iterative method in which the numerical results become more and more accurate the more iterative it is used.

The first order initial value problem (6) can be rewritten as an integral equation:

$$y(t) = y_0 + \int_{t_0}^t f(\tau, y(\tau)) d\tau. \quad (7)$$

Note that the exact solution of IVP is obtained for  $t = t_0$ . A sequence of approximations can be obtained as

$$y_{i+1}(t) = y_0 + \int_{t_0}^t f(\tau, y_i(\tau)) d\tau. \quad (8)$$

That is, the  $i^{\text{th}}$  approximation is inserted into the right hand side of the integral equation in place of the exact solution  $y(t)$  and used to compute the  $(i + 1)^{\text{st}}$  element of the sequence. This process is called Picard's Method.

### 2.3.2 Euler Method

Estimate  $y(t)$  by making the approximation  $f(t, y(t)) \approx f(t_0, y(t_0))$  for  $t \in [t_0, t_0 + h]$ , where  $h > 0$  and  $h$  is sufficiently small. Integrating (6),

$$y(t) = y(t_0) + \int_{t_0}^t f(\tau, y(\tau)) d\tau \approx y_0 + (t - t_0)f(t_0, y_0) \quad (9)$$

Given a sequence,  $t_0, t_1 = t_0 + h, t_2 = t_0 + 2h, \dots$ , where  $h$  is the time step ( $h > 0$ ), Denote by  $y$ , a numerical estimate of the exact solution  $y(t_i), i = 0, 1, \dots$  Motivated by (9), choose

$$y_1 = y_0 + hf(t_0, y_0)$$

This procedure can be continued to produce approximants at  $t_2, t_3$  and so on. In general, we obtain the recursive scheme

$$y_{i+1} = y_i + hf(t_i, y_i), \quad i = 0, 1, \dots, \quad (10)$$

this process is called Euler Method.

While the Modified Euler Method is given by

$$y_{i+1} = y_i + \frac{h}{2} [f(t_i, y_i) + f(t_{i+1}, y_{i+1})], \quad i = 0, 1, \dots \quad (11)$$

Use Euler's method (10) to approximate  $y_{i+1}$  and substitute in (11)

$$y_{i+1} = y_i + \frac{h}{2} [f(t_i, y_i) + f(t_{i+1}, y_i + hf(t_i, y_i))], \quad i = 0, 1, \dots \quad (12)$$

### 2.3.3 Runge-Kutta Method

The basis of all Runge-Kutta method is to express the difference between the value of  $y$  at  $t_{n+1}$  and  $t_n$  as

$$y_{n+1} - y_n = \sum_{i=1}^m w_i k_i \quad (13)$$

where for  $i = 1, 2, \dots, m$ , the  $w_i$ 's are constants and

$$\begin{aligned} k_i &= \\ &hf(t_n + \alpha_i h, y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j) \end{aligned} \quad (14)$$

Runge-Kutta of order four (see Figure 2.3.1)

$$\begin{aligned} k_1 &= hf(t_i, y_i), \\ k_2 &= hf\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right), \\ k_3 &= hf\left(t_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right), \\ k_4 &= hf(t_i + h, y_i + k_3), \\ y_{i+1} &= y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

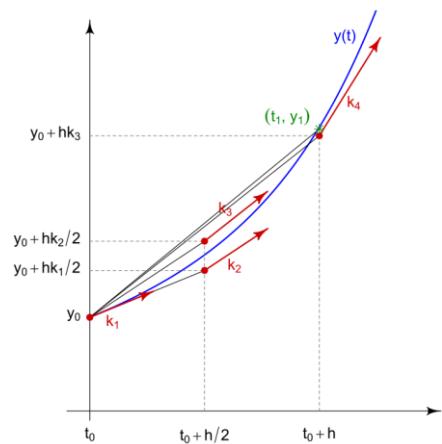


Figure 2.3.1: Slopes used by the Runge-Kutta of order four

Runge-Kutta of order five

$$\begin{aligned}
k_1 &= hf(t_i, y_i), \\
k_2 &= hf\left(t_i + \frac{h}{3}, y_i + \frac{k_1}{3}\right), \\
k_3 &= hf\left(t_i + \frac{h}{3}, y_i + \frac{k_1}{6} + \frac{k_2}{6}\right), \\
k_4 &= hf\left(t_i + \frac{h}{2}, y_i + \frac{k_1}{8} + \frac{3k_3}{8}\right), \\
k_5 &= hf\left(t_i + h, y_i + \frac{k_1}{2} - \frac{3k_3}{2} + 2k_4\right), \\
y_{i+1} &= y_i + \frac{1}{6}(k_1 + 4k_4 + k_5)
\end{aligned} \tag{15}$$

Where

$$a = t_0 < t_1 < \dots < t_N = b, \quad h = \frac{(b-a)}{N} = t_{i+1} - t_i, \quad y_0 = \alpha$$

### 2.3.4 Predictor-Corrector Method

If  $t_i$  and  $t_{i+1}$  are two consecutive mesh points, we have  $t_{i+1} = t_i + h$ . In Euler's method (10), we have

$$y_{i+1} = y_i + hf(t_i, y_i), \quad i = 0, 1, \dots, \tag{16}$$

The modified Euler's method (11), gives

$$y_{i+1} = y_i + \frac{h}{2}[f(t_i, y_i) + f(t_{i+1}, y_{i+1})], \quad i = 0, 1, \dots, \tag{17}$$

The value of  $y_{i+1}$  is first estimated by using (16), then this value is inserted on the right side of (17), giving a better approximation of  $y_{i+1}$ . This step is repeated until two consecutive values of  $y_{i+1}$  agree. This technique of refining an initially crude estimate of  $y_{i+1}$  by means of a more accurate formula is known as predictor-corrector method. The equation (16) is therefore called the predictor while (17) serves as a corrector of  $y_{i+1}$ .

### 2.3.5 Linear Multistep Method

The idea of extending the Euler method by allowing the approximate solution at a point to depend on the solution values and the derivative values at several

previous step values was originally proposed by Bashforth and Adams. Other special types of linear multistep methods were proposed by Nystrom and Milne.

The most important linear multistep methods are of Adams type. That is, the solution approximation at  $t_{i+1}$  is defined as

$$y_{i+1} = a_{m-1}y_i + a_{m-2}y_{i-1} + \cdots + a_0y_{i+1-m} + h(\beta_m f(t_{i+1}, y_{i+1}) + \beta_{m-1}f(t_i, y_i) + \cdots + \beta_0 f(t_{i+1-m}, y_{i+1-m})) \quad (18)$$

for  $i = m-1, m, \dots, N-1$  such that  $a = t_0 < t_1 < \cdots < t_N = b$

$h = \frac{b-a}{N}$  and  $a_0, a_1, a_2, \dots, a_{m-1}, \beta_0, \beta_1, \dots, \beta_m$  are constants with the starting values  $y_0 = \alpha_0, y_1 = \alpha_1, \dots, y_{m-1} = \alpha_{m-1}$ .

For  $\beta_m = 0$ , the method is known as explicit and when  $\beta_m \neq 0$ , the method is known as implicit.

**Definition 16:** Associated with the difference equation (18) the characteristic polynomial of the method is defined by

$$\rho(\lambda) = \lambda^m - a_{m-1}\lambda^{m-1} - a_{m-2}\lambda^{m-2} - \cdots - a_1\lambda - a_0$$

If  $|\lambda^i| \leq 1$  for  $i = 1, 2, 3, \dots, m$  and all roots with absolute value "1" are simple roots, then the difference method is said to satisfy the root condition.

**Theorem 5:** A multistep method of the form (18) is stable if and only it satisfies the root condition.

### Adams–Bashforth methods:

The Adams–Bashforth methods with  $k = 1, 2, 3, 4, 5$  are

$$y(t_i) = y(t_{i-1}) + hf(t_{i-1}, y_{i-1}) \quad \text{order 1}$$

(Euler method)

$$y(t_i) = y(t_{i-1}) + h \left( \frac{3}{2}f(t_{i-1}, y_{i-1}) - \frac{1}{2}f(t_{i-2}, y_{i-2}) \right) \quad \text{order 2}$$

$$y(t_i) = y(t_{i-1}) + h \left( \frac{23}{12}f(t_{i-1}, y_{i-1}) - \frac{16}{12}f(t_{i-2}, y_{i-2}) + \frac{5}{12}f(t_{i-3}, y_{i-3}) \right) \quad \text{order 3}$$

$$y(t_i) = y(t_{i-1}) + h \left( \frac{55}{24} f(t_{i-1}, y_{i-1}) - \frac{59}{24} f(t_{i-2}, y_{i-2}) + \frac{37}{24} f(t_{i-3}, y_{i-3}) - \frac{9}{24} f(t_{i-4}, y_{i-4}) \right) \quad \text{order 4}$$

$$y(t_i) = y(t_{i-1}) + h \left( \frac{1901}{720} f(t_{i-1}, y_{i-1}) - \frac{2774}{720} f(t_{i-2}, y_{i-2}) + \frac{2616}{720} f(t_{i-3}, y_{i-3}) - \frac{1274}{720} f(t_{i-4}, y_{i-4}) + \frac{251}{720} f(t_{i-5}, y_{i-5}) \right) \quad \text{order 5}$$

### Milne's fourth order method:

Explicit formula is

$$y(t_{i+3}) = y(t_{i-1}) + \frac{4h}{3} (2y'_{i+2} - y'_{i+1} + 2y'_i),$$

and implicit formula is

$$y(t_{i+3}) = y(t_{i+1}) + \frac{h}{3} (y'_{i+3} + 4y'_{i+2} + y'_{i+1}).$$

### Adams-Basforth fourth order method:

Explicit formula is

$$y(t_{i+3}) = y(t_{i+2}) + \frac{h}{24} (55y'_{i+2} - 59y'_{i+1} + 37y'_i - 9y'_{i-1}),$$

and implicit formula is

$$y(t_{i+3}) = y(t_{i+2}) + \frac{h}{24} (9y'_{i+3} + 19y'_{i+2} - 5y'_{i+1} + y'_i).$$

Sometimes an explicit multistep method is used to "predict" the value of  $y_{i+1}$ . That value is then used in an implicit formula to "correct" the value. The result is a predictor–corrector method.

### 2.3.6 General Linear Methods (GLM)

The name “general linear methods” applies to a large group of numerical methods for ordinary differential equations. Runge-Kutta (multi-stage) methods are examples of these methods. Linear multistep (multivalue) methods are other examples.

We will describe this in the diagram (Figure 2.3.2):

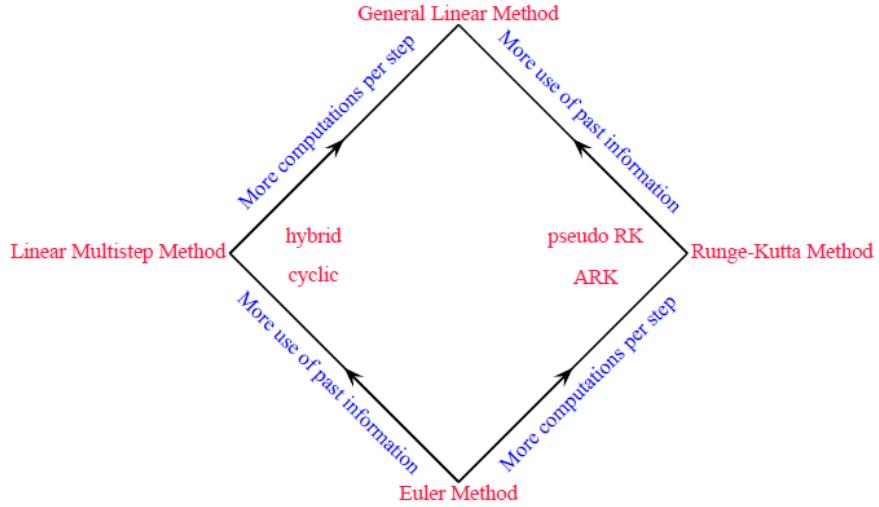


Figure 2.3.2 : General Linear Methods

We will consider  $r$ -value,  $s$ -stage methods, where  $r = 1$  for a Runge–Kutta method and  $s = 1$  for a linear multistep method.

Each step of the computation takes as input a certain number ( $r$ ) of items of data and generates for output the same number of items. The output items correspond to the input items but advanced through one time step ( $h$ ).

Within a step, a certain number ( $s$ ) of stages of computations are performed.

Denote by  $p$  the order of the method and by  $q$  the “stage-order”.

At the start of step number  $n$ , denote the input items by  $y_i^{[n-1]}$ ,  $i = 1, 2, \dots, r$ .

Denote the stages computed in the step and the stage derivatives by  $Y_i$  and  $F_i$  respectively,  $i = 1, 2, \dots, s$ .

For a compact notation, write

$$y^{[n-1]} = \begin{bmatrix} y_1^{[n-1]} \\ y_2^{[n-1]} \\ \vdots \\ y_r^{[n-1]} \end{bmatrix}, \quad y^{[n]} = \begin{bmatrix} y_1^{[n]} \\ y_2^{[n]} \\ \vdots \\ y_r^{[n]} \end{bmatrix}, \quad Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_s \end{bmatrix}, \quad F = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_s \end{bmatrix}$$

The stages are computed by the formula

$$Y_i = \sum_{j=1}^s a_{ij} h F_j + \sum_{j=1}^r u_{ij} y_j^{[n-1]}, \quad i = 1, 2, \dots, s \quad (19)$$

and the output approximations by the formula

$$y_i^{[n]} = \sum_{j=1}^s b_{ij} h F_j + \sum_{j=1}^r v_{ij} y_j^{[n-1]}, \quad i = 1, 2, \dots, r \quad (20)$$

In each case, the coefficients of the general linear formulation are presented in the  $(s+r) \times (s+r)$  partitioned matrix

$$\begin{bmatrix} Y \\ y^{[n]} \end{bmatrix} = \begin{bmatrix} A & U \\ B & V \end{bmatrix} \begin{bmatrix} hF \\ y^{[n-1]} \end{bmatrix} \quad (21)$$

where  $y^{[n-1]}$  and  $y^{[n]}$  are input and output approximations, respectively, and

$$A \in R^{s \times s}, U \in R^{s \times r}, B \in R^{r \times s}, V \in R^{r \times r}.$$

The matrices representing Euler method and implicit Euler methods are, respectively,

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

### 2.3.7 Variational Iteration Method (VIM)

To explain the basic concept of the method, consider the following general nonlinear differential equation given in the form

$$Ly + Ny = g(t) \quad (22)$$

where  $L$  is a linear operator,  $N$  is a nonlinear operator and  $g(t)$  is a known analytical function. The variational iteration method admits the use of a correction at function for equation (22) in the form

$$y_{n+1}(t) = y_n(t) + \int_0^t \lambda(\tau)(Ly_n(\tau) + N\tilde{y}_n(\tau) - g(\tau)) d\tau \quad (23)$$

where  $\lambda(\tau)$  is a general Lagrange multiplier which can be identified optimally via variational theory,  $y_n$  is the  $n$ th approximate solution and  $\tilde{y}_n$  is considered as a restricted variation, which means  $\delta\tilde{y}_n = 0$ . The Lagrange multiplier  $\lambda$  is crucial and critical in the method, and it can be a constant or a function. Having  $\lambda$  determined, an iteration formula should be used for the determination of the successive approximations  $y_{n+1}(t), n \geq 0$  of the solution  $y(t)$ . Consequently, the solution is given by

$$y(t) = \lim_{n \rightarrow \infty} y_n(t) \quad (24)$$

For linear problems, its exact solution can be obtained by only one iteration step due to the fact that the Lagrange multiplier can be exactly identified.

For first order differential equation of the form

$$y' + p(t)y = q(t), \quad y(0) = \alpha$$

$$y_{n+1}(t) = y_n(t) + \int_0^t \lambda(\tau)(y'_n + p(\tau)y_n - q(\tau)) d\tau,$$

Making the above correct functional stationary with respect to  $y_n$ , noticing that  $\delta y_n(0) = 0$  then

$$\begin{aligned} \delta y_{n+1}(t) &= \delta y_n(t) + \delta \int_0^t \lambda(\tau)(y'_n + p(\tau)y_n - q(\tau)) d\tau, \\ &= \delta y_n(t) + \lambda(t)\delta y_n(t) - \int_0^t (\lambda'(\tau) - p(\tau)\lambda(\tau))\delta y_n(\tau) d\tau, \end{aligned}$$

$$= (1 + \lambda(\tau))\delta y_n(t) - \int_0^t (\lambda'(\tau) - p(\tau)\lambda(\tau))\delta y_n(\tau) d\tau, \\ = 0.$$

Therefore, we have the following stationary conditions:

$$\lambda'(\tau) - p(\tau)\lambda(\tau) = 0, \quad 1 + \lambda(\tau) = 0.$$

So, the Lagrange multiplier

$$\begin{aligned} 1 + \lambda(\tau) &= 0 \\ \lambda(\tau) &= -1. \end{aligned}$$

### 2.3.8 Adomian Decomposition Method (ADM)

Consider the differential equation (22), then making  $Ly$  as the subject of the formula, gives

$$Ly = g(t) - Ny. \quad (25)$$

By solving (25) for  $Ly$ , since  $L$  is invertible, we can write

$$L^{-1}Ly = L^{-1}g(t) - L^{-1}Ny. \quad (26)$$

For initial value problems we conveniently define  $L^{-1}$ , for  $L = \frac{d^n}{dt^n}$  as the  $n$ -fold definite integration from 0 to  $t$ . If  $L$  is a second-order operator,  $L^{-1}$  is a two fold integral and so by solving (26) for  $y$ , gives

$$y = A + Bt + L^{-1}g(t) - L^{-1}Ny \quad (27)$$

where  $A$  and  $B$  are constants of integration and can be found from the initial or boundary conditions.

The Adomian method consists of approximating the solution of (22) as an infinite series

$$y(t) = \sum_{n=0}^{\infty} y_n(t) \quad (28)$$

and decomposing the non-linear operator  $N$  as

$$Ny = \sum_{n=0}^{\infty} A_n \quad (29)$$

where the  $A_n$ , depending on  $y_0, y_1, y_2$ , are called the Adomian polynomials, and are obtained for the nonlinearity  $Ny = f(y)$  by the definitional formula

$$A_n = \frac{1}{n!} \frac{\partial^n}{\partial \lambda^n} \left[ f \left( \sum_{k=0}^{\infty} y_k \lambda^k \right) \right]_{\lambda=0}, \quad n = 0, 1, 2, \dots,$$

substituting (28) and (29) into (27) yields

$$\sum_{n=0}^{\infty} y_n(t) = A + Bt + L^{-1} g(t) - L^{-1} \left( \sum_{n=0}^{\infty} A_n \right)$$

The recursive relationship is found to be

$$\begin{aligned} y_0 &= g(t) \\ y_{n+1} &= -L^{-1} A_n \end{aligned}$$

Using the above recursive relationship, one can construct the solution  $y$  as

$$y = \lim_{n \rightarrow \infty} \varphi_n(y) \quad (30)$$

where

$$\varphi_n(y) = \sum_{i=0}^n y_i$$

## Chapter Three

### Numerical Solutions for (HFDEs) with Triangular Fuzzy Number as Initial Conditions

In this Chapter, an example of HFDEs (29) will be solved with initial conditions as triangular fuzzy number by several numerical methods and 1 – differentiable and 2 – differentiable will be used to find exact solutions and compare them with approximate solutions.

#### 3.1 Numerical Example

##### Example

Let us consider the following hybrid fuzzy IVP [28, 35, 41],

$$y'(t) = y(t) + m(t)\lambda_k(y(t_k)), \quad t \in [t_k, t_{k+1}], \quad t_k = k, \quad k = 0, 1, 2, 3, \dots, \quad (29)$$

where

$$m(t) = \begin{cases} 2(t \bmod 1) & \text{if } t \bmod 1 \leq 0.5; \\ 2(1 - t \bmod 1) & \text{if } t \bmod 1 > 0.5; \end{cases}$$

$$\lambda_k(\mu) = \begin{cases} 0 & \text{if } k = 0 \\ \mu & \text{if } k \in \{1, 2, \dots\} \end{cases}$$

Using the following two triangular fuzzy numbers:

- a- Let  $y(0) = (0.75, 1, 1.125)$   
 $y(0, \alpha) = [(0.75 + 0.25\alpha), (1.125 - 0.125\alpha)], \quad 0 \leq \alpha \leq 1$
- b- Let  $y(0) = (0.25, 0.75, 4)$   
 $y(0, \alpha) = [(0.25 + 0.5\alpha), (4 - 3.25\alpha)], \quad 0 \leq \alpha \leq 1$

The hybrid fuzzy IVP (29) is equivalent to the following systems of fuzzy IVPs:

$$\begin{cases} y'(t) = y_0(t), & t \in [0, 1] \\ y_0(t, \alpha) = y(0, \alpha), & 0 \leq \alpha \leq 1 \\ y'_i(t) = y_i(t) + m(t)y_{i-1}(t), & t \in [t_i, t_{i+1}], \quad y_i(t) = y_{i-1}(t_i), \quad t = 1, 2, \dots, \end{cases}$$

In (29)  $y(t) + m(t)\lambda_k(y(t_k))$  is continuous function of  $t, y$  and  $\lambda_k(y(t_k))$ . Therefore by [32], for each  $k = 0, 1, 2, \dots$ , the fuzzy IVP

$$\begin{cases} y'(t) = y(t) + m(t)\lambda_k(y(t_k)), & t \in [t_k, t_{k+1}], \quad t_k = k, \\ y(t_k) = y_{t_k} \end{cases}$$

has a unique solution on  $[t_k, t_{k+1}]$ .

let  $f: [0, \infty) \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  be given by

$$f(t, x, \lambda_k(x(t_k))) = x(t) + m(t)\lambda_k(x(t_k)), \quad t_k = k, \quad k = 0, 1, 2, \dots,$$

where  $\lambda_k : \mathbb{R} \rightarrow \mathbb{R}$  is given by

$$\lambda_k(x) = \begin{cases} 0, & \text{if } k = 0 \\ x, & \text{if } k \in \{1, 2, \dots\} \end{cases}$$

We can write the HFDE (29) as a formula by

$$y'(t) = \begin{cases} y(t), & t \in [0, 1], \\ y(t) + 2y(1, \alpha)(t - 1), & t \in [1, 1.5], \\ y(t) + 2y(1, \alpha)(2 - t), & t \in [1.5, 2]. \end{cases} \quad (30)$$

By 1 – differentiable the exact solution for  $t \in [0, 2]$  is given by

$$y(t, \alpha) = \begin{cases} y(0, \alpha)e^t, & t \in [0, 1], \\ y(1, \alpha)(3e^{t-1} - 2t), & t \in [1, 1.5], \\ y(1, \alpha)(2t - 2 + e^{t-1.5}(3\sqrt{e} - 4)), & t \in [1.5, 2]. \end{cases} \quad (31)$$

By 2 – differentiable the exact solution for  $t \in [0, 2]$  is given by

$$y(t, \alpha) = [(0.9375 + 0.0625\alpha)e^t + (-0.1875 + 0.1875\alpha)e^{-t}, \\ (0.9375 + 0.0625\alpha)e^t - (-0.1875 + 0.1875\alpha)e^{-t}], \quad t \in [0, 1]$$

$$y(t, \alpha) = \left[ 1.5 \left( \underline{y}(0, \alpha) + \bar{y}(0, \alpha) \right) e^{t-1} + 0.5 \left( \bar{y}(0, \alpha) - \underline{y}(0, \alpha) \right) e^{1-t} - 2\underline{y}(0, \alpha)t \right.$$

$$+ 2 \left( \underline{y}(0, \alpha) - \bar{y}(0, \alpha) \right), 1.5 \left( \underline{y}(0, \alpha) + \bar{y}(0, \alpha) \right) e^{t-1}$$

$$- 0.5 \left( \bar{y}(0, \alpha) - \underline{y}(0, \alpha) \right) e^{1-t} - 2\bar{y}(0, \alpha)t - 2 \left( \underline{y}(0, \alpha) - \bar{y}(0, \alpha) \right) \right],$$

$$t \in [1, 1.5]$$

$$y(t, \alpha) = \left[ 0.5 \left( \underline{y}(1, \alpha) + \bar{y}(1, \alpha) - \underline{y}(0, \alpha) - \bar{y}(0, \alpha) \right) e^{t-1.5} \right. \\ + 0.5 \left( \underline{y}(1, \alpha) - \bar{y}(1, \alpha) + 3\underline{y}(0, \alpha) - 3\bar{y}(0, \alpha) \right) e^{1.5-t} + 2\underline{y}(0, \alpha)t \\ + 2\bar{y}(0, \alpha) - 4\underline{y}(0, \alpha), 0.5 \left( \underline{y}(1, \alpha) + \bar{y}(1, \alpha) - \underline{y}(0, \alpha) - \bar{y}(0, \alpha) \right) e^{t-1.5} \\ - 0.5 \left( \underline{y}(1, \alpha) - \bar{y}(1, \alpha) + 3\underline{y}(0, \alpha) - 3\bar{y}(0, \alpha) \right) e^{1.5-t} + 2\bar{y}(0, \alpha)t \\ \left. + 2\underline{y}(0, \alpha) - 4\bar{y}(0, \alpha) \right], \quad t \in [1.5, 2] \quad (32)$$

### 3.2 Picard's Method

Applying the Picard method for hybrid fuzzy differential equation first

by 1 – differentiable

$$\begin{cases} \underline{y}_{n+1} = \underline{y}_0 + \int_{t_0}^t f(\tau, \underline{y}_n(\tau, \alpha), \lambda_k(y_k)) d\tau \\ \bar{y}_{n+1} = \bar{y}_0 + \int_{t_0}^t f(\tau, \bar{y}_n(\tau, \alpha), \lambda_k(y_k)) d\tau \end{cases} \quad n = 0, 1, \dots, N. \quad (33)$$

Then by 2 – differentiable

$$\begin{cases} \underline{y}_{n+1} = \underline{y}_0 + \int_{t_0}^t f(\tau, \bar{y}_n(\tau, \alpha), \lambda_k(y_k)) d\tau \\ \bar{y}_{n+1} = \bar{y}_0 + \int_{t_0}^t f(\tau, \underline{y}_n(\tau, \alpha), \lambda_k(y_k)) d\tau \end{cases} \quad n = 0, 1, \dots, N \quad (34)$$

From the theory of differential equations, it can be proved that the above sequence of approximations converges to the exact solution of IVP.

Now to solve the previous Example

a- Let  $y(0) = (0.75, 1, 1.125)$

$$y(0, \alpha) = [(0.75 + 0.25\alpha), (1.125 - 0.125\alpha)], \quad 0 \leq \alpha \leq 1$$

Case 1: using 1 – differentiable

Solving through apply the Picard Method for hybrid fuzzy differential equations (29). Let  $N = 50$  when  $\alpha = 0$  and  $t \in [0, 1]$  then  $f(\tau, y_n(\tau, \alpha)) = y_n(\tau, \alpha)$  and  $t_0 = 0$  by (33)

$$\underline{y}_1 = \underline{y}_0 + \int_0^t \underline{y}_0(\tau) d\tau, \quad \bar{y}_1 = \bar{y}_0 + \int_0^t \bar{y}_0(\tau) d\tau$$

$$\underline{y}_1 = 0.75 + \int_0^t 0.75 d\tau, \quad \bar{y}_1 = 1.125 \int_0^t 1.125 d\tau$$

$$\underline{y}_1 = 0.75 + (0.75)t, \quad \bar{y}_1 = 1.125 + (1.125)t$$

$$\underline{y}_2 = \underline{y}_0 + \int_0^t \underline{y}_1(\tau) d\tau, \quad \bar{y}_2 = \bar{y}_0 + \int_0^t \bar{y}_1(\tau) d\tau$$

$$\underline{y}_2 = 0.75 + \int_0^t 0.75 + 0.75\tau d\tau, \quad \bar{y}_2 = 1.125 + \int_0^t 1.125 + 1.125\tau d\tau$$

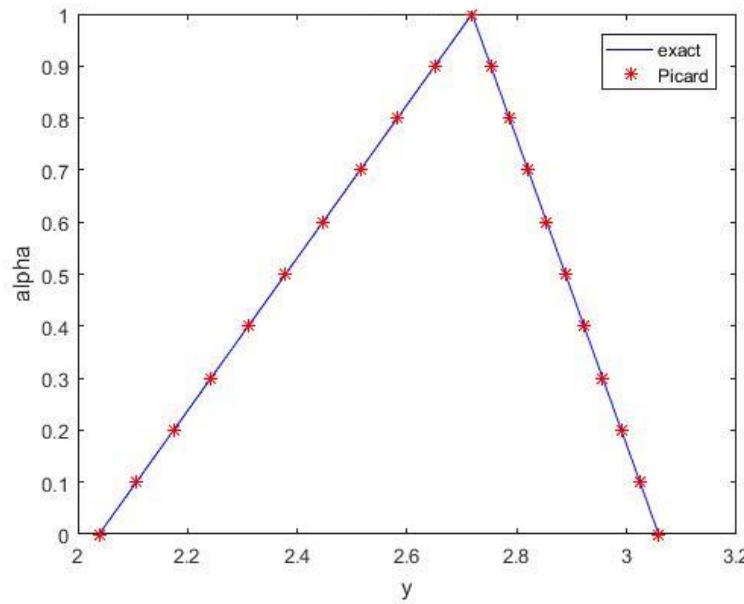
$$\underline{y}_2 = 0.75 + 0.75t + \frac{(0.75)t^2}{2}, \quad \bar{y}_2 = 1.125 + (1.125)t + \frac{1.125t^2}{2}$$

In the same manner to get  $\underline{y}_{50}$  and  $\bar{y}_{50}$  when  $t = 1, t = 1.5, t = 2$ .

By Matlab software, the exact solution with an approximate results of this example are presented in Tables 3.2.1-3 and in Figs. 3.2.1-3 respectively, and the absolute errors of the approximate results in Tables 3.2.4-6.

**Table 3.2.1:** Numerical values for the exact (1 – differentiable) and approximate solutions (Picard's method) for  $t = 1$

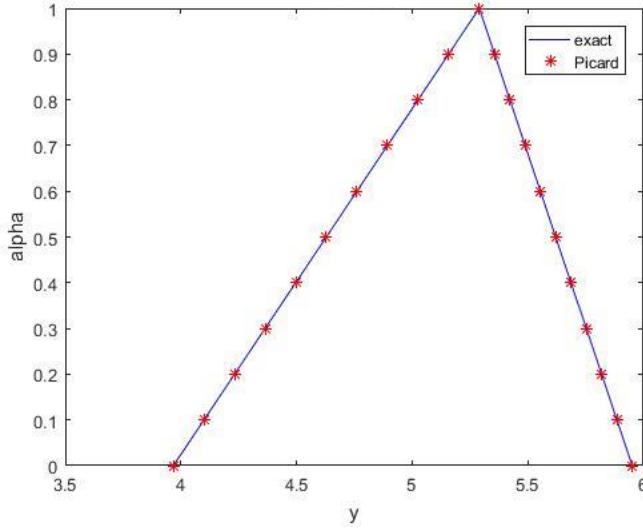
$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.038711371344284	3.058067057016426	2.038711371344284	3.058067057016426
0.1	2.106668417055760	3.024088534160689	2.106668417055760	3.024088534160688
0.2	2.174625462767236	2.990110011304950	2.174625462767236	2.990110011304950
0.3	2.242582508478713	2.956131488449212	2.242582508478713	2.956131488449211
0.4	2.310539554190189	2.922152965593474	2.310539554190189	2.922152965593474
0.5	2.378496599901665	2.888174442737736	2.378496599901665	2.888174442737736
0.6	2.446453645613141	2.854195919881998	2.446453645613141	2.854195919881997
0.7	2.514410691324617	2.820217397026260	2.514410691324617	2.820217397026260
0.8	2.582367737036093	2.786238874170521	2.582367737036093	2.786238874170521
0.9	2.650324782747569	2.752260351314784	2.650324782747569	2.752260351314783
1	2.718281828459046	2.718281828459046	2.718281828459045	2.718281828459045



**Figure 3.2.1 :** Exact and Picard solutions for  $t = 1$

**Table 3.2.2:** Numerical values for the exact (1 – differentiable) and approximate solutions (Picard's method) for  $t = 1.5$

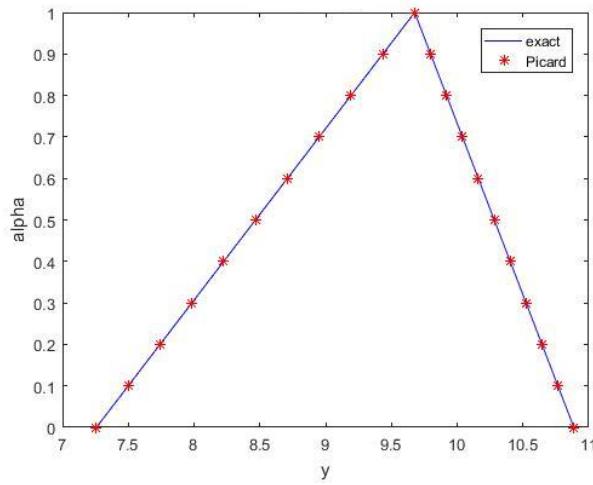
$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	3.967666294227795	5.951499441341692	3.967666294227794	5.951499441341692
0.1	4.099921837368721	5.885371669771230	4.099921837368721	5.885371669771228
0.2	4.232177380509648	5.819243898200766	4.232177380509647	5.819243898200766
0.3	4.364432923650575	5.753116126630302	4.364432923650575	5.753116126630300
0.4	4.496688466791501	5.686988355059839	4.496688466791502	5.686988355059840
0.5	4.628944009932427	5.620860583489376	4.628944009932427	5.620860583489376
0.6	4.761199553073354	5.554732811918913	4.761199553073354	5.554732811918911
0.7	4.893455096214280	5.488605040348450	4.893455096214280	5.488605040348450
0.8	5.025710639355206	5.422477268777985	5.025710639355206	5.422477268777984
0.9	5.157966182496133	5.356349497207523	5.157966182496131	5.356349497207522
1	5.290221725637060	5.290221725637060	5.290221725637058	5.290221725637058



**Figure 3.2.2 :** Exact and Picard solutions for  $t = 1.5$

**Table 3.2.3:** Numerical values for the exact (1 – differentiable) and approximate solutions (Picard's method) for  $t = 2$

$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	7.257731754268338	10.88659763140251	7.257731754268336	10.88659763140251
0.1	7.499656146077282	10.76563543549804	7.499656146077282	10.76563543549803
0.2	7.741580537886227	10.64467323959356	7.741580537886225	10.64467323959356
0.3	7.983504929695171	10.52371104368909	7.983504929695173	10.52371104368909
0.4	8.225429321504116	10.40274884778462	8.225429321504119	10.40274884778462
0.5	8.467353713313061	10.28178665188015	8.467353713313059	10.28178665188015
0.6	8.709278105122007	10.16082445597567	8.709278105122007	10.16082445597567
0.7	8.951202496930950	10.03986226007120	8.951202496930952	10.03986226007120
0.8	9.193126888739894	9.918900064166728	9.193126888739894	9.918900064166724
0.9	9.435051280548839	9.797937868262256	9.435051280548835	9.797937868262254
1	9.676975672357784	9.676975672357784	9.676975672357781	9.676975672357781



**Figure 3.2.3 : Exact and Picard solutions for  $t = 2$**

**Table 3.2.4 :The absolute errors of the Picard method (1 – differentiable) for  $t = 1$**

$\alpha$	Absolute Error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$10^{-16}$	$10^{-16}$
0.1	$10^{-16}$	$8.8818 \times 10^{-16}$
0.2	$10^{-16}$	$10^{-16}$
0.3	$10^{-16}$	$8.8818 \times 10^{-16}$
0.4	$10^{-16}$	$10^{-16}$
0.5	$10^{-16}$	$10^{-16}$
0.6	$10^{-16}$	$8.8818 \times 10^{-16}$
0.7	$10^{-16}$	$10^{-16}$
0.8	$10^{-16}$	$10^{-16}$
0.9	$10^{-16}$	$8.8818 \times 10^{-16}$
1	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$

**Table 3.2.5: The absolute errors of the Picard method (1 – differentiable) for  $t = 1.5$**

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$8.8818 \times 10^{-16}$	$10^{-16}$
0.1	$10^{-16}$	$1.7764 \times 10^{-15}$
0.2	$8.8818 \times 10^{-16}$	$10^{-16}$
0.3	$10^{-16}$	$1.7764 \times 10^{-15}$
0.4	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.5	$10^{-16}$	$10^{-16}$
0.6	$10^{-16}$	$1.7764 \times 10^{-15}$
0.7	$10^{-16}$	$10^{-16}$
0.8	$10^{-16}$	$1.7764 \times 10^{-15}$
0.9	$1.7764 \times 10^{-15}$	$8.8818 \times 10^{-16}$
1	$1.7764 \times 10^{-15}$	$1.7764 \times 10^{-15}$

**Table 3.2.6** :The absolute errors of the Picard method (1 – differentiable) for  $t = 2$

$\alpha$	Absolute errors	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.7764 \times 10^{-15}$	$10^{-16}$
0.1	$10^{-16}$	$1.0658 \times 10^{-14}$
0.2	$1.7764 \times 10^{-15}$	$10^{-16}$
0.3	$2.6645 \times 10^{-15}$	$10^{-16}$
0.4	$3.5527 \times 10^{-15}$	$10^{-16}$
0.5	$1.7764 \times 10^{-15}$	$10^{-16}$
0.6	$10^{-16}$	$10^{-16}$
0.7	$1.7764 \times 10^{-15}$	$10^{-16}$
0.8	$10^{-16}$	$3.5527 \times 10^{-15}$
0.9	$3.5527 \times 10^{-15}$	$1.7764 \times 10^{-15}$
1	$3.5527 \times 10^{-15}$	$3.5527 \times 10^{-15}$

As shown in Tables 3.2.1-6, the Picard method with triangular fuzzy number as initial condition gave high accurate results when used with high number of iteration.

### Case 2: using 2 – differentiable

Let  $N = 50$  when  $\alpha = 0$  and  $t \in [0,1]$  then  $f(\tau, y_n(\tau, \alpha)) = y_n(\tau, \alpha)$  and  $t_0 = 0$  by (34)

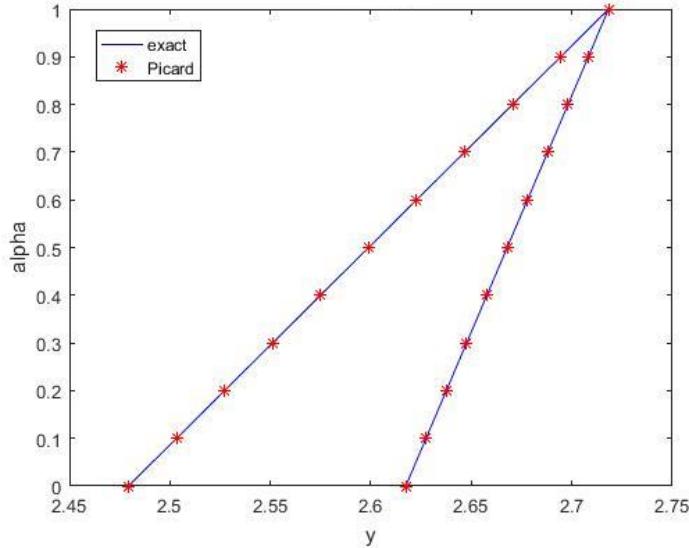
$$\begin{aligned}
\underline{y}_1 &= \underline{y}_0 + \int_0^t \bar{y}_0(\tau) d\tau, & \bar{y}_1 &= \bar{y}_0 + \int_0^t \underline{y}_0(\tau) d\tau \\
\underline{y}_1 &= 0.75 + \int_0^t 1.125 d\tau, & \bar{y}_1 &= 1.125 \int_0^t 0.75 d\tau \\
\underline{y}_1 &= 0.75 + (1.125)t, & \bar{y}_1 &= 1.125 + (0.75)t \\
\underline{y}_2 &= \underline{y}_0 + \int_0^t \bar{y}_1(\tau) d\tau, & \bar{y}_2 &= \bar{y}_0 + \int_0^t \underline{y}_1(\tau) d\tau \\
\underline{y}_2 &= 0.75 + \int_0^t 1.125 + 0.75\tau d\tau, & \bar{y}_2 &= 1.125 + \int_0^t 0.75 + 1.125\tau d\tau \\
\underline{y}_2 &= 0.75 + (1.125)t + \frac{(0.75)t^2}{2}, & \bar{y}_2 &= 1.125 + (0.75)t + \frac{1.125t^2}{2}
\end{aligned}$$

In the same manner to get  $\underline{y}_{50}$  and  $\bar{y}_{50}$  when  $t = 1, t = 1.5, t = 2$ .

By Matlab software, the exact solutions with approximate results of this example are presented in Tables 3.2.7-9 and Figs 3.2.4-6 respectively, and the absolute errors of the approximate results in Tables 3.2.10-12.

**Table 3.2.7:** Numerical values for the exact (2 – differentiable) and approximate solutions (Picard's method) for  $t = 1$

$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	2.479411818960710	2.617366609400000	2.479411818960710	2.617366609400000
0.1	2.503298819910544	2.627458131305905	2.503298819910544	2.627458131305905
0.2	2.527185820860377	2.637549653211810	2.527185820860376	2.637549653211810
0.3	2.551072821810211	2.647641175117714	2.551072821810210	2.647641175117714
0.4	2.574959822760044	2.657732697023619	2.574959822760043	2.657732697023618
0.5	2.598846823709878	2.667824218929523	2.598846823709878	2.667824218929523
0.6	2.622733824659711	2.677915740835428	2.622733824659711	2.677915740835428
0.7	2.646620825609545	2.688007262741332	2.646620825609544	2.688007262741332
0.8	2.670507826559379	2.698098784647236	2.670507826559379	2.698098784647236
0.9	2.694394827509212	2.708190306553141	2.694394827509211	2.708190306553141
1	2.718281828459046	2.718281828459046	2.718281828459045	2.718281828459045



**Figure 3.2.4:** Exact and Picard solutions for  $t = 1$

**Table 3.2.8:** Numerical values for the exact (2 – differentiable) and approximate solutions (Picard's method) for  $t = 1.5$

$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	4.932442377592928	4.986723357976558	4.932442377592928	4.986723357976557
0.1	4.968220312397341	5.017073194742609	4.968220312397342	5.017073194742609
0.2	5.003998247201754	5.047423031508658	5.003998247201754	5.047423031508657
0.3	5.039776182006168	5.077772868274710	5.039776182006166	5.077772868274708
0.4	5.075554116810579	5.108122705040760	5.075554116810579	5.108122705040757

0.5	5.111332051614995	5.138472541806809	5.111332051614995	5.138472541806809
0.6	5.147109986419408	5.168822378572860	5.147109986419407	5.168822378572860
0.7	5.182887921223820	5.199172215338908	5.182887921223819	5.199172215338908
0.8	5.218665856028234	5.229522052104959	5.218665856028233	5.229522052104960
0.9	5.254443790832648	5.259871888871011	5.254443790832645	5.259871888871008
1	5.290221725637062	5.290221725637062	5.290221725637058	5.290221725637058

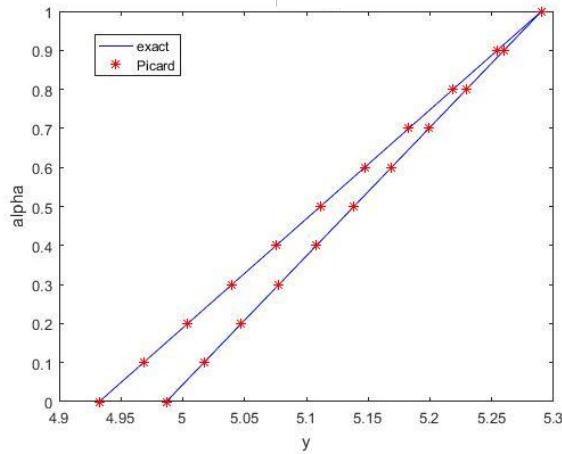


Figure 3.2.5: Exact and Picard solutions for  $t = 1.5$

Table 3.2.9: Numerical values for the exact solutions (2 – differentiable) and approximate solutions (Picard's method) for  $t = 2$

$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	9.068147228770254	9.076182156900588	9.068147228770254	9.076182156900588
0.1	9.129030073129007	9.136261508446307	9.129030073129009	9.136261508446310
0.2	9.189912917487760	9.196340859992027	9.189912917487760	9.196340859992025
0.3	9.250795761846515	9.256420211537749	9.250795761846513	9.256420211537746
0.4	9.311678606205266	9.316499563083468	9.311678606205263	9.316499563083465
0.5	9.372561450564021	9.376578914629187	9.372561450564021	9.376578914629185
0.6	9.433444294922772	9.436658266174906	9.433444294922772	9.436658266174907
0.7	9.494327139281523	9.496737617720623	9.494327139281523	9.496737617720623
0.8	9.555209983640278	9.556816969266345	9.555209983640278	9.556816969266347
0.9	9.616092827999033	9.616896320812066	9.616092827999028	9.616896320812062
1	9.676975672357788	9.676975672357788	9.676975672357781	9.676975672357781

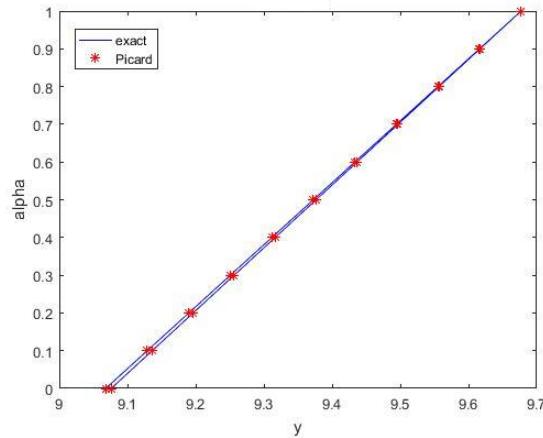


Figure 3.2.6: Exact and Picard solutions for  $t = 2$

**Table 3.2.10** :The absolute errors of the Picard method (2 – differentiable) for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$10^{-16}$	$10^{-16}$
0.1	$10^{-16}$	$10^{-16}$
0.2	$8.8818 \times 10^{-16}$	$10^{-16}$
0.3	$8.8818 \times 10^{-16}$	$10^{-16}$
0.4	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.5	$10^{-16}$	$10^{-16}$
0.6	$10^{-16}$	$10^{-16}$
0.7	$1.3323 \times 10^{-15}$	$10^{-16}$
0.8	$10^{-16}$	$10^{-16}$
0.9	$1.3323 \times 10^{-15}$	$10^{-16}$
1	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$

**Table 3.2.11** :The absolute errors of the Picard method (2 – differentiable) for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$10^{-16}$	$8.8818 \times 10^{-16}$
0.1	$8.8818 \times 10^{-16}$	$10^{-16}$
0.2	$10^{-16}$	$8.8818 \times 10^{-16}$
0.3	$1.7764 \times 10^{-15}$	$2.6645 \times 10^{-15}$
0.4	$10^{-16}$	$2.6645 \times 10^{-15}$
0.5	$10^{-16}$	$10^{-16}$
0.6	$8.8818 \times 10^{-16}$	$10^{-16}$
0.7	$8.8818 \times 10^{-16}$	$10^{-16}$
0.8	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.9	$2.6645 \times 10^{-15}$	$2.6645 \times 10^{-15}$
1	$4.4409 \times 10^{-15}$	$4.4409 \times 10^{-15}$

**Table 3.2.12**:The absolute errors of the Picard method (2 – differentiable) for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$10^{-16}$	$10^{-16}$
0.1	$1.7764 \times 10^{-15}$	$3.5527 \times 10^{-15}$
0.2	$10^{-16}$	$1.7764 \times 10^{-15}$
0.3	$1.7764 \times 10^{-15}$	$10^{-15}$
0.4	$3.5527 \times 10^{-15}$	$10^{-15}$
0.5	$10^{-16}$	$3.5527 \times 10^{-15}$
0.6	$10^{-16}$	$3.5527 \times 10^{-15}$
0.7	$10^{-16}$	$10^{-16}$
0.8	$10^{-16}$	$1.7764 \times 10^{-15}$
0.9	$5.3291 \times 10^{-15}$	$3.5527 \times 10^{-15}$
1	$7.1054 \times 10^{-15}$	$7.1054 \times 10^{-15}$

As shown in Tables 3.2.7-12 , the Picard method with triangular fuzzy number as initial condition gave high accurate results when used with high number of iteration.

b- Let  $y(0) = (0.25, 0.75, 4)$

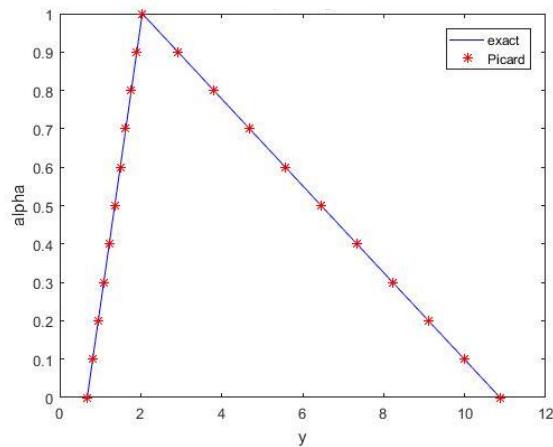
$$y(0, \alpha) = [(0.25 + 0.5\alpha), (4 - 3.25\alpha)], \quad 0 \leq \alpha \leq 1$$

using 1 – differentiable

We solve by the Matlab software the exact solutions with approximate results of this example are presented in Tables 3.2.13-15 and Figs 3.2.7-9 respectively, and the absolute errors of the approximate results in Tables 3.2.16-18

**Table 3.2.13:** Numerical values for the exact and approximate solutions (Picard's method) for  $t = 1$

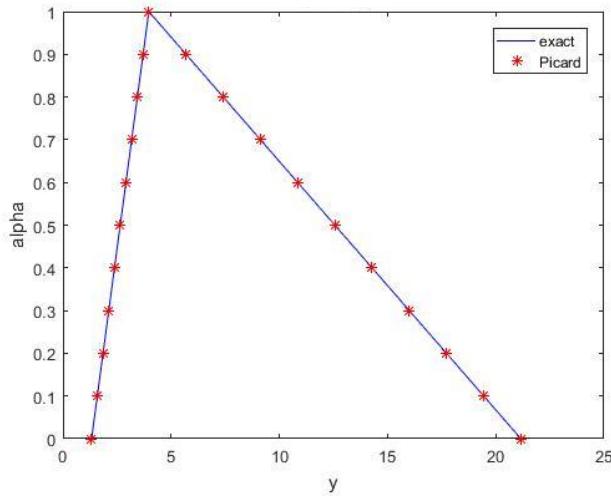
$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	0.6795704571147614	10.87312731383618	0.6795704571147613	10.87312731383618
0.1	0.8154845485377137	9.989685719586992	0.8154845485377136	9.989685719586991
0.2	0.9513986399606659	9.106244125337803	0.9513986399606658	9.106244125337803
0.3	1.087312731383618	8.222802531088613	1.087312731383618	8.222802531088611
0.4	1.223226822806571	7.339360936839423	1.223226822806570	7.339360936839422
0.5	1.359140914229523	6.455919342590233	1.359140914229523	6.455919342590232
0.6	1.495055005652475	5.572477748341043	1.495055005652475	5.572477748341043
0.7	1.630969097075428	4.689036154091854	1.630969097075427	4.689036154091853
0.8	1.766883188498380	3.805594559842664	1.766883188498379	3.805594559842663
0.9	1.902797279921332	2.922152965593473	1.902797279921332	2.922152965593474
1	2.038711371344284	2.038711371344284	2.038711371344284	2.038711371344284



**Figure 3.2.7:** Exact and Picard solutions for  $t = 1$

**Table 3.2.14:** Numerical values for the exact and approximate solutions (Picard's method) for  $t = 1.5$

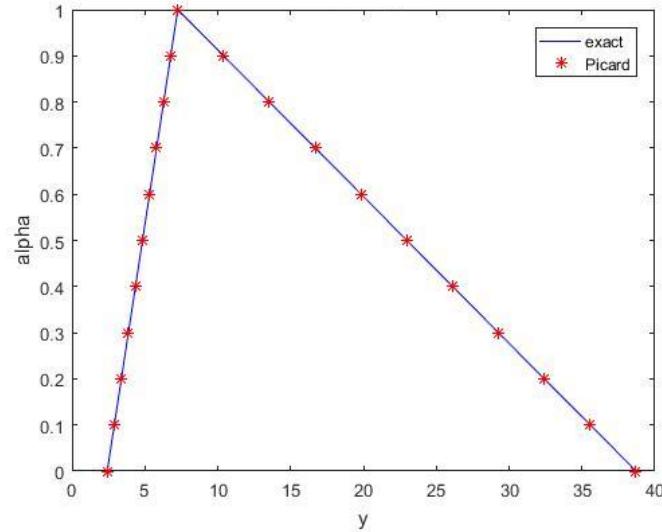
$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	1.322555431409265	21.16088690254824	1.322555431409265	21.16088690254823
0.1	1.587066517691118	19.44156484171619	1.587066517691118	19.44156484171619
0.2	1.851577603972971	17.72224278088415	1.851577603972971	17.72224278088415
0.3	2.116088690254824	16.00292072005211	2.116088690254824	16.00292072005210
0.4	2.380599776536677	14.28359865922006	2.380599776536676	14.28359865922006
0.5	2.645110862818530	12.56427659838802	2.645110862818530	12.56427659838801
0.6	2.909621949100383	10.84495453755597	2.909621949100383	10.84495453755597
0.7	3.174133035382237	9.125632476723929	3.174133035382234	9.125632476723926
0.8	3.438644121664089	7.406310415891883	3.438644121664088	7.406310415891882
0.9	3.703155207945942	5.686988355059838	3.703155207945942	5.686988355059840
1	3.967666294227795	3.967666294227795	3.967666294227794	3.967666294227794



**Figure 3.2.8:** Exact and Picard solutions for  $t = 1.5$

**Table 3.2.15:** Numerical values for the exact and approximate solutions (Picard's method) for  $t = 2$

$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.419243918089446	38.70790268943114	2.419243918089446	38.70790268943112
0.1	2.903092701707335	35.56288559591486	2.903092701707335	35.56288559591485
0.2	3.386941485325224	32.41786850239858	3.386941485325225	32.41786850239858
0.3	3.870790268943114	29.27285140888230	3.870790268943114	29.27285140888229
0.4	4.354639052561003	26.12783431536602	4.354639052561001	26.12783431536602
0.5	4.838487836178892	22.98281722184974	4.838487836178891	22.98281722184973
0.6	5.322336619796782	19.83780012833346	5.322336619796781	19.83780012833346
0.7	5.806185403414671	16.69278303481718	5.806185403414667	16.69278303481718
0.8	6.290034187032560	13.54776594130090	6.290034187032558	13.54776594130090
0.9	6.773882970650448	10.40274884778462	6.773882970650449	10.40274884778462
1	7.257731754268338	7.257731754268338	7.257731754268336	7.257731754268336



**Figure 3.2.9:** Exact and Picard solutions for  $t = 2$

**Table 3.2.16 :**The absolute errors of the Picard method for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.1102 \times 10^{-15}$	$10^{-16}$
0.1	$1.1102 \times 10^{-15}$	$1.7764 \times 10^{-15}$
0.2	$1.1102 \times 10^{-15}$	$10^{-16}$
0.3	$10^{-16}$	$1.7764 \times 10^{-15}$
0.4	$1.1102 \times 10^{-15}$	$10^{-16}$
0.5	$10^{-16}$	$8.8818 \times 10^{-16}$
0.6	$10^{-16}$	$10^{-16}$
0.7	$1.1102 \times 10^{-15}$	$8.8818 \times 10^{-16}$
0.8	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.9	$10^{-16}$	$1.3323 \times 10^{-15}$
1	$10^{-16}$	$10^{-16}$

**Table 3.2.17 :**The absolute errors of the Picard method for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$10^{-16}$	$7.1054 \times 10^{-15}$
0.1	$10^{-16}$	$10^{-16}$
0.2	$10^{-16}$	$10^{-16}$
0.3	$10^{-16}$	$1.0658 \times 10^{-15}$
0.4	$8.8818 \times 10^{-16}$	$10^{-16}$
0.5	$10^{-16}$	$8.8818 \times 10^{-16}$
0.6	$10^{-16}$	$10^{-16}$
0.7	$3.1086 \times 10^{-15}$	$3.1086 \times 10^{-15}$
0.8	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.9	$10^{-16}$	$1.7764 \times 10^{-15}$
1	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$

**Table 3.2.18 :The absolute errors of the Picard method for  $t = 2$**

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$10^{-16}$	$1.4211 \times 10^{-14}$
0.1	$10^{-16}$	$7.1054 \times 10^{-15}$
0.2	$8.8818 \times 10^{-16}$	$10^{-16}$
0.3	$10^{-16}$	$1.0658 \times 10^{-14}$
0.4	$2.6645 \times 10^{-15}$	$10^{-16}$
0.5	$8.8818 \times 10^{-16}$	$1.0658 \times 10^{-14}$
0.6	$8.8818 \times 10^{-16}$	$10^{-16}$
0.7	$3.5527 \times 10^{-15}$	$10^{-16}$
0.8	$1.7764 \times 10^{-15}$	$10^{-16}$
0.9	$8.8818 \times 10^{-16}$	$10^{-16}$
1	$1.7764 \times 10^{-15}$	$1.7764 \times 10^{-15}$

As shown in Tables 3.2.13-18 , the Picard method with another triangular fuzzy number as initial condition gave high accurate results when used with high number of iteration.

### 3.3 Runge-Kutta of Order Five

In this section, for a hybrid fuzzy differential equation (3) we developed the Runge-Kutta method via an application of the Runge-Kutta method for fuzzy differential equation.

For a fixed  $\alpha$ , to integrate the system in (5) for  $[t_0, t_1], [t_1, t_2], \dots, [t_k, t_{k+1}], \dots$ , we replace each interval by a set of  $N_k + 1$  discrete equally spaced grid points (including the end points) at which exact solution  $(\underline{Y}_k(t, r), \bar{Y}_k(t, r))$ . For each the chosen grid points on  $[t_k, t_{k+1}]$  at  $t_{k,n} = t_k + nh_k = \frac{t_{k+1} - t_k}{N_k}, 0 \leq n \leq N_k$ .

Let  $(\underline{Y}_k(t, r), \bar{Y}_k(t, r))$  and  $(\underline{y}_k(t, r), \bar{y}_k(t, r))$  are exact solution by (1) –differentiable and approximate solution respectively, and may be denoted by  $(\underline{Y}_{k,n}(t, r), \bar{Y}_{k,n}(t, r))$  and  $(\underline{y}_{k,n}(t, r), \bar{y}_{k,n}(t, r))$  respectively. We allow the  $N_k$ 's to vary over the  $[t_k, t_{k+1}]$ 's so that the  $h_k$ 's may be comparable.

The Runge-Kutta method is a fifth order approximation of  $\underline{Y}'_k(t, r)$  and  $\bar{Y}'_k(t, r)$ . To develop the Runge-Kutta method under 1 – differentiable for (1), we define:

$$\underline{y}_{k,n+1}(r) - \underline{y}_{k,n}(r) = \sum_{i=1}^5 \omega_i \underline{k}_i(t_{k,n}, y_{k,n}(r)),$$

$$\bar{y}_{k,n+1}(r) - \bar{y}_{k,n}(r) = \sum_{i=1}^5 \omega_i \bar{k}_i(t_{k,n}, y_{k,n}(r)),$$

Where  $\omega_1, \omega_2, \omega_3, \omega_4$ , and  $\omega_5$  are constants and

$$\underline{k}_1(t_{k,n}, y_{k,n}(r)) = \min\left\{h_k f(t_{k,n}, u, \lambda_k(u_k)) \mid u \in [\underline{y}_{k,n}(r), \bar{y}_{k,n}(r)], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\right\},$$

$$\bar{k}_1(t_{k,n}, y_{k,n}(r)) = \max\left\{h_k f(t_{k,n}, u, \lambda_k(u_k)) \mid u \in [\underline{y}_{k,n}(r), \bar{y}_{k,n}(r)], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\right\},$$

$$\underline{k}_2(t_{k,n}, y_{k,n}(r)) = \min\left\{h_k f\left(t_{k,n} + \frac{h_k}{3}, u, \lambda_k(u_k)\right) \mid u \in [\underline{z}_{k_1}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_1}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\right\},$$

$$\bar{k}_2(t_{k,n}, y_{k,n}(r)) = \max\left\{h_k f\left(t_{k,n} + \frac{h_k}{3}, u, \lambda_k(u_k)\right) \mid u \in [\underline{z}_{k_1}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_1}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\right\},$$

$$\underline{k}_3(t_{k,n}, y_{k,n}(r)) = \min\left\{h_k f\left(t_{k,n} + \frac{h_k}{3}, u, \lambda_k(u_k)\right) \mid u \in [\underline{z}_{k_2}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_2}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\right\},$$

$$\bar{k}_3(t_{k,n}, y_{k,n}(r)) = \max\left\{h_k f\left(t_{k,n} + \frac{h_k}{3}, u, \lambda_k(u_k)\right) \mid u \in [\underline{z}_{k_2}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_2}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\right\},$$

$$\underline{k}_4(t_{k,n}, y_{k,n}(r)) = \min\left\{h_k f\left(t_{k,n} + \frac{h_k}{2}, u, \lambda_k(u_k)\right) \mid u \in [\underline{z}_{k_3}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_3}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\right\},$$

$$\bar{k}_4(t_{k,n}, y_{k,n}(r)) = \max\left\{h_k f\left(t_{k,n} + \frac{h_k}{2}, u, \lambda_k(u_k)\right) \mid u \in [\underline{z}_{k_3}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_3}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\right\},$$

$$\underline{k}_5(t_{k,n}, y_{k,n}(r)) = \min\left\{h_k f\left(t_{k,n} + h_k, u, \lambda_k(u_k)\right) \mid$$

$$u \in [\underline{z}_{k_3}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_3}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\},$$

$$\bar{k}_5(t_{k,n}, y_{k,n}(r)) = \max\{h_k f(t_{k,n} + h_k, u, \lambda_k(u_k))\}$$

$$u \in [\underline{z}_{k_3}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_3}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\},$$

$$\underline{z}_{k_1}(t_{k,n}, y_{k,n}(r)) = \underline{y}_{k,n}(r) + \frac{1}{3}\underline{k}_1(t_{k,n}, y_{k,n}(r)),$$

$$\bar{z}_{k_1}(t_{k,n}, y_{k,n}(r)) = \bar{y}_{k,n}(r) + \frac{1}{3}\bar{k}_1(t_{k,n}, y_{k,n}(r)),$$

$$\underline{z}_{k_2}(t_{k,n}, y_{k,n}(r)) = \underline{y}_{k,n}(r) + \frac{1}{6}[\underline{k}_1(t_{k,n}, y_{k,n}(r)) + \underline{k}_2(t_{k,n}, y_{k,n}(r))],$$

$$\bar{z}_{k_2}(t_{k,n}, y_{k,n}(r)) = \bar{y}_{k,n}(r) + \frac{1}{6}[\bar{k}_1(t_{k,n}, y_{k,n}(r)) + \bar{k}_2(t_{k,n}, y_{k,n}(r))],$$

$$\underline{z}_{k_3}(t_{k,n}, y_{k,n}(r)) = \underline{y}_{k,n}(r) + \frac{1}{8}[\underline{k}_1(t_{k,n}, y_{k,n}(r)) + 3\underline{k}_3(t_{k,n}, y_{k,n}(r))],$$

$$\bar{z}_{k_3}(t_{k,n}, y_{k,n}(r)) = \bar{y}_{k,n}(r) + \frac{1}{8}[\bar{k}_1(t_{k,n}, y_{k,n}(r)) + 3\bar{k}_3(t_{k,n}, y_{k,n}(r))],$$

$$\begin{aligned} \underline{z}_{k_4}(t_{k,n}, y_{k,n}(r)) \\ = \underline{y}_{k,n}(r) + \frac{1}{2}\underline{k}_1(t_{k,n}, y_{k,n}(r)) - \frac{3}{2}\underline{k}_3(t_{k,n}, y_{k,n}(r)) + 2\underline{k}_4(t_{k,n}, y_{k,n}(r)), \end{aligned}$$

$$\begin{aligned} \bar{z}_{k_4}(t_{k,n}, y_{k,n}(r)) \\ = \bar{y}_{k,n}(r) + \frac{1}{2}\bar{k}_1(t_{k,n}, y_{k,n}(r)) - \frac{3}{2}\bar{k}_3(t_{k,n}, y_{k,n}(r)) + 2\bar{k}_4(t_{k,n}, y_{k,n}(r)), \end{aligned}$$

Next define

$$S_k[t_{k,n}, \underline{y}_{k,n}(r), \bar{y}_{k,n}(r)] = \underline{k}_1(t_{k,n}, y_{k,n}(r)) + 4\underline{k}_4(t_{k,n}, y_{k,n}(r)) + \underline{k}_5(t_{k,n}, y_{k,n}(r))$$

$$T_k[t_{k,n}, \underline{y}_{k,n}(r), \bar{y}_{k,n}(r)] = \bar{k}_1(t_{k,n}, y_{k,n}(r)) + 4\bar{k}_4(t_{k,n}, y_{k,n}(r)) + \bar{k}_5(t_{k,n}, y_{k,n}(r))$$

The exact solution at  $t_{k,n+1}$  is given by

$$\begin{cases} \underline{Y}_{k,n+1}(r) \approx \underline{Y}_{k,n}(r) + \frac{1}{6}S_k[t_{k,n}, \underline{Y}_{k,n}(r), \bar{Y}_{k,n}(r)], \\ \bar{Y}_{k,n+1}(r) \approx \bar{Y}_{k,n}(r) + \frac{1}{6}T_k[t_{k,n}, \underline{Y}_{k,n}(r), \bar{Y}_{k,n}(r)], \end{cases}$$

The approximate solution is given by

$$\begin{cases} \underline{y}_{k,n+1}(r) \approx \underline{y}_{k,n}(r) + \frac{1}{6} S_k [t_{k,n}, \underline{y}_{k,n}(r), \bar{y}_{k,n}(r)], \\ \bar{y}_{k,n+1}(r) \approx \bar{y}_{k,n}(r) + \frac{1}{6} T_k [t_{k,n}, \underline{y}_{k,n}(r), \bar{y}_{k,n}(r)], \end{cases} \quad (35)$$

### The Runge-Kutta method under 2 – differentiable

$$\underline{y}_{k,n+1}(r) - \underline{y}_{k,n}(r) = \sum_{i=1}^5 \omega_i k_i(t_{k,n}; y_{k,n}(r)),$$

$$\bar{y}_{k,n+1}(r) - \bar{y}_{k,n}(r) = \sum_{i=1}^5 \omega_i \bar{k}_i(t_{k,n}; y_{k,n}(r)),$$

Where  $\omega_1, \omega_2, \omega_3, \omega_4$ , and  $\omega_5$  are constants and

$$k_1(t_{k,n}, y_{k,n}(r)) = \max\{h_k f(t_{k,n}, u, \lambda_k(u_k)) \mid u \in [\underline{y}_{k,n}(r), \bar{y}_{k,n}(r)], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\},$$

$$\bar{k}_1(t_{k,n}, y_{k,n}(r)) = \min\{h_k f(t_{k,n}, u, \lambda_k(u_k)) \mid u \in [\underline{y}_{k,n}(r), \bar{y}_{k,n}(r)], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\},$$

$$\begin{aligned} \underline{k}_2(t_{k,n}, y_{k,n}(r)) &= \max\left\{h_k f\left(t_{k,n} + \frac{h_k}{3}, u, \lambda_k(u_k)\right) \mid \right. \\ &\quad \left. u \in [\underline{z}_{k_1}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_1}(t_{k,n}, y_{k,n}(r))] \right\}, u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)], \end{aligned}$$

$$\begin{aligned} \bar{k}_2(t_{k,n}, y_{k,n}(r)) &= \min\left\{h_k f\left(t_{k,n} + \frac{h_k}{3}, u, \lambda_k(u_k)\right) \mid \right. \\ &\quad \left. u \in [\underline{z}_{k_1}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_1}(t_{k,n}, y_{k,n}(r))] \right\}, u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)], \end{aligned}$$

$$\begin{aligned} \underline{k}_3(t_{k,n}, y_{k,n}(r)) &= \max\left\{h_k f\left(t_{k,n} + \frac{h_k}{3}, u, \lambda_k(u_k)\right) \mid \right. \\ &\quad \left. u \in [\underline{z}_{k_2}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_2}(t_{k,n}, y_{k,n}(r))] \right\}, u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)], \end{aligned}$$

$$\begin{aligned} \bar{k}_3(t_{k,n}, y_{k,n}(r)) &= \min\left\{h_k f\left(t_{k,n} + \frac{h_k}{3}, u, \lambda_k(u_k)\right) \mid \right. \\ &\quad \left. u \in [\underline{z}_{k_2}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_2}(t_{k,n}, y_{k,n}(r))] \right\}, u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)], \end{aligned}$$

$$\underline{k}_4(t_{k,n}, y_{k,n}(r)) = \max\left\{h_k f\left(t_{k,n} + \frac{h_k}{2}, u, \lambda_k(u_k)\right) \mid \right.$$

$$u \in [\underline{z}_{k_3}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_3}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\},$$

$$\bar{k}_4(t_{k,n}, y_{k,n}(r)) = \min\left\{h_k f\left(t_{k,n} + \frac{h_k}{2}, u, \lambda_k(u_k)\right)\right\}$$

$$u \in [\underline{z}_{k_3}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_3}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\},$$

$$\underline{k}_5(t_{k,n}, y_{k,n}(r)) = \max\left\{h_k f\left(t_{k,n} + h_k, u, \lambda_k(u_k)\right)\right\}$$

$$u \in [\underline{z}_{k_4}(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_4}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\},$$

$$\bar{k}_5(t_{k,n}, y_{k,n}(r)) = \min\left\{h_k f\left(t_{k,n} + h_k, u, \lambda_k(u_k)\right)\right\}$$

$$u[\underline{z}_4(t_{k,n}, y_{k,n}(r)), \bar{z}_{k_4}(t_{k,n}, y_{k,n}(r))], u_k \in [\underline{y}_{k,0}(r), \bar{y}_{k,0}(r)]\},$$

$$\underline{z}_{k_1}(t_{k,n}, y_{k,n}(r)) = \underline{y}_{k,n}(r) + \frac{1}{3}\underline{k}_1(t_{k,n}, y_{k,n}(r)),$$

$$\bar{z}_{k_1}(t_{k,n}, y_{k,n}(r)) = \bar{y}_{k,n}(r) + \frac{1}{3}\bar{k}_1(t_{k,n}, y_{k,n}(r)),$$

$$\underline{z}_{k_2}(t_{k,n}, y_{k,n}(r)) = \underline{y}_{k,n}(r) + \frac{1}{6}[\underline{k}_1(t_{k,n}, y_{k,n}(r)) + \underline{k}_2(t_{k,n}, y_{k,n}(r))],$$

$$\bar{z}_{k_2}(t_{k,n}, y_{k,n}(r)) = \bar{y}_{k,n}(r) + \frac{1}{6}[\bar{k}_1(t_{k,n}, y_{k,n}(r)) + \bar{k}_2(t_{k,n}, y_{k,n}(r))],$$

$$\underline{z}_{k_3}(t_{k,n}, y_{k,n}(r)) = \underline{y}_{k,n}(r) + \frac{1}{8}[\underline{k}_1(t_{k,n}, y_{k,n}(r)) + 3\underline{k}_3(t_{k,n}, y_{k,n}(r))],$$

$$\bar{z}_{k_3}(t_{k,n}, y_{k,n}(r)) = \bar{y}_{k,n}(r) + \frac{1}{8}[\bar{k}_1(t_{k,n}, y_{k,n}(r)) + 3\bar{k}_3(t_{k,n}, y_{k,n}(r))],$$

$$\begin{aligned} \underline{z}_{k_4}(t_{k,n}, y_{k,n}(r)) \\ = \underline{y}_{k,n}(r) + \frac{1}{2}\underline{k}_1(t_{k,n}, y_{k,n}(r)) - \frac{3}{2}\underline{k}_3(t_{k,n}, y_{k,n}(r)) + 2\underline{k}_4(t_{k,n}, y_{k,n}(r)), \end{aligned}$$

$$\begin{aligned} \bar{z}_{k_4}(t_{k,n}, y_{k,n}(r)) \\ = \bar{y}_{k,n}(r) + \frac{1}{2}\bar{k}_1(t_{k,n}, y_{k,n}(r)) - \frac{3}{2}\bar{k}_3(t_{k,n}, y_{k,n}(r)) + 2\bar{k}_4(t_{k,n}, y_{k,n}(r)), \end{aligned}$$

Next define

$$S_k \left[ t_{k,n}, \underline{y}_{k,n}(r), \bar{y}_{k,n}(r) \right] = \underline{k}_1(t_{k,n}, y_{k,n}(r)) + 4\underline{k}_4(t_{k,n}, y_{k,n}(r)) + \underline{k}_5(t_{k,n}, y_{k,n}(r))$$

$$T_k \left[ t_{k,n}, \underline{y}_{k,n}(r), \bar{y}_{k,n}(r) \right] = \bar{k}_1(t_{k,n}, y_{k,n}(r)) + 4\bar{k}_4(t_{k,n}, y_{k,n}(r)) + \bar{k}_5(t_{k,n}, y_{k,n}(r))$$

The exact solution at  $t_{k,n+1}$  is given by

$$\begin{cases} \underline{Y}_{k,n+1}(r) \approx \underline{Y}_{k,n}(r) + \frac{1}{6}S_k[t_{k,n}, \underline{Y}_{k,n}(r), \bar{Y}_{k,n}(r)], \\ \bar{Y}_{k,n+1}(r) \approx \bar{Y}_{k,n}(r) + \frac{1}{6}T_k[t_{k,n}, \underline{Y}_{k,n}(r), \bar{Y}_{k,n}(r)], \end{cases}$$

Finally the approximate solution is given by

$$\begin{cases} \underline{y}_{k,n+1}(r) \approx \underline{y}_{k,n}(r) + \frac{1}{6}S_k[t_{k,n}, \underline{y}_{k,n}(r), \bar{y}_{k,n}(r)], \\ \bar{y}_{k,n+1}(r) \approx \bar{y}_{k,n}(r) + \frac{1}{6}T_k[t_{k,n}, \underline{y}_{k,n}(r), \bar{y}_{k,n}(r)], \end{cases} \quad (36)$$

To solve numerically the hybrid fuzzy IVP (29) we will apply the Runge-Kutta method of order five for hybrid fuzzy differential equation with  $N = 100$  and  $h = \frac{2-0}{100} = 0.02$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 3.3.1-3 and Figs 3.3.1-3 respectively, and the absolute errors of the approximate results in Tables 3.3.4-6

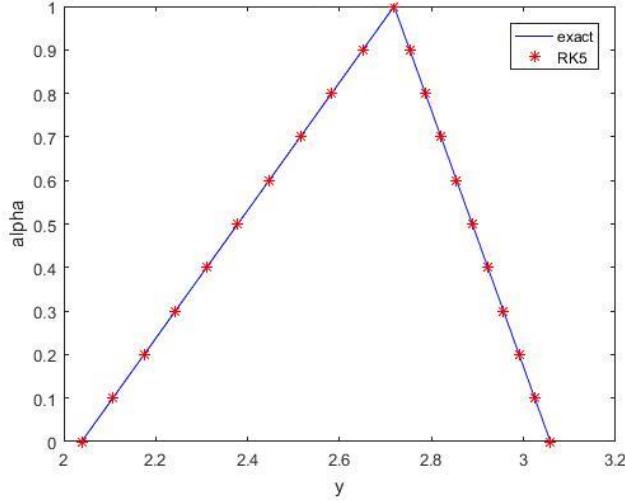
a- Let  $y(0) = (0.75, 1, 1.125)$

$$y(0, \alpha) = [(0.75 + 0.25\alpha), (1.125 - 0.125\alpha)], \quad 0 \leq \alpha \leq 1$$

Case 1: using 1 – differentiable

**Table 3.3.1:** Numerical values for the exact (1 – differentiable) and approximate solutions (Runge-kutta) for  $t = 1$

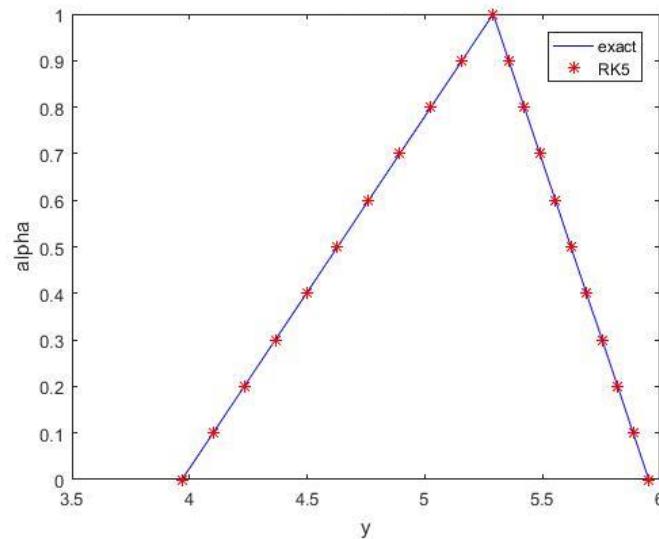
$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.038711371344284	3.058067057016426	2.038711370891301	3.058067056336951
0.1	2.106668417055760	3.024088534160689	2.106668416587678	3.024088533488763
0.2	2.174625462767236	2.990110011304950	2.174625462284054	2.990110010640575
0.3	2.242582508478713	2.956131488449212	2.242582507980431	2.956131487792387
0.4	2.310539554190189	2.922152965593474	2.310539553676808	2.922152964944198
0.5	2.378496599901665	2.888174442737736	2.378496599373185	2.888174442096010
0.6	2.446453645613141	2.854195919881998	2.446453645069561	2.854195919247821
0.7	2.514410691324617	2.820217397026260	2.514410690765938	2.820217396399633
0.8	2.582367737036093	2.786238874170521	2.582367736462314	2.786238873551445
0.9	2.650324782747569	2.752260351314784	2.650324782158691	2.752260350703256
1	2.718281828459046	2.718281828459046	2.718281827855068	2.718281827855068



**Figure 3.3.1:** Exact and RK5 solutions for  $t = 1$

**Table 3.3.2:** Numerical values for the exact ( $1 - \text{differentiable}$ ) and approximate solutions (Runge-kutta) for  $t = 1.5$

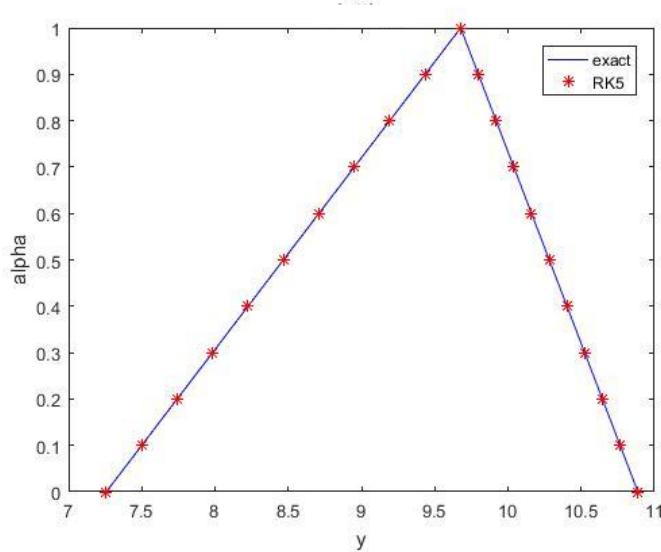
$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	3.967666294227795	5.951499441341692	3.967666292225951	5.951499438338925
0.1	4.099921837368721	5.885371669771230	4.099921835300150	5.885371666801826
0.2	4.232177380509648	5.819243898200766	4.232177378374347	5.819243895264729
0.3	4.364432923650575	5.753116126630302	4.364432921448546	5.753116123727630
0.4	4.496688466791501	5.686988355059839	4.496688464522745	5.686988352190530
0.5	4.628944009932427	5.620860583489376	4.628944007596944	5.620860580653431
0.6	4.761199553073354	5.554732811918913	4.761199550671140	5.554732809116330
0.7	4.893455096214280	5.488605040348450	4.893455093745340	5.488605037579232
0.8	5.025710639355206	5.422477268777985	5.025710636819537	5.422477266042134
0.9	5.157966182496133	5.356349497207523	5.157966179893736	5.356349494505033
1	5.290221725637060	5.290221725637060	5.290221722967934	5.290221722967934



**Figure 3.3.2:** Exact and RK5 solutions for  $t = 1.5$

**Table 3.3.3:** Numerical values for the exact (1 – differentiable) and approximate solutions (Runge-kutta) for  $t = 2$

$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	7.257731754268338	10.88659763140251	7.257731750455414	10.88659762568312
0.1	7.499656146077282	10.76563543549804	7.499656142137263	10.76563542984220
0.2	7.741580537886227	10.64467323959356	7.741580533819107	10.64467323400128
0.3	7.983504929695171	10.52371104368909	7.983504925500955	10.52371103816035
0.4	8.225429321504116	10.40274884778462	8.225429317182805	10.40274884231943
0.5	8.467353713313061	10.28178665188015	8.467353708864652	10.28178664647850
0.6	8.709278105122007	10.16082445597567	8.709278100546495	10.16082445063758
0.7	8.951202496930950	10.03986226007120	8.951202492228344	10.03986225479666
0.8	9.193126888739894	9.918900064166728	9.193126883910189	9.918900058955734
0.9	9.435051280548839	9.797937868262256	9.435051275592038	9.797937863114807
1	9.676975672357784	9.676975672357784	9.676975667273885	9.676975667273885



**Figure 3.3.3:** Exact and RK5 solutions for  $t = 2$

**Table 3.3.4:** The absolute errors of the Runge-kutta method (1 – differentiable) for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$4.5298 \times 10^{-10}$	$6.7948 \times 10^{-10}$
0.1	$4.6808 \times 10^{-10}$	$6.7193 \times 10^{-10}$
0.2	$4.8318 \times 10^{-10}$	$6.6437 \times 10^{-10}$
0.3	$4.9828 \times 10^{-10}$	$6.5682 \times 10^{-10}$
0.4	$5.1338 \times 10^{-10}$	$6.4928 \times 10^{-10}$
0.5	$5.2848 \times 10^{-10}$	$6.4173 \times 10^{-10}$
0.6	$5.4358 \times 10^{-10}$	$6.3418 \times 10^{-10}$
0.7	$5.5868 \times 10^{-10}$	$6.2663 \times 10^{-10}$
0.8	$5.7378 \times 10^{-10}$	$6.1908 \times 10^{-10}$
0.9	$5.8888 \times 10^{-10}$	$6.1153 \times 10^{-10}$
1	$6.0398 \times 10^{-10}$	$6.0398 \times 10^{-10}$

**Table 3.3.5:** The absolute errors of the Runge-kutta method (1 – differentiable) for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$2.0018 \times 10^{-9}$	$3.0028 \times 10^{-9}$
0.1	$2.0686 \times 10^{-9}$	$2.9694 \times 10^{-9}$
0.2	$2.1353 \times 10^{-9}$	$2.9360 \times 10^{-9}$
0.3	$2.2020 \times 10^{-9}$	$2.9027 \times 10^{-9}$
0.4	$2.2688 \times 10^{-9}$	$2.8693 \times 10^{-9}$
0.5	$2.3355 \times 10^{-9}$	$2.8359 \times 10^{-9}$
0.6	$2.4022 \times 10^{-9}$	$2.8026 \times 10^{-9}$
0.7	$2.4689 \times 10^{-9}$	$2.7692 \times 10^{-9}$
0.8	$2.5357 \times 10^{-9}$	$2.7359 \times 10^{-9}$
0.9	$2.6024 \times 10^{-9}$	$2.7025 \times 10^{-9}$
1	$2.6691 \times 10^{-9}$	$2.6691 \times 10^{-9}$

**Table 3.3.6 :** The absolute errors of the Runge-kutta method (1 – differentiable) for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$3.8129 \times 10^{-9}$	$5.7194 \times 10^{-9}$
0.1	$3.9400 \times 10^{-9}$	$5.6558 \times 10^{-9}$
0.2	$4.0671 \times 10^{-9}$	$5.5923 \times 10^{-9}$
0.3	$4.1942 \times 10^{-9}$	$5.5287 \times 10^{-9}$
0.4	$4.3213 \times 10^{-9}$	$5.4652 \times 10^{-9}$
0.5	$4.4484 \times 10^{-9}$	$5.4017 \times 10^{-9}$
0.6	$4.5755 \times 10^{-9}$	$5.3381 \times 10^{-9}$
0.7	$4.7026 \times 10^{-9}$	$5.2745 \times 10^{-9}$
0.8	$4.8297 \times 10^{-9}$	$5.2110 \times 10^{-9}$
0.9	$4.9568 \times 10^{-9}$	$5.1474 \times 10^{-9}$
1	$5.0839 \times 10^{-9}$	$5.0839 \times 10^{-9}$

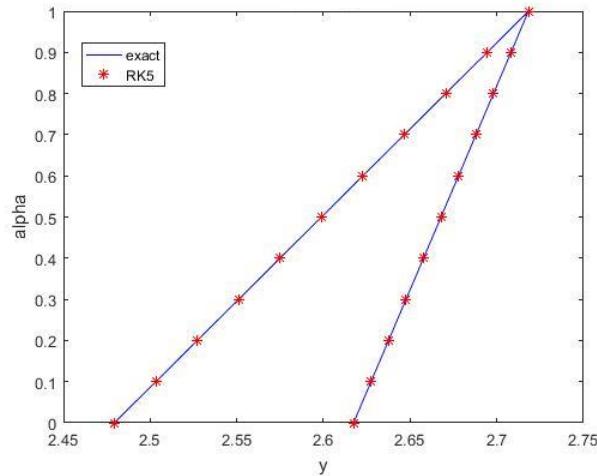
As shown in Tables 3.3.1-6 , the Runge-Kutta of order five method with triangular fuzzy number as initial condition gave accurate results with small  $h$ .

Case 2: using 2 – differentiable

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 3.3.7-9 and Figs 3.3.4-6 respectively, and the absolute errors of the approximate results in Tables 3.3.10-12.

**Table 3.3.7:** Numerical values for the exact (2 – differentiable) and approximate solution (Runge-kutta) for  $t = 1$

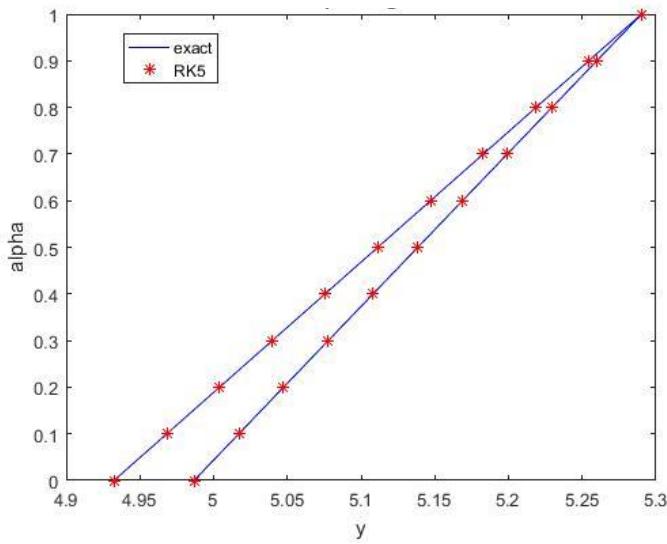
$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.479411818960710	2.617366609400000	2.479411818379155	2.617366608849098
0.1	2.503298819910544	2.627458131305905	2.503298819326746	2.627458130749695
0.2	2.527185820860377	2.637549653211810	2.527185820274337	2.637549652650292
0.3	2.551072821810211	2.647641175117714	2.551072821221929	2.647641174550889
0.4	2.574959822760044	2.657732697023619	2.574959822169520	2.657732696451486
0.5	2.598846823709878	2.667824218929523	2.598846823117111	2.667824218352083
0.6	2.622733824659711	2.677915740835428	2.622733824064703	2.677915740252680
0.7	2.646620825609545	2.688007262741332	2.646620825012294	2.688007262153277
0.8	2.670507826559379	2.698098784647236	2.670507825959885	2.698098784053874
0.9	2.694394827509212	2.708190306553141	2.694394826907477	2.708190305954471
1	2.718281828459046	2.718281828459046	2.718281827855068	2.718281827855068



**Figure 3.3.4:** Exact and RK5 solutions for  $t = 1$

**Table 3.3.8:** Numerical values for the exact (2 – differentiable) and approximate solutions (Runge-kutta) for  $t = 1.5$

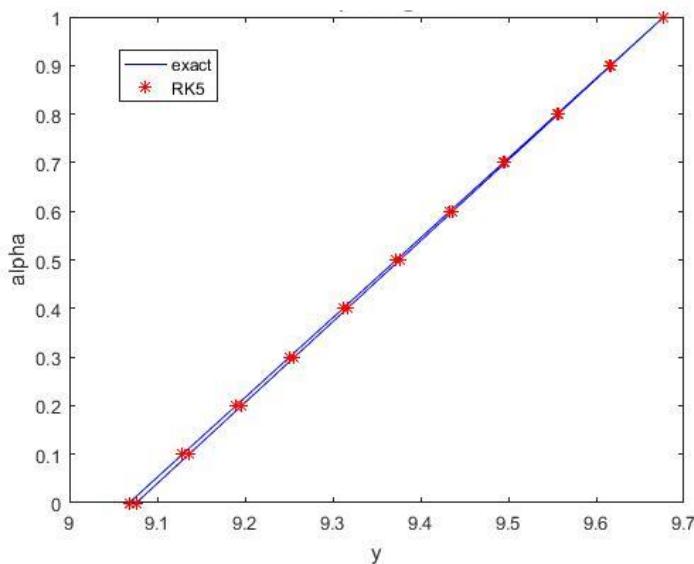
$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	4.932442377592928	4.986723357976558	4.932442375089241	4.986723355475636
0.1	4.968220312397341	5.017073194742609	4.968220309877111	5.017073192224866
0.2	5.003998247201754	5.047423031508658	5.003998244664980	5.047423028974096
0.3	5.039776182006168	5.077772868274710	5.039776179452850	5.077772865723326
0.4	5.075554116810579	5.108122705040760	5.075554114240719	5.108122702472556
0.5	5.111332051614995	5.138472541806809	5.111332049028587	5.138472539221786
0.6	5.147109986419408	5.168822378572860	5.147109983816458	5.168822375971016
0.7	5.182887921223820	5.199172215338908	5.182887918604327	5.199172212720246
0.8	5.218665856028234	5.229522052104959	5.218665853392196	5.229522049469475
0.9	5.254443790832648	5.259871888871011	5.254443788180066	5.259871886218705
1	5.290221725637062	5.290221725637062	5.290221722967934	5.290221722967934



**Figure 3.3.5:** Exact and RK5 solutions for  $t = 1.5$

**Table 3.3.9:** Numerical values for the exact (2 – differentiable) and approximate solutions (Runge-kutta) for  $t = 2$

$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	9.068147228770254	9.076182156900588	9.068147223990255	9.076182152148281
0.1	9.129030073129007	9.136261508446307	9.129030068318619	9.136261503660840
0.2	9.189912917487760	9.196340859992027	9.189912912646982	9.196340855173402
0.3	9.250795761846515	9.256420211537749	9.250795756975345	9.256420206685963
0.4	9.311678606205266	9.316499563083468	9.311678601303708	9.316499558198522
0.5	9.372561450564021	9.376578914629187	9.372561445632069	9.376578909711082
0.6	9.433444294922772	9.436658266174906	9.433444289960434	9.436658261223645
0.7	9.494327139281523	9.496737617720623	9.494327134288797	9.496737612736204
0.8	9.555209983640278	9.556816969266345	9.555209978617159	9.556816964248764
0.9	9.616092827999033	9.616896320812066	9.616092822945523	9.616896315761325
1	9.676975672357788	9.676975672357788	9.676975667273885	9.676975667273885



**Figure 3.3.6:** Exact and RK5 solutions for  $t = 2$

**Table 3.3.10:** The absolute errors of the Runge-kutta method (2 – differentiable) for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$5.8156 \times 10^{-10}$	$5.5090 \times 10^{-10}$
0.1	$5.8380 \times 10^{-10}$	$5.5621 \times 10^{-10}$
0.2	$5.8604 \times 10^{-10}$	$5.6152 \times 10^{-10}$
0.3	$5.8828 \times 10^{-10}$	$5.6682 \times 10^{-10}$
0.4	$5.9052 \times 10^{-10}$	$5.7213 \times 10^{-10}$
0.5	$5.9277 \times 10^{-10}$	$5.7744 \times 10^{-10}$
0.6	$5.9501 \times 10^{-10}$	$5.8275 \times 10^{-10}$
0.7	$5.9725 \times 10^{-10}$	$5.8805 \times 10^{-10}$
0.8	$5.9949 \times 10^{-10}$	$5.9336 \times 10^{-10}$
0.9	$6.0174 \times 10^{-10}$	$5.9867 \times 10^{-10}$
1	$6.0398 \times 10^{-10}$	$6.0398 \times 10^{-10}$

**Table 3.3.11:** The absolute errors of the Runge-kutta method (2 – differentiable) for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$2.5037 \times 10^{-9}$	$2.5009 \times 10^{-9}$
0.1	$2.5202 \times 10^{-9}$	$2.5177 \times 10^{-9}$
0.2	$2.5368 \times 10^{-9}$	$2.5346 \times 10^{-9}$
0.3	$2.5533 \times 10^{-9}$	$2.5514 \times 10^{-9}$
0.4	$2.5699 \times 10^{-9}$	$2.5682 \times 10^{-9}$
0.5	$2.5864 \times 10^{-9}$	$2.5850 \times 10^{-9}$
0.6	$2.6029 \times 10^{-9}$	$2.6018 \times 10^{-9}$
0.7	$2.6195 \times 10^{-9}$	$2.6187 \times 10^{-9}$
0.8	$2.6360 \times 10^{-9}$	$2.6355 \times 10^{-9}$
0.9	$2.6526 \times 10^{-9}$	$2.6523 \times 10^{-9}$
1	$2.6691 \times 10^{-9}$	$2.6691 \times 10^{-9}$

**Table 3.3.12:** The absolute errors of the Runge-kutta method (2 – differentiable) for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$4.7800 \times 10^{-9}$	$4.7523 \times 10^{-9}$
0.1	$4.8104 \times 10^{-9}$	$4.7855 \times 10^{-9}$
0.2	$4.8408 \times 10^{-9}$	$4.8186 \times 10^{-9}$
0.3	$4.8712 \times 10^{-9}$	$4.8518 \times 10^{-9}$
0.4	$4.9016 \times 10^{-9}$	$4.8849 \times 10^{-9}$
0.5	$4.9320 \times 10^{-9}$	$4.9181 \times 10^{-9}$
0.6	$4.9623 \times 10^{-9}$	$4.9513 \times 10^{-9}$
0.7	$4.9927 \times 10^{-9}$	$4.9844 \times 10^{-9}$
0.8	$5.0231 \times 10^{-9}$	$5.0176 \times 10^{-9}$
0.9	$5.0535 \times 10^{-9}$	$5.0507 \times 10^{-9}$
1	$5.0839 \times 10^{-9}$	$5.0839 \times 10^{-9}$

As shown in Tables 3.3.7-12 , the Runge-Kutta of order five method with triangular fuzzy number as initial condition gave accurate results with small  $h$ .

b- Let  $y(0) = (0.25, 0.75, 4)$

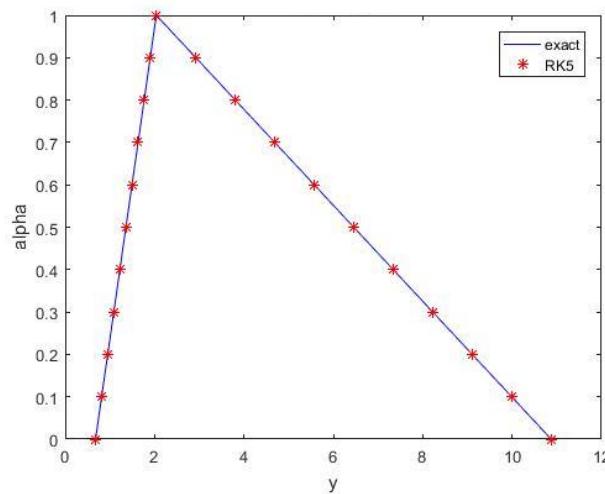
$$y(0, \alpha) = [0.25 + 0.5\alpha, 4 - 3.25\alpha], \quad 0 \leq \alpha \leq 1$$

using 1 – differentiable

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 3.3.13-15 and Figs 3.3.7-9 respectively, and the absolute errors of the approximate results in Tables 3.3.16-18.

**Table 3.3.13:** Numerical values for the exact and approximate solutions (Runge-kutta) for  $t = 1$

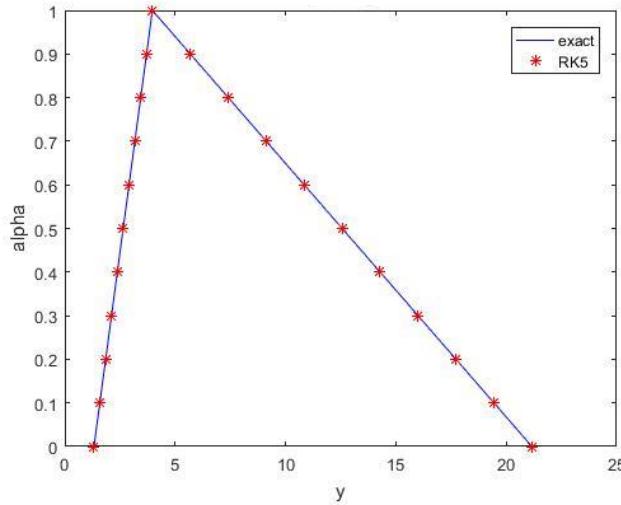
$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	0.6795704571147614	10.87312731383618	0.6795704569637669	10.87312731142027
0.1	0.8154845485377137	9.989685719586992	0.8154845483565204	9.989685717367374
0.2	0.9513986399606659	9.106244125337803	0.9513986397492737	9.106244123314477
0.3	1.087312731383618	8.222802531088613	1.087312731142027	8.222802529261580
0.4	1.223226822806571	7.339360936839423	1.223226822534781	7.339360935208683
0.5	1.359140914229523	6.455919342590233	1.359140913927534	6.455919341155786
0.6	1.495055005652475	5.572477748341043	1.495055005320287	5.572477747102889
0.7	1.630969097075428	4.689036154091854	1.630969096713041	4.689036153049992
0.8	1.766883188498380	3.805594559842664	1.766883188105794	3.805594558997095
0.9	1.902797279921332	2.922152965593473	1.902797279498548	2.922152964944198
1	2.038711371344284	2.038711371344284	2.038711370891301	2.038711370891301



**Figure 3.3.7:** Exact and RK5 solutions for  $t = 1$

**Table 3.3.14:** Numerical values for the exact and approximate solutions (Runge-kutta) for  $t = 1.5$

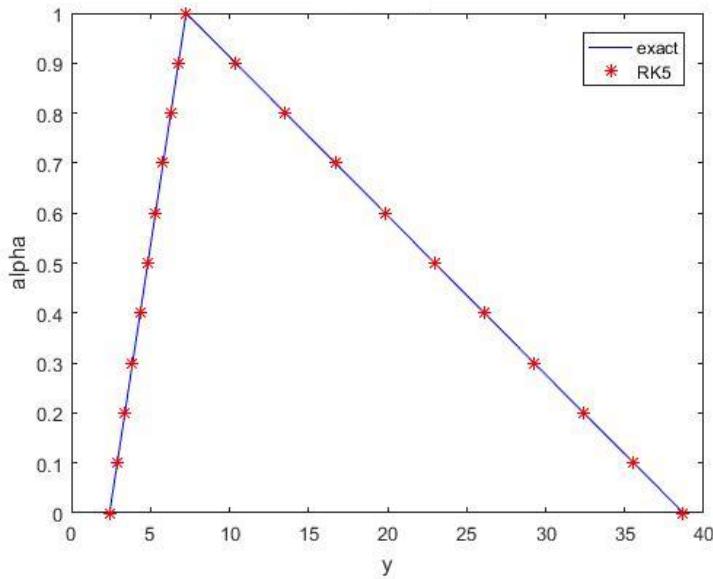
$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	1.322555431409265	21.16088690254824	1.322555430741984	21.16088689187173
0.1	1.587066517691118	19.44156484171619	1.587066516890380	19.44156483190716
0.2	1.851577603972971	17.72224278088415	1.851577603038777	17.72224277194258
0.3	2.116088690254824	16.00292072005211	2.116088689187174	16.00292071197800
0.4	2.380599776536677	14.28359865922006	2.380599775335571	14.28359865201342
0.5	2.645110862818530	12.56427659838802	2.645110861483967	12.56427659204884
0.6	2.909621949100383	10.84495453755597	2.909621947632363	10.84495453208427
0.7	3.174133035382237	9.125632476723929	3.174133033780761	9.125632472119687
0.8	3.438644121664089	7.406310415891883	3.438644119929157	7.406310412155108
0.9	3.703155207945942	5.686988355059838	3.703155206077555	5.686988352190530
1	3.967666294227795	3.967666294227795	3.967666292225951	3.967666292225951



**Figure 3.3.8:** Exact and RK5 solutions for  $t = 1.5$

**Table 3.3.15:** Numerical values for the exact and approximate solutions (Runge-kutta) for  $t = 2$

$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.419243918089446	38.70790268943114	2.419243916818472	38.70790266909553
0.1	2.903092701707335	35.56288559591486	2.903092700182165	35.56288557723153
0.2	3.386941485325224	32.41786850239858	3.386941483545860	32.41786848536751
0.3	3.870790268943114	29.27285140888230	3.870790266909554	29.27285139350350
0.4	4.354639052561003	26.12783431536602	4.354639050273249	26.12783430163949
0.5	4.838487836178892	22.98281722184974	4.838487833636942	22.98281720977547
0.6	5.322336619796782	19.83780012833346	5.322336617000635	19.83780011791147
0.7	5.806185403414671	16.69278303481718	5.806185400364332	16.69278302604745
0.8	6.290034187032560	13.54776594130090	6.290034183728024	13.54776593418344
0.9	6.773882970650448	10.40274884778462	6.773882967091722	10.40274884231943
1	7.257731754268338	7.257731754268338	7.257731750455414	7.257731750455414



**Figure 3.3.9:** Exact and RK5 solutions for  $t = 2$

**Table 3.3.16:** The absolute errors of the Runge-kutta method for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.5099 \times 10^{-10}$	$2.4159 \times 10^{-9}$
0.1	$1.8119 \times 10^{-10}$	$2.2196 \times 10^{-9}$
0.2	$2.1139 \times 10^{-10}$	$2.0233 \times 10^{-9}$
0.3	$2.4159 \times 10^{-10}$	$1.8270 \times 10^{-9}$
0.4	$2.7179 \times 10^{-10}$	$1.6307 \times 10^{-9}$
0.5	$3.0199 \times 10^{-10}$	$1.4344 \times 10^{-9}$
0.6	$3.3219 \times 10^{-10}$	$1.2382 \times 10^{-9}$
0.7	$3.6239 \times 10^{-10}$	$1.0419 \times 10^{-9}$
0.8	$3.9259 \times 10^{-10}$	$8.4557 \times 10^{-10}$
0.9	$4.2278 \times 10^{-10}$	$6.4927 \times 10^{-10}$
1	$4.5298 \times 10^{-10}$	$4.5298 \times 10^{-10}$

**Table 3.3.17:** The absolute errors of the Runge-kutta method for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$6.6728 \times 10^{-10}$	$1.0677 \times 10^{-8}$
0.1	$8.0074 \times 10^{-10}$	$9.8090 \times 10^{-9}$
0.2	$9.3419 \times 10^{-10}$	$8.9416 \times 10^{-9}$
0.3	$1.0676 \times 10^{-9}$	$8.0741 \times 10^{-9}$
0.4	$1.2011 \times 10^{-9}$	$7.2066 \times 10^{-9}$
0.5	$1.3346 \times 10^{-9}$	$6.3392 \times 10^{-9}$
0.6	$1.4680 \times 10^{-9}$	$5.4717 \times 10^{-9}$
0.7	$1.6015 \times 10^{-9}$	$4.6042 \times 10^{-9}$
0.8	$1.7349 \times 10^{-9}$	$3.7368 \times 10^{-9}$
0.9	$1.8684 \times 10^{-9}$	$2.8693 \times 10^{-9}$
1	$2.0018 \times 10^{-9}$	$2.0018 \times 10^{-9}$

**Table 3.3.18:** The absolute errors of the Runge-kutta method for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.2710 \times 10^{-9}$	$2.0336 \times 10^{-8}$
0.1	$1.5252 \times 10^{-9}$	$1.8683 \times 10^{-8}$
0.2	$1.7794 \times 10^{-9}$	$1.7031 \times 10^{-8}$
0.3	$2.0336 \times 10^{-9}$	$1.5379 \times 10^{-8}$
0.4	$2.2878 \times 10^{-9}$	$1.3727 \times 10^{-8}$
0.5	$2.5419 \times 10^{-9}$	$1.2074 \times 10^{-8}$
0.6	$2.7961 \times 10^{-9}$	$1.0422 \times 10^{-8}$
0.7	$3.0503 \times 10^{-9}$	$8.7697 \times 10^{-9}$
0.8	$3.3045 \times 10^{-9}$	$7.1175 \times 10^{-9}$
0.9	$3.5587 \times 10^{-9}$	$5.4652 \times 10^{-9}$
1	$3.8129 \times 10^{-9}$	$3.8129 \times 10^{-9}$

As shown in Tables 3.3.13-18 , the Runge-Kutta of order five method with another triangular fuzzy number as initial condition gave accurate results with small  $h$ .

### 3.4 General Linear Methods (GLM)

In this section, we present the derivation of a GLM based on linear  $k$ -step Adams schemes for solving hybrid fuzzy initial value problem. Assume that for an equally spaced points  $0 = t_0 < t_1 < \dots < t_N = T$  at  $t_n$  the exact solutions are indicated by  $Y(t_n, \alpha) = [\underline{Y}(t_n, \alpha), \bar{Y}(t_n, \alpha)]$ . Also assume that  $y(t_n, \alpha) = [\underline{y}(t_n, \alpha), \bar{y}(t_n, \alpha)]$  are the approximate value. The  $k$ -step Adams–Bashforth methods can be written as:

By 1 –differentiable

$$\begin{cases} \underline{y}_\alpha(t_{n+k}, \alpha) = \underline{y}_\alpha(t_{n+k-1}, \alpha) + h \sum_{j=0}^k \beta_j f(t_{n+j}, \underline{y}_\alpha(t_{n+j}, \alpha)) \\ \bar{y}_\alpha(t_{n+k}, \alpha) = \bar{y}_\alpha(t_{n+k-1}, \alpha) + h \sum_{j=0}^k \beta_j f(t_{n+j}, \bar{y}_\alpha(t_{n+j}, \alpha)) \end{cases} \quad (37)$$

By 2 –differentiable

$$\begin{cases} \underline{y}_\alpha(t_{n+k}, \alpha) = \underline{y}_\alpha(t_{n+k-1}, \alpha) + h \sum_{j=0}^k \beta_j f(t_{n+j}, \bar{y}_\alpha(t_{n+j}, \alpha)) \\ \bar{y}_\alpha(t_{n+k}, \alpha) = \bar{y}_\alpha(t_{n+k-1}, \alpha) + h \sum_{j=0}^k \beta_j f(t_{n+j}, \underline{y}_\alpha(t_{n+j}, \alpha)) \end{cases} \quad (38)$$

The Adams schemes are characterized by their first characteristic polynomial as  $\rho(\alpha) = \alpha^k - \alpha^{k-1}$ . Therefore, we have  $y_{n+k} = y_{n+k-1} + h \sum_{j=0}^k \beta_j f_{n+j}$ . In this setting we can find their corresponding general linear method framework. In GLM representation, first determine the input and output vectors and then find the corresponding matrices. For this end consider the input and the output approximation of GLM as follows:

$$y^{[n-1]} = \begin{bmatrix} y_{n+k-1} \\ hf_{n+k-1} \\ hf_{n+k-2} \\ \vdots \\ hf_{n+1} \\ hf_n \end{bmatrix}, \quad y^{[n]} = \begin{bmatrix} y_{n+k} \\ hf_{n+k} \\ hf_{n+k-1} \\ \vdots \\ hf_{n+2} \\ hf_{n+1} \end{bmatrix}$$

Similarly, a linear  $k$ -steps Method under strongly generalized differentiability (37) and (38) can be represented in the form of GLM. For this representation the input vectors for the GLM form of (37) and (38) are indicated by  $y_{1\alpha}^{[n-1]} = [\underline{y}_{1\alpha}^{[n-1]}, \bar{y}_{1\alpha}^{[n-1]}]$  and  $y_{2\alpha}^{[n-1]} = [\underline{y}_{2\alpha}^{[n-1]}, \bar{y}_{2\alpha}^{[n-1]}]$  under 1 and 2 – differentiability, respectively. Corresponding to the input vectors, the output vectors are indicated by  $y_{1\alpha}^{[n]} = [\underline{y}_{1\alpha}^{[n]}, \bar{y}_{1\alpha}^{[n]}]$  and  $y_{2\alpha}^{[n]} = [\underline{y}_{2\alpha}^{[n]}, \bar{y}_{2\alpha}^{[n]}]$  under 1 and 2 – differentiability, respectively. Now, consider the input approximation of general linear methods in terms of 1 – differentiability as:

$$\underline{y}_{1\alpha}^{[n-1]} = \begin{bmatrix} \underline{y}_{n+k-1\alpha} \\ h\underline{f}_{n+k-1\alpha} \\ h\underline{f}_{n+k-2\alpha} \\ \vdots \\ h\underline{f}_{n+1\alpha} \\ h\underline{f}_n \end{bmatrix}, \quad \bar{y}_{1\alpha}^{[n-1]} = \begin{bmatrix} \bar{y}_{n+k-1\alpha} \\ h\bar{f}_{n+k-1\alpha} \\ h\bar{f}_{n+k-2\alpha} \\ \vdots \\ h\bar{f}_{n+1\alpha} \\ h\bar{f}_n \end{bmatrix}$$

and under the 2 – differentiability we obtain the following input vectors:

$$\underline{y}_{2\alpha}^{[n-1]} = \begin{bmatrix} \underline{y}_{n+k-1} \\ h\underline{f}_{n+k-1} \\ h\underline{f}_{n+k-2} \\ \vdots \\ \vdots \\ h\underline{f}_{n+1} \\ h\underline{f}_n \end{bmatrix}, \quad \bar{y}_{2\alpha}^{[n-1]} = \begin{bmatrix} \bar{y}_{n+k-1} \\ h\underline{f}_{n+k-1} \\ h\underline{f}_{n+k-2} \\ \vdots \\ \vdots \\ h\underline{f}_{n+1} \\ h\underline{f}_n \end{bmatrix}$$

By considering the above input vectors, the fuzzy GLM (FGLM) form of (37) and (38) can be formulated in case of 1 – differentiability as:

$$\begin{bmatrix} Y_{1\alpha} \\ y_{1\alpha}^{[n]} \end{bmatrix} = \begin{bmatrix} A & U \\ B & V \end{bmatrix} \begin{bmatrix} hf_{1\alpha}(Y_{1\alpha}) \\ y_{1\alpha}^{[n-1]} \end{bmatrix}$$

and in case of 2 – differentiability has the form:

$$\begin{bmatrix} Y_{2\alpha} \\ y_{2\alpha}^{[n]} \end{bmatrix} = \begin{bmatrix} A & U \\ B & V \end{bmatrix} \begin{bmatrix} hf_{2\alpha}(Y_{2\alpha}) \\ y_{2\alpha}^{[n-1]} \end{bmatrix}$$

Where  $Y_{1\alpha} = [\underline{Y}_{1\alpha}, \bar{Y}_{1\alpha}]$  and  $Y_{2\alpha} = [\underline{Y}_{2\alpha}, \bar{Y}_{2\alpha}]$  are internal stages under 1 and 2 –differentiability, respectively. Also

$$\begin{bmatrix} A & U \\ B & V \end{bmatrix} = \begin{bmatrix} 0 & 1 & \beta_{k-1} & \cdots & \beta_1 & \beta_0 \\ 0 & 1 & \beta_{k-1} & \cdots & \beta_1 & \beta_0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

Now, we consider two examples of FGLMs form of k-step methods under strongly generalized differentiability for  $k = 4, 5$ . First, Consider  $k = 4$ . The input vectors for  $k = 4$  under 1 and 2 – differentiability are as follow, respectively:

$$\underline{y}_{1\alpha}^{[n-1]} = \begin{bmatrix} \underline{y}_{1\alpha}(t_{n+3}) \\ h f_{1\alpha}(t_{n+3}, \underline{y}_{1\alpha}(t_{n+3})) \\ h f_{1\alpha}(t_{n+2}, \underline{y}_{1\alpha}(t_{n+2})) \\ h f_{1\alpha}(t_{n+1}, \underline{y}_{1\alpha}(t_{n+1})) \\ h f_{1\alpha}(t_n, \underline{y}_{1\alpha}(t_n)) \end{bmatrix}, \quad \bar{y}_{\alpha 1}^{[n-1]} = \begin{bmatrix} \bar{y}_{1\alpha}(t_{n+3}) \\ h f_{1\alpha}(t_{n+3}, \bar{y}_{1\alpha}(t_{n+3})) \\ h f_{1\alpha}(t_{n+2}, \bar{y}_{1\alpha}(t_{n+2})) \\ h f_{1\alpha}(t_{n+1}, \bar{y}_{1\alpha}(t_{n+1})) \\ h f_{1\alpha}(t_n, \bar{y}_{1\alpha}(t_n)) \end{bmatrix}$$

$$\underline{y}_{2\alpha}^{[n-1]} = \begin{bmatrix} \underline{y}_{2\alpha}(t_{n+3}) \\ h f_{2\alpha}(t_{n+3}, \bar{y}_{2\alpha}(t_{n+3})) \\ h f_{2\alpha}(t_{n+2}, \bar{y}_{2\alpha}(t_{n+2})) \\ h f_{2\alpha}(t_{n+1}, \bar{y}_{2\alpha}(t_{n+1})) \\ h f_{2\alpha}(t_n, \bar{y}_{2\alpha}(t_n)) \end{bmatrix}, \quad \bar{y}_{2\alpha}^{[n-1]} = \begin{bmatrix} \bar{y}_{2\alpha}(t_{n+3}) \\ h f_{2\alpha}(t_{n+3}, \underline{y}_{2\alpha}(t_{n+3})) \\ h f_{2\alpha}(t_{n+2}, \underline{y}_{2\alpha}(t_{n+2})) \\ h f_{2\alpha}(t_{n+1}, \underline{y}_{2\alpha}(t_{n+1})) \\ h f_{2\alpha}(t_n, \underline{y}_{2\alpha}(t_n)) \end{bmatrix}$$

And

$$\begin{bmatrix} 0 & 1 & \frac{55}{24} & \frac{-59}{24} & \frac{37}{24} & \frac{-9}{24} \\ 0 & 1 & \frac{55}{24} & \frac{-59}{24} & \frac{37}{24} & \frac{-9}{24} \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

similarly, for  $k = 5$  we obtain

$$\underline{y}_{1\alpha}^{[n-1]} = \begin{bmatrix} \underline{y}_{1\alpha}(t_{n+4}) \\ h f_{1\alpha}(t_{n+4}, \underline{y}_{1\alpha}(t_{n+4})) \\ h f_{1\alpha}(t_{n+3}, \underline{y}_{1\alpha}(t_{n+3})) \\ h f_{1\alpha}(t_{n+2}, \underline{y}_{1\alpha}(t_{n+2})) \\ h f_{1\alpha}(t_{n+1}, \underline{y}_{1\alpha}(t_{n+1})) \\ h f_{1\alpha}(t_n, \underline{y}_{1\alpha}(t_n)) \end{bmatrix}, \quad \bar{y}_{1\alpha}^{[n-1]} = \begin{bmatrix} \bar{y}_{1\alpha}(t_{n+4}) \\ h f_{1\alpha}(t_{n+4}, \bar{y}_{1\alpha}(t_{n+4})) \\ h f_{1\alpha}(t_{n+3}, \bar{y}_{1\alpha}(t_{n+3})) \\ h f_{1\alpha}(t_{n+2}, \bar{y}_{1\alpha}(t_{n+2})) \\ h f_{1\alpha}(t_{n+1}, \bar{y}_{1\alpha}(t_{n+1})) \\ h f_{1\alpha}(t_n, \bar{y}_{1\alpha}(t_n)) \end{bmatrix}$$

$$\underline{y}_{2\alpha}^{[n-1]} = \begin{bmatrix} \underline{y}_{2\alpha}(t_{n+4}) \\ h f_{2\alpha}(t_{n+4}, \underline{y}_{2\alpha}(t_{n+4})) \\ h f_{2\alpha}(t_{n+3}, \bar{y}_{2\alpha}(t_{n+3})) \\ h f_{2\alpha}(t_{n+2}, \bar{y}_{2\alpha}(t_{n+2})) \\ h f_{2\alpha}(t_{n+1}, \bar{y}_{2\alpha}(t_{n+1})) \\ h f_{2\alpha}(t_n, \bar{y}_{2\alpha}(t_n)) \end{bmatrix}, \quad \bar{y}_{2\alpha}^{[n-1]} = \begin{bmatrix} \bar{y}_{2\alpha}(t_{n+4}) \\ h f_{2\alpha}(t_{n+4}, \underline{y}_{2\alpha}(t_{n+4})) \\ h f_{2\alpha}(t_{n+3}, \underline{y}_{2\alpha}(t_{n+3})) \\ h f_{2\alpha}(t_{n+2}, \underline{y}_{2\alpha}(t_{n+2})) \\ h f_{2\alpha}(t_{n+1}, \underline{y}_{2\alpha}(t_{n+1})) \\ h f_{2\alpha}(t_n, \underline{y}_{2\alpha}(t_n)) \end{bmatrix}$$

And

$$\begin{bmatrix} 0 & 1 & \frac{1901}{720} & \frac{-2774}{720} & \frac{2616}{720} & \frac{-1274}{7200} & \frac{251}{720} \\ 0 & 1 & \frac{1901}{720} & \frac{-2774}{720} & \frac{2616}{720} & \frac{-1274}{720} & \frac{251}{720} \\ 0 & 1 & \frac{720}{720} & \frac{720}{720} & \frac{720}{720} & \frac{720}{720} & \frac{720}{720} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

**Theorem 6:** A FGLM is stable

The proof is elaborated in [22]

Solve Example (29) to show the numerical results of FGLMs for solving fuzzy differential equations. using the FGLMs ( $k = 4, 5$ ) presented in this section.

Let  $N = 100$  and  $h = 0.02$ .

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 3.4.1-3 and Figs 3.4.1-3 respectively, and the absolute errors of the approximate results in Tables 3.4.4-6.

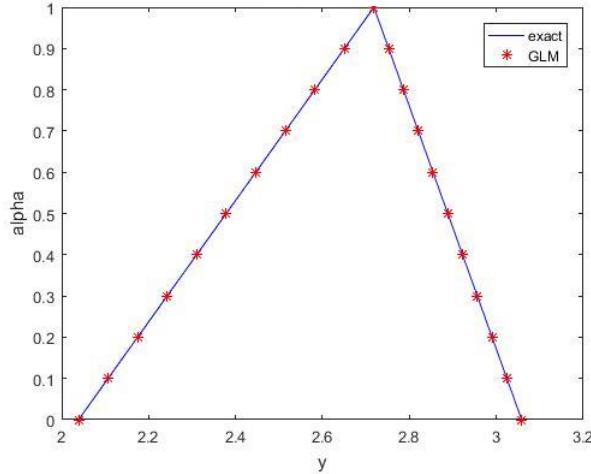
a- Let  $y(0) = (0.75, 1, 1.125)$

$$y(0, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha], \quad 0 \leq \alpha \leq 1$$

Case 1: using 1 – differentiable

**Table 3.4.1:** Numerical values for the exact (1 – differentiable) and approximate solutions (FGLMs) for  $t = 1$

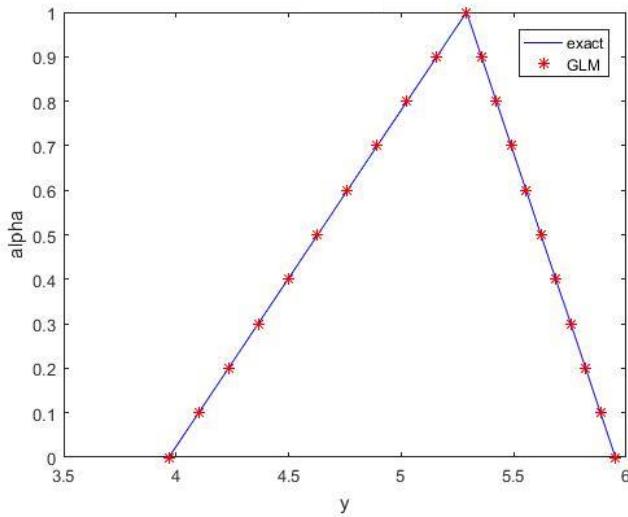
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0.0	2.038711371344284	3.058067057016426	2.038711267561744	3.058066901342616
0.1	2.106668417055760	3.024088534160689	2.106668309813803	3.024088380216586
0.2	2.174625462767236	2.990110011304950	2.174625352065861	2.990109859090558
0.3	2.242582508478713	2.956131488449212	2.242582394317919	2.956131337964530
0.4	2.310539554190189	2.922152965593474	2.310539436569978	2.922152816838501
0.5	2.378496599901665	2.888174442737736	2.378496478822036	2.888174295712471
0.6	2.446453645613141	2.854195919881998	2.446453521074094	2.854195774586440
0.7	2.514410691324617	2.820217397026260	2.514410563326151	2.820217253460412
0.8	2.582367737036093	2.786238874170521	2.582367605578210	2.786238732334387
0.9	2.650324782747569	2.752260351314784	2.650324647830267	2.752260211208354
1	2.718281828459046	2.718281828459046	2.718281690082326	2.718281690082326



**Figure 3.4.1:** Exact and FGLM solutions for  $t = 1$

**Table 3.4.2:** Numerical values for the exact (1 – differentiable) and approximate solutions (FGLMs) for  $t = 1.5$

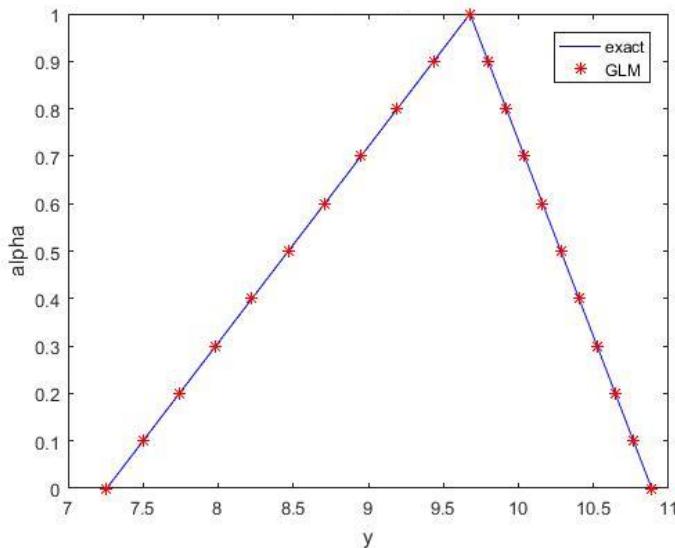
$\alpha$	Exact		FGLM	
	$\underline{Y}(t, \alpha)$	$\bar{Y}(t, \alpha)$	$\underline{y}(t, \alpha)$	$\bar{y}(t, \alpha)$
0.0	3.967666294227795	5.951499441341692	3.967665851557501	5.951498777336254
0.1	4.099921837368721	5.885371669771230	4.099921379942754	5.885371013143627
0.2	4.232177380509648	5.819243898200766	4.232176908328005	5.819243248951001
0.3	4.364432923650575	5.753116126630302	4.364432436713252	5.753115484758379
0.4	4.496688466791501	5.686988355059839	4.496687965098505	5.686987720565753
0.5	4.628944009932427	5.620860583489376	4.628943493483756	5.620859956373129
0.6	4.761199553073354	5.554732811918913	4.761199021869006	5.554732192180500
0.7	4.893455096214280	5.488605040348450	4.8934550254253	5.488604427987878
0.8	5.025710639355206	5.422477268777985	5.025710078639504	5.422476663795258
0.9	5.157966182496133	5.356349497207523	5.157965607024751	5.356348899602628
1	5.290221725637060	5.290221725637060	5.290221135410005	5.290221135410005



**Figure 3.4.2:** Exact and FGML solutions for  $t = 1.5$

**Table 3.4.3:** Numerical values for the exact (1 – differentiable) and approximate solutions (FGMLs) for  $t = 2$

$\alpha$	Exact		FGML	
	$\underline{Y}(t, \alpha)$	$\bar{Y}(t, \alpha)$	$\underline{y}(t, \alpha)$	$\bar{y}(t, \alpha)$
0.0	7.257731754268338	10.88659763140251	7.257730912060317	10.88659636809048
0.1	7.499656146077282	10.76563543549804	7.499655275795665	10.76563418622281
0.2	7.741580537886227	10.64467323959356	7.741579639531012	10.64467200435513
0.3	7.983504929695171	10.52371104368909	7.983504003266352	10.52370982248747
0.4	8.225429321504116	10.40274884778462	8.225428367001701	10.40274764061979
0.5	8.467353713313061	10.28178665188015	8.467352730737046	10.28178545875212
0.6	8.709278105122007	10.16082445597567	8.709277094472389	10.16082327688444
0.7	8.951202496930950	10.03986226007120	8.951201458207732	10.03986109501678
0.8	9.193126888739894	9.918900064166728	9.193125821943072	9.918898913149114
0.9	9.435051280548839	9.797937868262256	9.435050185678415	9.797936731281430
1	9.676975672357784	9.676975672357784	9.676974549413766	9.676974549413766



**Figure 3.4.3:** Exact and FGML solutions for  $t = 2$

**Table 3.4.4:** The absolute errors of the FGML (1 – differentiable) for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.0378 \times 10^{-7}$	$1.5567 \times 10^{-7}$
0.1	$1.0724 \times 10^{-7}$	$1.5394 \times 10^{-7}$
0.2	$1.1070 \times 10^{-7}$	$1.5221 \times 10^{-7}$
0.3	$1.1416 \times 10^{-7}$	$1.5048 \times 10^{-7}$
0.4	$1.1762 \times 10^{-7}$	$1.4875 \times 10^{-7}$
0.5	$1.2108 \times 10^{-7}$	$1.4703 \times 10^{-7}$
0.6	$1.2454 \times 10^{-7}$	$1.4530 \times 10^{-7}$
0.7	$1.2800 \times 10^{-7}$	$1.4357 \times 10^{-7}$
0.8	$1.3146 \times 10^{-7}$	$1.4184 \times 10^{-7}$
0.9	$1.3492 \times 10^{-7}$	$1.4011 \times 10^{-7}$
1	$1.3838 \times 10^{-7}$	$1.3838 \times 10^{-7}$

**Table 3.4.5:** The absolute errors of the FGML (1 – differentiable) for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$4.4267 \times 10^{-7}$	$6.6401 \times 10^{-7}$
0.1	$4.5743 \times 10^{-7}$	$6.5663 \times 10^{-7}$
0.2	$4.7218 \times 10^{-7}$	$6.4925 \times 10^{-7}$
0.3	$4.8694 \times 10^{-7}$	$6.4187 \times 10^{-7}$
0.4	$5.0169 \times 10^{-7}$	$6.3449 \times 10^{-7}$
0.5	$5.1645 \times 10^{-7}$	$6.2712 \times 10^{-7}$
0.6	$5.3120 \times 10^{-7}$	$6.1974 \times 10^{-7}$
0.7	$5.4596 \times 10^{-7}$	$6.1236 \times 10^{-7}$
0.8	$5.6072 \times 10^{-7}$	$6.0498 \times 10^{-7}$
0.9	$5.7547 \times 10^{-7}$	$5.9760 \times 10^{-7}$
1	$5.9023 \times 10^{-7}$	$5.9023 \times 10^{-7}$

**Table 3.4.6:** The absolute errors of the FGML (1 – differentiable) for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$8.4221 \times 10^{-7}$	$1.2633 \times 10^{-6}$
0.1	$8.7028 \times 10^{-7}$	$1.2493 \times 10^{-6}$
0.2	$8.9836 \times 10^{-7}$	$1.2352 \times 10^{-6}$
0.3	$9.2643 \times 10^{-7}$	$1.2212 \times 10^{-6}$
0.4	$9.5450 \times 10^{-7}$	$1.2072 \times 10^{-6}$
0.5	$9.8258 \times 10^{-7}$	$1.1931 \times 10^{-6}$
0.6	$1.0106 \times 10^{-6}$	$1.1791 \times 10^{-6}$
0.7	$1.0387 \times 10^{-6}$	$1.1651 \times 10^{-6}$
0.8	$1.0668 \times 10^{-6}$	$1.1510 \times 10^{-6}$
0.9	$1.0949 \times 10^{-6}$	$1.1370 \times 10^{-6}$
1	$1.1229 \times 10^{-6}$	$1.1229 \times 10^{-6}$

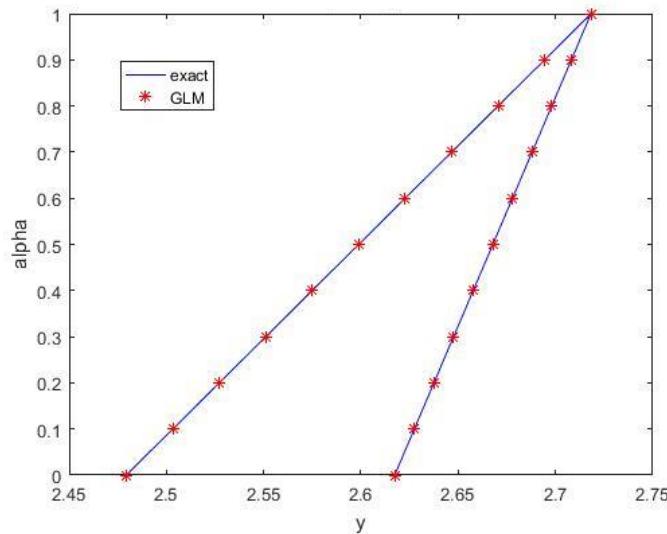
As shown in Tables 3.4.1-6, the fuzzy general linear method with triangular fuzzy number as initial condition gave less accurate than Runge-Kutta but it needed less number of steps.

### Case 2: using 2 –differentiable

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 3.4.7-9 and Figs 3.4.4-6 respectively, and the absolute errors of the approximate results in Tables 3.4.10-12.

**Table 3.4.7:** Numerical values for the exact (2 – differentiable) and approximate solutions (FGLMs) for  $t = 1$

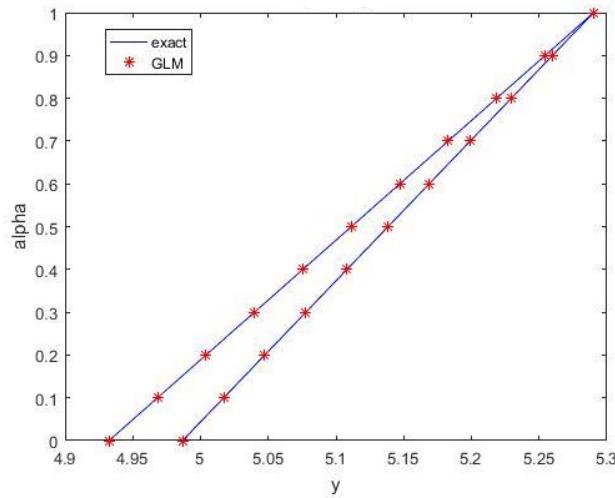
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.479411818960710	2.617366609400000	2.479411685495919	2.617366483408444
0.1	2.503298819910544	2.627458131305905	2.503298685954558	2.627458004075831
0.2	2.527185820860377	2.637549653211810	2.527185686413198	2.637549524743220
0.3	2.551072821810211	2.647641175117714	2.551072686871839	2.647641045410608
0.4	2.574959822760044	2.657732697023619	2.574959687330481	2.657732566077996
0.5	2.598846823709878	2.667824218929523	2.598846687789121	2.667824086745385
0.6	2.622733824659711	2.677915740835428	2.622733688247763	2.677915607412773
0.7	2.646620825609545	2.688007262741332	2.646620688706403	2.688007128080161
0.8	2.670507826559379	2.698098784647236	2.670507689165044	2.698098648747549
0.9	2.694394827509212	2.708190306553141	2.694394689623684	2.708190169414938
1	2.718281828459046	2.718281828459046	2.718281690082326	2.718281690082326



**Figure 3.4.4:** Exact and FGLM solutions for  $t = 1$

**Table 3.4.8:** Numerical values for the exact (2 – differentiable) and approximate solutions (FGLMs) for  $t = 1.5$

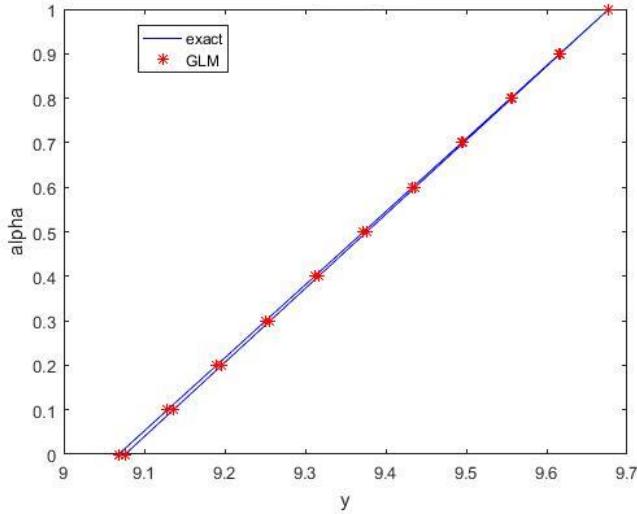
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	4.932442377592928	4.986723357976558	4.932441823847529	4.986722805046231
0.1	4.968220312397341	5.017073194742609	4.968219755003775	5.017072638082606
0.2	5.003998247201754	5.047423031508658	5.003997686160021	5.047422471118983
0.3	5.039776182006168	5.077772868274710	5.039775617316268	5.077772304155361
0.4	5.075554116810579	5.108122705040760	5.075553548472518	5.108122137191739
0.5	5.111332051614995	5.138472541806809	5.111331479628763	5.138471970228117
0.6	5.147109986419408	5.168822378572860	5.147109410785012	5.168821803264494
0.7	5.182887921223820	5.199172215338908	5.182887341941259	5.199171636300870
0.8	5.218665856028234	5.229522052104959	5.218665273097508	5.229521469337248
0.9	5.254443790832648	5.259871888871011	5.254443204253756	5.259871302373626
1	5.290221725637062	5.290221725637062	5.290221135410005	5.290221135410004



**Figure 3.4.5:** Exact and FGLM solutions for  $t = 1.5$

**Table 3.4.9 :** Numerical values for the exact (2 – differentiable) and approximate solutions (FGLMs) for  $t = 2$

$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	9.068147228770254	9.076182156900588	9.068146172830902	9.076181107319904
0.1	9.129030073129007	9.136261508446307	9.129029010489184	9.136260451529287
0.2	9.189912917487760	9.196340859992027	9.189911848147473	9.196339795738671
0.3	9.250795761846515	9.256420211537749	9.250794685805758	9.256419139948058
0.4	9.311678606205266	9.316499563083468	9.311677523464043	9.316498484157446
0.5	9.372561450564021	9.376578914629187	9.372560361122329	9.376577828366830
0.6	9.433444294922772	9.436658266174906	9.433443198780616	9.436657172576217
0.7	9.494327139281523	9.496737617720623	9.494326036438903	9.496736516785601
0.8	9.555209983640278	9.556816969266345	9.555208874097186	9.556815860994988
0.9	9.616092827999033	9.616896320812066	9.616091711755473	9.616895205204374
1	9.676975672357788	9.676975672357788	9.676974549413764	9.676974549413762



**Figure 3.4.6:** Exact and FGLM solutions for  $t = 2$

**Table 3.4.10:** The absolute errors of the FGLM (2 – differentiable) for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.3346 \times 10^{-7}$	$1.2599 \times 10^{-7}$
0.1	$1.3396 \times 10^{-7}$	$1.2723 \times 10^{-7}$
0.2	$1.3445 \times 10^{-7}$	$1.2847 \times 10^{-7}$
0.3	$1.3494 \times 10^{-7}$	$1.2971 \times 10^{-7}$
0.4	$1.3543 \times 10^{-7}$	$1.3095 \times 10^{-7}$
0.5	$1.3592 \times 10^{-7}$	$1.3218 \times 10^{-7}$
0.6	$1.3641 \times 10^{-7}$	$1.3342 \times 10^{-7}$
0.7	$1.3690 \times 10^{-7}$	$1.3466 \times 10^{-7}$
0.8	$1.3739 \times 10^{-7}$	$1.3590 \times 10^{-7}$
0.9	$1.3789 \times 10^{-7}$	$1.3714 \times 10^{-7}$
1	$1.3838 \times 10^{-7}$	$1.3838 \times 10^{-7}$

**Table 3.4.11 :** The absolute errors of the FGLM (2 – differentiable) for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$5.5375 \times 10^{-7}$	$5.5293 \times 10^{-7}$
0.1	$5.5739 \times 10^{-7}$	$5.5666 \times 10^{-7}$
0.2	$5.6104 \times 10^{-7}$	$5.6039 \times 10^{-7}$
0.3	$5.6469 \times 10^{-7}$	$5.6412 \times 10^{-7}$
0.4	$5.6834 \times 10^{-7}$	$5.6785 \times 10^{-7}$
0.5	$5.7199 \times 10^{-7}$	$5.7158 \times 10^{-7}$
0.6	$5.7563 \times 10^{-7}$	$5.7531 \times 10^{-7}$
0.7	$5.7928 \times 10^{-7}$	$5.7904 \times 10^{-7}$
0.8	$5.8293 \times 10^{-7}$	$5.8277 \times 10^{-7}$
0.9	$5.8658 \times 10^{-7}$	$5.8650 \times 10^{-7}$
1	$5.9023 \times 10^{-7}$	$5.9023 \times 10^{-7}$

**Table 3.4.12:** The absolute errors of the FGLM (2 – differentiable) for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.0559 \times 10^{-6}$	$1.0496 \times 10^{-6}$
0.1	$1.0626 \times 10^{-6}$	$1.0569 \times 10^{-6}$
0.2	$1.0693 \times 10^{-6}$	$1.0643 \times 10^{-6}$
0.3	$1.0760 \times 10^{-6}$	$1.0716 \times 10^{-6}$
0.4	$1.0827 \times 10^{-6}$	$1.0789 \times 10^{-6}$
0.5	$1.0894 \times 10^{-6}$	$1.0863 \times 10^{-6}$
0.6	$1.0961 \times 10^{-6}$	$1.0936 \times 10^{-6}$
0.7	$1.1028 \times 10^{-6}$	$1.1009 \times 10^{-6}$
0.8	$1.1095 \times 10^{-6}$	$1.1083 \times 10^{-6}$
0.9	$1.1162 \times 10^{-6}$	$1.1156 \times 10^{-6}$
1	$1.1229 \times 10^{-6}$	$1.1229 \times 10^{-6}$

As shown in Tables 3.4.7-12, the fuzzy general linear method with triangular fuzzy number as initial condition gave less accurate than Runge-Kutta but it needed less number of steps.

b- Let  $y(0) = (0.25, 0.75, 4)$

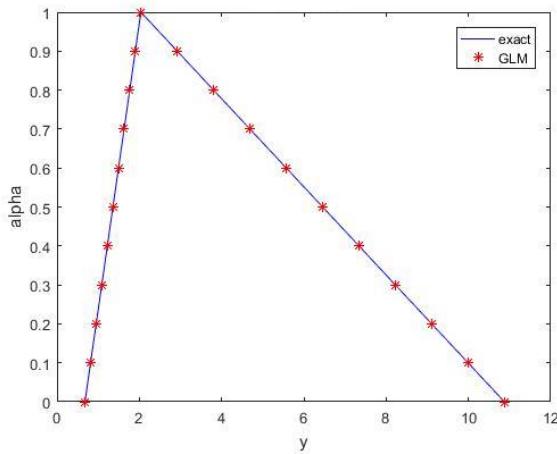
$$y(0, \alpha) = [0.25 + 0.5\alpha, 4 - 3.25\alpha], \quad 0 \leq \alpha \leq 1$$

using 1 –differentiable

We solve by MATLAB software the exact solutions with approximate results of this example are presented in Tables 3.4.13-15 and Figs 3.4.7-9 respectively, and the absolute errors of the approximate results in Tables 3.4.16-18.

**Table 3.4.13:** Numerical values for the exact and approximate solutions (FGLMs) for  $t = 1$

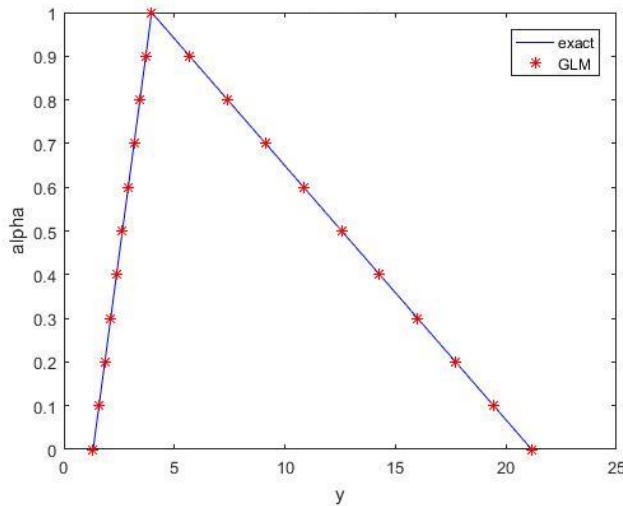
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	0.6795704571147614	10.87312731383618	0.6795704225205815	10.87312676032930
0.1	0.8154845485377137	9.989685719586992	0.8154845070246972	9.989685211052548
0.2	0.9513986399606659	9.106244125337803	0.9513985915288141	9.106243661775789
0.3	1.087312731383618	8.222802531088613	1.087312676032931	8.222802112499039
0.4	1.223226822806571	7.339360936839423	1.223226760537047	7.339360563222281
0.5	1.359140914229523	6.455919342590233	1.359140845041163	6.455919013945520
0.6	1.495055005652475	5.572477748341043	1.495054929545279	5.572477464668773
0.7	1.630969097075428	4.689036154091854	1.630969014049394	4.689035915392013
0.8	1.766883188498380	3.805594559842664	1.766883098553512	3.805594366115257
0.9	1.902797279921332	2.922152965593473	1.902797183057628	2.922152816838501
1	2.038711371344284	2.038711371344284	2.038711267561744	2.038711267561744



**Figure 3.4.7:** Exact and FGLM solutions for  $t = 1$

**Table 3.4.14:** Numerical values for the exact and approximate solutions (FGLMs) for  $t = 1.5$

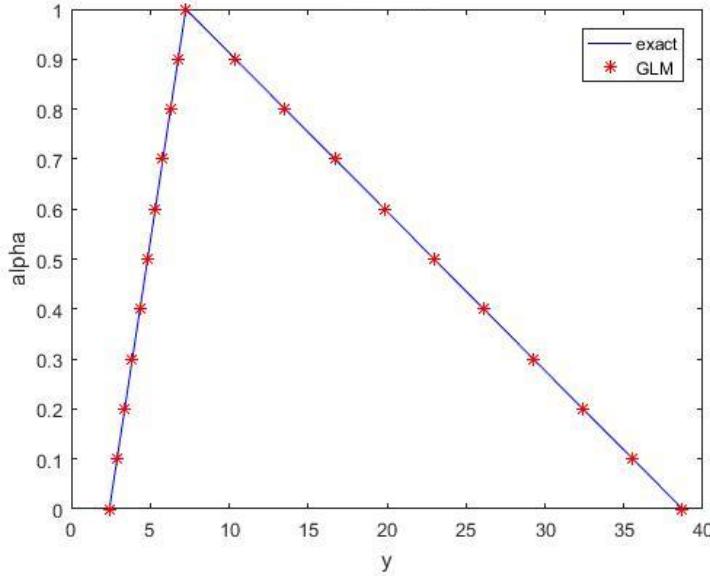
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	1.322555431409265	21.16088690254824	1.322555283852501	21.16088454164001
0.1	1.587066517691118	19.44156484171619	1.587066340623000	19.44156267263176
0.2	1.851577603972971	17.72224278088415	1.851577397393502	17.72224080362351
0.3	2.116088690254824	16.00292072005211	2.116088454164003	16.00291893461526
0.4	2.380599776536677	14.28359865922006	2.380599510934503	14.28359706560701
0.5	2.645110862818530	12.56427659838802	2.645110567705002	12.56427519659875
0.6	2.909621949100383	10.84495453755597	2.909621624475501	10.84495332759052
0.7	3.174133035382237	9.125632476723929	3.174132681245999	9.125631458582259
0.8	3.438644121664089	7.406310415891883	3.438643738016503	7.406309589574008
0.9	3.703155207945942	5.686988355059838	3.703154794787003	5.686987720565753
1	3.967666294227795	3.967666294227795	3.967665851557501	3.967665851557501



**Figure 3.4.8:** Exact and FGLM solutions for  $t = 1.5$

**Table 3.4.15:** Numerical values for the exact and approximate solutions (FGLMs) for  $t = 2$

$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.419243918089446	38.70790268943114	2.419243637353441	38.70789819765503
0.1	2.903092701707335	35.56288559591486	2.903092364824126	35.56288146909557
0.2	3.386941485325224	32.41786850239858	3.386941092294817	32.41786474053610
0.3	3.870790268943114	29.27285140888230	3.870789819765507	29.27284801197664
0.4	4.354639052561003	26.12783431536602	4.354638547236195	26.12783128341714
0.5	4.838487836178892	22.98281722184974	4.838487274706883	22.98281455485767
0.6	5.322336619796782	19.83780012833346	5.322336002177562	19.83779782629823
0.7	5.806185403414671	16.69278303481718	5.806184729648250	16.69278109773874
0.8	6.290034187032560	13.54776594130090	6.290033457118945	13.54776436917927
0.9	6.773882970650448	10.40274884778462	6.773882184589633	10.40274764061979
1	7.257731754268338	7.257731754268338	7.257730912060317	7.257730912060317



**Figure 3.4.9:** Exact and FGLM solutions for  $t = 2$

**Table 3.4.16:** The absolute errors of the FGLM for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$3.4594 \times 10^{-8}$	$5.5351 \times 10^{-7}$
0.1	$4.1513 \times 10^{-8}$	$5.0853 \times 10^{-7}$
0.2	$4.8432 \times 10^{-8}$	$4.6356 \times 10^{-7}$
0.3	$5.5351 \times 10^{-8}$	$4.1859 \times 10^{-7}$
0.4	$6.2270 \times 10^{-8}$	$3.7362 \times 10^{-7}$
0.5	$6.9188 \times 10^{-8}$	$3.2864 \times 10^{-7}$
0.6	$7.6107 \times 10^{-8}$	$2.8367 \times 10^{-7}$
0.7	$8.3026 \times 10^{-8}$	$2.3870 \times 10^{-7}$
0.8	$8.9945 \times 10^{-8}$	$1.9373 \times 10^{-7}$
0.9	$9.6864 \times 10^{-8}$	$1.4875 \times 10^{-7}$
1	$1.0378 \times 10^{-7}$	$1.0378 \times 10^{-7}$

**Table 3.4.17:** The absolute errors of the FGML for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.4756 \times 10^{-7}$	$2.3609 \times 10^{-6}$
0.1	$1.7707 \times 10^{-7}$	$2.1691 \times 10^{-6}$
0.2	$2.0658 \times 10^{-7}$	$1.9773 \times 10^{-6}$
0.3	$2.3609 \times 10^{-7}$	$1.7854 \times 10^{-6}$
0.4	$2.6560 \times 10^{-7}$	$1.5936 \times 10^{-6}$
0.5	$2.9511 \times 10^{-7}$	$1.4018 \times 10^{-6}$
0.6	$3.2462 \times 10^{-7}$	$1.2100 \times 10^{-6}$
0.7	$3.5414 \times 10^{-7}$	$1.0181 \times 10^{-6}$
0.8	$3.8365 \times 10^{-7}$	$8.2632 \times 10^{-7}$
0.9	$4.1316 \times 10^{-7}$	$6.3449 \times 10^{-7}$
1	$4.4267 \times 10^{-7}$	$4.4267 \times 10^{-7}$

**Table 3.4.18:** The absolute errors of the FGML for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$2.8074 \times 10^{-7}$	$4.4918 \times 10^{-6}$
0.1	$3.3688 \times 10^{-7}$	$4.1268 \times 10^{-6}$
0.2	$3.9303 \times 10^{-7}$	$3.7619 \times 10^{-6}$
0.3	$4.4918 \times 10^{-7}$	$3.3969 \times 10^{-6}$
0.4	$5.0532 \times 10^{-7}$	$3.0319 \times 10^{-6}$
0.5	$5.6147 \times 10^{-7}$	$2.6670 \times 10^{-6}$
0.6	$6.1762 \times 10^{-7}$	$2.3020 \times 10^{-6}$
0.7	$6.7377 \times 10^{-7}$	$1.9371 \times 10^{-6}$
0.8	$7.2991 \times 10^{-7}$	$1.5721 \times 10^{-6}$
0.9	$7.8606 \times 10^{-7}$	$1.2072 \times 10^{-6}$
1	$8.4221 \times 10^{-7}$	$8.4221 \times 10^{-7}$

As shown in Tables 3.4.13-18, the fuzzy general linear method with another triangular fuzzy number as initial condition gave less accurate than Runge-Kutta but it needed less number of steps.

For  $k = 5$

a-  $y(0) = (0.75, 1, 1.125)$

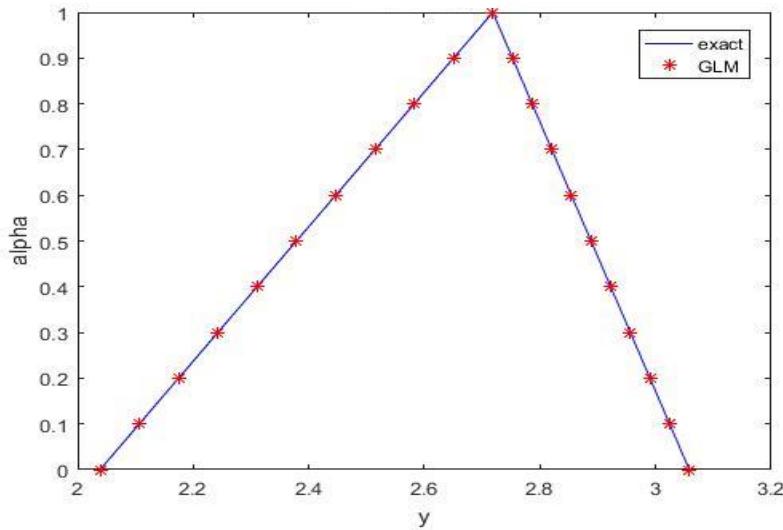
Let  $y(0, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha]$

Case 1: using 1 – differentiable

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 3.4.19-21 and Figs 3.4.10-12 respectively, and the absolute errors of the approximate results in Tables 3.4.22-24.

**Table 3.4.19:** Numerical values for the exact ( $1 - \text{differentiable}$ ) and approximate solutions (FGLMs) for  $t = 1$

$\alpha$	Exact		FGLMs	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	2.038711371344284	3.058067057016426	2.038711369407443	3.058067054111163
0.1	2.106668417055760	3.024088534160689	2.106668415054357	3.024088531287706
0.2	2.174625462767236	2.990110011304950	2.174625460701272	2.990110008464247
0.3	2.242582508478713	2.956131488449212	2.242582506348186	2.956131485640793
0.4	2.310539554190189	2.922152965593474	2.310539551995102	2.922152962817334
0.5	2.378496599901665	2.888174442737736	2.378496597642015	2.888174439993877
0.6	2.446453645613141	2.854195919881998	2.446453643288931	2.854195917170421
0.7	2.514410691324617	2.820217397026260	2.514410688935846	2.820217394346960
0.8	2.582367737036093	2.786238874170521	2.582367734582760	2.786238871523505
0.9	2.650324782747569	2.752260351314784	2.650324780229675	2.752260348700048
1	2.718281828459046	2.718281828459046	2.718281825876591	2.718281825876591



**Figure 3.4.10:** Exact and FGLM solutions for  $t = 1$

**Table 3.4.20:** Numerical values for the exact ( $1 - \text{differentiable}$ ) and approximate solutions (FGLMs) for  $t = 1.5$

$\alpha$	Exact		FGLMs	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	3.967666294227795	5.951499441341692	3.967666285987665	5.951499428981495
0.1	4.099921837368721	5.885371669771230	4.099921828853920	5.885371657548369
0.2	4.232177380509648	5.819243898200766	4.232177371720176	5.819243886115239
0.3	4.364432923650575	5.753116126630302	4.364432914586430	5.753116114682116
0.4	4.496688466791501	5.686988355059839	4.496688457452688	5.686988343248985

0.5	4.628944009932427	5.620860583489376	4.628944000318938	5.620860571815860
0.6	4.761199553073354	5.554732811918913	4.761199543185199	5.554732800382736
0.7	4.893455096214280	5.488605040348450	4.893455086051453	5.488605028949600
0.8	5.025710639355206	5.422477268777985	5.025710628917707	5.422477257516476
0.9	5.157966182496133	5.356349497207523	5.157966171783964	5.356349486083349
1	5.290221725637060	5.290221725637060	5.290221714650222	5.290221714650222

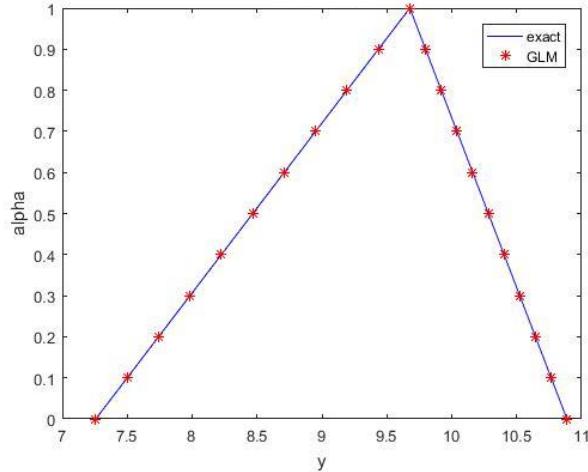


Figure 3.4.11: Exact and FGLM solutions for  $t = 1.5$

Table 3.4.21: Numerical values for the exact (1 – differentiable )and approximate solutions (FGLMs) for  $t = 2$

$\alpha$	Exact		FGLMs	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	7.257731754268338	10.88659763140251	7.257731738592279	10.88659760788841
0.1	7.499656146077282	10.76563543549804	7.499656129878690	10.76563541224522
0.2	7.741580537886227	10.64467323959356	7.741580521165099	10.64467321660200
0.3	7.983504929695171	10.52371104368909	7.983504912451502	10.52371102095881
0.4	8.225429321504116	10.40274884778462	8.225429303737919	10.40274882531560
0.5	8.467353713313061	10.28178665188015	8.467353695024317	10.28178662967240
0.6	8.709278105122007	10.16082445597567	8.709278086310736	10.16082443402920
0.7	8.951202496930950	10.03986226007120	8.951202477597144	10.03986223838597
0.8	9.193126888739894	9.918900064166728	9.193126868883550	9.918900042742784
0.9	9.435051280548839	9.797937868262256	9.435051260169963	9.797937847099576
1	9.676975672357784	9.676975672357784	9.676975651456381	9.676975651456381

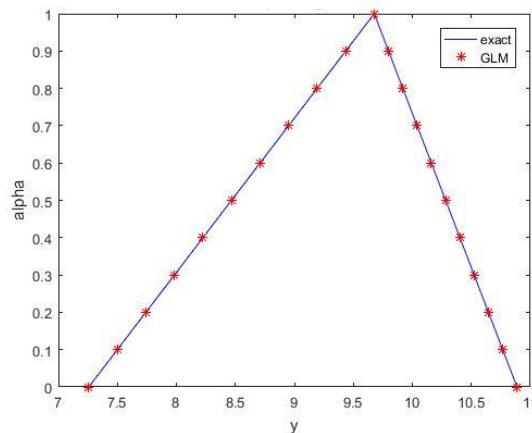


Figure 3.4.12: Exact and FGLM solutions for  $t = 2$

**Table 3.4.22:** The absolute errors of the FGLM (1 – differentiable) for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.9368 \times 10^{-9}$	$2.9053 \times 10^{-9}$
0.1	$2.0014 \times 10^{-9}$	$2.8730 \times 10^{-9}$
0.2	$2.0660 \times 10^{-9}$	$2.8407 \times 10^{-9}$
0.3	$2.1305 \times 10^{-9}$	$2.8084 \times 10^{-9}$
0.4	$2.1951 \times 10^{-9}$	$2.7761 \times 10^{-9}$
0.5	$2.2596 \times 10^{-9}$	$2.7439 \times 10^{-9}$
0.6	$2.3242 \times 10^{-9}$	$2.7116 \times 10^{-9}$
0.7	$2.3888 \times 10^{-9}$	$2.6793 \times 10^{-9}$
0.8	$2.4533 \times 10^{-9}$	$2.6470 \times 10^{-9}$
0.9	$2.5179 \times 10^{-9}$	$2.6147 \times 10^{-9}$
1	$2.5825 \times 10^{-9}$	$2.5825 \times 10^{-9}$

**Table 3.4.23:** The absolute errors of the FGLM (1 – differentiable) for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$8.2401 \times 10^{-9}$	$1.2360 \times 10^{-8}$
0.1	$8.5148 \times 10^{-9}$	$1.2223 \times 10^{-8}$
0.2	$8.7895 \times 10^{-9}$	$1.2086 \times 10^{-8}$
0.3	$9.0641 \times 10^{-9}$	$1.1948 \times 10^{-8}$
0.4	$9.3388 \times 10^{-9}$	$1.1811 \times 10^{-8}$
0.5	$9.6135 \times 10^{-9}$	$1.1674 \times 10^{-8}$
0.6	$9.8882 \times 10^{-9}$	$1.1536 \times 10^{-8}$
0.7	$1.0163 \times 10^{-8}$	$1.1399 \times 10^{-8}$
0.8	$1.0437 \times 10^{-8}$	$1.1262 \times 10^{-8}$
0.9	$1.0712 \times 10^{-8}$	$1.1124 \times 10^{-8}$
1	$1.0987 \times 10^{-8}$	$1.0987 \times 10^{-8}$

**Table 3.4.24:** The absolute errors of the FGLM (1 – differentiable) for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.5676 \times 10^{-8}$	$2.3514 \times 10^{-8}$
0.1	$1.6199 \times 10^{-8}$	$2.3253 \times 10^{-8}$
0.2	$1.6721 \times 10^{-8}$	$2.2992 \times 10^{-8}$
0.3	$1.7244 \times 10^{-8}$	$2.2730 \times 10^{-8}$
0.4	$1.7766 \times 10^{-8}$	$2.2469 \times 10^{-8}$
0.5	$1.8289 \times 10^{-8}$	$2.2208 \times 10^{-8}$
0.6	$1.8811 \times 10^{-8}$	$2.1946 \times 10^{-8}$
0.7	$1.9334 \times 10^{-8}$	$2.1685 \times 10^{-8}$
0.8	$1.9856 \times 10^{-8}$	$2.1424 \times 10^{-8}$
0.9	$2.0379 \times 10^{-8}$	$2.1163 \times 10^{-8}$
1	$2.0901 \times 10^{-8}$	$2.0901 \times 10^{-8}$

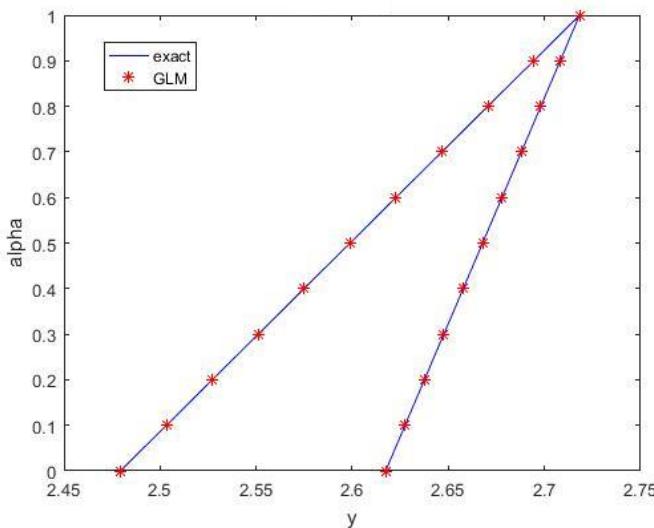
As shown in Tables 3.4.19-24, the fuzzy general linear method with triangular fuzzy number as initial condition gave less accurate than Runge-Kutta but it needed less number of steps.

Case 2: using 2 –differentiable

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 3.4.25-27 and Figs 3.4.13-15 respectively, and the absolute errors of the approximate results in Tables 3.4.28-30.

**Table 3.4.25:** Numerical values for the exact (2 – differentiable )and approximate solutions (FGLMs) for  $t = 1$

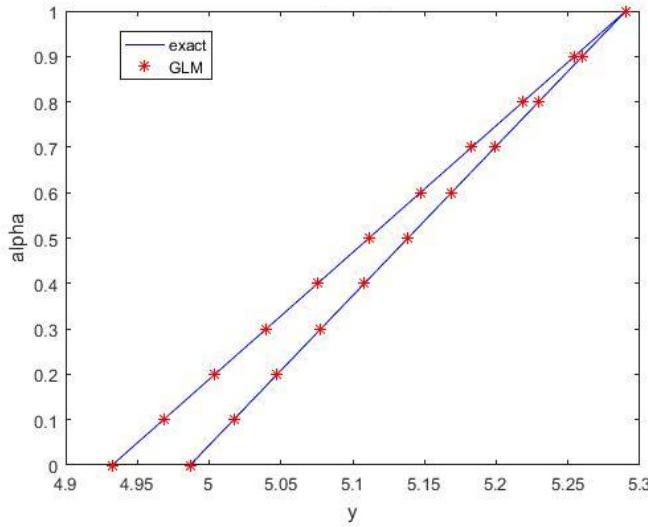
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.479411818960710	2.617366609400000	2.479411816608218	2.617366606910388
0.1	2.503298819910544	2.627458131305905	2.503298817535054	2.627458128807009
0.2	2.527185820860377	2.637549653211810	2.527185818461892	2.637549650703629
0.3	2.551072821810211	2.647641175117714	2.551072819388729	2.647641172600248
0.4	2.574959822760044	2.657732697023619	2.574959820315566	2.657732694496869
0.5	2.598846823709878	2.667824218929523	2.598846821242404	2.667824216393489
0.6	2.622733824659711	2.677915740835428	2.622733822169241	2.677915738290109
0.7	2.646620825609545	2.688007262741332	2.646620823096077	2.688007260186729
0.8	2.670507826559379	2.698098784647236	2.670507824022915	2.698098782083350
0.9	2.694394827509212	2.708190306553141	2.694394824949753	2.708190303979970
1	2.718281828459046	2.718281828459046	2.718281825876591	2.718281825876590



**Figure 3.4.13:** Exact and FGLM solutions at  $t = 1$

**Table 3.4.26:** Numerical values for the exact (2 – differentiable) and approximate solutions (FGLMs) for  $t = 1.5$

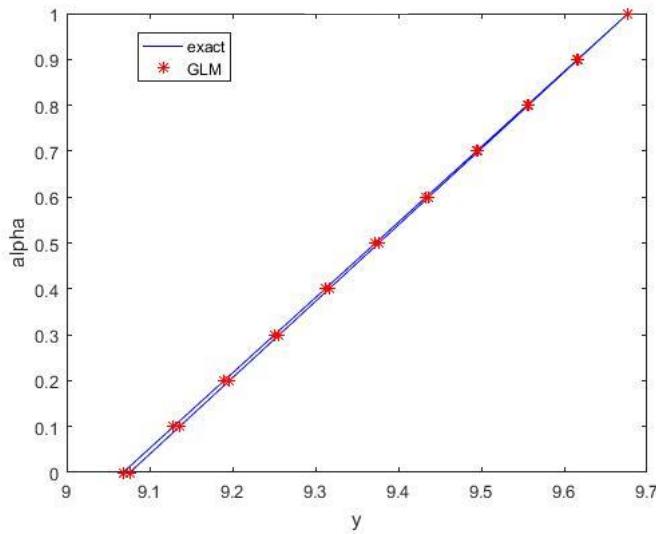
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	4.932442377592928	4.986723357976558	4.932442367301161	4.986723347668000
0.1	4.968220312397341	5.017073194742609	4.968220302036066	5.017073184366222
0.2	5.003998247201754	5.047423031508658	5.003998236770974	5.047423021064444
0.3	5.039776182006168	5.077772868274710	5.039776171505876	5.077772857762664
0.4	5.075554116810579	5.108122705040760	5.075554106240784	5.108122694460887
0.5	5.111332051614995	5.138472541806809	5.111332040975690	5.138472531159110
0.6	5.147109986419408	5.168822378572860	5.147109975710596	5.168822367857331
0.7	5.182887921223820	5.199172215338908	5.182887910445476	5.199172204555550
0.8	5.218665856028234	5.229522052104959	5.218665845180407	5.229522041253775
0.9	5.254443790832648	5.259871888871011	5.254443779915314	5.259871877951998
1	5.290221725637062	5.290221725637062	5.290221714650222	5.290221714650220



**Figure 3.4.14:** Exact and FGLM solutions for  $t = 1.5$

**Table 3.4.27:** Numerical values for the exact (2 – differentiable) and approximate solutions (FGLMs) for  $t = 2$

$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	9.068147228770254	9.076182156900588	9.068147209230954	9.076182137249736
0.1	9.129030073129007	9.136261508446307	9.129030053453498	9.136261488670399
0.2	9.189912917487760	9.196340859992027	9.189912897676042	9.196340840091065
0.3	9.250795761846515	9.256420211537749	9.250795741898577	9.256420191511722
0.4	9.311678606205266	9.316499563083468	9.311678586121120	9.316499542932389
0.5	9.372561450564021	9.376578914629187	9.372561430343666	9.376578894353054
0.6	9.433444294922772	9.436658266174906	9.433444274566204	9.436658245773716
0.7	9.494327139281523	9.496737617720623	9.494327118788714	9.496737597194363
0.8	9.555209983640278	9.556816969266345	9.555209963011288	9.556816948615042
0.9	9.616092827999033	9.616896320812066	9.616092807233830	9.616896300035707
1	9.676975672357788	9.676975672357788	9.676975651456381	9.676975651456374



**Figure 3.4.15:** Exact and FGLM solutions for  $t = 2$

**Table 3.4.28:** The absolute errors of the FGLM (2 – differentiable) for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$2.3525 \times 10^{-9}$	$2.4896 \times 10^{-9}$
0.1	$2.3755 \times 10^{-9}$	$2.4989 \times 10^{-9}$
0.2	$2.3985 \times 10^{-9}$	$2.5082 \times 10^{-9}$
0.3	$2.4215 \times 10^{-9}$	$2.5175 \times 10^{-9}$
0.4	$2.4445 \times 10^{-9}$	$2.5267 \times 10^{-9}$
0.5	$2.4675 \times 10^{-9}$	$2.5360 \times 10^{-9}$
0.6	$2.4905 \times 10^{-9}$	$2.5453 \times 10^{-9}$
0.7	$2.5135 \times 10^{-9}$	$2.5546 \times 10^{-9}$
0.8	$2.5365 \times 10^{-9}$	$2.5639 \times 10^{-9}$
0.9	$2.5595 \times 10^{-9}$	$2.5732 \times 10^{-9}$
1	$2.5825 \times 10^{-9}$	$2.5825 \times 10^{-9}$

**Table 3.4.29:** The absolute errors of the FGLM (2 – differentiable) for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.0292 \times 10^{-8}$	$1.0309 \times 10^{-8}$
0.1	$1.0361 \times 10^{-8}$	$1.0376 \times 10^{-8}$
0.2	$1.0431 \times 10^{-8}$	$1.0444 \times 10^{-8}$
0.3	$1.0500 \times 10^{-8}$	$1.0512 \times 10^{-8}$
0.4	$1.0570 \times 10^{-8}$	$1.0580 \times 10^{-8}$
0.5	$1.0639 \times 10^{-8}$	$1.0648 \times 10^{-8}$
0.6	$1.0709 \times 10^{-8}$	$1.0716 \times 10^{-8}$
0.7	$1.0778 \times 10^{-8}$	$1.0783 \times 10^{-8}$
0.8	$1.0848 \times 10^{-8}$	$1.0851 \times 10^{-8}$
0.9	$1.0917 \times 10^{-8}$	$1.0919 \times 10^{-8}$
1	$1.0987 \times 10^{-8}$	$1.0987 \times 10^{-8}$

**Table 3.4.30:** The absolute errors of the FGLM (2 – differentiable) for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.9539 \times 10^{-8}$	$1.9651 \times 10^{-8}$
0.1	$1.9676 \times 10^{-8}$	$1.9776 \times 10^{-8}$
0.2	$1.9812 \times 10^{-8}$	$1.9901 \times 10^{-8}$
0.3	$1.9948 \times 10^{-8}$	$2.0026 \times 10^{-8}$
0.4	$2.0084 \times 10^{-8}$	$2.0151 \times 10^{-8}$
0.5	$2.0220 \times 10^{-8}$	$2.0276 \times 10^{-8}$
0.6	$2.0357 \times 10^{-8}$	$2.0401 \times 10^{-8}$
0.7	$2.0493 \times 10^{-8}$	$2.0526 \times 10^{-8}$
0.8	$2.0629 \times 10^{-8}$	$2.0651 \times 10^{-8}$
0.9	$2.0765 \times 10^{-8}$	$2.0776 \times 10^{-8}$
1	$2.0901 \times 10^{-8}$	$2.0901 \times 10^{-8}$

As shown in Tables 3.4.25-30, the fuzzy general linear method with triangular fuzzy number as initial condition (under 2 – differentiable) gave less accurate than Runge-Kutta but it needed less number of steps.

b- Let  $y(0) = (0.25, 0.75, 4)$

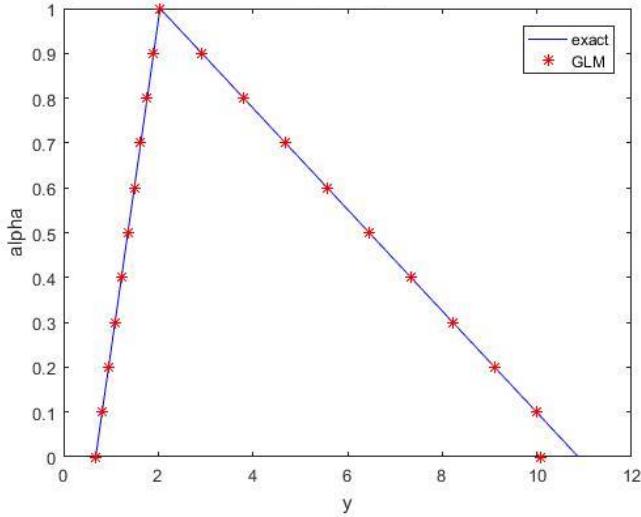
$$y(0, \alpha) = [(0.25 + 0.5\alpha), (4 - 3.25\alpha)], \quad 0 \leq \alpha \leq 1$$

using 1 –differentiable

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 3.4.31-33 and Figs 3.4.16-18 respectively, and the absolute errors of the approximate results in Tables 3.4.34-36.

**Table 3.4.31:** Numerical values for the exact and approximate solutions (FGLMs) for  $t = 1$

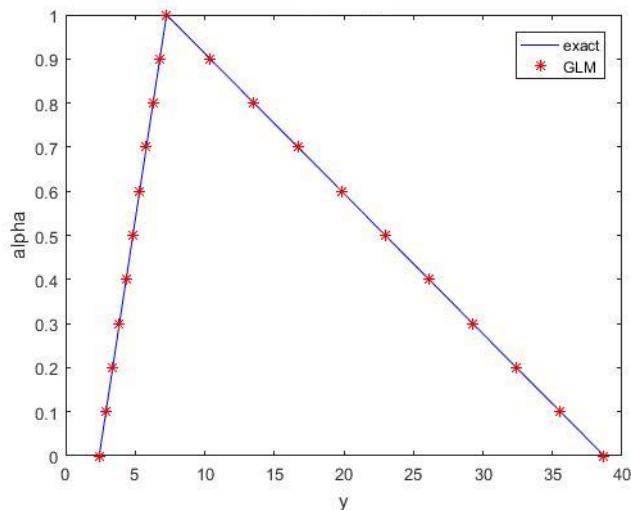
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	0.6795704571147614	10.87312731383618	0.6795704564691478	10.87312730350637
0.1	0.8154845485377137	9.989685719586992	0.8154845477629767	9.989685710096467
0.2	0.9513986399606659	9.106244125337803	0.9513986390568062	9.106244116686574
0.3	1.087312731383618	8.222802531088613	1.087312730350636	8.222802523276682
0.4	1.223226822806571	7.339360936839423	1.223226821644466	7.339360929866792
0.5	1.359140914229523	6.455919342590233	1.359140912938296	6.455919336456899
0.6	1.495055005652475	5.572477748341043	1.495055004232123	5.572477743047009
0.7	1.630969097075428	4.689036154091854	1.630969095525954	4.689036149637117
0.8	1.766883188498380	3.805594559842664	1.766883186819783	3.805594556227225
0.9	1.902797279921332	2.922152965593473	1.902797278113612	2.922152962817334
1	2.038711371344284	2.038711371344284	2.038711369407443	2.038711369407443



**Figure 3.4.16:** Exact and FGML solutions for  $t = 1$

**Table 3.4.32:** Numerical values for the exact and approximate solutions (FGMLs) for  $t = 1.5$

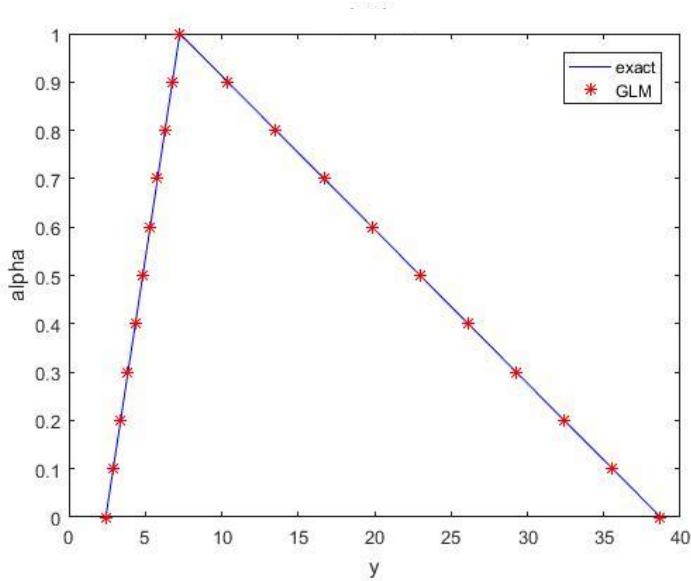
$\alpha$	Exact		FGML	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	1.322555431409265	21.16088690254824	1.322555428662556	21.16088685860090
0.1	1.587066517691118	19.44156484171619	1.587066514395066	19.44156480133955
0.2	1.851577603972971	17.72224278088415	1.851577600127576	17.72224274407822
0.3	2.116088690254824	16.00292072005211	2.116088685860088	16.00292068681691
0.4	2.380599776536677	14.28359865922006	2.380599771592600	14.28359862955559
0.5	2.645110862818530	12.56427659838802	2.645110857325112	12.56427657229427
0.6	2.909621949100383	10.84495453755597	2.909621943057618	10.84495451503295
0.7	3.174133035382237	9.125632476723929	3.174133028790132	9.125632457771628
0.8	3.438644121664089	7.406310415891883	3.438644114522641	7.406310400510304
0.9	3.703155207945942	5.686988355059838	3.703155200255151	5.686988343248985
1	3.967666294227795	3.967666294227795	3.967666285987665	3.967666285987665



**Figure 3.4.17:** Exact and FGML solutions for  $t = 1.5$

**Table 3.4.33:** Numerical values for the exact and approximate solutions (FGLMs) for  $t = 2$

$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.419243918089446	38.70790268943114	2.419243912864096	38.70790260582554
0.1	2.903092701707335	35.56288559591486	2.903092695436911	35.56288551910215
0.2	3.386941485325224	32.41786850239858	3.386941478009728	32.41786843237880
0.3	3.870790268943114	29.27285140888230	3.870790260582549	29.27285134565551
0.4	4.354639052561003	26.12783431536602	4.354639043155369	26.12783425893220
0.5	4.838487836178892	22.98281722184974	4.838487825728191	22.98281717220888
0.6	5.322336619796782	19.83780012833346	5.322336608300999	19.83780008548556
0.7	5.806185403414671	16.69278303481718	5.806185390873822	16.69278299876224
0.8	6.290034187032560	13.54776594130090	6.290034173446638	13.54776591203891
0.9	6.773882970650448	10.40274884778462	6.773882956019454	10.40274882531560
1	7.257731754268338	7.257731754268338	7.257731738592279	7.257731738592279



**Figure 3.4.18:** Exact and FGLM solutions for  $t = 2$

**Table 3.4.34:** The absolute errors of the FGLM for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$6.4561 \times 10^{-10}$	$1.0330 \times 10^{-8}$
0.1	$7.7474 \times 10^{-10}$	$9.4905 \times 10^{-9}$
0.2	$9.0386 \times 10^{-10}$	$8.6512 \times 10^{-9}$
0.3	$1.0330 \times 10^{-9}$	$7.8119 \times 10^{-9}$
0.4	$1.1621 \times 10^{-9}$	$6.9726 \times 10^{-9}$
0.5	$1.2912 \times 10^{-9}$	$6.1333 \times 10^{-9}$
0.6	$1.4204 \times 10^{-9}$	$5.2940 \times 10^{-9}$
0.7	$1.5495 \times 10^{-9}$	$4.4547 \times 10^{-9}$
0.8	$1.6786 \times 10^{-9}$	$3.6154 \times 10^{-9}$
0.9	$1.8077 \times 10^{-9}$	$2.7761 \times 10^{-9}$
1	$1.9368 \times 10^{-9}$	$1.9368 \times 10^{-9}$

**Table 3.4.35:** The absolute errors of the FGML for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$2.7467 \times 10^{-9}$	$4.3947 \times 10^{-8}$
0.1	$3.2961 \times 10^{-9}$	$4.0377 \times 10^{-8}$
0.2	$3.8454 \times 10^{-9}$	$3.6806 \times 10^{-8}$
0.3	$4.3947 \times 10^{-9}$	$3.3235 \times 10^{-8}$
0.4	$4.9441 \times 10^{-9}$	$2.9664 \times 10^{-8}$
0.5	$5.4934 \times 10^{-9}$	$2.6094 \times 10^{-8}$
0.6	$6.0428 \times 10^{-9}$	$2.2523 \times 10^{-8}$
0.7	$6.5921 \times 10^{-9}$	$1.8952 \times 10^{-8}$
0.8	$7.1414 \times 10^{-9}$	$1.5382 \times 10^{-8}$
0.9	$7.6908 \times 10^{-9}$	$1.1811 \times 10^{-8}$
1	$8.2401 \times 10^{-9}$	$8.2401 \times 10^{-9}$

**Table 3.4.36 :** The absolute errors of the FGML for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$5.2253 \times 10^{-9}$	$8.3606 \times 10^{-8}$
0.1	$6.2704 \times 10^{-9}$	$7.6813 \times 10^{-8}$
0.2	$7.3155 \times 10^{-9}$	$7.0020 \times 10^{-8}$
0.3	$8.3606 \times 10^{-9}$	$6.3227 \times 10^{-8}$
0.4	$9.4056 \times 10^{-9}$	$5.6434 \times 10^{-8}$
0.5	$1.0451 \times 10^{-8}$	$4.9641 \times 10^{-8}$
0.6	$1.1496 \times 10^{-8}$	$4.2848 \times 10^{-8}$
0.7	$1.2541 \times 10^{-8}$	$3.6055 \times 10^{-8}$
0.8	$1.3586 \times 10^{-8}$	$2.9262 \times 10^{-8}$
0.9	$1.4631 \times 10^{-8}$	$2.2469 \times 10^{-8}$
1	$1.5676 \times 10^{-8}$	$1.5676 \times 10^{-8}$

As shown in Tables 3.4.31-36, the fuzzy general linear method with another triangular fuzzy number as initial condition gave less accurate than Runge-Kutta but it needed less number of steps.

### 3.5 Variational Iteration Method (VIM)

In this section, we solve hybrid fuzzy differential equations (3) by Variational Iteration Method. For linear problems, its exact solution can be obtained by only one iteration step due to the fact the Lagrange multiplier can be exactly identified.

**Theorem 7** Consider the fuzzy initial-value problem

$$\begin{cases} \underline{y}'(t) = (a(t) \odot \underline{y}(t)) \oplus b(t) \\ \underline{y}(t_0) = y_0 \end{cases} \quad (39)$$

By 1 – differentiable, replace equation (39) by the equivalent system:

$$\begin{cases} \underline{y}'(t, \alpha) = a(t)\underline{y}(t, \alpha) + \underline{b}(t, \alpha), & \underline{y}(t_0, \alpha) = \underline{y}_0(\alpha) \\ \bar{y}'(t, \alpha) = a(t)\bar{y}(t, \alpha) + \bar{b}(t, \alpha), & \bar{y}(t_0, \alpha) = \bar{y}_0(\alpha) \end{cases} \quad (40)$$

By 2 – differentiable, replace equation (39) by the equivalent system:

$$\begin{cases} \underline{y}'(t, \alpha) = a(t)\bar{y}(t, \alpha) + \bar{b}(t, \alpha), & \underline{y}(t_0, \alpha) = \underline{y}_0(\alpha) \\ \bar{y}'(t, \alpha) = a(t)\underline{y}(t, \alpha) + \underline{b}(t, \alpha), & \bar{y}(t_0, \alpha) = \bar{y}_0(\alpha) \end{cases} \quad (41)$$

where  $a(t) > 0$

For every prefixed  $\alpha \in [0,1]$ , the above systems represent ordinary initial value problems for which any converging classical numerical procedure can be applied to solve this system.

For solving equations. (40) by VIM, construct the following correction function

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \underline{y}_n(t, \alpha) + \int_{t_0}^t \lambda_1(\tau) (\underline{y}'_n(\tau, \alpha) - a(\tau)\underline{y}_n(\tau, \alpha) - \underline{b}(\tau, \alpha)) d\tau \\ \bar{y}_{n+1}(t, \alpha) = \bar{y}_n(t, \alpha) + \int_{t_0}^t \lambda_2(\tau) (\bar{y}'_n(\tau, \alpha) - a(\tau)\bar{y}_n(\tau, \alpha) - \bar{b}(\tau, \alpha)) d\tau \end{cases} \quad (42)$$

Calculating variation with respect to  $\underline{y}_n$  and  $\bar{y}_n$ , noticing that  $\delta \underline{y}_n = 0$  and  $\delta \bar{y}_n = 0$  yields

$$\begin{aligned} \delta \underline{y}_{n+1}(t, \alpha) &= \delta \underline{y}_n(t, \alpha) + \delta \int_{t_0}^t \lambda_1(\tau) (\underline{y}'_n(\tau, \alpha) - a(\tau)\underline{y}_n(\tau, \alpha) - \underline{b}(\tau, \alpha)) d\tau \\ &= (1 + \lambda_1(t)) \delta \underline{y}_n(t, \alpha) - \int_{t_0}^t (\lambda'_1(\tau) + a(\tau)\lambda_1(\tau)) \delta \underline{y}_n(\tau, \alpha) d\tau \\ &= 0 \end{aligned}$$

And

$$\begin{aligned}\delta\bar{y}_{n+1}(t, \alpha) &= \delta\bar{y}_n(t, \alpha) + \delta \int_{t_0}^t \lambda_2(\tau) \left( \bar{y}'_n(\tau, \alpha) - a(\tau)\bar{y}_n(\tau, \alpha) - \bar{b}(\tau, \alpha) \right) d\tau \\ &= (1 + \lambda_2(\tau))\delta\bar{y}_n(\tau, \alpha)|_{\tau=t} - \int_{t_0}^t (\lambda'_2(\tau) + a(\tau)\lambda_2(\tau))\delta\bar{y}_n(\tau, \alpha) d\tau \\ &= 0\end{aligned}$$

Therefore, the following stationary conditions exists:

$$\lambda'_i(\tau) + a(\tau)\lambda_i(\tau) = 0, \quad 1 + \lambda_i(\tau)|_{\tau=t} = 0, \quad i = 1, 2$$

So, the Lagrange multipliers can be readily identified

$$\lambda_1(\tau) = \lambda_2(\tau) = -e^{\int_{\tau}^t a(s) ds}$$

Substituting these values of the Lagrange multipliers into function (42) gives the iteration formulas:

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \underline{y}_n(t, \alpha) - \int_{t_0}^t e^{\int_{\tau}^t a(s) ds} \left( \underline{y}'_n(\tau, \alpha) - a(\tau)\underline{y}_n(\tau, \alpha) - \underline{b}(\tau, \alpha) \right) d\tau \\ \bar{y}_{n+1}(t, \alpha) = \bar{y}_n(t, \alpha) - \int_{t_0}^t e^{\int_{\tau}^t a(s) ds} \left( \bar{y}'_n(\tau, \alpha) - a(\tau)\bar{y}_n(\tau, \alpha) - \bar{b}(\tau, \alpha) \right) d\tau \end{cases} \quad (43)$$

the values of  $\underline{y}_0(t, \alpha)$  and  $\bar{y}_0(t, \alpha)$  are initial approximations and chosen as follows:

$$\underline{y}_0(t, \alpha) = \underline{y}(t_0, \alpha) = \underline{y}_0(\alpha), \quad \bar{y}_0(t, \alpha) = \bar{y}(t_0, \alpha) = \bar{y}_0(\alpha)$$

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{y}_n(t, \alpha), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{y}_n(t, \alpha)$$

where  $y(t) = (\underline{y}(t, \alpha), \bar{y}(t, \alpha))$  is exact solution.

For solving equations (41) by VIM, construct the following correction functional

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \underline{y}_n(t, \alpha) + \int_{t_0}^t \lambda_1(\tau) \left( \underline{y}'_n(\tau, \alpha) - a(\tau)\underline{y}_n(\tau, \alpha) - \underline{b}(\tau, \alpha) \right) d\tau \\ \bar{y}_{n+1}(t, \alpha) = \bar{y}_n(t, \alpha) + \int_{t_0}^t \lambda_2(\tau) \left( \bar{y}'_n(\tau, \alpha) - a(\tau)\bar{y}_n(\tau, \alpha) - \bar{b}(\tau, \alpha) \right) d\tau \end{cases} \quad (44)$$

Calculating variation with respect to  $\underline{y}_n$  and  $\bar{y}_n$ , noticing that  $\delta\underline{y}_n = 0$  and  $\delta\bar{y}_n = 0$  which yields.

So, the Lagrange multipliers can be readily identified

$$\lambda_1(\tau) = \lambda_2(\tau) = -1.$$

Substituting these values of the Lagrange multipliers into functional (44) gives the iteration formulas:

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \underline{y}_n(t, \alpha) - \int_{t_0}^t (\underline{y}'_n(\tau, \alpha) - a(\tau)\bar{\underline{y}}_n(\tau, \alpha) - \bar{b}(\tau, \alpha)) d\tau \\ \bar{y}_{n+1}(t, \alpha) = \bar{y}_n(t, \alpha) - \int_{t_0}^t (\bar{y}'_n(\tau, \alpha) - a(\tau)\underline{y}_n(\tau, \alpha) - \underline{b}(\tau, \alpha)) d\tau \end{cases} \quad (45)$$

the values of  $\underline{y}_0(t, \alpha)$  and  $\bar{y}_0(t, \alpha)$  are initial approximations chosen as follows:

$$\underline{y}_0(t, \alpha) = \underline{y}(t_0, \alpha) = \underline{y}_0(\alpha), \quad \bar{y}_0(t, \alpha) = \bar{y}(t_0, \alpha) = \bar{y}_0(\alpha)$$

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{y}_n(t, \alpha), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{y}_n(t, \alpha)$$

where  $y(t) = (\underline{y}(t, \alpha), \bar{y}(t, \alpha))$  is exact solution.

To solve the example (29) by VIM

Let

$$y'(t) = \begin{cases} y(t), & t \in [0,1], \\ y(t) + 2y(1, \alpha)(t-1), & t \in [1,1.5], \\ y(t) + 2y(1, \alpha)(2-t), & t \in [1.5,2]. \end{cases}$$

a- Let  $y(0) = (0.75, 1, 1.125)$

$$y(0, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha], \quad 0 \leq \alpha \leq 1$$

Case 1: using 1 – differentiable

According to Equations (43), when  $t \in [0,1]$  the following iteration formulas can be obtained:

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \underline{y}_n(t, \alpha) - \int_0^t e^{\int_\tau^t 1 ds} (\underline{y}'_n(\tau, \alpha) - \underline{y}_n(\tau, \alpha)) d\tau \\ \bar{y}_{n+1}(t, \alpha) = \bar{y}_n(t, \alpha) - \int_0^t e^{\int_\tau^t 1 ds} (\bar{y}'_n(\tau, \alpha) - \bar{y}_n(\tau, \alpha)) d\tau \end{cases}$$

Start by initial approximations

$$\underline{y}(0, \alpha) = (0.75 + 0.25\alpha), \quad \bar{y}(0, \alpha) = (1.125 - 0.125\alpha), \quad 0 \leq \alpha \leq 1$$

and by the above iteration formulas, the following can obtained:

$$\begin{aligned}
\underline{y}_1(t, \alpha) &= \underline{y}_0(t, \alpha) - \int_0^t e^{t-\tau} \left( \underline{y}'_0(\tau, \alpha) - \underline{y}_0(\tau, \alpha) \right) d\tau, \\
\bar{y}_1(t, \alpha) &= \bar{y}_0(t, \alpha) - \int_0^t e^{t-\tau} \left( \bar{y}'_0(\tau, \alpha) - \bar{y}_0(\tau, \alpha) \right) d\tau \\
\underline{y}_1(t, \alpha) &= 0.75 + 0.25\alpha + \int_0^t e^{t-\tau} (0.75 + 0.25\alpha) d\tau, \\
\bar{y}_1(t, \alpha) &= 1.125 - 0.125\alpha + \int_0^t e^{t-\tau} (1.125 - 0.125\alpha) d\tau \\
\underline{y}_1(t, \alpha) &= e^t (0.75 + 0.25\alpha), \quad \bar{y}_1(t, \alpha) = e^t (1.125 - 0.125\alpha) \\
\underline{y}_2(t, \alpha) &= e^t (0.75 + 0.25\alpha), \quad \bar{y}_2(t, \alpha) = e^t (1.125 - 0.125\alpha) \\
&\vdots
\end{aligned}$$

Which these formulas are exactly the same as components of equation (31). Therefore, by only one iteration, the exact solution is obtained.

When  $t \in [1, 1.5]$  the following iteration formulas are obtained:

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \underline{y}_n(t, \alpha) - \int_1^t e^{\int_\tau^t 1 ds} \left( \underline{y}'_n(\tau, \alpha) - \underline{y}_n(\tau, \alpha) - 2\underline{y}(1, \alpha) \cdot (\tau - 1) \right) d\tau \\ \bar{y}_{n+1}(t, \alpha) = \bar{y}_n(t, \alpha) - \int_1^t e^{\int_\tau^t 1 ds} \left( \bar{y}'_n(\tau, \alpha) - \bar{y}_n(\tau, \alpha) - 2\bar{y}(1, \alpha) \cdot (\tau - 1) \right) d\tau \end{cases}$$

Start by the initial approximations:

$$\underline{y}(1, \alpha) = (0.75 + 0.25\alpha)e, \quad \bar{y}(1, \alpha) = (1.125 - 0.125\alpha)e, \quad 0 \leq \alpha \leq 1$$

and by the above iteration formulas, it can be obtained:

$$\begin{aligned}
\underline{y}_1(t, \alpha) &= \underline{y}_0(t, \alpha) - \int_1^t e^{t-\tau} \left( \underline{y}'_0(\tau, \alpha) - \underline{y}_0(\tau, \alpha) - 2\underline{y}(1, \alpha)(\tau - 1) \right) d\tau \\
\underline{y}_1(t, \alpha) &= (0.75 + 0.25\alpha)e + \int_1^t e^{t-\tau} (0.75 + 0.25\alpha)e + 2e^{t-\tau} (0.75 + 0.25\alpha)(\tau - 1)ed\tau \\
\underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 2t) \\
\underline{y}_2(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 2t) \\
\bar{y}_1(t, \alpha) &= \bar{y}(1, \alpha)(3e^{t-1} - 2t)
\end{aligned}$$

$$\bar{y}_2(t, \alpha) = \bar{y}(1, \alpha)(3e^{t-1} - 2t)$$

:

which are exactly the same as components of Equations (31). Therefore, by only one iteration, the exact solution is obtained.

when  $t \in [1.5, 2]$  we can obtain the following iteration formulas:

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \underline{y}_n(t, \alpha) - \int_{1.5}^t e^{\int_{\tau}^t 1 ds} \left( \underline{y}'_n(\tau, \alpha) - \underline{y}_n(\tau, \alpha) - 2\underline{y}(1, \alpha) \cdot (2 - \tau) \right) d\tau \\ \bar{y}_{n+1}(t, \alpha) = \bar{y}_n(t, \alpha) - \int_{1.5}^t e^{\int_{\tau}^t 1 ds} \left( \bar{y}'_n(\tau, \alpha) - \bar{y}_n(\tau, \alpha) - 2\bar{y}(1, \alpha) \cdot (2 - \tau) \right) d\tau \end{cases}$$

We start with initial approximations

$$\underline{y}(1.5, \alpha) = \underline{y}(1, \alpha)(3e^{0.5} - 3), \quad \bar{y}(1, \alpha) = \bar{y}(1, \alpha)(3e^{0.5} - 3), \quad 0 \leq \alpha \leq 1$$

and by the above iteration formulas, we can obtain

$$\begin{aligned} \underline{y}_1(t, \alpha) &= \underline{y}_0(t, \alpha) - \int_{1.5}^t e^{t-\tau} \left( \underline{y}'_0(\tau, \alpha) - \underline{y}_0(\tau, \alpha) - 2\underline{y}(1, \alpha)(2 - \tau) \right) d\tau \\ \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha)(3e^{0.5} - 3) + \int_{1.5}^t e^{t-\tau} \underline{y}(1, \alpha)(3e^{0.5} - 3) + 2e^{t-\tau} \underline{y}(1, \alpha)(2 - \tau) d\tau \\ \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha)(3e^{0.5} - 3) + \underline{y}(1, \alpha) \int_{1.5}^t e^{t-\tau} (3e^{0.5} - 3) + 2e^{t-\tau} (2 - \tau) d\tau \\ \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha)(3e^{0.5} - 3) \\ &\quad + \underline{y}(1, \alpha)(-e^{t-\tau}(3e^{0.5} - 3)|_{1.5}^t + 2(-e^{t-\tau}(2 - \tau) + e^{t-\tau})|_{1.5}^t) \end{aligned}$$

$$\underline{y}_1(t, \alpha) = \underline{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t)$$

$$\underline{y}_2(t, \alpha) = \underline{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t)$$

$$\bar{y}_1(t, \alpha) = \bar{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t)$$

$$\bar{y}_2(t, \alpha) = \bar{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t)$$

:

which are exactly the same as the components of equations (31). Therefore, by only one iteration, the exact solution is obtained.

Case 2: using 2 – differentiable

By Matlab software we can obtain the exact solutions by VIM when  $n = 18$ .

According to equations (45), when  $t \in [0,1]$  the following iteration formulas can be obtained:

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \underline{y}_n(t, \alpha) - \int_0^t (\underline{y}'_n(\tau, \alpha) - \bar{y}_n(\tau, \alpha)) d\tau \\ \bar{y}_{n+1}(t, \alpha) = \bar{y}_n(t, \alpha) - \int_0^t (\bar{y}'_n(\tau, \alpha) - \underline{y}_n(\tau, \alpha)) d\tau \end{cases}$$

start with initial approximations

$$\underline{y}(0, \alpha) = (0.75 + 0.25\alpha), \quad \bar{y}(0, \alpha) = (1.125 - 0.125\alpha), \quad 0 \leq \alpha \leq 1$$

**Table 3.5.1:** Numerical values for the exact and approximate solutions (VIM) at  $t = 1$

$\alpha$	Exact		VIM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.479411818960710	2.617366609400000	2.479411818960710	2.617366609400000
0.1	2.503298819910544	2.627458131305905	2.503298819910544	2.627458131305905
0.2	2.527185820860377	2.637549653211810	2.527185820860377	2.637549653211810
0.3	2.551072821810211	2.647641175117714	2.551072821810211	2.647641175117714
0.4	2.574959822760044	2.657732697023619	2.574959822760044	2.657732697023619
0.5	2.598846823709878	2.667824218929523	2.598846823709878	2.667824218929523
0.6	2.622733824659711	2.677915740835428	2.622733824659711	2.677915740835428
0.7	2.646620825609545	2.688007262741332	2.646620825609545	2.688007262741332
0.8	2.670507826559379	2.698098784647236	2.670507826559379	2.698098784647236
0.9	2.694394827509212	2.708190306553141	2.694394827509212	2.708190306553141
1	2.718281828459046	2.718281828459046	2.718281828459046	2.718281828459046

when  $t \in [1,1.5]$  we can obtain the following iteration formulas:

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \underline{y}_n(t, \alpha) - \int_0^t (\underline{y}'_n(\tau, \alpha) - \bar{y}_n(\tau, \alpha) - 2\bar{y}(1, \alpha)(\tau - 1)) d\tau \\ \bar{y}_{n+1}(t, \alpha) = \bar{y}_n(t, \alpha) - \int_0^t (\bar{y}'_n(\tau, \alpha) - \underline{y}_n(\tau, \alpha) - 2\underline{y}(1, \alpha)(\tau - 1)) d\tau \end{cases}$$

We start with initial approximations by *Table (3.5.1)*

**Table 3.5.2:** Numerical values for the exact and approximate solutions (VIM) at  $t = 1.5$

$\alpha$	Exact		VIM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	4.932442377592928	4.986723357976558	4.932442377592928	4.986723357976558
0.1	4.968220312397341	5.017073194742609	4.968220312397341	5.017073194742609
0.2	5.003998247201754	5.047423031508658	5.003998247201754	5.047423031508658
0.3	5.039776182006168	5.077772868274710	5.039776182006168	5.077772868274710
0.4	5.075554116810579	5.108122705040760	5.075554116810579	5.108122705040760
0.5	5.111332051614995	5.138472541806809	5.111332051614995	5.138472541806809
0.6	5.147109986419408	5.168822378572860	5.147109986419408	5.168822378572860
0.7	5.182887921223820	5.199172215338908	5.182887921223820	5.199172215338908
0.8	5.218665856028234	5.229522052104959	5.218665856028234	5.229522052104959
0.9	5.254443790832648	5.259871888871011	5.254443790832648	5.259871888871011
1	5.290221725637062	5.290221725637062	5.290221725637062	5.290221725637062

when  $t \in [1.5, 2]$  we can obtain the following iteration formulas:

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \underline{y}_n(t, \alpha) - \int_0^t (\underline{y}'_n(\tau, \alpha) - \bar{y}_n(\tau, \alpha) - 2\bar{y}(1, \alpha)(2-\tau)) d\tau \\ \bar{y}_{n+1}(t, \alpha) = \bar{y}_n(t, \alpha) - \int_0^t (\bar{y}'_n(\tau, \alpha) - \underline{y}_n(\tau, \alpha) - 2\underline{y}(1, \alpha)(2-\tau)) d\tau \end{cases}$$

We start with initial approximations by *Table (3.5.2)*

**Table 3.5.3:** Numerical values for the exact and approximate solutions (VIM) at  $t = 2$

$\alpha$	Exact		VIM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	9.068147228770254	9.076182156900588	9.068147228770254	9.076182156900588
0.1	9.129030073129007	9.136261508446307	9.129030073129007	9.136261508446307
0.2	9.189912917487760	9.196340859992027	9.189912917487760	9.196340859992027
0.3	9.250795761846515	9.256420211537749	9.250795761846515	9.256420211537749
0.4	9.311678606205266	9.316499563083468	9.311678606205266	9.316499563083468
0.5	9.372561450564021	9.376578914629187	9.372561450564021	9.376578914629187
0.6	9.433444294922772	9.436658266174906	9.433444294922772	9.436658266174906
0.7	9.494327139281523	9.496737617720623	9.494327139281523	9.496737617720623
0.8	9.555209983640278	9.556816969266345	9.555209983640278	9.556816969266345
0.9	9.616092827999033	9.616896320812066	9.616092827999033	9.616896320812066
1	9.676975672357788	9.676975672357788	9.676975672357788	9.676975672357788

As shown in Tables 3.5.1-3, the VIM gave exact solutions.

b- Let  $y(0) = (0.25, 0.75, 4)$

$$y(0, \alpha) = [0.25 + 0.5\alpha, 4 - 3.25\alpha], \quad 0 \leq \alpha \leq 1$$

using 1 – differentiable

When  $t \in [0,1]$ ,

$$\underline{y}(0, \alpha) = 0.25 + 0.5\alpha, \quad \bar{y}(1, \alpha) = 4 - 3.25\alpha, \quad 0 \leq \alpha \leq 1$$

by iteration formulas (43), we can obtain

$$\underline{y}_1(t, \alpha) = \underline{y}_0(t, \alpha) - \int_0^t e^{t-\tau} \left( \underline{y}'_0(\tau, \alpha) - \underline{y}_0(\tau, \alpha) \right) d\tau$$

$$\underline{y}_1(t, \alpha) = 0.25 + 0.5\alpha + \int_0^t e^{t-\tau} (0.25 + 0.5\alpha) d\tau$$

$$\underline{y}_1(t, \alpha) = e^t (0.25 + 0.5\alpha), \quad \bar{y}_1(t, \alpha) = e^t (4 - 3.25\alpha)$$

$$\underline{y}_2(t, \alpha) = e^t (0.25 + 0.5\alpha), \quad \bar{y}_2(t, \alpha) = e^t (4 - 3.25\alpha)$$

:

Therefore, by only one iteration, the exact solution is obtained.

When  $t \in [1,1.5]$ , start with initial approximations

$$\underline{y}(1, \alpha) = (0.25 + 0.5\alpha)e, \quad \bar{y}(1, \alpha) = (4 - 3.25\alpha)e, \quad 0 \leq \alpha \leq 1$$

by the iteration formulas (43), we can obtain

$$\underline{y}_1(t, \alpha) = \underline{y}_0(t, \alpha) - \int_1^t e^{t-\tau} \left( \underline{y}'_0(\tau, \alpha) - \underline{y}_0(\tau, \alpha) - 2\underline{y}(1, \alpha)(\tau - 1) \right) d\tau$$

$$\underline{y}_1(t, \alpha) = (0.25 + 0.5\alpha)e + \int_1^t e^{t-\tau} (0.25 + 0.5\alpha)e + 2e^{t-\tau} (0.25 + 0.5\alpha)(\tau - 1)e d\tau$$

$$\underline{y}_1(t, \alpha) = \underline{y}(1, \alpha)(3e^{t-1} - 2t), \quad \bar{y}_1(t, \alpha) = \bar{y}(1, \alpha)(3e^{t-1} - 2t)$$

$$\underline{y}_2(t, \alpha) = \underline{y}(1, \alpha)(3e^{t-1} - 2t), \quad \bar{y}_2(t, \alpha) = \bar{y}(1, \alpha)(3e^{t-1} - 2t)$$

:

By only one iteration, the exact solution is obtained.

When  $t \in [1.5, 2]$ , start with initial approximations

$$\underline{y}(1.5, \alpha) = \underline{y}(1, \alpha)(3e^{0.5} - 3), \quad \bar{y}(1, \alpha) = \bar{y}(1, \alpha)(3e^{0.5} - 3), \quad 0 \leq \alpha \leq 1$$

and by the iteration formulas (43), obtain

$$\begin{aligned}\underline{y}_1(t, \alpha) &= \underline{y}_0(t, \alpha) - \int_{1.5}^t e^{t-\tau} \left( \underline{y}'_0(\tau, \alpha) - \underline{y}_0(\tau, \alpha) - 2\underline{y}(1, \alpha) \cdot (2 - \tau) \right) d\tau \\ \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha)(3e^{0.5} - 3) + \int_{1.5}^t e^{t-\tau} \underline{y}(1, \alpha)(3e^{0.5} - 3) + 2e^{t-\tau} \underline{y}(1, \alpha)(2 - \tau) d\tau \\ \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t) \\ \bar{y}_1(t, \alpha) &= \bar{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t) \\ \underline{y}_2(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t) \\ \bar{y}_2(t, \alpha) &= \bar{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t) \\ &\vdots\end{aligned}$$

By only one iteration, the exact solution is obtained.

### 3.6 Adomian Decomposition Method (ADM)

In this section, hybrid fuzzy differential equations (3) are solved by Adomian Decomposition Method.

By 1 – differentiable, we rewrite Equations (40) in the following form:

$$\begin{cases} \underline{y}(t, \alpha) = \underline{y}_0(\alpha) + \int_{t_0}^t a(s) \cdot \underline{y}(s, \alpha) ds + \int_{t_0}^t \underline{b}(s, \alpha) ds \\ \bar{y}(t, \alpha) = \bar{y}_0(\alpha) + \int_{t_0}^t a(s) \cdot \bar{y}(s, \alpha) ds + \int_{t_0}^t \bar{b}(s, \alpha) ds. \end{cases} \quad (46)$$

By 2 –differentiable, we rewrite Equations (41) in the following form:

$$\begin{cases} \underline{y}(t, \alpha) = \underline{y}_0(\alpha) + \int_{t_0}^t a(s) \cdot \bar{y}(s, \alpha) ds + \int_{t_0}^t \bar{b}(s, \alpha) ds \\ \bar{y}(t, \alpha) = \bar{y}_0(\alpha) + \int_{t_0}^t a(s) \cdot \underline{y}(s, \alpha) ds + \int_{t_0}^t \underline{b}(s, \alpha) ds. \end{cases} \quad (47)$$

To use ADM by 1-differentiable let

$$\underline{y}(t, \alpha) = \sum_{n=0}^{\infty} \underline{y}_n(t, \alpha), \quad \bar{y}(t, \alpha) = \sum_{n=0}^{\infty} \bar{y}_n(t, \alpha), \quad (48)$$

substituting (48) in (46) we have:

$$\begin{cases} \sum_{n=0}^{\infty} \underline{y}_n(t, \alpha) = \underline{y}_0(\alpha) + \int_{t_0}^t \underline{b}(s, \alpha) ds + \sum_{n=0}^{\infty} \left( \int_{t_0}^t a(s) \cdot \underline{y}(s, \alpha) ds \right) \\ \sum_{n=0}^{\infty} \bar{y}_n(t, \alpha) = \bar{y}_0(\alpha) + \int_{t_0}^t \bar{b}(s, \alpha) ds + \sum_{n=0}^{\infty} \left( \int_{t_0}^t a(s) \cdot \bar{y}(s, \alpha) ds \right). \end{cases}$$

Identifying the zeroth components

$$\underline{y}_0(t, \alpha) = \underline{y}_0(\alpha) + \int_{t_0}^t \underline{b}(s, \alpha) ds, \quad \bar{y}_0(t, \alpha) = \bar{y}_0(\alpha) + \int_{t_0}^t \bar{b}(s, \alpha) ds$$

the remaining components  $\underline{y}_n(t, \alpha)$  and  $\bar{y}_n(t, \alpha), n > 1$ , can be determined by using the recurrence relations

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \int_{t_0}^t a(s) \underline{y}_n(s, \alpha) ds, & n \geq 0 \\ \bar{y}_{n+1}(t, \alpha) = \int_{t_0}^t a(s) \bar{y}_n(s, \alpha) ds, & n \geq 0 \end{cases} \quad (49)$$

By Equations (49), we approximate  $\underline{y}(x, r)$  with

$$\underline{\Phi}_n(t) = \sum_{k=0}^n \underline{y}_k(t, \alpha)$$

and approximate  $\bar{y}(x, r)$  with

$$\bar{\Phi}_n(t) = \sum_{k=0}^n \bar{y}_k(t, \alpha)$$

and the exact solution is obtained at the limit of the resulting approximations, i.e.,

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{\Phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\Phi}_n(t)$$

and  $y(t) = (\underline{y}(t, \alpha); \bar{y}(t, \alpha))$  is exact solution.

## By 2-differentiable

$$\begin{cases} \underline{y}_0(t, \alpha) = \underline{y}_0(\alpha) + \int_{t_0}^t \bar{b}(s, \alpha) ds, & \bar{y}_0(t, \alpha) = \bar{y}_0(\alpha) + \int_{t_0}^t \underline{b}(s, \alpha) ds \\ \underline{y}_{n+1}(t, \alpha) = \int_{t_0}^t a(s) \bar{y}_n(s, \alpha) ds, & n \geq 0 \\ \bar{y}_{n+1}(t, \alpha) = \int_{t_0}^t a(s) \underline{y}_n(s, \alpha) ds, & n \geq 0. \end{cases} \quad (50)$$

We approximate  $\underline{y}(x, r)$  with

$$\underline{\phi}_n(t) = \sum_{k=0}^n \underline{y}_k(t, \alpha)$$

and approximate  $\bar{y}(x, r)$  with

$$\bar{\phi}_n(t) = \sum_{k=0}^n \bar{y}_k(t, \alpha)$$

and the exact solution is obtained at the limit of the resulting approximations, i.e.,

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{\phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\phi}_n(t)$$

and  $y(t) = (\underline{y}(t, \alpha); \bar{y}(t, \alpha))$  is exact solution.

To solve example (29) by ADM (calculations)

Let

$$y'(t) = \begin{cases} y(t), & t \in [0,1], \\ y(t) + 2y(1, \alpha)(t-1), & t \in [1,1.5], \\ y(t) + 2y(1, \alpha)(2-t), & t \in [1.5,2]. \end{cases}$$

a- Let  $y(0) = (0.75, 1, 1.125)$

$$y(0, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha], \quad 0 \leq \alpha \leq 1$$

Case 1: using 1-differentiable

According to Equations (49), when  $t \in [0,1]$  we have

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \int_{t_0}^t \underline{y}_n(s, \alpha) ds, & n \geq 0 \\ \bar{y}_{n+1}(t, \alpha) = \int_{t_0}^t \bar{y}_n(s, \alpha) ds, & n \geq 0 \end{cases}$$

where

$$\underline{y}_0(t, \alpha) = \underline{y}_0(\alpha), \quad \bar{y}_0(t, \alpha) = \bar{y}_0(\alpha)$$

$$\underline{y}_0(t, \alpha) = 0.75 + 0.25\alpha, \quad \bar{y}_0(t, \alpha) = 1.125 - 0.125\alpha$$

We approximate  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\underline{\phi}_6(t) = \sum_{k=0}^6 \underline{y}_k(t, \alpha), \quad \bar{\phi}_6(t) = \sum_{k=0}^6 \bar{y}_k(t, \alpha)$$

$$\begin{aligned}
\underline{y}_1(t, \alpha) &= \int_0^t \underline{y}_0(s, \alpha) ds, & \bar{y}_1(t, \alpha) &= \int_0^t \bar{y}_0(s, \alpha) ds \\
\underline{y}_1(t, \alpha) &= \int_0^t 0.75 + 0.25\alpha s ds, & \bar{y}_1(t, \alpha) &= \int_0^t 1.125 - 0.125\alpha s ds \\
\underline{y}_1(t, \alpha) &= (0.75 + 0.25\alpha)t, & \bar{y}_1(t, \alpha) &= (1.125 - 0.125\alpha)t \\
\underline{y}_2(t, \alpha) &= \int_0^t \underline{y}_1(s, \alpha) ds, & \bar{y}_2(t, \alpha) &= \int_0^t \bar{y}_1(s, \alpha) ds \\
\underline{y}_2(t, \alpha) &= \int_0^t (0.75 + 0.25\alpha)s ds, & \bar{y}_2(t, \alpha) &= \int_0^t (1.125 - 0.125\alpha)s ds \\
\underline{y}_2(t, \alpha) &= (0.75 + 0.25\alpha)\frac{t^2}{2}, & \bar{y}_2(t, \alpha) &= (1.125 - 0.125\alpha)\frac{t^2}{2} \\
\underline{y}_3(t, \alpha) &= (0.75 + 0.25\alpha)\frac{t^3}{6}, & \bar{y}_3(t, \alpha) &= (1.125 - 0.125\alpha)\frac{t^3}{6} \\
&\vdots \\
\underline{y}_6(t, \alpha) &= (0.75 + 0.25\alpha)\frac{t^6}{720}, & \bar{y}_6(t, \alpha) &= (1.125 - 0.125\alpha)\frac{t^6}{720} \\
\underline{\Phi}_6(t) &= \sum_{k=0}^6 \underline{y}_0(t, \alpha) \frac{t^k}{k!}, & \bar{\Phi}_6(t) &= \sum_{k=0}^6 \bar{y}_0(t, \alpha) \frac{t^k}{k!}
\end{aligned}$$

To find exact solution

$$\begin{aligned}
\underline{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \underline{\Phi}_n(t), & \bar{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \bar{\Phi}_n(t) \\
\underline{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \sum_{k=0}^n \underline{y}_0(t, \alpha) \frac{t^k}{k!}, & \bar{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \sum_{k=0}^n \bar{y}_0(t, \alpha) \frac{t^k}{k!} \\
\underline{y}(t, \alpha) &= \underline{y}_0(t, \alpha) e^t, & \bar{y}(t, \alpha) &= \bar{y}_0(t, \alpha) e^t
\end{aligned}$$

where

$$e^t = \sum_{k=0}^{\infty} \frac{t^k}{k!}$$

When  $t \in [1, 1.5]$  we have

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \int_{t_0}^t \underline{y}_n(s, \alpha) ds, & n \geq 0 \\ \bar{y}_{n+1}(t, \alpha) = \int_{t_0}^t \bar{y}_n(s, \alpha) ds, & n \geq 0 \end{cases}$$

where

$$y_0(\alpha) = y(1, \alpha) = [(0.75 + 0.25\alpha)e, (1.125 - 0.125\alpha)e]$$

$$\begin{aligned} \underline{y}_0(t, \alpha) &= \underline{y}_0(\alpha) + \int_1^t 2\underline{y}(1, \alpha)(s-1) ds, \\ \bar{y}_0(t, \alpha) &= \bar{y}_0(\alpha) + \int_1^t 2\bar{y}(1, \alpha)(s-1) ds \end{aligned}$$

$$\underline{y}_0(t, \alpha) = \underline{y}(1, \alpha)(t^2 - 2t + 2), \quad \bar{y}_0(t, \alpha) = \bar{y}(1, \alpha)(t^2 - 2t + 2)$$

$$\underline{y}_0(t, \alpha) = \underline{y}(1, \alpha)((t-1)^2 + 1), \quad \bar{y}_0(t, \alpha) = \bar{y}(1, \alpha)((t-1)^2 + 1)$$

We approximate  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\begin{aligned} \underline{\phi}_6(t) &= \sum_{k=0}^6 \underline{y}_k(t, \alpha), \quad \bar{\phi}_6(t) = \sum_{k=0}^6 \bar{y}_k(t, \alpha) \\ \underline{y}_1(t, \alpha) &= \int_1^t \underline{y}_0(s, \alpha) ds, \quad \bar{y}_1(t, \alpha) = \int_1^t \bar{y}_0(s, \alpha) ds \\ \underline{y}_1(t, \alpha) &= \int_1^t \underline{y}(1, \alpha)((s-1)^2 + 1) ds, \quad \bar{y}_1(t, \alpha) = \int_1^t \bar{y}(1, \alpha)((s-1)^2 + 1) ds \\ \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^3}{3} + t - 1 \right), \quad \bar{y}_1(t, \alpha) = \bar{y}(1, \alpha) \left( \frac{(t-1)^3}{3} + t - 1 \right) \\ \underline{y}_2(t, \alpha) &= \int_1^t \underline{y}_1(s, \alpha) ds, \quad \bar{y}_2(t, \alpha) = \int_1^t \bar{y}_1(s, \alpha) ds, \\ \underline{y}_2(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^4}{12} + \frac{t^2}{2} - t + \frac{1}{2} \right) \\ \underline{y}_2(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^4}{12} + \frac{1}{2}(t^2 - 2t + 1) \right) \\ \underline{y}_2(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^4}{12} + \frac{(t-1)^2}{2} \right), \quad \bar{y}_2(t, \alpha) = \bar{y}(1, \alpha) \left( \frac{(t-1)^4}{12} + \frac{(t-1)^2}{2} \right) \end{aligned}$$

$$\begin{aligned}\underline{y}_3(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^5}{60} + \frac{(t-1)^3}{6} \right), & \bar{y}_3(t, \alpha) &= \bar{y}(1, \alpha) \left( \frac{(t-1)^5}{60} + \frac{(t-1)^3}{6} \right) \\ &\vdots \\ \underline{y}_6(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^8}{20160} + \frac{(t-1)^6}{720} \right), & \bar{y}_6(t, \alpha) &= \bar{y}(1, \alpha) \left( \frac{(t-1)^8}{20160} + \frac{(t-1)^6}{720} \right)\end{aligned}$$

Then

$$\begin{aligned}\underline{\Phi}_6(t) &= \sum_{k=0}^6 \underline{y}(1, \alpha) \left( \frac{2(t-1)^{k+2}}{(k+2)!} + \frac{(t-1)^k}{(k)!} \right), \\ \bar{\Phi}_6(t) &= \sum_{k=0}^6 \bar{y}(1, \alpha) \left( \frac{2(t-1)^{k+2}}{(k+2)!} + \frac{(t-1)^k}{(k)!} \right)\end{aligned}$$

To find exact solution

$$\begin{aligned}\underline{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \underline{\Phi}_n(t), & \bar{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \bar{\Phi}_n(t) \\ \underline{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \sum_{k=0}^n \underline{y}(1, \alpha) \left( \frac{2(t-1)^{k+2}}{(k+2)!} + \frac{(t-1)^k}{(k)!} \right) \\ \underline{y}(t, \alpha) &= 2\underline{y}(1, \alpha)(e^{t-1} - 1 - (t-1)) + \underline{y}(1, \alpha)e^{t-1} \\ \underline{y}(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 2t)\end{aligned}$$

where

$$\begin{aligned}e^{t-1} &= \sum_{k=0}^{\infty} \frac{(t-1)^k}{k!} \\ \bar{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \sum_{k=0}^n \bar{y}(1, \alpha) \left( \frac{2(t-1)^{k+2}}{(k+2)!} + \frac{(t-1)^k}{(k)!} \right) \\ \bar{y}(t, \alpha) &= \bar{y}(1, \alpha)(3e^{t-1} - 2t)\end{aligned}$$

when  $t \in [1.5, 2]$  we have

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \int_{t_0}^t \underline{y}_n(s, \alpha) ds, & n \geq 0 \\ \bar{y}_{n+1}(t, \alpha) = \int_{t_0}^t \bar{y}_n(s, \alpha) ds, & n \geq 0 \end{cases}$$

where

$$\begin{aligned}
y_0(\alpha) &= y(1.5, \alpha) = [\underline{y}(1, \alpha)(3e^{0.5} - 3), \bar{y}(1, \alpha)(3e^{0.5} - 3)] \\
\underline{y}_0(t, \alpha) &= \underline{y}_0(\alpha) + \int_{1.5}^t 2 \underline{y}(1, \alpha)(2-s) ds \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) + 2\underline{y}(1, \alpha) \left( 2s - \frac{s^2}{2} \Big|_{1.5}^t \right) \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) + 2\underline{y}(1, \alpha) \left( 2t - \frac{t^2}{2} - 3 + \frac{9}{8} \right) \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) - \underline{y}(1, \alpha) \left( t^2 - 4t + \frac{15}{4} \right) \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) - \underline{y}(1, \alpha) \left( t^2 - 3t + \frac{9}{4} - t + \frac{6}{4} \right) \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) - \underline{y}(1, \alpha)((t-1.5)^2 - (t-1.5)) \\
\bar{y}_0(t, \alpha) &= \bar{y}(1.5, \alpha) - \bar{y}(1, \alpha)((t-1.5)^2 - (t-1.5))
\end{aligned}$$

We approximate  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\begin{aligned}
\underline{\phi}_6(t) &= \sum_{k=0}^6 \underline{y}_k(t, \alpha), & \bar{\phi}_6(t) &= \sum_{k=0}^6 \bar{y}_k(t, \alpha) \\
\underline{y}_1(t, \alpha) &= \int_{1.5}^t \underline{y}_0(s, \alpha) ds, & \bar{y}_1(t, \alpha) &= \int_{1.5}^t \bar{y}_0(s, \alpha) ds \\
\underline{y}_1(t, \alpha) &= \int_{1.5}^t \left( \underline{y}(1.5, \alpha) - \underline{y}(1, \alpha)((s-1.5)^2 - (s-1.5)) \right) ds \\
\underline{y}_1(t, \alpha) &= \underline{y}(1.5, \alpha)(t-1.5) - \underline{y}(1, \alpha) \left( \frac{(t-1.5)^3}{3} - \frac{(t-1.5)^2}{2} \right) \\
\bar{y}_1(t, \alpha) &= \bar{y}(1.5, \alpha)(t-1.5) - \bar{y}(1, \alpha) \left( \frac{(t-1.5)^3}{3} - \frac{(t-1.5)^2}{2} \right) \\
\underline{y}_2(t, \alpha) &= \int_{1.5}^t \underline{y}_1(s, \alpha) ds, & \bar{y}_2(t, \alpha) &= \int_{1.5}^t \bar{y}_1(s, \alpha) ds \\
\underline{y}_2(t, \alpha) &= \underline{y}(1.5, \alpha) \frac{(t-1.5)^2}{2} - \underline{y}(1, \alpha) \left( \frac{(t-1.5)^4}{12} - \frac{(t-1.5)^3}{6} \right) \\
\bar{y}_2(t, \alpha) &= \bar{y}(1.5, \alpha) \frac{(t-1.5)^2}{2} - \bar{y}(1, \alpha) \left( \frac{(t-1.5)^4}{12} - \frac{(t-1.5)^3}{6} \right)
\end{aligned}$$

$$\underline{y}_3(t, \alpha) = \underline{y}(1.5, \alpha) \frac{(t - 1.5)^3}{6} - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^5}{60} - \frac{(t - 1.5)^4}{24} \right)$$

$$\bar{y}_3(t, \alpha) = \bar{y}(1.5, \alpha) \frac{(t - 1.5)^3}{6} - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^5}{60} - \frac{(t - 1.5)^4}{24} \right)$$

⋮

$$\underline{y}_6(t, \alpha) = \underline{y}(1.5, \alpha) \frac{(t - 1.5)^6}{720} - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^8}{20160} - \frac{(t - 1.5)^7}{5040} \right)$$

$$\bar{y}_6(t, \alpha) = \bar{y}(1.5, \alpha) \frac{(t - 1.5)^6}{720} - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^8}{20160} - \frac{(t - 1.5)^7}{5040} \right).$$

Then

$$\underline{\Phi}_6(t) = \sum_{k=0}^6 \underline{y}(1.5, \alpha) \frac{(t - 1.5)^k}{k!} - \underline{y}(1, \alpha) \left( \frac{2(t - 1.5)^{k+2}}{(k+2)!} - \frac{(t - 1.5)^{k+1}}{(k+1)!} \right)$$

$$\bar{\Phi}_6(t) = \sum_{k=0}^6 \bar{y}(1.5, \alpha) \frac{(t - 1.5)^k}{k!} - \bar{y}(1, \alpha) \left( \frac{2(t - 1.5)^{k+2}}{(k+2)!} - \frac{(t - 1.5)^{k+1}}{(k+1)!} \right).$$

To find exact solution

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{\Phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\Phi}_n(t)$$

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \sum_{k=0}^n \underline{y}(1.5, \alpha) \frac{(t - 1.5)^k}{k!} - \underline{y}(1, \alpha) \left( \frac{2(t - 1.5)^{k+2}}{(k+2)!} - \frac{(t - 1.5)^{k+1}}{(k+1)!} \right)$$

$$\underline{y}(t, \alpha) = \underline{y}(1.5, \alpha) \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{(t - 1.5)^k}{k!} - \underline{y}(1, \alpha) \lim_{n \rightarrow \infty} \sum_{k=0}^n \left( \frac{2(t - 1.5)^{k+2}}{(k+2)!} - \frac{(t - 1.5)^{k+1}}{(k+1)!} \right)$$

$$\underline{y}(t, \alpha) = \bar{y}(1.5, \alpha) e^{t-1.5} - 2\underline{y}(1, \alpha)(e^{t-1.5} - 1 - t + 1.5) + \underline{y}(1, \alpha)(e^{t-1.5} - 1)$$

$$\underline{y}(t, \alpha) = \underline{y}(1, \alpha)(3e^{0.5} - 3)e^{t-1.5} - 2\underline{y}(1, \alpha)(e^{t-1.5} - 1 - t + 1.5) + \underline{y}(1, \alpha)(e^{t-1.5} - 1)$$

$$\underline{y}(t, \alpha) = \underline{y}(1, \alpha)(3e^{t-1} - 3e^{t-1.5} - 2e^{t-1.5} + 2t - 1 + e^{t-1.5} - 1)$$

$$\underline{y}(t, \alpha) = \underline{y}(1, \alpha)(2t - 2 + 3e^{t-1} - 4e^{t-1.5})$$

where

$$e^{t-1.5} = \sum_{k=0}^{\infty} \frac{(t - 1.5)^k}{k!}$$

$$\bar{y}(t, \alpha) = \bar{y}(1, \alpha)(2t - 2 + 3e^{t-1} - 4e^{t-1.5})$$

Case 2: using 2 – differentiable

When  $t \in [0,1]$ , According to Equations (50)

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \int_{t_0}^t \bar{y}_n(s, \alpha) ds, & n \geq 0 \\ \bar{y}_{n+1}(t, \alpha) = \int_{t_0}^t \underline{y}_n(s, \alpha) ds, & n \geq 0 \end{cases}$$

where

$$\underline{y}_0(t, \alpha) = \underline{y}_0(\alpha), \quad \bar{y}_0(t, \alpha) = \bar{y}_0(\alpha)$$

$$\underline{y}_0(t, \alpha) = 0.75 + 0.25\alpha, \quad \bar{y}_0(t, \alpha) = 1.125 - 0.125\alpha$$

We approximate  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\underline{\phi}_6(t) = \sum_{k=0}^6 \underline{y}_k(t, \alpha), \quad \bar{\phi}_6(t) = \sum_{k=0}^6 \bar{y}_k(t, \alpha)$$

$$\underline{y}_1(t, \alpha) = \int_0^t \bar{y}_0(s, \alpha) ds, \quad \bar{y}_1(t, \alpha) = \int_0^t \underline{y}_0(s, \alpha) ds$$

$$\underline{y}_1(t, \alpha) = (1.125 - 0.125\alpha)t, \quad \bar{y}_1(t, \alpha) = (0.75 + 0.25\alpha)t$$

$$\underline{y}_2(t, \alpha) = \int_0^t \bar{y}_1(s, \alpha) ds, \quad \bar{y}_2(t, \alpha) = \int_0^t \underline{y}_1(s, \alpha) ds$$

$$\underline{y}_2(t, \alpha) = (0.75 + 0.25\alpha)\frac{t^2}{2}, \quad \bar{y}_2(t, \alpha) = (1.125 - 0.125\alpha)\frac{t^2}{2}$$

$$\underline{y}_3(t, \alpha) = (1.125 - 0.125\alpha)\frac{t^3}{6}, \quad \bar{y}_3(t, \alpha) = (0.75 + 0.25\alpha)\frac{t^3}{6}$$

⋮

$$\underline{y}_6(t, \alpha) = (0.75 + 0.25\alpha)\frac{t^6}{720}, \quad \bar{y}_6(t, \alpha) = (1.125 - 0.125\alpha)\frac{t^6}{720}$$

$$\underline{\phi}_6(t) = 0.75 + 0.25\alpha + (1.125 - 0.125\alpha)t + (0.75 + 0.25\alpha)\frac{t^2}{2} + (1.125 - 0.125\alpha)\frac{t^3}{6} \\ + (0.75 + 0.25\alpha)\frac{t^4}{24} + (1.125 - 0.125\alpha)\frac{t^5}{120} + (0.75 + 0.25\alpha)\frac{t^6}{720}$$

$$\underline{\phi}_6(t) = 0.9375 - 0.1875 + 0.0625\alpha + 0.1875\alpha + (0.9375 + 0.1875)t \\ + (0.0625\alpha - 0.1875\alpha)t + (0.9375 - 0.1875)\frac{t^2}{2} \\ + (0.0625\alpha + 0.1875\alpha)\frac{t^2}{2} + (0.9375 + 0.1875)\frac{t^3}{6} \\ + (0.0625\alpha - 0.1875\alpha)\frac{t^3}{6} + \dots + (0.9375 - 0.1875)\frac{t^6}{720} \\ + (0.0625\alpha + 0.1875\alpha)\frac{t^6}{720}$$

$$\underline{\phi}_6(t) = (0.9375 + 0.0625\alpha) + (0.9375 + 0.0625\alpha)t + (0.9375 + 0.0625\alpha)\frac{t^2}{2} \\ + (0.9375 + 0.0625\alpha)\frac{t^3}{6} + \dots + (0.9375 + 0.0625\alpha)\frac{t^6}{720} \\ + (-0.1875 + 0.1875\alpha) - (-0.1875 + 0.1875\alpha)t \\ + (-0.1875 + 0.1875\alpha)\frac{t^2}{2} - (-0.1875 + 0.1875\alpha)\frac{t^3}{6} + \dots \\ + (-0.1875 + 0.1875\alpha)\frac{t^6}{720}$$

$$\underline{\phi}_6(t) = \sum_{k=0}^6 (0.9375 + 0.0625\alpha)\frac{t^k}{k!} + (-0.1875 + 0.1875\alpha)\frac{(-t)^k}{k!}$$

$$\bar{\phi}_6(t) = 1.125 - 0.125\alpha + (0.75 + 0.25\alpha)t + (1.125 - 0.125\alpha)\frac{t^2}{2} + (0.75 + 0.25\alpha)\frac{t^3}{6} \\ + (1.125 - 0.125\alpha)\frac{t^4}{24} + (0.75 + 0.25\alpha)\frac{t^5}{120} + (1.125 - 0.125\alpha)\frac{t^6}{720}$$

$$\bar{\phi}_6(t) = 0.9375 + 0.1875 + 0.0625\alpha - 0.1875\alpha + (0.9375 - 0.1875)t \\ + (0.0625\alpha + 0.1875\alpha)t + (0.9375 + 0.1875)\frac{t^2}{2} \\ + (0.0625\alpha - 0.1875\alpha)\frac{t^2}{2} + (0.9375 - 0.1875)\frac{t^3}{6} \\ + (0.0625\alpha + 0.1875\alpha)\frac{t^3}{6} + \dots + (0.9375 + 0.1875)\frac{t^6}{720} \\ + (0.0625\alpha - 0.1875\alpha)\frac{t^6}{720}$$

$$\begin{aligned}\bar{\phi}_6(t) &= (0.9375 + 0.0625\alpha) + (0.9375 + 0.0625\alpha)t + (0.9375 + 0.0625\alpha)\frac{t^2}{2} \\ &\quad + (0.9375 + 0.0625\alpha)\frac{t^3}{6} + \cdots + (0.9375 + 0.0625\alpha)\frac{t^6}{720} \\ &\quad + (0.1875 - 0.1875\alpha) - (0.1875 - 0.1875\alpha)t + (0.1875 - 0.1875\alpha)\frac{t^2}{2} \\ &\quad - (0.1875 - 0.1875\alpha)\frac{t^3}{6} + \cdots + (0.1875 - 0.1875\alpha)\frac{t^6}{720}\end{aligned}$$

$$\bar{\phi}_6(t) = \sum_{k=0}^6 (0.9375 + 0.0625\alpha)\frac{t^k}{k!} - (-0.1875 + 0.1875\alpha)\frac{(-t)^k}{k!}$$

To find exact solution

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{\phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\phi}_n(t)$$

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \sum_{k=0}^n (0.9375 + 0.0625\alpha)\frac{t^k}{k!} + (-0.1875 + 0.1875\alpha)\frac{(-t)^k}{k!}$$

$$\underline{y}(t, \alpha) = (0.9375 + 0.0625\alpha)e^t + (-0.1875 + 0.1875\alpha)e^{-t}$$

where

$$e^{-t} = \sum_{k=0}^{\infty} \frac{(-t)^k}{k!}$$

$$\bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \sum_{k=0}^n (0.9375 + 0.0625\alpha)\frac{t^k}{k!} - (-0.1875 + 0.1875\alpha)\frac{(-t)^k}{k!}$$

$$\bar{y}(t, \alpha) = (0.9375 + 0.0625\alpha)e^t - (-0.1875 + 0.1875\alpha)e^{-t}$$

When  $t \in [1, 1.5]$  we have

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \int_{t_0}^t \bar{y}_n(s, \alpha) ds, & n \geq 0 \\ \bar{y}_{n+1}(t, \alpha) = \int_{t_0}^t \underline{y}_n(s, \alpha) ds, & n \geq 0 \end{cases}$$

where

$$\underline{y}_0(\alpha) = \underline{y}(1, \alpha) = (0.9375 + 0.0625\alpha)e + (-0.1875 + 0.1875\alpha)e^{-1}$$

$$\bar{y}_0(\alpha) = \bar{y}(1, \alpha) = (0.9375 + 0.0625\alpha)e - (-0.1875 + 0.1875\alpha)e^{-1}$$

$$\begin{aligned}\underline{y}_0(t, \alpha) &= \underline{y}_0(\alpha) + \int_1^t 2\underline{y}(1, \alpha)(s-1)ds, & \bar{y}_0(t, \alpha) &= \bar{y}_0(\alpha) + \int_1^t 2\underline{y}(1, \alpha)(s-1)ds \\ \underline{y}_0(t, \alpha) &= \underline{y}(1, \alpha) + 2\bar{y}(1, \alpha)\left(\frac{s^2}{2} - s|_1^t\right), & \bar{y}_0(t, \alpha) &= \bar{y}(1, \alpha) + 2\underline{y}(1, \alpha)\left(\frac{s^2}{2} - s|_1^t\right)\end{aligned}$$

$$\underline{y}_0(t, \alpha) = \underline{y}(1, \alpha) + \bar{y}(1, \alpha)(t-1)^2, \quad \bar{y}_0(t, \alpha) = \bar{y}(1, \alpha) + \underline{y}(1, \alpha)(t-1)^2$$

We approximate  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\underline{\phi}_6(t) = \sum_{k=0}^6 \underline{y}_k(t, \alpha), \quad \bar{\phi}_6(t) = \sum_{k=0}^6 \bar{y}_k(t, \alpha)$$

$$\underline{y}_1(t, \alpha) = \int_1^t \bar{y}_0(s, \alpha)ds, \quad \bar{y}_1(t, \alpha) = \int_1^t \underline{y}_0(s, \alpha)ds$$

$$\underline{y}_1(t, \alpha) = \int_1^t \bar{y}(1, \alpha) + \underline{y}(1, \alpha)(t-1)^2 ds$$

$$\underline{y}_1(t, \alpha) = \bar{y}(1, \alpha)(t-1) + \underline{y}(1, \alpha) \frac{(t-1)^3}{3}$$

$$\bar{y}_1(t, \alpha) = \underline{y}(1, \alpha)(t-1) + \bar{y}(1, \alpha) \frac{(t-1)^3}{3}$$

$$\underline{y}_2(t, \alpha) = \int_1^t \bar{y}_1(s, \alpha)ds, \quad \bar{y}_2(t, \alpha) = \int_1^t \underline{y}_1(s, \alpha)ds$$

$$\underline{y}_2(t, \alpha) = \int_1^t \underline{y}(1, \alpha)(t-1) + \bar{y}(1, \alpha) \frac{(t-1)^3}{3} ds$$

$$\underline{y}_2(t, \alpha) = \underline{y}(1, \alpha) \frac{(t-1)^2}{2} + \bar{y}(1, \alpha) \frac{(t-1)^4}{12}$$

$$\bar{y}_2(t, \alpha) = \bar{y}(1, \alpha) \frac{(t-1)^2}{2} + \underline{y}(1, \alpha) \frac{(t-1)^4}{12}$$

$$\underline{y}_3(t, \alpha) = \bar{y}(1, \alpha) \frac{(t-1)^3}{6} + \underline{y}(1, \alpha) \frac{(t-1)^5}{60}$$

$$\bar{y}_3(t, \alpha) = \underline{y}(1, \alpha) \frac{(t-1)^3}{6} + \bar{y}(1, \alpha) \frac{(t-1)^5}{60}$$

⋮

$$\underline{y}_6(t, \alpha) = \underline{y}(1, \alpha) \frac{(t-1)^6}{720} + \bar{y}(1, \alpha) \frac{(t-1)^8}{20160}$$

$$\bar{y}_6(t, \alpha) = \bar{y}(1, \alpha) \frac{(t-1)^6}{720} + \underline{y}(1, \alpha) \frac{(t-1)^8}{20160}$$

Then

$$\begin{aligned} \underline{\phi}_6(t) &= \underline{y}(1, \alpha) + \bar{y}(1, \alpha)(t-1)^2 + \bar{y}(1, \alpha)(t-1) + \underline{y}(1, \alpha) \frac{(t-1)^3}{3} + \underline{y}(1, \alpha) \frac{(t-1)^2}{2} \\ &\quad + \bar{y}(1, \alpha) \frac{(t-1)^4}{12} + \bar{y}(1, \alpha) \frac{(t-1)^3}{6} + \underline{y}(1, \alpha) \frac{(t-1)^5}{60} + \underline{y}(1, \alpha) \frac{(t-1)^4}{24} \\ &\quad + \bar{y}(1, \alpha) \frac{(t-1)^6}{360} + \bar{y}(1, \alpha) \frac{(t-1)^5}{120} + \underline{y}(1, \alpha) \frac{(t-1)^7}{2520} + \underline{y}(1, \alpha) \frac{(t-1)^6}{720} \\ &\quad + \bar{y}(1, \alpha) \frac{(t-1)^8}{20160} \end{aligned}$$

$$\begin{aligned} \underline{\phi}_6(t) &= \underline{y}(1, \alpha) + \frac{3}{4} \bar{y}(1, \alpha)(t-1)^2 + \frac{1}{4} \bar{y}(1, \alpha)(t-1)^2 + \frac{3}{2} \bar{y}(1, \alpha)(t-1) \\ &\quad - \frac{1}{2} \bar{y}(1, \alpha)(t-1) + \frac{3}{4} \underline{y}(1, \alpha) \frac{(t-1)^3}{3} + \frac{1}{4} \underline{y}(1, \alpha) \frac{(t-1)^3}{3} \\ &\quad + \frac{3}{2} \underline{y}(1, \alpha) \frac{(t-1)^2}{2} - \frac{1}{2} \underline{y}(1, \alpha) \frac{(t-1)^2}{2} + \frac{3}{4} \bar{y}(1, \alpha) \frac{(t-1)^4}{12} + \frac{1}{4} \bar{y}(1, \alpha) \frac{(t-1)^4}{12} \\ &\quad + \frac{3}{2} \bar{y}(1, \alpha) \frac{(t-1)^3}{6} - \frac{1}{2} \bar{y}(1, \alpha) \frac{(t-1)^3}{6} + \frac{3}{4} \underline{y}(1, \alpha) \frac{(t-1)^5}{60} + \frac{1}{4} \underline{y}(1, \alpha) \frac{(t-1)^5}{60} \\ &\quad + \frac{3}{2} \underline{y}(1, \alpha) \frac{(t-1)^4}{24} - \frac{1}{2} \underline{y}(1, \alpha) \frac{(t-1)^4}{24} + \frac{3}{4} \bar{y}(1, \alpha) \frac{(t-1)^6}{360} + \frac{1}{4} \bar{y}(1, \alpha) \frac{(t-1)^6}{360} \\ &\quad + \frac{3}{2} \bar{y}(1, \alpha) \frac{(t-1)^5}{120} - \frac{1}{2} \bar{y}(1, \alpha) \frac{(t-1)^5}{120} + \frac{3}{4} \underline{y}(1, \alpha) \frac{(t-1)^7}{2520} + \frac{1}{4} \underline{y}(1, \alpha) \frac{(t-1)^7}{2520} \\ &\quad + \frac{3}{2} \underline{y}(1, \alpha) \frac{(t-1)^6}{720} - \frac{1}{2} \underline{y}(1, \alpha) \frac{(t-1)^6}{720} + \frac{3}{4} \bar{y}(1, \alpha) \frac{(t-1)^8}{20160} + \frac{1}{4} \bar{y}(1, \alpha) \frac{(t-1)^8}{20160} \\ \underline{\phi}_6(t) &= \frac{3}{2} \bar{y}(1, \alpha) + \frac{1}{2} \bar{y}(1, \alpha) - 2 \bar{y}(1, \alpha) + \frac{3}{2} \bar{y}(1, \alpha)(t-1) - \frac{1}{2} \bar{y}(1, \alpha)(t-1) \\ &\quad + \frac{3}{2} \bar{y}(1, \alpha) \frac{(t-1)^2}{2} + \frac{1}{2} \bar{y}(1, \alpha) \frac{(t-1)^2}{2} + \frac{3}{2} \bar{y}(1, \alpha) \frac{(t-1)^3}{6} - \frac{1}{2} \bar{y}(1, \alpha) \frac{(t-1)^3}{6} \end{aligned}$$

$$\begin{aligned}
& \frac{3}{2}\bar{y}(1, \alpha) \frac{(t-1)^4}{24} + \frac{1}{2}\bar{y}(1, \alpha) \frac{(t-1)^4}{24} + \frac{3}{2}\bar{y}(1, \alpha) \frac{(t-1)^5}{120} - \frac{1}{2}\bar{y}(1, \alpha) \frac{(t-1)^5}{120} \\
& \frac{3}{2}\bar{y}(1, \alpha) \frac{(t-1)^6}{720} + \frac{1}{2}\bar{y}(1, \alpha) \frac{(t-1)^6}{720} + \frac{3}{2}\underline{y}(1, \alpha) - \frac{1}{2}\bar{y}(1, \alpha) + \frac{3}{2}\underline{y}(1, \alpha)(t-1) \\
& + \frac{1}{2}\underline{y}(1, \alpha)(t-1) - 2\underline{y}(1, \alpha)(t-1) + \frac{3}{2}\underline{y}(1, \alpha) \frac{(t-1)^2}{2} - \frac{1}{2}\underline{y}(1, \alpha) \frac{(t-1)^2}{2} \\
& + \frac{3}{2}\underline{y}(1, \alpha) \frac{(t-1)^3}{6} + \frac{1}{2}\underline{y}(1, \alpha) \frac{(t-1)^3}{6} + \frac{3}{2}\underline{y}(1, \alpha) \frac{(t-1)^4}{24} \\
& - \frac{1}{2}\underline{y}(1, \alpha) \frac{(t-1)^4}{24} + \frac{3}{2}\underline{y}(1, \alpha) \frac{(t-1)^5}{120} + \frac{1}{2}\underline{y}(1, \alpha) \frac{(t-1)^5}{120} \\
& + \frac{3}{2}\underline{y}(1, \alpha) \frac{(t-1)^6}{720} - \frac{1}{2}\underline{y}(1, \alpha) \frac{(t-1)^6}{720} + \frac{3}{2}\underline{y}(1, \alpha) \frac{(t-1)^7}{5040} + \frac{1}{2}\underline{y}(1, \alpha) \frac{(t-1)^7}{5040}
\end{aligned}$$

$$\begin{aligned}
\underline{\Phi}_6(t) = & \frac{3}{2} \left( \bar{y}(1, \alpha) + \bar{y}(1, \alpha)(t-1) + \bar{y}(1, \alpha) \frac{(t-1)^2}{2} + \bar{y}(1, \alpha) \frac{(t-1)^3}{6} + \dots \right. \\
& + \bar{y}(1, \alpha) \frac{(t-1)^6}{720} + \underline{y}(1, \alpha) + \underline{y}(1, \alpha)(t-1) + \underline{y}(1, \alpha) \frac{(t-1)^2}{2} \\
& + \underline{y}(1, \alpha) \frac{(t-1)^3}{6} + \dots + \underline{y}(1, \alpha) \frac{(t-1)^6}{720} \Big) \\
& - \frac{1}{2} \left( \underline{y}(1, \alpha) + \underline{y}(1, \alpha)(1-t) + \underline{y}(1, \alpha) \frac{(1-t)^2}{2} + \underline{y}(1, \alpha) \frac{(1-t)^3}{6} + \dots \right. \\
& + \underline{y}(1, \alpha) \frac{(1-t)^6}{720} - \bar{y}(1, \alpha)(1-t) - \bar{y}(1, \alpha) \frac{(1-t)^2}{2} - \bar{y}(1, \alpha) \frac{(1-t)^3}{6} \\
& \left. - \dots - \bar{y}(1, \alpha) \frac{(t-1)^6}{720} \right) - 2\underline{y}(1, \alpha)(t-1) - 2\bar{y}(1, \alpha)
\end{aligned}$$

$$\begin{aligned}
\underline{\Phi}_6(t) = & \sum_{k=0}^6 \frac{3}{2} \left( \bar{y}(1, \alpha) \frac{(t-1)^k}{k!} + \underline{y}(1, \alpha) \frac{(t-1)^k}{k!} \right) \\
& - \frac{1}{2} \left( \underline{y}(1, \alpha) \frac{(1-t)^k}{k!} - \bar{y}(1, \alpha) \frac{(1-t)^k}{k!} \right) - 2\underline{y}(1, \alpha)(t-1) - 2\bar{y}(1, \alpha)
\end{aligned}$$

$$\begin{aligned}
\bar{\Phi}_6(t) = & \bar{y}(1, \alpha) + \underline{y}(1, \alpha)(t-1)^2 + \underline{y}(1, \alpha)(t-1) + \bar{y}(1, \alpha) \frac{(t-1)^3}{3} + \bar{y}(1, \alpha) \frac{(t-1)^2}{2} \\
& + \underline{y}(1, \alpha) \frac{(t-1)^4}{12} + \underline{y}(1, \alpha) \frac{(t-1)^3}{6} + \bar{y}(1, \alpha) \frac{(t-1)^5}{60} + \bar{y}(1, \alpha) \frac{(t-1)^4}{24} \\
& + \underline{y}(1, \alpha) \frac{(t-1)^6}{360} + \underline{y}(1, \alpha) \frac{(t-1)^5}{120} + \bar{y}(1, \alpha) \frac{(t-1)^7}{2520} + \bar{y}(1, \alpha) \frac{(t-1)^6}{720} \\
& + \underline{y}(1, \alpha) \frac{(t-1)^8}{20160}
\end{aligned}$$

$$\begin{aligned}\bar{\Phi}_6(t) = & \sum_{k=0}^6 \frac{3}{2} \left( \underline{y}(1, \alpha) \frac{(t-1)^k}{k!} + \bar{y}(1, \alpha) \frac{(t-1)^k}{k!} \right) \\ & + \frac{1}{2} \left( \bar{y}(1, \alpha) \frac{(1-t)^k}{k!} - \underline{y}(1, \alpha) \frac{(1-t)^k}{k!} \right) - 2\bar{y}(1, \alpha)(t-1) - 2\underline{y}(1, \alpha)\end{aligned}$$

To find exact solution

$$\begin{aligned}\underline{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \underline{\phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\Phi}_n(t) \\ \underline{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \sum_{k=0}^6 \frac{3}{2} \left( \bar{y}(1, \alpha) \frac{(t-1)^k}{k!} + \underline{y}(1, \alpha) \frac{(t-1)^k}{k!} \right) \\ &\quad - \frac{1}{2} \left( \underline{y}(1, \alpha) \frac{(1-t)^k}{k!} - \bar{y}(1, \alpha) \frac{(1-t)^k}{k!} \right) - 2\underline{y}(1, \alpha)(t-1) - 2\bar{y}(1, \alpha) \\ \underline{y}(t, \alpha) &= \frac{3}{2} \left( \bar{y}(1, \alpha) e^{t-1} + \underline{y}(1, \alpha) e^{t-1} \right) - \frac{1}{2} \left( \underline{y}(1, \alpha) e^{1-t} - \bar{y}(1, \alpha) e^{1-t} \right) \\ &\quad - 2\underline{y}(1, \alpha)(t-1) - 2\bar{y}(1, \alpha) \\ \underline{y}(t, \alpha) &= \frac{3}{2} e^{t-1} \left( \underline{y}(1, \alpha) + \bar{y}(1, \alpha) \right) + \frac{1}{2} e^{1-t} \left( \bar{y}(1, \alpha) - \underline{y}(1, \alpha) \right) - 2\underline{y}(1, \alpha)(t-1) \\ &\quad - 2\bar{y}(1, \alpha) \\ \bar{y}(t, \alpha) &= \frac{3}{2} e^{t-1} \left( \underline{y}(1, \alpha) + \bar{y}(1, \alpha) \right) - \frac{1}{2} e^{1-t} \left( \bar{y}(1, \alpha) - \underline{y}(1, \alpha) \right) - 2\bar{y}(1, \alpha)(t-1) \\ &\quad - 2\underline{y}(1, \alpha)\end{aligned}$$

when  $t \in [1.5, 2]$  we have

$$\begin{cases} \underline{y}_{n+1}(t, \alpha) = \int_{t_0}^t \bar{y}_n(s, \alpha) ds, & n \geq 0 \\ \bar{y}_{n+1}(t, \alpha) = \int_{t_0}^t \underline{y}_n(s, \alpha) ds, & n \geq 0 \end{cases}$$

where

$$\begin{aligned}\underline{y}_0(\alpha) &= \underline{y}(1.5, \alpha) \\ &= \frac{3}{2} e^{t-1} \left( \underline{y}(1, \alpha) + \bar{y}(1, \alpha) \right) + \frac{1}{2} e^{1-t} \left( \bar{y}(1, \alpha) - \underline{y}(1, \alpha) \right) - 2\underline{y}(1, \alpha)(t-1) \\ &\quad - 2\bar{y}(1, \alpha)\end{aligned}$$

$$\begin{aligned}
\bar{y}_0(\alpha) &= \bar{y}(1.5, \alpha) \\
&= \frac{3}{2} e^{t-1} \left( \underline{y}(1, \alpha) + \bar{y}(1, \alpha) \right) - \frac{1}{2} e^{1-t} \left( \bar{y}(1, \alpha) - \underline{y}(1, \alpha) \right) - 2\bar{y}(1, \alpha)(t-1) \\
&\quad - 2\underline{y}(1, \alpha) \\
\underline{y}_0(t, \alpha) &= \underline{y}_0(\alpha) + \int_{1.5}^t 2\bar{y}(1, \alpha)(2-s)ds \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) + 2\bar{y}(1, \alpha) \left( 2s - \frac{s^2}{2} \Big|_{1.5}^t \right) \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) + 2\bar{y}(1, \alpha) \left( 2t - \frac{t^2}{2} - 3 + \frac{9}{8} \right) \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) - 2\bar{y}(1, \alpha) \left( \frac{t^2}{2} - 2t + \frac{15}{8} \right) \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) - \bar{y}(1, \alpha) \left( t^2 - 4t + \frac{15}{4} \right) \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) - \bar{y}(1, \alpha) \left( t^2 - 3t + \frac{9}{4} + \frac{3}{2} - t \right) \\
\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) - \bar{y}(1, \alpha) \left( \left( t - \frac{3}{2} \right)^2 - \left( t - \frac{3}{2} \right) \right) \\
\bar{y}_0(t, \alpha) &= \bar{y}_0(\alpha) + \int_{1.5}^t 2\underline{y}(1, \alpha)(2-s)ds \\
\bar{y}_0(t, \alpha) &= \bar{y}(1.5, \alpha) - \underline{y}(1, \alpha) \left( \left( t - \frac{3}{2} \right)^2 - \left( t - \frac{3}{2} \right) \right)
\end{aligned}$$

We approximate  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\begin{aligned}
\underline{\phi}_6(t) &= \sum_{k=0}^6 \underline{y}_k(t, \alpha), \quad \bar{\phi}_6(t) = \sum_{k=0}^6 \bar{y}_k(t, \alpha) \\
\underline{y}_1(t, \alpha) &= \int_{1.5}^t \bar{y}_0(s, \alpha)ds, \quad \bar{y}_1(t, \alpha) = \int_{1.5}^t \underline{y}_0(s, \alpha)ds \\
\underline{y}_1(t, \alpha) &= \bar{y}(1.5, \alpha)(t-1.5) - \underline{y}(1, \alpha) \left( \frac{(t-1.5)^3}{3} - \frac{(t-1.5)^2}{2} \right) \\
\bar{y}_1(t, \alpha) &= \underline{y}(1.5, \alpha)(t-1.5) - \bar{y}(1, \alpha) \left( \frac{(t-1.5)^3}{3} - \frac{(t-1.5)^2}{2} \right) \\
\underline{y}_2(t, \alpha) &= \underline{y}(1.5, \alpha) \frac{(t-1.5)^2}{2} - \bar{y}(1, \alpha) \left( \frac{(t-1.5)^4}{12} - \frac{(t-1.5)^3}{6} \right)
\end{aligned}$$

$$\bar{y}_2(t, \alpha) = \bar{y}(1.5, \alpha) \frac{(t - 1.5)^2}{2} - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^4}{12} - \frac{(t - 1.5)^3}{6} \right)$$

$$\underline{y}_3(t, \alpha) = \bar{y}(1.5, \alpha) \frac{(t - 1.5)^3}{6} - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^5}{60} - \frac{(t - 1.5)^4}{24} \right)$$

$$\bar{y}_3(t, \alpha) = \underline{y}(1.5, \alpha) \frac{(t - 1.5)^3}{6} - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^5}{60} - \frac{(t - 1.5)^4}{24} \right)$$

⋮

$$\underline{y}_6(t, \alpha) = \underline{y}(1.5, \alpha) \frac{(t - 1.5)^6}{720} - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^8}{20160} - \frac{(t - 1.5)^7}{5040} \right)$$

$$\bar{y}_6(t, \alpha) = \bar{y}(1.5, \alpha) \frac{(t - 1.5)^6}{720} - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^8}{20160} - \frac{(t - 1.5)^7}{5040} \right)$$

then

$$\begin{aligned} \underline{\phi}_6(t) &= \underline{y}(1.5, \alpha) - \bar{y}(1, \alpha) ((t - 1.5)^2 - (t - 1.5)) + \bar{y}(1.5, \alpha)(t - 1.5) \\ &\quad - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^3}{3} - \frac{(t - 1.5)^2}{2} \right) + \underline{y}(1.5, \alpha) \frac{(t - 1.5)^2}{2} \\ &\quad - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^4}{12} - \frac{(t - 1.5)^3}{6} \right) + \bar{y}(1.5, \alpha) \frac{(t - 1.5)^3}{6} \\ &\quad - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^5}{60} - \frac{(t - 1.5)^4}{24} \right) + \underline{y}(1.5, \alpha) \frac{(t - 1.5)^4}{24} \\ &\quad - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^6}{360} - \frac{(t - 1.5)^5}{120} \right) + \bar{y}(1.5, \alpha) \frac{(t - 1.5)^5}{120} \\ &\quad - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^7}{2520} - \frac{(t - 1.5)^6}{720} \right) + \underline{y}(1.5, \alpha) \frac{(t - 1.5)^6}{720} \\ &\quad - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^8}{20160} - \frac{(t - 1.5)^7}{5040} \right) \end{aligned}$$

$$\begin{aligned} \underline{\phi}_6(t) &= \frac{1}{2} \underline{y}(1.5, \alpha) + \frac{1}{2} \underline{y}(1.5, \alpha) + \frac{1}{2} \underline{y}(1.5, \alpha)(t - 1.5) + \frac{1}{2} \underline{y}(1.5, \alpha)(1.5 - t) \\ &\quad + \frac{1}{2} \underline{y}(1.5, \alpha) \frac{(t - 1.5)^2}{2} + \frac{1}{2} \underline{y}(1.5, \alpha) \frac{(1.5 - t)^2}{2} + \frac{1}{2} \underline{y}(1.5, \alpha) \frac{(t - 1.5)^3}{6} \\ &\quad + \frac{1}{2} \underline{y}(1.5, \alpha) \frac{(1.5 - t)^3}{6} + \frac{1}{2} \underline{y}(1.5, \alpha) \frac{(t - 1.5)^4}{24} + \frac{1}{2} \underline{y}(1.5, \alpha) \frac{(1.5 - t)^4}{24} \end{aligned}$$

$$\begin{aligned}
& + \frac{1}{2} \underline{y}(1.5, \alpha) \frac{(t-1.5)^5}{120} + \frac{1}{2} \underline{y}(1.5, \alpha) \frac{(1.5-t)^5}{120} + \frac{1}{2} \underline{y}(1.5, \alpha) \frac{(t-1.5)^6}{720} \\
& + \frac{1}{2} \underline{y}(1.5, \alpha) \frac{(1.5-t)^6}{720} + \frac{1}{2} \bar{y}(1.5, \alpha) - \frac{1}{2} \bar{y}(1.5, \alpha) + \frac{1}{2} \bar{y}(1.5, \alpha)(t-1.5) \\
& - \frac{1}{2} \bar{y}(1.5, \alpha)(1.5-t) + \frac{1}{2} \bar{y}(1.5, \alpha) \frac{(t-1.5)^2}{2} - \frac{1}{2} \bar{y}(1.5, \alpha) \frac{(1.5-t)^2}{2} \\
& + \frac{1}{2} \bar{y}(1.5, \alpha) \frac{(t-1.5)^3}{6} - \frac{1}{2} \bar{y}(1.5, \alpha) \frac{(1.5-t)^3}{6} + \frac{1}{2} \bar{y}(1.5, \alpha) \frac{(t-1.5)^4}{24} \\
& - \frac{1}{2} \bar{y}(1.5, \alpha) \frac{(1.5-t)^4}{24} + \frac{1}{2} \bar{y}(1.5, \alpha) \frac{(t-1.5)^5}{120} - \frac{1}{2} \bar{y}(1.5, \alpha) \frac{(1.5-t)^5}{120} \\
& + \frac{1}{2} \bar{y}(1.5, \alpha) \frac{(t-1.5)^6}{720} - \frac{1}{2} \bar{y}(1.5, \alpha) \frac{(1.5-t)^6}{720} - \frac{1}{2} \bar{y}(1, \alpha) - \frac{3}{2} \bar{y}(1, \alpha) + 2\bar{y}(1, \alpha) \\
& - \frac{1}{2} \bar{y}(1, \alpha)(t-1.5) - \frac{3}{2} \bar{y}(1, \alpha)(1.5-t) - \frac{1}{2} \bar{y}(1, \alpha) \frac{(t-1.5)^2}{2} - \frac{3}{2} \bar{y}(1, \alpha) \frac{(1.5-t)^2}{2} \\
& - \frac{1}{2} \bar{y}(1, \alpha) \frac{(t-1.5)^3}{6} - \frac{3}{2} \bar{y}(1, \alpha) \frac{(1.5-t)^3}{6} - \frac{1}{2} \bar{y}(1, \alpha) \frac{(t-1.5)^4}{24} - \frac{3}{2} \bar{y}(1, \alpha) \frac{(1.5-t)^4}{24} \\
& - \dots - \frac{1}{2} \bar{y}(1, \alpha) \frac{(t-1.5)^8}{40320} - \frac{3}{2} \bar{y}(1, \alpha) \frac{(1.5-t)^8}{40320} \\
& - \frac{1}{2} \underline{y}(1, \alpha) + \frac{3}{2} \underline{y}(1, \alpha) - \underline{y}(1, \alpha) - \frac{1}{2} \underline{y}(1, \alpha)(t-1.5) + \frac{3}{2} \underline{y}(1, \alpha)(1.5-t) \\
& + 2\underline{y}(1, \alpha)(t-1.5) - \frac{1}{2} \underline{y}(1, \alpha) \frac{(t-1.5)^2}{2} + \frac{3}{2} \underline{y}(1, \alpha) \frac{(1.5-t)^2}{2} - \frac{1}{2} \underline{y}(1, \alpha) \frac{(t-1.5)^3}{6} \\
& + \frac{3}{2} \underline{y}(1, \alpha) \frac{(1.5-t)^3}{6} - \dots - \frac{1}{2} \underline{y}(1, \alpha) \frac{(t-1.5)^6}{720} + \frac{3}{2} \underline{y}(1, \alpha) \frac{(1.5-t)^6}{720} \\
\Phi_6(t) = & \sum_{k=0}^6 \frac{1}{2} \left( \underline{y}(1.5, \alpha) + \bar{y}(1.5, \alpha) - \underline{y}(1, \alpha) - \bar{y}(1, \alpha) \right) \frac{(t-1.5)^k}{k!} \\
& + \frac{1}{2} \left( \underline{y}(1.5, \alpha) - \bar{y}(1.5, \alpha) + 3\underline{y}(1, \alpha) - 3\bar{y}(1, \alpha) \right) \frac{(1.5-t)^k}{k!} \\
& + 2\underline{y}(1, \alpha)(t-2) + 2\bar{y}(1, \alpha)
\end{aligned}$$

$$\begin{aligned}
\bar{\Phi}_6(t) = & \bar{y}(1.5, \alpha) - \underline{y}(1, \alpha)((t-1.5)^2 - (t-1.5)) + \underline{y}(1.5, \alpha)(t-1.5) \\
& - \bar{y}(1, \alpha) \left( \frac{(t-1.5)^3}{3} - \frac{(t-1.5)^2}{2} \right) + \bar{y}(1.5, \alpha) \frac{(t-1.5)^2}{2}
\end{aligned}$$

$$\begin{aligned}
& -\underline{y}(1, \alpha) \left( \frac{(t-1.5)^4}{12} - \frac{(t-1.5)^3}{6} \right) + \underline{y}(1.5, \alpha) \frac{(t-1.5)^3}{6} \\
& -\bar{y}(1, \alpha) \left( \frac{(t-1.5)^5}{60} - \frac{(t-1.5)^4}{24} \right) + \bar{y}(1.5, \alpha) \frac{(t-1.5)^4}{24} \\
& -\underline{y}(1, \alpha) \left( \frac{(t-1.5)^6}{360} - \frac{(t-1.5)^5}{120} \right) + \underline{y}(1.5, \alpha) \frac{(t-1.5)^5}{120} \\
& -\bar{y}(1, \alpha) \left( \frac{(t-1.5)^7}{2520} - \frac{(t-1.5)^6}{720} \right) + \bar{y}(1.5, \alpha) \frac{(t-1.5)^6}{720} \\
& -\underline{y}(1, \alpha) \left( \frac{(t-1.5)^8}{20160} - \frac{(t-1.5)^7}{5040} \right)
\end{aligned}$$

$$\begin{aligned}
\bar{\Phi}_6(t) = & \sum_{k=0}^6 \frac{1}{2} \left( \underline{y}(1.5, \alpha) + \bar{y}(1.5, \alpha) - \underline{y}(1, \alpha) - \bar{y}(1, \alpha) \right) \frac{(t-1.5)^k}{k!} \\
& - \frac{1}{2} \left( \underline{y}(1.5, \alpha) - \bar{y}(1.5, \alpha) + 3\underline{y}(1, \alpha) - 3\bar{y}(1, \alpha) \right) \frac{(1.5-t)^k}{k!} \\
& + 2\bar{y}(1, \alpha)(t-2) + 2\underline{y}(1, \alpha)
\end{aligned}$$

To find exact solution

$$\begin{aligned}
\underline{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \underline{\phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\phi}_n(t) \\
\underline{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \sum_{k=0}^6 \frac{1}{2} \left( \underline{y}(1.5, \alpha) + \bar{y}(1.5, \alpha) - \underline{y}(1, \alpha) - \bar{y}(1, \alpha) \right) \frac{(t-1.5)^k}{k!} \\
& + \frac{1}{2} \left( \underline{y}(1.5, \alpha) - \bar{y}(1.5, \alpha) + 3\underline{y}(1, \alpha) - 3\bar{y}(1, \alpha) \right) \frac{(1.5-t)^k}{k!} \\
& + 2\underline{y}(1, \alpha)(t-2) + 2\bar{y}(1, \alpha) \\
\underline{y}(t, \alpha) &= \frac{1}{2} \left( \underline{y}(1.5, \alpha) + \bar{y}(1.5, \alpha) - \underline{y}(1, \alpha) - \bar{y}(1, \alpha) \right) e^{t-1.5} \\
& + \frac{1}{2} \left( \underline{y}(1.5, \alpha) - \bar{y}(1.5, \alpha) + 3\underline{y}(1, \alpha) - 3\bar{y}(1, \alpha) \right) e^{1.5-t} + 2\underline{y}(1, \alpha)(t-2) \\
& + 2\bar{y}(1, \alpha) \\
\bar{y}(t, \alpha) &= \frac{1}{2} \left( \underline{y}(1.5, \alpha) + \bar{y}(1.5, \alpha) - \underline{y}(1, \alpha) - \bar{y}(1, \alpha) \right) e^{t-1.5} \\
& - \frac{1}{2} \left( \underline{y}(1.5, \alpha) - \bar{y}(1.5, \alpha) + 3\underline{y}(1, \alpha) - 3\bar{y}(1, \alpha) \right) e^{1.5-t} + 2\bar{y}(1, \alpha)(t-2) \\
& + 2\underline{y}(1, \alpha)
\end{aligned}$$

b- Let  $y(0) = (0.25, 0.75, 4)$

$$y(0, \alpha) = [0.25 + 0.5\alpha, 4 - 3.25\alpha], \quad 0 \leq \alpha \leq 1$$

using 1 – differentiable

According to Equations (49), when  $t \in [0,1]$  we have

$$\underline{y}_0(t, \alpha) = \underline{y}_0(\alpha), \quad \bar{y}_0(t, \alpha) = \bar{y}_0(\alpha)$$

$$\underline{y}_0(t, \alpha) = 0.25 + 0.5\alpha, \quad \bar{y}_0(t, \alpha) = 4 - 3.25\alpha$$

We approximate  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\Phi}_6(t)$  and  $\bar{\Phi}_6(t)$ , respectively, as follows:

$$\underline{\Phi}_6(t) = \sum_{k=0}^6 \underline{y}_k(t, \alpha), \quad \bar{\Phi}_6(t) = \sum_{k=0}^6 \bar{y}_k(t, \alpha)$$

$$\underline{y}_1(t, \alpha) = \int_0^t \underline{y}_0(s, \alpha) ds, \quad \bar{y}_1(t, \alpha) = \int_0^t \bar{y}_0(s, \alpha) ds$$

$$\underline{y}_1(t, \alpha) = (0.25 + 0.5\alpha)t, \quad \bar{y}_1(t, \alpha) = (4 - 3.25\alpha)t$$

$$\underline{y}_2(t, \alpha) = (0.25 + 0.5\alpha) \frac{t^2}{2}, \quad \bar{y}_2(t, \alpha) = (4 - 3.25\alpha) \frac{t^2}{2}$$

⋮

$$\underline{y}_6(t, \alpha) = (0.25 + 0.5\alpha) \frac{t^6}{720}, \quad \bar{y}_6(t, \alpha) = (4 - 3.25\alpha) \frac{t^6}{720}$$

$$\underline{\Phi}_6(t) = \sum_{k=0}^6 \underline{y}_0(t, \alpha) \frac{t^k}{k!}, \quad \bar{\Phi}_6(t) = \sum_{k=0}^6 \bar{y}_0(t, \alpha) \frac{t^k}{k!}$$

To find exact solution

$$\underline{y}(t, \alpha) = \underline{y}_0(t, \alpha)e^t, \quad \bar{y}(t, \alpha) = \bar{y}_0(t, \alpha)e^t,$$

when  $t \in [1, 1.5]$  we have

$$y_0(\alpha) = y(1, \alpha) = [(0.25 + 0.5\alpha)e, (4 - 3.25\alpha)e]$$

$$\underline{y}_0(t, \alpha) = \underline{y}_0(\alpha) + \int_1^t 2\underline{y}(1, \alpha)(s-1) ds,$$

$$\bar{y}_0(t, \alpha) = \bar{y}_0(\alpha) + \int_1^t 2\bar{y}(1, \alpha)(s-1) ds$$

$$\underline{y}_0(t, \alpha) = \underline{y}(1, \alpha)((t-1)^2 + 1), \quad \bar{y}_0(t, \alpha) = \bar{y}(1, \alpha)((t-1)^2 + 1),$$

We approximate  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\begin{aligned}\underline{\phi}_6(t) &= \sum_{k=0}^6 \underline{y}_k(t, \alpha), & \bar{\phi}_6(t) &= \sum_{k=0}^6 \bar{y}_k(t, \alpha) \\ \underline{y}_1(t, \alpha) &= \int_1^t \underline{y}_0(s, \alpha) ds, & \bar{y}_1(t, \alpha) &= \int_1^t \bar{y}_0(s, \alpha) ds \\ \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^3}{3} + t - 1 \right), & \bar{y}_1(t, \alpha) &= \bar{y}(1, \alpha) \left( \frac{(t-1)^3}{3} + t - 1 \right) \\ \underline{y}_2(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^4}{12} + \frac{(t-1)^2}{2} \right), & \bar{y}_2(t, \alpha) &= \bar{y}(1, \alpha) \left( \frac{(t-1)^4}{12} + \frac{(t-1)^2}{2} \right) \\ &\vdots \\ \underline{y}_6(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^8}{20160} + \frac{(t-1)^6}{720} \right), & \bar{y}_6(t, \alpha) &= \bar{y}(1, \alpha) \left( \frac{(t-1)^8}{20160} + \frac{(t-1)^6}{720} \right)\end{aligned}$$

Then

$$\begin{aligned}\underline{\phi}_6(t) &= \sum_{k=0}^6 \underline{y}(1, \alpha) \left( \frac{2(t-1)^{k+2}}{(k+2)!} + \frac{(t-1)^k}{(k)!} \right) \\ \bar{\phi}_6(t) &= \sum_{k=0}^6 \bar{y}(1, \alpha) \left( \frac{2(t-1)^{k+2}}{(k+2)!} + \frac{(t-1)^k}{(k)!} \right)\end{aligned}$$

To find exact solution

$$\underline{y}(t, \alpha) = \underline{y}(1, \alpha)(3e^{t-1} - 2t)$$

$$\bar{y}(t, \alpha) = \bar{y}(1, \alpha)(3e^{t-1} - 2t)$$

when  $t \in [1.5, 2]$  we have

$$\begin{aligned}y_0(\alpha) &= y(1.5, \alpha) = [\underline{y}(1, \alpha)(3e^{0.5} - 3), \bar{y}(1, \alpha)(3e^{0.5} - 3)] \\ \underline{y}_0(t, \alpha) &= \underline{y}_0(\alpha) + \int_{1.5}^t 2\underline{y}(1, \alpha)(2-s) ds \\ \underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) - \underline{y}(1, \alpha)((t-1.5)^2 - (t-1.5)) \\ \bar{y}_0(t, \alpha) &= \bar{y}_0(\alpha) + \int_{1.5}^t 2\bar{y}(1, \alpha)(2-s) ds\end{aligned}$$

$$\bar{y}_0(t, \alpha) = \bar{y}(1.5, \alpha) - \bar{y}(1, \alpha)((t - 1.5)^2 - (t - 1.5))$$

We approximate  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\begin{aligned}\underline{\phi}_6(t) &= \sum_{k=0}^6 \underline{y}_k(t, \alpha), & \bar{\phi}_6(t) &= \sum_{k=0}^6 \bar{y}_k(t, \alpha) \\ \underline{y}_1(t, \alpha) &= \int_{1.5}^t \underline{y}_0(s, \alpha) ds, & \bar{y}_1(t, \alpha) &= \int_{1.5}^t \bar{y}_0(s, \alpha) ds \\ \underline{y}_1(t, \alpha) &= \int_{1.5}^t \left( \underline{y}(1.5, \alpha) - \underline{y}(1, \alpha)((s - 1.5)^2 - (s - 1.5)) \right) ds \\ \underline{y}_1(t, \alpha) &= \underline{y}(1.5, \alpha)(t - 1.5) - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^3}{3} - \frac{(t - 1.5)^2}{2} \right) \\ \bar{y}_1(t, \alpha) &= \bar{y}(1.5, \alpha)(t - 1.5) - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^3}{3} - \frac{(t - 1.5)^2}{2} \right) \\ \underline{y}_2(t, \alpha) &= \underline{y}(1.5, \alpha) \frac{(t - 1.5)^2}{2} - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^4}{12} - \frac{(t - 1.5)^3}{6} \right) \\ \bar{y}_2(t, \alpha) &= \bar{y}(1.5, \alpha) \frac{(t - 1.5)^2}{2} - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^4}{12} - \frac{(t - 1.5)^3}{6} \right) \\ &\vdots \\ \underline{y}_6(t, \alpha) &= \underline{y}(1.5, \alpha) \frac{(t - 1.5)^6}{720} - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^8}{20160} - \frac{(t - 1.5)^7}{5040} \right) \\ \bar{y}_6(t, \alpha) &= \bar{y}(1.5, \alpha) \frac{(t - 1.5)^6}{720} - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^8}{20160} - \frac{(t - 1.5)^7}{5040} \right)\end{aligned}$$

then

$$\begin{aligned}\underline{\phi}_6(t) &= \sum_{k=0}^6 \underline{y}(1.5, \alpha) \frac{(t - 1.5)^k}{k!} - \underline{y}(1, \alpha) \left( \frac{2(t - 1.5)^{k+2}}{(k + 2)!} - \frac{(t - 1.5)^{k+1}}{(k + 1)!} \right) \\ \bar{\phi}_6(t) &= \sum_{k=0}^6 \bar{y}(1.5, \alpha) \frac{(t - 1.5)^k}{k!} - \bar{y}(1, \alpha) \left( \frac{2(t - 1.5)^{k+2}}{(k + 2)!} - \frac{(t - 1.5)^{k+1}}{(k + 1)!} \right)\end{aligned}$$

To find exact solution

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{\phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\phi}_n(t)$$

$$\underline{y}(t, \alpha) = \bar{y}(1, \alpha)(2t - 2 + 3e^{t-1} - 4e^{t-1.5})$$

$$\bar{y}(t, \alpha) = \bar{y}(1, \alpha)(2t - 2 + 3e^{t-1} - 4e^{t-1.5})$$

### 3.7 Predictor-Corrector Method

In this section, we solve hybrid fuzzy differential equations (3) by Milne's Fourth Order Predictor-Corrector Method and Adams-Bashforth order four Predictor-Corrector Method.

#### 3.7.1 Milne's Four Step Method

This section will introduce how to solve hybrid fuzzy differential equations (29) by Milne's four step method.

Let the fuzzy initial values be

$$y(t_{i-1}), \quad y(t_i), \quad y(t_{i+1}), \quad y(t_{i+2}),$$

$$\text{i.e. } f(t_{i-1}, y(t_{i-1})), f(t_i, y(t_i)), f(t_{i+1}, y(t_{i+1})), f(t_{i+2}, y(t_{i+2})),$$

The predictor used the Lagrange polynomial approximation for  $f(t, y(t))$ . This process produced the Milne's four -step method, Therefore, the Milne's explicit four step method is obtained as follows:

$$\underline{y}^\alpha(t_{i+3}) = \underline{y}^\alpha(t_{i-1}) + \frac{4h}{3} \left( 2\underline{f}^\alpha(t_{i+2}, y(t_{i+2})) - \underline{f}^\alpha(t_{i+1}, y(t_{i+1})) + 2\underline{f}^\alpha(t_i, y(t_i)) \right)$$

$$\bar{y}^\alpha(t_{i+3}) = \bar{y}^\alpha(t_{i-1}) + \frac{4h}{3} \left( 2\bar{f}^\alpha(t_{i+2}, y(t_{i+2})) - \bar{f}^\alpha(t_{i+1}, y(t_{i+1})) + 2\bar{f}^\alpha(t_i, y(t_i)) \right)$$

The corrector is developed similarly, The Implicit three step method is obtained as follows:

$$\underline{y}^\alpha(t_{i+3}) = \underline{y}^\alpha(t_{i+1}) + \frac{h}{3} \left( \underline{f}^\alpha(t_{i+3}, y(t_{i+3})) + 4\underline{f}^\alpha(t_{i+2}, y(t_{i+2})) + \underline{f}^\alpha(t_{i+1}, y(t_{i+1})) \right)$$

$$\bar{y}^\alpha(t_{i+3}) = \bar{y}^\alpha(t_{i+1}) + \frac{h}{3} \left( \bar{f}^\alpha(t_{i+3}, y(t_{i+3})) + 4\bar{f}^\alpha(t_{i+2}, y(t_{i+2})) + \bar{f}^\alpha(t_{i+1}, y(t_{i+1})) \right)$$

## Predictor-corrector four step method

The following algorithm is based on Explicit four-step method as a predictor and also an iteration of implicit three-step method as a corrector.

### ALGORITHM:

By fix  $k \in \mathbb{Z}^+$ . To approximate the solution of the following hybrid fuzzy initial value problem.

$$\begin{cases} y'_k(t) = f(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)), & t_k \leq t \leq t_{k+1} \\ \underline{y}^\alpha(t_{k,i-1}) = \underline{\alpha}_0, & \underline{y}^\alpha(t_{k,i}) = \underline{\alpha}_1, & \underline{y}^\alpha(t_{k,i+1}) = \underline{\alpha}_2, & \underline{y}^\alpha(t_{k,i+2}) = \underline{\alpha}_3 \\ \bar{y}^\alpha(t_{k,i-1}) = \bar{\alpha}_0, & \bar{y}^\alpha(t_{k,i}) = \bar{\alpha}_1, & \bar{y}^\alpha(t_{k,i+1}) = \bar{\alpha}_2, & \bar{y}^\alpha(t_{k,i+2}) = \bar{\alpha}_3 \end{cases}$$

an arbitrary positive integer  $N_k$  is chosen.

Step 1: Let  $h = \frac{t_{k+1} - t_k}{N_k}$ ,

$$\begin{aligned} \underline{w}^\alpha(t_{k,0}) &= \underline{\alpha}_0, & \underline{w}^\alpha(t_{k,1}) &= \underline{\alpha}_1, & \underline{w}^\alpha(t_{k,2}) &= \underline{\alpha}_2, & \underline{w}^\alpha(t_{k,3}) &= \underline{\alpha}_3, \\ \bar{w}^\alpha(t_{k,0}) &= \bar{\alpha}_0, & \bar{w}^\alpha(t_{k,1}) &= \bar{\alpha}_1, & \bar{w}^\alpha(t_{k,2}) &= \bar{\alpha}_2, & \bar{w}^\alpha(t_{k,3}) &= \bar{\alpha}_3, \end{aligned}$$

Step 2: Let  $i = 1$ .

Step 3: Let

$$\begin{aligned} \underline{w}^{(0)\alpha}(t_{k,i+3}) &= \underline{w}^\alpha(t_{k,i-1}) \\ &+ \frac{4h}{3} \left[ 2\underline{f}^\alpha(t_{k,i+2}, w(t_{k,i+2}), \lambda_k(w_k)) - \underline{f}^\alpha(t_{k,i+1}, w(t_{k,i+1}), \lambda_k(w_k)) \right. \\ &\quad \left. + 2\underline{f}^\alpha(t_{k,i}, w(t_{k,i}), \lambda_k(w_k)) \right] \end{aligned}$$

$$\begin{aligned} \bar{w}^{(0)\alpha}(t_{k,i+3}) &= \bar{w}^\alpha(t_{k,i-1}) \\ &+ \frac{4h}{3} \left[ 2\bar{f}^\alpha(t_{k,i+2}, w(t_{k,i+2}), \lambda_k(w_k)) - \bar{f}^\alpha(t_{k,i+1}, w(t_{k,i+1}), \lambda_k(w_k)) \right. \\ &\quad \left. + 2\bar{f}^\alpha(t_{k,i}, w(t_{k,i}), \lambda_k(w_k)) \right] \end{aligned}$$

Step 4: Let  $t_{k,i+3} = t_{k,0} + (i + 3)h$

Step 5: Let

$$\begin{aligned} \underline{w}^\alpha(t_{k,i+3}) &= \underline{w}^\alpha(t_{k,i+1}) + \frac{h}{3} [\underline{f}^\alpha(t_{k,i+3}, w(t_{k,i+3}), \lambda_k(w_k)) \\ &+ 4\underline{f}^\alpha(t_{k,i+2}, w(t_{k,i+2}), \lambda_k(w_k)) + \underline{f}^\alpha(t_{k,i+1}, w(t_{k,i+1}), \lambda_k(w_k))] \end{aligned}$$

$$\begin{aligned}\bar{w}^\alpha(t_{k,i+3}) &= \bar{w}^\alpha(t_{k,i+1}) + \frac{h}{3} [\bar{f}^\alpha(t_{k,i+3}, w(t_{k,i+3}), \lambda_k(w_k)) \\ &\quad + 4\bar{f}^\alpha(t_{k,i+2}, w(t_{k,i+2}), \lambda_k(w_k)) + \bar{f}^\alpha(t_{k,i+1}, w(t_{k,i+1}), \lambda_k(w_k))]\end{aligned}$$

Step 6:  $i = i + 1$ .

Step 7: if  $i \leq N - 3$ , go to step 3.

Step 8: The algorithm ends, and  $(\underline{w}^\alpha(t_{k+1}), \bar{w}^\alpha(t_{k+1}))$  approximates the value of  $(\underline{Y}^\alpha(t_{k+1}), \bar{Y}^\alpha(t_{k+1}))$

### 3.7.2 Adams-Bashforth Four Step Method

The Adams-Bashforth explicit four step method is obtained as follows:

$$\begin{aligned}\underline{y}^\alpha(t_{i+3}) &= \underline{y}^\alpha(t_{i+2}) \\ &\quad + \frac{h}{24} (55\underline{f}^\alpha(t_{i+2}, y(t_{i+2})) - 59\underline{f}^\alpha(t_{i+1}, y(t_{i+1})) + 37\underline{f}^\alpha(t_i, y(t_i)) \\ &\quad - 9\underline{f}^\alpha(t_{i-1}, y(t_{i-1})))\end{aligned}$$

$$\begin{aligned}\bar{y}^\alpha(t_{i+3}) &= \bar{y}^\alpha(t_{i+2}) \\ &\quad + \frac{h}{24} (55\bar{f}^\alpha(t_{i+2}, y(t_{i+2})) - 59\bar{f}^\alpha(t_{i+1}, y(t_{i+1})) + 37\bar{f}^\alpha(t_i, y(t_i)) \\ &\quad - 9\bar{f}^\alpha(t_{i-1}, y(t_{i-1})))\end{aligned}$$

The corrector is developed similarly, The Implicit three step method is obtained as follows:

$$\begin{aligned}\underline{y}^\alpha(t_{i+3}) &= \underline{y}^\alpha(t_{i+2}) \\ &\quad + \frac{h}{24} (9\underline{f}^\alpha(t_{i+3}, y(t_{i+3})) + 19\underline{f}^\alpha(t_{i+2}, y(t_{i+2})) - 5\underline{f}^\alpha(t_{i+1}, y(t_{i+1})) \\ &\quad + \underline{f}^\alpha(t_i, y(t_i)))\end{aligned}$$

$$\begin{aligned}\bar{y}^\alpha(t_{i+3}) &= \bar{y}^\alpha(t_{i+2}) \\ &\quad + \frac{h}{24} (9\bar{f}^\alpha(t_{i+3}, y(t_{i+3})) + 19\bar{f}^\alpha(t_{i+2}, y(t_{i+2})) - 5\bar{f}^\alpha(t_{i+1}, y(t_{i+1})) \\ &\quad + \bar{f}^\alpha(t_i, y(t_i)))\end{aligned}$$

## ALGORITHM:

By fix  $k \in \mathbb{Z}^+$ . To approximate the solution of the following hybrid fuzzy initial value problem.

$$\begin{cases} y'_k(t) = f(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)), & t_k \leq t \leq t_{k+1} \\ \underline{y}^\alpha(t_{k,i-1}) = \underline{\alpha}_0, & \underline{y}^\alpha(t_{k,i}) = \underline{\alpha}_1, & \underline{y}^\alpha(t_{k,i+1}) = \underline{\alpha}_2, & \underline{y}^\alpha(t_{k,i+2}) = \underline{\alpha}_3 \\ \bar{y}^\alpha(t_{k,i-1}) = \bar{\alpha}_0, & \bar{y}^\alpha(t_{k,i}) = \bar{\alpha}_1, & \bar{y}^\alpha(t_{k,i+1}) = \bar{\alpha}_2, & \bar{y}^\alpha(t_{k,i+2}) = \bar{\alpha}_3 \end{cases}$$

an arbitrary positive integer  $N_k$  is chosen.

Step 1: Let  $h = \frac{t_{k+1} - t_k}{N_k}$ ,

$$\begin{aligned} \underline{w}^\alpha(t_{k,0}) &= \underline{\alpha}_0, & \underline{w}^\alpha(t_{k,1}) &= \underline{\alpha}_1, & \underline{w}^\alpha(t_{k,2}) &= \underline{\alpha}_2, & \underline{w}^\alpha(t_{k,3}) &= \underline{\alpha}_3, \\ \bar{w}^\alpha(t_{k,0}) &= \bar{\alpha}_0, & \bar{w}^\alpha(t_{k,1}) &= \bar{\alpha}_1, & \bar{w}^\alpha(t_{k,2}) &= \bar{\alpha}_2, & \bar{w}^\alpha(t_{k,3}) &= \bar{\alpha}_3, \end{aligned}$$

Step 2: Let  $i = 1$ .

Step 3: Let

$$\begin{aligned} \underline{w}^{(0)\alpha}(t_{k,i+3}) &= \underline{w}^\alpha(t_{k,i-1}) \\ &+ \frac{h}{24} [55\underline{f}^\alpha(t_{k,i+2}, w(t_{k,i+2}), \lambda_k(w_k)) - 59\underline{f}^\alpha(t_{k,i+1}, w(t_{k,i+1}), \lambda_k(w_k)) \\ &+ 37\underline{f}^\alpha(t_{k,i}, w(t_{k,i}), \lambda_k(w_k)) - 9\underline{f}^\alpha(t_{k,i-1}, w(t_{k,i-1}), \lambda_k(w_k))] \end{aligned}$$

$$\begin{aligned} \bar{w}^{(0)\alpha}(t_{k,i+3}) &= \bar{w}^\alpha(t_{k,i-1}) \\ &+ \frac{h}{24} [55\bar{f}^\alpha(t_{k,i+2}, w(t_{k,i+2}), \lambda_k(w_k)) - 59\bar{f}^\alpha(t_{k,i+1}, w(t_{k,i+1}), \lambda_k(w_k)) \\ &+ 37\bar{f}^\alpha(t_{k,i}, w(t_{k,i}), \lambda_k(w_k)) - 9\bar{f}^\alpha(t_{k,i-1}, w(t_{k,i-1}), \lambda_k(w_k))] \end{aligned}$$

Step 4: Let  $t_{k,i+3} = t_{k,0} + (i + 3)h$

Step 5: Let

$$\begin{aligned} \underline{w}^\alpha(t_{k,i+3}) &= \underline{w}^\alpha(t_{k,i+2}) + \frac{h}{24} [9\underline{f}^\alpha(t_{k,i+3}, w(t_{k,i+3}), \lambda_k(w_k)) \\ &+ 19\underline{f}^\alpha(t_{k,i+2}, w(t_{k,i+2}), \lambda_k(w_k)) - 5\underline{f}^\alpha(t_{k,i+1}, w(t_{k,i+1}), \lambda_k(w_k)) \\ &+ \underline{f}^\alpha(t_{k,i}, w(t_{k,i}), \lambda_k(w_k))] \end{aligned}$$

$$\bar{w}^\alpha(t_{k,i+3}) = \bar{w}^\alpha(t_{k,i+2}) + \frac{h}{24} [9\bar{f}^\alpha(t_{k,i+3}, w(t_{k,i+3}), \lambda_k(w_k))]$$

$$+19\bar{f}^\alpha(t_{k,i+2}, w(t_{k,i+2}), \lambda_k(w_k)) - 5\bar{f}^\alpha(t_{k,i+1}, w(t_{k,i+1}), \lambda_k(w_k)) \\ + \bar{f}^\alpha(t_{k,i}, w(t_{k,i}), \lambda_k(w_k))]$$

Step 6:  $i = i + 1$ .

Step 7: if  $i \leq N - 3$ , go to step 3.

Step 8: The algorithm ends, and  $(\underline{w}^\alpha(t_{k+1}), \bar{w}^\alpha(t_{k+1}))$  approximates the value of  $(\underline{Y}^\alpha(t_{k+1}), \bar{Y}^\alpha(t_{k+1}))$

**Theorem 8:** The Adams-Bashforth explicit four step method is stable method.

**Proof:** there exists only one characteristic polynomial  $\rho(\lambda) = \lambda^4 - \lambda^3$ , so it satisfies the root condition. And therefore, it is a stable method.

**Theorem 9:** The Adams-Bashforth implicit three step method.

**Proof:** there exists only one characteristic polynomial  $\rho(\lambda) = \lambda^4 - \lambda^3$ , so it satisfies the root condition. And therefore, it is a stable method.

**Theorem 10:** The Milne's explicit four step method is stable method.

**Proof:** there exists only one characteristic polynomial  $\rho(\lambda) = \lambda^4 - 1$ , so it satisfies the root condition. And therefore, it is a stable method.

**Theorem 11:** The Milne's implicit three step method.

**Proof:** there exists only one characteristic polynomial  $\rho(\lambda) = \lambda^4 - \lambda^2$ , so it satisfies the root condition. And therefore, it is a stable method.

To solve HFDEs (29) by Milne's Fourth Order Predictor-Corrector Method

Suppose given the IVP

$$\begin{cases} y(0, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha] \\ y(0.02, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha]e^{0.02} \\ y(0.04, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha]e^{0.04} \\ y(0.06, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha]e^{0.06} \end{cases} \quad (51)$$

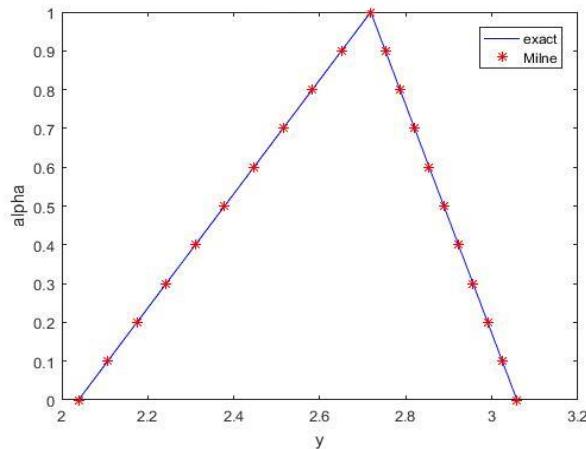
Let  $N = 100$  then  $h = 0.02$

When  $t \in [0,1]$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 3.7.1-3 and Figs 3.7.1-3 respectively, and the absolute errors of the approximate results in Tables 3.7.4-6.

**Table 3.7.1:** Numerical values for the exact and approximate solutions (Milne's) for  $t = 1$

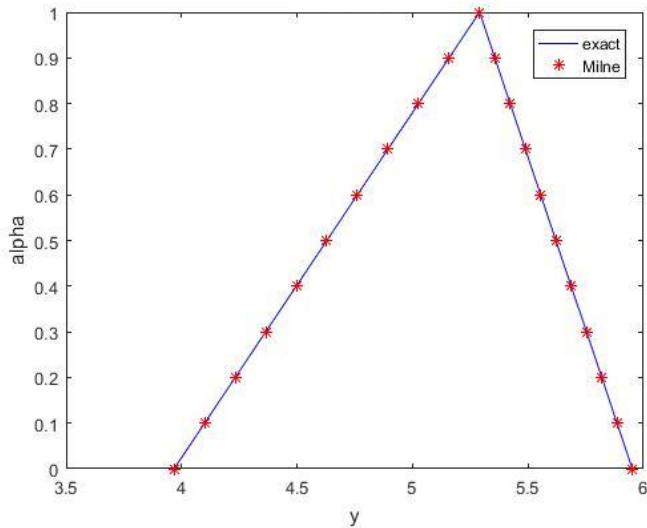
$\alpha$	Exact		Milne's of order four	
	$\underline{Y}(t, \alpha)$	$\bar{Y}(t, \alpha)$	$\underline{y}(t, \alpha)$	$\bar{y}(t, \alpha)$
0	2.038711371344284	3.058067057016426	2.038711372736861	3.058067059105292
0.1	2.106668417055760	3.024088534160689	2.106668418494757	3.024088536226344
0.2	2.174625462767236	2.990110011304950	2.174625464252652	2.990110013347397
0.3	2.242582508478713	2.956131488449212	2.242582510010547	2.956131490468449
0.4	2.310539554190189	2.922152965593474	2.310539555768443	2.922152967589501
0.5	2.378496599901665	2.888174442737736	2.378496601526338	2.888174444710553
0.6	2.446453645613141	2.854195919881998	2.446453647284233	2.854195921831606
0.7	2.514410691324617	2.820217397026260	2.514410693042129	2.820217398952658
0.8	2.582367737036093	2.786238874170521	2.582367738800024	2.786238876073711
0.9	2.650324782747569	2.752260351314784	2.650324784557919	2.752260353194763
1	2.718281828459046	2.718281828459046	2.718281830315815	2.718281830315815



**Figure 3.7.1:** Exact and Milne solutions for  $t = 1.5$

**Table 3.7.2:** Numerical values for the exact and approximate solutions (Milne's) for  $t = 1.5$

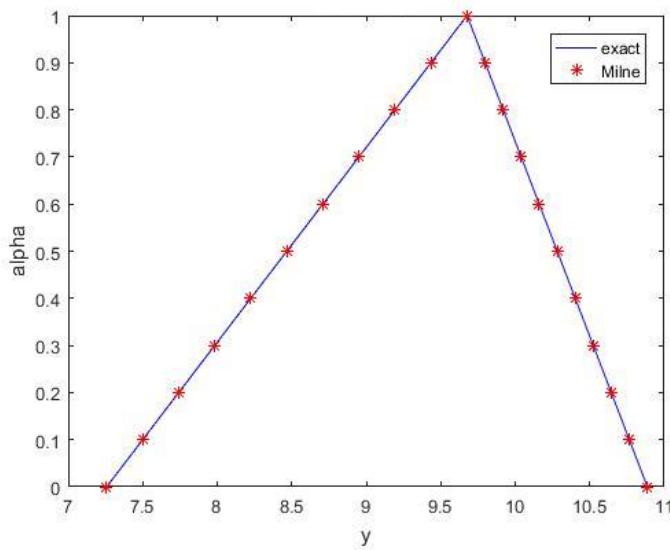
$\alpha$	Exact		Milne's of order four	
	$\underline{Y}(t, \alpha)$	$\bar{Y}(t, \alpha)$	$\underline{y}(t, \alpha)$	$\bar{y}(t, \alpha)$
0	3.967666294227795	5.951499441341692	3.967666299420739	5.951499449131109
0.1	4.099921837368721	5.885371669771230	4.099921842734765	5.885371677474096
0.2	4.232177380509648	5.819243898200766	4.232177386048789	5.819243905817085
0.3	4.364432923650575	5.753116126630302	4.364432929362812	5.753116134160072
0.4	4.496688466791501	5.686988355059839	4.496688472676839	5.686988362503059
0.5	4.628944009932427	5.620860583489376	4.628944015990862	5.620860590846046
0.6	4.761199553073354	5.554732811918913	4.761199559304886	5.554732819189035
0.7	4.893455096214280	5.488605040348450	4.893455102618911	5.488605047532023
0.8	5.025710639355206	5.422477268777985	5.025710645932936	5.422477275875011
0.9	5.157966182496133	5.356349497207523	5.157966189246960	5.356349504217998
1	5.290221725637060	5.290221725637060	5.290221732560985	5.290221732560985



**Figure 3.7.2:** Exact and Milne solutions for  $t = 1.5$

**Table 3.7.3:** Numerical values for the exact and approximate solutions (Milne's) for  $t = 2$

$\alpha$	Exact		Milne's of order four	
	$Y$	$\bar{Y}$	$y$	$\bar{y}$
0	7.257731754268338	10.88659763140251	7.257731764102271	10.88659764615341
0.1	7.499656146077282	10.76563543549804	7.499656156239015	10.76563545008503
0.2	7.741580537886227	10.64467323959356	7.741580548375756	10.64467325401667
0.3	7.983504929695171	10.52371104368909	7.983504940512497	10.52371105794829
0.4	8.225429321504116	10.40274884778462	8.225429332649243	10.40274886187992
0.5	8.467353713313061	10.28178665188015	8.467353724785982	10.28178666581155
0.6	8.709278105122007	10.16082445597567	8.709278116922723	10.16082446974318
0.7	8.951202496930950	10.03986226007120	8.951202509059467	10.03986227367481
0.8	9.193126888739894	9.918900064166728	9.193126901196209	9.918900077606438
0.9	9.435051280548839	9.797937868262256	9.435051293332952	9.797937881538067
1	9.676975672357784	9.676975672357784	9.676975685469694	9.676975685469694



**Figure 3.7.3:** Exact and Milne solutions for  $t = 2$

**Table 3.7.4:** The absolute errors of the Milne's method (1 – differentiable) for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.3926 \times 10^{-9}$	$2.0889 \times 10^{-9}$
0.1	$1.4390 \times 10^{-9}$	$2.0657 \times 10^{-9}$
0.2	$1.4854 \times 10^{-9}$	$2.0424 \times 10^{-9}$
0.3	$1.5318 \times 10^{-9}$	$2.0192 \times 10^{-9}$
0.4	$1.5783 \times 10^{-9}$	$1.9960 \times 10^{-9}$
0.5	$1.6247 \times 10^{-9}$	$1.9728 \times 10^{-9}$
0.6	$1.6711 \times 10^{-9}$	$1.9496 \times 10^{-9}$
0.7	$1.7175 \times 10^{-9}$	$1.9264 \times 10^{-9}$
0.8	$1.7639 \times 10^{-9}$	$1.9032 \times 10^{-9}$
0.9	$1.8104 \times 10^{-9}$	$1.8800 \times 10^{-9}$
1	$1.8568 \times 10^{-9}$	$1.8568 \times 10^{-9}$

**Table 3.7.5:** The absolute errors of the Milne's method (1 – differentiable) for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$5.1929 \times 10^{-9}$	$7.7894 \times 10^{-9}$
0.1	$5.3660 \times 10^{-9}$	$7.7029 \times 10^{-9}$
0.2	$5.5391 \times 10^{-9}$	$7.6163 \times 10^{-9}$
0.3	$5.7122 \times 10^{-9}$	$7.5298 \times 10^{-9}$
0.4	$5.8853 \times 10^{-9}$	$7.4432 \times 10^{-9}$
0.5	$6.0584 \times 10^{-9}$	$7.3567 \times 10^{-9}$
0.6	$6.2315 \times 10^{-9}$	$7.2701 \times 10^{-9}$
0.7	$6.4046 \times 10^{-9}$	$7.1836 \times 10^{-9}$
0.8	$6.5777 \times 10^{-9}$	$7.0970 \times 10^{-9}$
0.9	$6.7508 \times 10^{-9}$	$7.0105 \times 10^{-9}$
1	$6.9239 \times 10^{-9}$	$6.9239 \times 10^{-9}$

**Table 3.7.6:** The absolute errors of the Milne's method (1 – differentiable) for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$9.8339 \times 10^{-9}$	$1.4751 \times 10^{-8}$
0.1	$1.0162 \times 10^{-8}$	$1.4587 \times 10^{-8}$
0.2	$1.0490 \times 10^{-8}$	$1.4423 \times 10^{-8}$
0.3	$1.0817 \times 10^{-8}$	$1.4259 \times 10^{-8}$
0.4	$1.1145 \times 10^{-8}$	$1.4095 \times 10^{-8}$
0.5	$1.1473 \times 10^{-8}$	$1.3931 \times 10^{-8}$
0.6	$1.1801 \times 10^{-8}$	$1.3768 \times 10^{-8}$
0.7	$1.2129 \times 10^{-8}$	$1.3604 \times 10^{-8}$
0.8	$1.2456 \times 10^{-8}$	$1.3440 \times 10^{-8}$
0.9	$1.2784 \times 10^{-8}$	$1.3276 \times 10^{-8}$
1	$1.3112 \times 10^{-8}$	$1.3112 \times 10^{-8}$

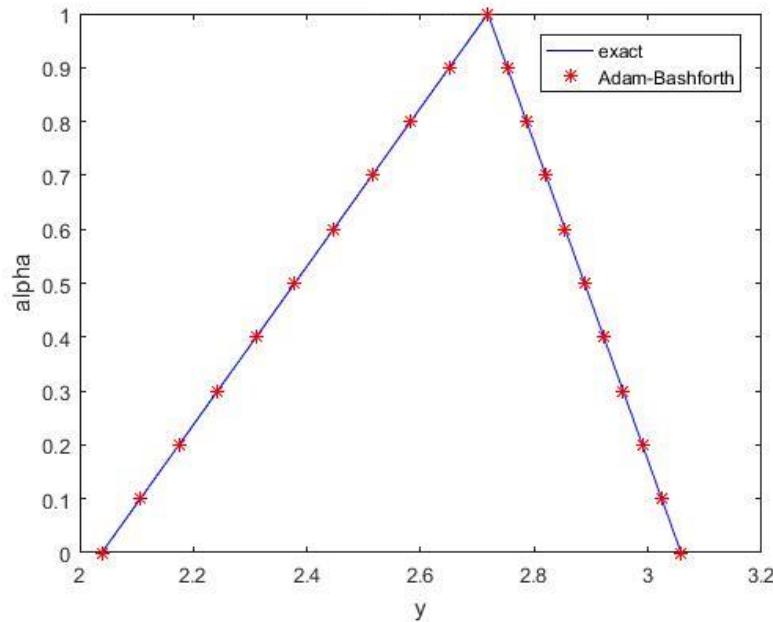
As shown in Tables 3.7.1-6, the PCM using Milne's 4<sup>th</sup> order obtained accurate results.

To solve HFDEs (29) by Adams-Bashforth four step method Predictor-Corrector Method. Let IVP (51),  $N = 100$  then  $h = 0.02$ .

By Matlab software the exact solutions with approximate results of this example are presented in Tables 3.7.7-9 and Figs 3.7.4-6 respectively., and the absolute errors of the approximate results in Tables 3.7.10-12.

**Table 3.7.7:** Numerical values for the exact and approximate solutions (Adams-Bashforth) for  $t = 1$

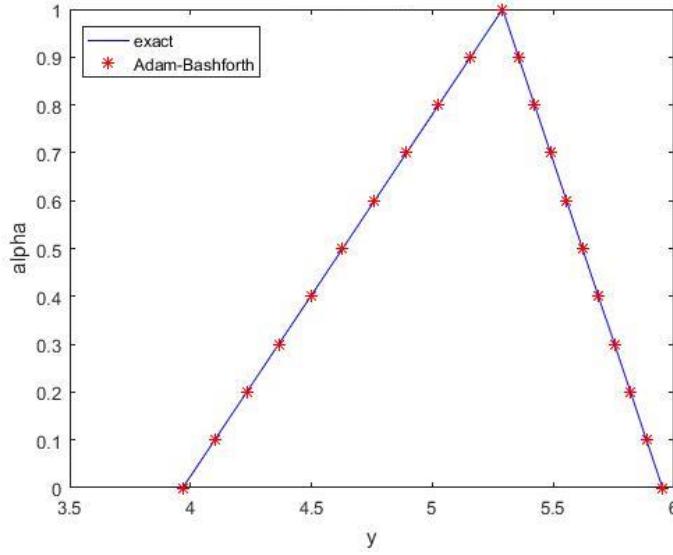
$\alpha$	Exact		Adams-Bashforth of order four	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.038711371344284	3.058067057016426	2.038711378478180	3.058067067717269
0.1	2.106668417055760	3.024088534160689	2.106668424427452	3.024088544742633
0.2	2.174625462767236	2.990110011304950	2.174625470376725	2.990110021767997
0.3	2.242582508478713	2.956131488449212	2.242582516325997	2.956131498793360
0.4	2.310539554190189	2.922152965593474	2.310539562275270	2.922152975818724
0.5	2.378496599901665	2.888174442737736	2.378496608224543	2.888174452844087
0.6	2.446453645613141	2.854195919881998	2.446453654173816	2.854195929869452
0.7	2.514410691324617	2.820217397026260	2.514410700123088	2.820217406894816
0.8	2.582367737036093	2.786238874170521	2.582367746072360	2.786238883920178
0.9	2.650324782747569	2.752260351314784	2.650324792021633	2.752260360945542
1	2.718281828459046	2.718281828459046	2.718281837970906	2.718281837970906



**Figure 3.7.4:** Exact and Adams-Bashforth solutions for  $t = 1$

**Table 3.7.8:** Numerical values for the exact and approximate solutions (Adams-Bashforth) for  $t = 1.5$

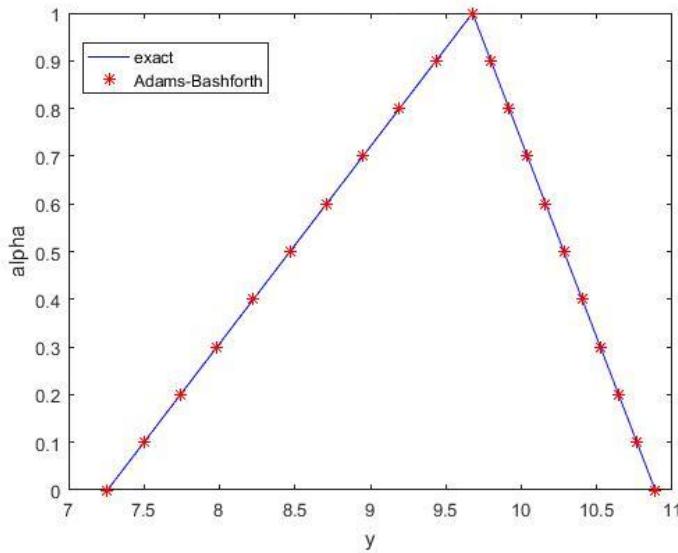
$\alpha$	Exact		Adams-Bashforth of order four	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	3.967666294227795	5.951499441341692	3.967666323833623	5.951499485750433
0.1	4.099921837368721	5.885371669771230	4.099921867961409	5.885371713686539
0.2	4.232177380509648	5.819243898200766	4.232177412089197	5.819243941622646
0.3	4.364432923650575	5.753116126630302	4.364432956216984	5.753116169558751
0.4	4.496688466791501	5.686988355059839	4.496688500344771	5.686988397494859
0.5	4.628944009932427	5.620860583489376	4.628944044472560	5.620860625430963
0.6	4.761199553073354	5.554732811918913	4.761199588600348	5.554732853367072
0.7	4.893455096214280	5.488605040348450	4.893455132728134	5.488605081303179
0.8	5.025710639355206	5.422477268777985	5.025710676855920	5.422477309239282
0.9	5.157966182496133	5.356349497207523	5.157966220983708	5.356349537175389
1	5.290221725637060	5.290221725637060	5.290221765111497	5.290221765111497



**Figure 3.7.5:** Exact and Adams-Bashforth solutions for  $t = 1.5$

**Table 3.7.9:** Numerical values for the exact and approximate solutions (Adams-Bashforth) for  $t = 2$

$\alpha$	Exact		Adams-Bashforth of order four	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	7.257731754268338	10.88659763140251	7.257731810544643	10.88659771581696
0.1	7.499656146077282	10.76563543549804	7.499656204229461	10.76563551897455
0.2	7.741580537886227	10.64467323959356	7.741580597914283	10.64467332213214
0.3	7.983504929695171	10.52371104368909	7.983504991599104	10.52371112528973
0.4	8.225429321504116	10.40274884778462	8.225429385283924	10.40274892844732
0.5	8.467353713313061	10.28178665188015	8.467353778968748	10.28178673160491
0.6	8.709278105122007	10.16082445597567	8.709278172653573	10.16082453476250
0.7	8.951202496930950	10.03986226007120	8.951202566338390	10.03986233792009
0.8	9.193126888739894	9.918900064166728	9.193126960023209	9.918900141077673
0.9	9.435051280548839	9.797937868262256	9.435051353708031	9.797937944235263
1	9.676975672357784	9.676975672357784	9.676975747392856	9.676975747392856



**Figure 3.7.6:** Exact and Adams-Bashforth solutions for  $t = 2$

**Table 3.7.10:** The absolute errors of the Adams-Bashforth method (1 – differentiable) for  $t = 1$

$\alpha$	Absolute error	
	$ Y - y  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$7.1339 \times 10^{-8}$	$1.0701 \times 10^{-8}$
0.1	$7.3717 \times 10^{-9}$	$1.0582 \times 10^{-8}$
0.2	$7.6095 \times 10^{-9}$	$1.0463 \times 10^{-8}$
0.3	$7.8473 \times 10^{-9}$	$1.0344 \times 10^{-8}$
0.4	$8.0851 \times 10^{-9}$	$1.0225 \times 10^{-8}$
0.5	$8.3229 \times 10^{-9}$	$1.0106 \times 10^{-8}$
0.6	$8.5607 \times 10^{-9}$	$9.9875 \times 10^{-9}$
0.7	$8.7985 \times 10^{-9}$	$9.8686 \times 10^{-9}$
0.8	$9.0363 \times 10^{-9}$	$9.7497 \times 10^{-9}$
0.9	$9.2741 \times 10^{-9}$	$9.6308 \times 10^{-9}$
1	$9.5119 \times 10^{-9}$	$9.5119 \times 10^{-9}$

**Table 3.7.11:** The absolute errors of the Adams-Bashforth method (1 – differentiable) for  $t = 1.5$

$\alpha$	Absolute error	
	$ Y - y  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$7.1339 \times 10^{-8}$	$1.0701 \times 10^{-8}$
0.1	$7.3717 \times 10^{-9}$	$1.0582 \times 10^{-8}$
0.2	$7.6095 \times 10^{-9}$	$1.0463 \times 10^{-8}$
0.3	$7.8473 \times 10^{-9}$	$1.0344 \times 10^{-8}$
0.4	$8.0851 \times 10^{-9}$	$1.0225 \times 10^{-8}$
0.5	$8.3229 \times 10^{-9}$	$1.0106 \times 10^{-8}$
0.6	$8.5607 \times 10^{-9}$	$9.9875 \times 10^{-9}$
0.7	$8.7985 \times 10^{-9}$	$9.8686 \times 10^{-9}$
0.8	$9.0363 \times 10^{-9}$	$9.7497 \times 10^{-9}$
0.9	$9.2741 \times 10^{-9}$	$9.6308 \times 10^{-9}$
1	$9.5119 \times 10^{-9}$	$9.5119 \times 10^{-9}$

**Table 3.7.12:** The absolute errors of the Adams-Bashforth method (1 – differentiable) for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$5.6276 \times 10^{-8}$	$8.4414 \times 10^{-8}$
0.1	$5.8152 \times 10^{-8}$	$8.3477 \times 10^{-8}$
0.2	$6.0028 \times 10^{-8}$	$8.2539 \times 10^{-8}$
0.3	$6.1904 \times 10^{-8}$	$8.1601 \times 10^{-8}$
0.4	$6.3780 \times 10^{-8}$	$8.0663 \times 10^{-8}$
0.5	$6.5656 \times 10^{-8}$	$7.9725 \times 10^{-8}$
0.6	$6.7532 \times 10^{-8}$	$7.8787 \times 10^{-8}$
0.7	$6.9407 \times 10^{-8}$	$7.7849 \times 10^{-8}$
0.8	$7.1283 \times 10^{-8}$	$7.6911 \times 10^{-8}$
0.9	$7.3159 \times 10^{-8}$	$7.5973 \times 10^{-8}$
1	$7.5035 \times 10^{-8}$	$7.5035 \times 10^{-8}$

As shown in Tables 3.7.7-12, the PCM using Adams-Bashforth 4<sup>th</sup> order gave accurate results.

### 3.8 Improved Predictor-Corrector (IPC) Method

In this section an improved predictor-corrector (IPC) method will be applied to solve hybrid fuzzy differential equations. The algorithm is illustrated by an example.

**Notation 2:** Triangular fuzzy numbers are those fuzzy sets in  $R_F$  which are characterized by an ordered triple  $(x^l, x^c, x^r) \in \mathbb{R}^3$  with  $x^l \leq x^c \leq x^r$  such that  $U_0 = [x^l, x^r]$  and  $U_1 = \{x^c\}$ , then

$$U_\alpha = [x^c - (1 - \alpha)(x^c - x^l), x^c + (1 - \alpha)(x^r - x^c)]$$

**Theorem 12 :** Let  $(t_i, U_i), i = 0, 1, 2, \dots, n$  be the observed data and suppose that each of  $U_i = (u_i^l, u_i^c, u_i^r)$  is an element of  $R_F$ . Then for each  $t \in [t_0, t_n]$ ,  $f(t) = (f^l(t), f^c(t), f^r(t)) \in R_F$ ,

$$f^l(t) = \sum_{l_i(t) \geq 0} l_i(t) u_i^l + \sum_{l_i(t) < 0} l_i(t) u_i^r,$$

$$f^c(t) = \sum_{i=0}^n l_i(t) u_i^c$$

$$f^r(t) = \sum_{l_i(t) \geq 0} l_i(t) u_i^r + \sum_{l_i(t) < 0} l_i(t) u_i^l,$$

### Explicit three-step method

For the hybrid fuzzy differential equation (3), implemented the explicit three-step method by an application of the explicit three-step method for fuzzy differential equation.

For fixed  $\alpha$ , replace each interval  $[t_k, t_{k+1}]$  by a set of  $N_{k+1}$  discrete equally spaced grid points,  $t_k = t_{k,0} < t_{k,1} < \dots < t_{k,N} = t_{k+1}$  (including the end points) at which the exact solution  $Y(t)$  is approximated by some  $y_k(t)$ . By fix  $k \in \mathbb{Z}^+$ . The hybrid fuzzy initial value problem (3) can be solved by explicit three-step method. Let the fuzzy initial values be  $y(t_{k,i-1}), y(t_{k,i}), y(t_{k,i+1})$ , that is

$$\begin{aligned} & f\left(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)\right), f\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right), f\left(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)\right), \\ & \left\{ f^l\left(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)\right), f^c\left(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)\right), f^r\left(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)\right) \right\} \\ & \left\{ f^l\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right), f^c\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right), f^r\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right) \right\} \\ & \left\{ f^l\left(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)\right), f^c\left(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)\right), f^r\left(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)\right) \right\} \end{aligned}$$

Also, integrating equation (3) from  $t_{k,i-1}$  to  $t_{k,i+2}$ , we get

$$y(t_{k,i+2}) = y(t_{k,i-1}) + \int_{t_{k,i-1}}^{t_{k,i+2}} f(t, y_k(t), \lambda_k(y_k)) dt. \quad (52)$$

By fuzzy linear spline interpolation for

$$\begin{aligned} & f\left(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)\right), f\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right), f\left(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)\right), \\ & f_1(t_k, y_k(t), \lambda_k(y_k)) = \frac{t_{k,i} - t}{t_{k,i} - t_{k,i-1}} f\left(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)\right) \\ & + \frac{t - t_{k,i-1}}{t_{k,i} - t_{k,i-1}} f\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right), t \in [t_{k,i-1}, t_{k,i}] \end{aligned}$$

$$f_2(t_k, y_k(t), \lambda_k(y_k)) = \frac{t_{k,i+1} - t}{t_{k,i} - t_{k,i-1}} f\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right)$$

$$+ \frac{t - t_{k,i}}{t_{k,i+1} - t_{k,i}} f\left(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)\right), t \in [t_{k,i}, t_{k,i+1}]$$

Thus for  $t \in [t_{k,i-1}, t_{k,i}]$  we have

$$\begin{cases} f_1^l(t_k, y_k(t), \lambda_k(y_k)) = \\ \frac{t_{k,i} - t}{t_{k,i} - t_{k,i-1}} f_1^l\left(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)\right) + \frac{t - t_{k,i-1}}{t_{k,i} - t_{k,i-1}} f_1^l\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right) \\ f_1^c(t_k, y_k(t), \lambda_k(y_k)) = \\ \frac{t_{k,i} - t}{t_{k,i} - t_{k,i-1}} f_1^c\left(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)\right) + \frac{t - t_{k,i-1}}{t_{k,i} - t_{k,i-1}} f_1^c\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right) \\ f_1^r(t_k, y_k(t), \lambda_k(y_k)) = \\ \frac{t_{k,i} - t}{t_{k,i} - t_{k,i-1}} f_1^r\left(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)\right) + \frac{t - t_{k,i-1}}{t_{k,i} - t_{k,i-1}} f_1^r\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right) \end{cases} \quad (53)$$

And for  $t \in [t_{k,i}, t_{k,i+1}]$  we have

$$\begin{cases} f_2^l(t_k, y_k(t), \lambda_k(y_k)) = \\ \frac{t_{k,i+1} - t}{t_{k,i} - t_{k,i-1}} f_2^l\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right) + \frac{t - t_{k,i}}{t_{k,i+1} - t_{k,i}} f_2^l\left(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)\right) \\ f_2^c(t_k, y_k(t), \lambda_k(y_k)) = \\ \frac{t_{k,i+1} - t}{t_{k,i} - t_{k,i-1}} f_2^c\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right) + \frac{t - t_{k,i}}{t_{k,i+1} - t_{k,i}} f_2^c\left(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)\right) \\ f_2^r(t_k, y_k(t), \lambda_k(y_k)) = \\ \frac{t_{k,i+1} - t}{t_{k,i} - t_{k,i-1}} f_2^r\left(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)\right) + \frac{t - t_{k,i}}{t_{k,i+1} - t_{k,i}} f_2^r\left(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)\right) \end{cases} \quad (54)$$

From equations (52) it follows that

$$y^\alpha(t_{k,i+2}) = [\underline{y}^\alpha(t_{k,i+2}), \bar{y}^\alpha(t_{k,i+2})]$$

Where

$$\begin{cases} \underline{y}^\alpha(t_{k,i+2}) = \underline{y}^\alpha(t_{k,i-1}) + \int_{t_{k,i-1}}^{t_{k,i}} \{\alpha f_1^c(t_k, y_k(t), \lambda_k(y_k)) + (1 - \alpha) f_1^l(t_k, y_k(t), \lambda_k(y_k))\} dt \\ + \int_{t_{k,i}}^{t_{k,i+2}} \{\alpha f_2^c(t_k, y_k(t), \lambda_k(y_k)) + (1 - \alpha) f_2^l(t_k, y_k(t), \lambda_k(y_k))\} dt \\ \bar{y}^\alpha(t_{k,i+2}) = \bar{y}^\alpha(t_{k,i-1}) + \int_{t_{k,i-1}}^{t_{k,i}} \{\alpha f_1^c(t_k, y_k(t), \lambda_k(y_k)) + (1 - \alpha) f_1^r(t_k, y_k(t), \lambda_k(y_k))\} dt \\ + \int_{t_{k,i}}^{t_{k,i+2}} \{\alpha f_2^c(t_k, y_k(t), \lambda_k(y_k)) + (1 - \alpha) f_2^r(t_k, y_k(t), \lambda_k(y_k))\} dt \end{cases} \quad (55)$$

Using equations (53) and (54) in (55) we get

$$\begin{aligned}
& \underline{y}^\alpha(t_{k,i+2}) = \underline{y}^\alpha(t_{k,i-1}) \\
& + \int_{t_{k,i-1}}^{t_{k,i}} \left\{ \alpha \left\{ \frac{t_{k,i} - t}{t_{k,i} - t_{k,i-1}} f_1^c(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)) + \frac{t - t_{k,i-1}}{t_{k,i} - t_{k,i-1}} f_1^c(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) \right\} \right. \\
& \quad + (1 - \alpha) \left\{ \frac{t_{k,i} - t}{t_{k,i} - t_{k,i-1}} f_1^l(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)) \right. \\
& \quad \left. \left. + \frac{t - t_{k,i-1}}{t_{k,i} - t_{k,i-1}} f_1^l(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) \right\} dt \right. \\
& + \int_{t_{k,i}}^{t_{k,i+2}} \left\{ \alpha \left\{ \frac{t_{k,i+1} - t}{t_{k,i} - t_{k,i-1}} f_2^c(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) + \frac{t - t_{k,i}}{t_{k,i+1} - t_{k,i}} f_2^c(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)) \right\} \right. \\
& \quad + (1 - \alpha) \left\{ \frac{t_{k,i+1} - t}{t_{k,i} - t_{k,i-1}} f_2^l(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) \right. \\
& \quad \left. \left. + \frac{t - t_{k,i}}{t_{k,i+1} - t_{k,i}} f_2^l(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)) \right\} dt \right.
\end{aligned}$$

By integrating we get the following result

$$\begin{cases} \underline{y}^\alpha(t_{k,i+2}) = \\ \underline{y}^\alpha(t_{k,i-1}) + \frac{h}{2} [\alpha f^c(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)) + (1 - \alpha) f^l(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k))] \\ \quad + \frac{h}{2} [\alpha f^c(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) + (1 - \alpha) f^l(t_{k,i}, y(t_{k,i}), \lambda_k(y_k))] \\ 2h [\alpha f^c(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)) + (1 - \alpha) f^l(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k))] \end{cases} \quad (56)$$

and similarly it can be deduced that

$$\begin{cases} \bar{y}^\alpha(t_{k,i+2}) = \\ \bar{y}^\alpha(t_{k,i-1}) + \frac{h}{2} [\alpha f^c(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)) + (1 - \alpha) f^l(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k))] \\ \quad + \frac{h}{2} [\alpha f^c(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) + (1 - \alpha) f^l(t_{k,i}, y(t_{k,i}), \lambda_k(y_k))] \\ 2h [\alpha f^c(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)) + (1 - \alpha) f^l(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k))] \end{cases} \quad (57)$$

and thus

$$\begin{aligned}
\underline{y}^\alpha(t_{k,i+2}) &= \underline{y}^\alpha(t_{k,i-1}) \\
&\quad + \frac{h}{2} [\underline{f}^\alpha(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)) + \underline{f}^\alpha(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) \\
&\quad + 4\underline{f}^\alpha(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k))]
\end{aligned} \tag{58}$$

$$\begin{aligned}
\bar{y}^\alpha(t_{k,i+2}) &= \bar{y}^\alpha(t_{k,i-1}) \\
&\quad + \frac{h}{2} [\bar{f}^\alpha(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)) + \bar{f}^\alpha(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) \\
&\quad + 4\bar{f}^\alpha(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k))]
\end{aligned} \tag{59}$$

Therefore, the explicit three-step method is as follows:

$$\left\{
\begin{aligned}
\underline{y}^\alpha(t_{k,i+2}) &= \underline{y}^\alpha(t_{k,i-1}) + \frac{h}{2} [\underline{f}^\alpha(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)) + \underline{f}^\alpha(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) \\
&\quad + 4\underline{f}^\alpha(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k))] \\
\bar{y}^\alpha(t_{k,i+2}) &= \bar{y}^\alpha(t_{k,i-1}) + \frac{h}{2} [\bar{f}^\alpha(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)) + \bar{f}^\alpha(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) \\
&\quad + 4\bar{f}^\alpha(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k))] \\
\underline{y}^\alpha(t_{k,i-1}) &= \alpha_0, & \underline{y}^\alpha(t_{k,i}) &= \alpha_1, & \underline{y}^\alpha(t_{k,i+1}) &= \alpha_2 \\
\bar{y}^\alpha(t_{k,i-1}) &= \alpha_3, & \bar{y}^\alpha(t_{k,i}) &= \alpha_4, & \bar{y}^\alpha(t_{k,i+1}) &= \alpha_5
\end{aligned} \right. \tag{60}$$

### Implicit two-step method

Fix  $k \in \mathbb{Z}^+$ . The hybrid fuzzy initial value problem (3) can be solved by implicit two-step method. Let the fuzzy initial values be  $y(t_{k,i-1})$ ,  $y(t_{k,i})$  subject to given:

$$f(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)), f(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)),$$

which are triangular fuzzy numbers and are shown as

$$\begin{aligned}
&\{f^l(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)), f^c(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)), f^r(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k))\} \\
&\{f^l(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)), f^c(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)), f^r(t_{k,i}, y(t_{k,i}), \lambda_k(y_k))\}
\end{aligned}$$

Consider the following hybrid fuzzy equation,

$$y(t_{k,i+1}) = y(t_{k,i-1}) + \int_{t_{k,i-1}}^{t_{k,i+1}} f(t, y_k(t), \lambda_k(y_k)) dt \quad (61)$$

By fuzzy linear spline interpolation for

$$f(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)), f(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)), f(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k)),$$

and from (61) we have

$$y^\alpha(t_{k,i+1}) = [\underline{y}^\alpha(t_{k,i+1}), \bar{y}^\alpha(t_{k,i+1})]$$

where

$$\begin{aligned} \underline{y}^\alpha(t_{k,i+1}) &= \underline{y}^\alpha(t_{k,i-1}) \\ &\quad + \int_{t_{k,i-1}}^{t_{k,i}} [\alpha f_1^c(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) + (1 - \alpha) f_1^l(t_{k,i}, y(t_{k,i}), \lambda_k(y_k))] dt \\ &\quad + \int_{t_{k,i}}^{t_{k,i+1}} [\alpha f_2^c(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) + (1 - \alpha) f_2^l(t_{k,i}, y(t_{k,i}), \lambda_k(y_k))] dt, \end{aligned} \quad (62)$$

$$\begin{aligned} \bar{y}^\alpha(t_{k,i+1}) &= \bar{y}^\alpha(t_{k,i-1}) \\ &\quad + \int_{t_{k,i-1}}^{t_{k,i}} [\alpha f_1^c(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) + (1 - \alpha) f_1^l(t_{k,i}, y(t_{k,i}), \lambda_k(y_k))] dt \\ &\quad + \int_{t_{k,i}}^{t_{k,i+1}} [\alpha f_2^c(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) + (1 - \alpha) f_2^l(t_{k,i}, y(t_{k,i}), \lambda_k(y_k))] dt, \end{aligned} \quad (63)$$

Using equations (53) and (54) in (62) and (63), by the Explicit three-step method we obtained the following implicit two-step method.

$$\begin{cases} \underline{y}^\alpha(t_{k,i+1}) = \underline{y}^\alpha(t_{k,i-1}) + \frac{h}{2} [\underline{f}^\alpha(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)) + 2\underline{f}^\alpha(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) \\ \quad + \underline{f}^\alpha(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k))] \\ \bar{y}^\alpha(t_{k,i+1}) = \bar{y}^\alpha(t_{k,i-1}) + \frac{h}{2} [\bar{f}^\alpha(t_{k,i-1}, y(t_{k,i-1}), \lambda_k(y_k)) + 2\bar{f}^\alpha(t_{k,i}, y(t_{k,i}), \lambda_k(y_k)) \\ \quad + \bar{f}^\alpha(t_{k,i+1}, y(t_{k,i+1}), \lambda_k(y_k))] \\ \underline{y}^\alpha(t_{k,i-1}) = \alpha_0, \quad \underline{y}^\alpha(t_{k,i}) = \alpha_1, \quad \bar{y}^\alpha(t_{k,i-1}) = \alpha_2, \quad \bar{y}^\alpha(t_{k,i}) = \alpha_3 \end{cases} \quad (64)$$

**Theorem 13 :** The explicit three- step method is stable.

The proof is elaborated in [35]

**Theorem 14 :** The implicit- two step method is stable.

The proof is elaborated in [35]

To solve HFDEs (29) by Milne's Fourth Order Predictor-Corrector Method

Let IVP

$$\begin{cases} y(0, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha] \\ y(0.02, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha]e^{0.02} \\ y(0.04, \alpha) = [0.75 + 0.25\alpha, 1.125 - 0.125\alpha]e^{0.04} \end{cases}$$

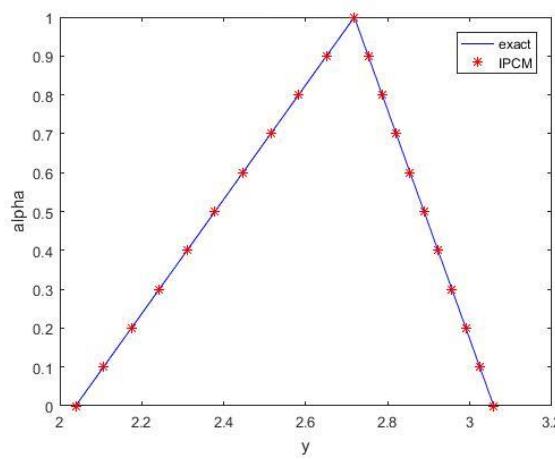
Let  $N = 100$  then  $h = 0.02$ .

When  $t \in [0,1]$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 3.8.1-3 and Figs 3.8.1-3 respectively, and the absolute errors of the approximate results in Tables 3.8.4-6.

**Table 3.8.1:** Numerical values for the exact and approximate solutions (IPCM) for  $t = 1$

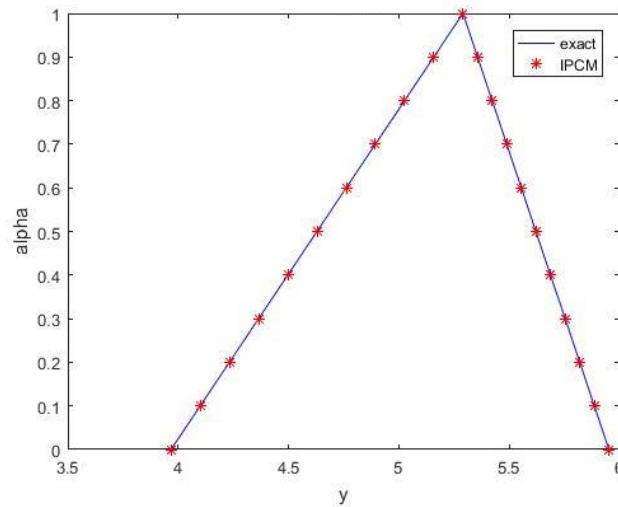
$\alpha$	Exact		IPCM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.038711371344284	3.058067057016426	2.038775193568358	3.058162790352537
0.1	2.106668417055760	3.024088534160689	2.106734366687303	3.024183203793064
0.2	2.174625462767236	2.990110011304950	2.174693539806248	2.990203617233592
0.3	2.242582508478713	2.956131488449212	2.242652712925193	2.956224030674118
0.4	2.310539554190189	2.922152965593474	2.310611886044139	2.922244444114646
0.5	2.378496599901665	2.888174442737736	2.378571059163084	2.888264857555173
0.6	2.446453645613141	2.854195919881998	2.446530232282029	2.854285270995701
0.7	2.514410691324617	2.820217397026260	2.514489405400975	2.820305684436228
0.8	2.582367737036093	2.786238874170521	2.582448578519919	2.786326097876755
0.9	2.650324782747569	2.752260351314784	2.650407751638865	2.752346511317283
1	2.718281828459046	2.718281828459046	2.718366924757810	2.718366924757810



**Figure 3.8.1:** Exact and IPCM solutions for  $t = 1$

**Table 3.8.2:** Numerical values for the exact and approximate solutions (IPCM) for  $t = 1.5$

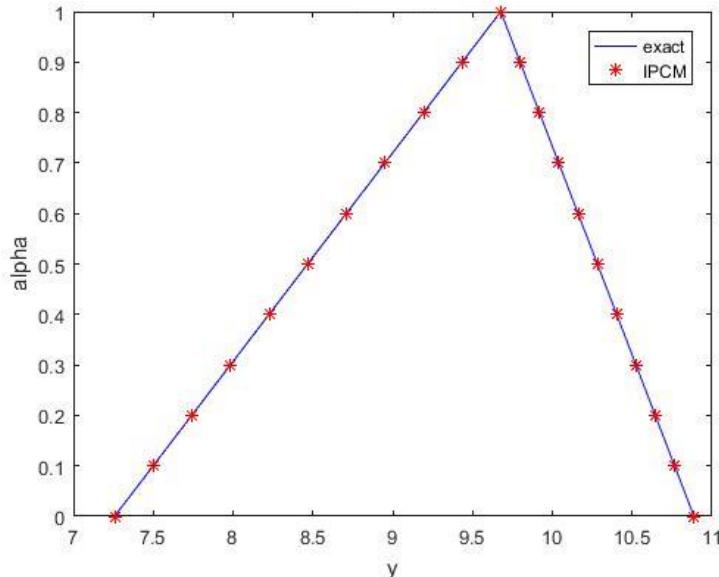
$\alpha$	Exact		IPCM	
	$Y$	$\bar{Y}$	$y$	$\bar{y}$
0	3.967666294227795	5.951499441341692	3.967945475290978	5.951918212936468
0.1	4.099921837368721	5.885371669771230	4.100210324467344	5.885785788348284
0.2	4.232177380509648	5.819243898200766	4.232475173643709	5.819653363760102
0.3	4.364432923650575	5.753116126630302	4.364740022820075	5.753520939171917
0.4	4.496688466791501	5.686988355059839	4.497004871996443	5.687388514583735
0.5	4.628944009932427	5.620860583489376	4.629269721172808	5.621256089995551
0.6	4.761199553073354	5.554732811918913	4.761534570349173	5.555123665407369
0.7	4.893455096214280	5.488605040348450	4.893799419525541	5.488991240819185
0.8	5.025710639355206	5.422477268777985	5.026064268701903	5.422858816231002
0.9	5.157966182496133	5.356349497207523	5.158329117878271	5.356726391642821
1	5.290221725637060	5.290221725637060	5.290593967054637	5.290593967054637



**Figure 3.8.2:** Exact and IPCM solutions for  $t = 1.5$

**Table 3.8.3:** Numerical values for the exact and approximate solutions (IPCM) for  $t = 2$

$\alpha$	Exact		IPCM	
	$Y$	$\bar{Y}$	$y$	$\bar{y}$
0	7.257731754268338	10.88659763140251	7.258263345818116	10.88739501872717
0.1	7.499656146077282	10.76563543549804	7.500205457345387	10.76642396296354
0.2	7.741580537886227	10.64467323959356	7.742147568872656	10.64545290719991
0.3	7.983504929695171	10.52371104368909	7.984089680399926	10.52448185143627
0.4	8.225429321504116	10.40274884778462	8.226031791927200	10.40351079567263
0.5	8.467353713313061	10.28178665188015	8.467973903454469	10.28253973990900
0.6	8.709278105122007	10.16082445597567	8.709916014981738	10.16156868414536
0.7	8.951202496930950	10.03986226007120	8.951858126509013	10.04059762838173
0.8	9.193126888739894	9.918900064166728	9.193800238036276	9.919626572618089
0.9	9.435051280548839	9.797937868262256	9.435742349563549	9.798655516854458
1	9.676975672357784	9.676975672357784	9.677684461090820	9.677684461090820



**Figure 3.7.3:** Exact and IPCM solutions for  $t = 2$

**Table 3.8.4:** The absolute errors of the IPCM (1 – differentiable) for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$6.3822 \times 10^{-5}$	$9.5733 \times 10^{-5}$
0.1	$6.5950 \times 10^{-5}$	$9.4670 \times 10^{-5}$
0.2	$6.8077 \times 10^{-5}$	$9.3606 \times 10^{-5}$
0.3	$7.0204 \times 10^{-5}$	$9.2542 \times 10^{-5}$
0.4	$7.2332 \times 10^{-5}$	$9.1479 \times 10^{-5}$
0.5	$7.4459 \times 10^{-5}$	$9.0415 \times 10^{-5}$
0.6	$7.6587 \times 10^{-5}$	$8.9351 \times 10^{-5}$
0.7	$7.8714 \times 10^{-5}$	$8.8287 \times 10^{-5}$
0.8	$8.0841 \times 10^{-5}$	$8.7224 \times 10^{-5}$
0.9	$8.2969 \times 10^{-5}$	$8.6160 \times 10^{-5}$
1	$8.5096 \times 10^{-5}$	$8.5096 \times 10^{-5}$

**Table 3.8.5:** The absolute errors of the IPCM (1 – differentiable) for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$2.7918 \times 10^{-4}$	$2.7918 \times 10^{-4}$
0.1	$2.8849 \times 10^{-4}$	$2.8849 \times 10^{-4}$
0.2	$2.9779 \times 10^{-4}$	$2.9779 \times 10^{-4}$
0.3	$3.0710 \times 10^{-4}$	$3.0710 \times 10^{-4}$
0.4	$3.1641 \times 10^{-4}$	$3.1641 \times 10^{-4}$
0.5	$3.2571 \times 10^{-4}$	$3.2571 \times 10^{-4}$
0.6	$3.3502 \times 10^{-4}$	$3.3502 \times 10^{-4}$
0.7	$3.4432 \times 10^{-4}$	$3.4432 \times 10^{-4}$
0.8	$3.5363 \times 10^{-4}$	$3.5363 \times 10^{-4}$
0.9	$3.6294 \times 10^{-4}$	$3.6294 \times 10^{-4}$
1	$3.7224 \times 10^{-4}$	$3.7224 \times 10^{-4}$

**Table 3.8.6:** The absolute errors of the IPCM (1-differentiable) for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$5.3159 \times 10^{-4}$	$7.9739 \times 10^{-4}$
0.1	$5.4931 \times 10^{-4}$	$7.8853 \times 10^{-4}$
0.2	$5.6703 \times 10^{-4}$	$7.7967 \times 10^{-4}$
0.3	$5.8475 \times 10^{-4}$	$7.7081 \times 10^{-4}$
0.4	$6.0247 \times 10^{-4}$	$7.6195 \times 10^{-4}$
0.5	$6.2019 \times 10^{-4}$	$7.5309 \times 10^{-4}$
0.6	$6.3791 \times 10^{-4}$	$7.4423 \times 10^{-4}$
0.7	$6.5563 \times 10^{-4}$	$7.3537 \times 10^{-4}$
0.8	$6.7335 \times 10^{-4}$	$7.2651 \times 10^{-4}$
0.9	$6.9107 \times 10^{-4}$	$7.1765 \times 10^{-4}$
1	$7.0879 \times 10^{-4}$	$7.0879 \times 10^{-4}$

As shown in Tables 3.8.1-6, the IPCM gave less accurate results than the other methods.

### 3.9 Summery

An example of the HFDEs was solved by several numerical methods using triangular fuzzy number as initial conditions. A Matlab code was constructed for each numerical method. The exact and approximate solutions were compared under generalized Hukuhara derivative. Finally, the results were compared for the used numerical methods.

## Chapter Four

### Numerical Solutions for HFDEs with Selected Types of Fuzzy Numbers as Initial Conditions

In this Chapter, the HFDEs (29) will be solved with different initial conditions as trapezoidal and triangular shaped fuzzy numbers by several numerical methods and Hukuhara derivative will be used to find exact solution and compare them with approximate solutions.

#### 4.1 Picard's Method

We will solve a hybrid fuzzy differential equations (29) by the Picard Method using  $N = 50$ .

a- Trapezoidal Fuzzy Number

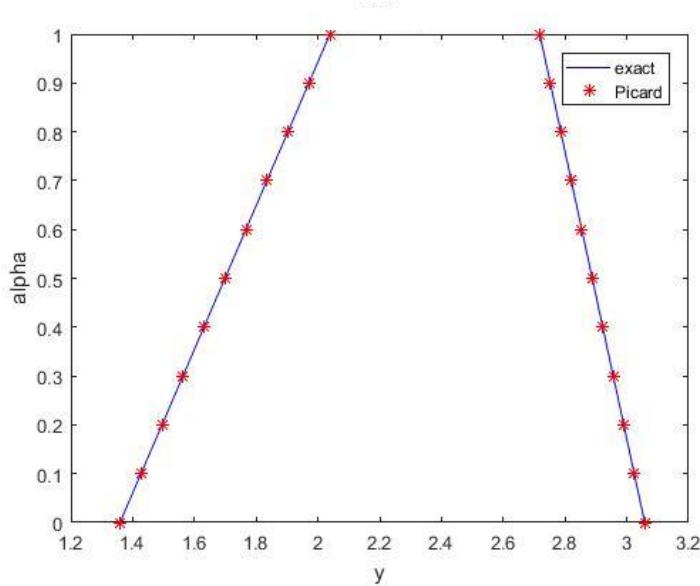
$$\text{let } y(0) = (0.5, 0.75, 1, 1.125)$$

$$y(0, \alpha) = [0.5 + 0.25\alpha, 1.125 - 0.125\alpha], \quad 0 \leq \alpha \leq 1$$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 4.1.1-3 and Figs 4.1.1-3 respectively, and the absolute errors of the approximate results in Tables 4.1.4-6.

**Table 4.1.1:** Numerical values for the exact and approximate solutions (Picard) for  $t = 1$

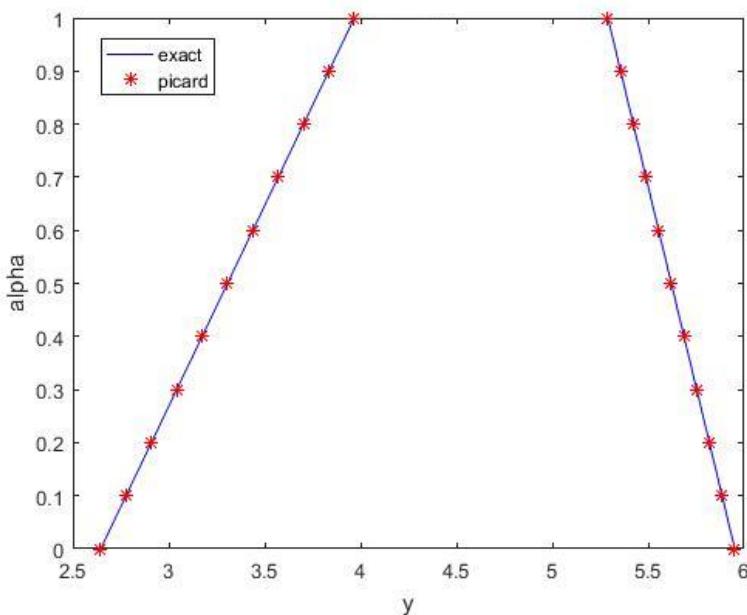
$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	1.359140914229523	3.058067057016426	1.359140914229523	3.058067057016426
0.1	1.427097959940999	3.024088534160689	1.427097959940999	3.024088534160688
0.2	1.495055005652475	2.990110011304950	1.495055005652475	2.990110011304950
0.3	1.563012051363951	2.956131488449212	1.563012051363951	2.956131488449211
0.4	1.630969097075427	2.922152965593474	1.630969097075427	2.922152965593474
0.5	1.698926142786903	2.888174442737736	1.698926142786903	2.888174442737736
0.6	1.766883188498380	2.854195919881998	1.766883188498379	2.854195919881997
0.7	1.834840234209856	2.820217397026260	1.834840234209856	2.820217397026260
0.8	1.902797279921332	2.786238874170521	1.902797279921332	2.786238874170521
0.9	1.970754325632808	2.752260351314784	1.970754325632808	2.752260351314783
1	2.038711371344284	2.718281828459046	2.038711371344284	2.718281828459045



**Figure 4.1.1:** Exact and Picard solutions for  $t = 1$

**Table 4.1.2:** Numerical values for the exact and approximate solutions (Picard) for  $t = 1.5$

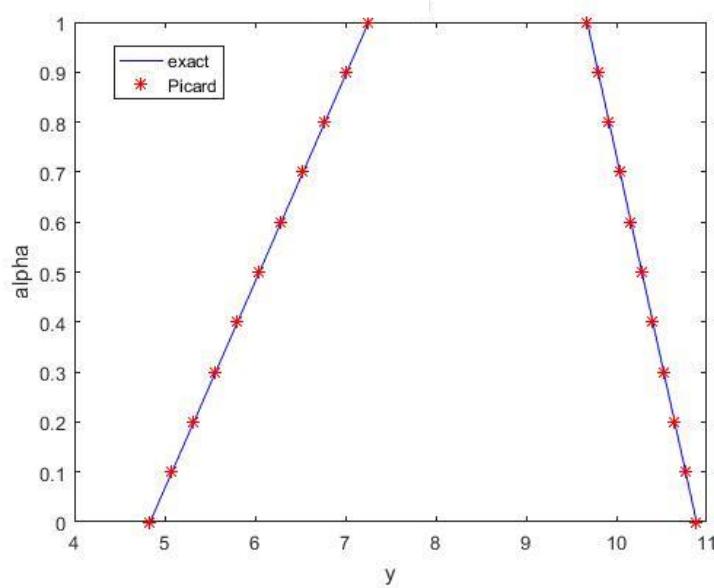
$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	2.645110862818530	5.951499441341692	2.645110862818530	5.951499441341692
0.1	2.777366405959457	5.885371669771230	2.777366405959456	5.885371669771228
0.2	2.909621949100383	5.819243898200766	2.909621949100383	5.819243898200766
0.3	3.041877492241309	5.753116126630302	3.041877492241309	5.753116126630300
0.4	3.174133035382236	5.686988355059839	3.174133035382234	5.686988355059840
0.5	3.306388578523162	5.620860583489376	3.306388578523161	5.620860583489376
0.6	3.438644121664089	5.554732811918913	3.438644121664088	5.554732811918911
0.7	3.570899664805015	5.488605040348450	3.570899664805015	5.488605040348450
0.8	3.703155207945942	5.422477268777985	3.703155207945942	5.422477268777984
0.9	3.835410751086868	5.356349497207523	3.835410751086868	5.356349497207522
1	3.967666294227795	5.290221725637060	3.967666294227794	5.290221725637058



**Figure 4.1.2:** Exact and Picard solutions for  $t = 1.5$

**Table 4.1.3:** Numerical values for the exact and approximate solutions (Picard) for  $t = 2$

$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	4.838487836178892	10.88659763140251	4.838487836178891	10.88659763140251
0.1	5.080412227987837	10.76563543549804	5.080412227987836	10.76563543549803
0.2	5.322336619796782	10.64467323959356	5.322336619796781	10.64467323959356
0.3	5.564261011605725	10.52371104368909	5.564261011605725	10.52371104368909
0.4	5.806185403414671	10.40274884778462	5.806185403414667	10.40274884778462
0.5	6.048109795223614	10.28178665188015	6.048109795223613	10.28178665188015
0.6	6.290034187032560	10.16082445597567	6.290034187032558	10.16082445597567
0.7	6.531958578841504	10.03986226007120	6.531958578841503	10.03986226007120
0.8	6.773882970650448	9.918900064166728	6.773882970650449	9.918900064166724
0.9	7.015807362459393	9.797937868262256	7.015807362459393	9.797937868262254
1	7.257731754268338	9.676975672357784	7.257731754268336	9.676975672357781



**Figure 4.1.3:** Exact and Picard solutions for  $t = 2$

**Table 4.1.4:** The absolute errors of the Picard method for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$10^{-16}$	$10^{-16}$
0.1	$10^{-16}$	$8.8818 \times 10^{-16}$
0.2	$10^{-16}$	$10^{-16}$
0.3	$10^{-16}$	$8.8818 \times 10^{-16}$
0.4	$10^{-16}$	$10^{-16}$
0.5	$10^{-16}$	$10^{-16}$
0.6	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.7	$10^{-16}$	$10^{-16}$
0.8	$10^{-16}$	$10^{-16}$
0.9	$10^{-16}$	$8.8818 \times 10^{-16}$
1	$10^{-16}$	$8.8818 \times 10^{-16}$

**Table 4.1.5:** The absolute errors of the Picard method for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$10^{-16}$	$10^{-16}$
0.1	$8.8818 \times 10^{-16}$	$1.7764 \times 10^{-15}$
0.2	$10^{-16}$	$10^{-16}$
0.3	$10^{-16}$	$1.7764 \times 10^{-15}$
0.4	$2.2204 \times 10^{-15}$	$8.8818 \times 10^{-16}$
0.5	$8.8818 \times 10^{-16}$	$10^{-16}$
0.6	$8.8818 \times 10^{-16}$	$1.7764 \times 10^{-15}$
0.7	$10^{-16}$	$10^{-16}$
0.8	$10^{-16}$	$1.7764 \times 10^{-15}$
0.9	$10^{-16}$	$8.8818 \times 10^{-16}$
1	$8.8818 \times 10^{-16}$	$1.7764 \times 10^{-15}$

**Table 4.1.6:** The absolute errors of the Picard method for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$8.8818 \times 10^{-16}$	$10^{-16}$
0.1	$1.7764 \times 10^{-15}$	$1.0658 \times 10^{-14}$
0.2	$8.8818 \times 10^{-16}$	$10^{-16}$
0.3	$10^{-16}$	$10^{-16}$
0.4	$3.5527 \times 10^{-15}$	$10^{-16}$
0.5	$8.8818 \times 10^{-16}$	$10^{-16}$
0.6	$1.7764 \times 10^{-15}$	$10^{-16}$
0.7	$8.8818 \times 10^{-16}$	$10^{-16}$
0.8	$8.8818 \times 10^{-16}$	$3.5527 \times 10^{-15}$
0.9	$10^{-16}$	$1.7764 \times 10^{-15}$
1	$1.7764 \times 10^{-15}$	$3.5527 \times 10^{-15}$

As shown in Tables 4.1.1-6 , the Picard method with trapezoidal fuzzy number as initial condition gave high accurate results when used with high number of iteration.

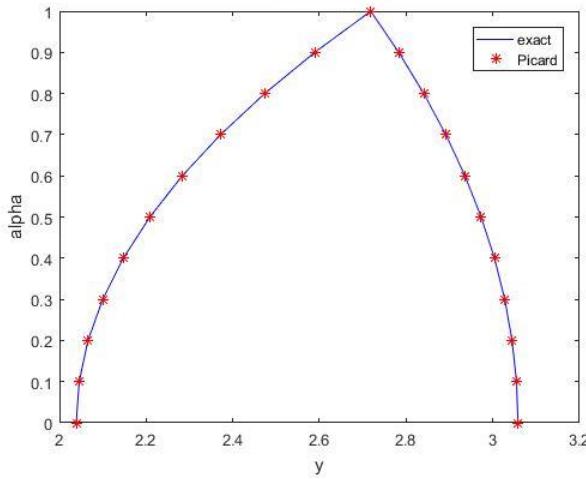
### b- Triangular Shaped Fuzzy Number

$$\text{let } y(0, \alpha) = [0.75 + 0.25\alpha^2, 1.125 - 0.125\alpha^2], 0 \leq \alpha \leq 1$$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 4.1.7-9 and Figs 4.1.4-6 respectively, and the absolute errors of the approximate results in Tables 4.1.10-12.

**Table 4.1.7:** Numerical values for the exact and approximate solutions (Picard) for  $t = 1$

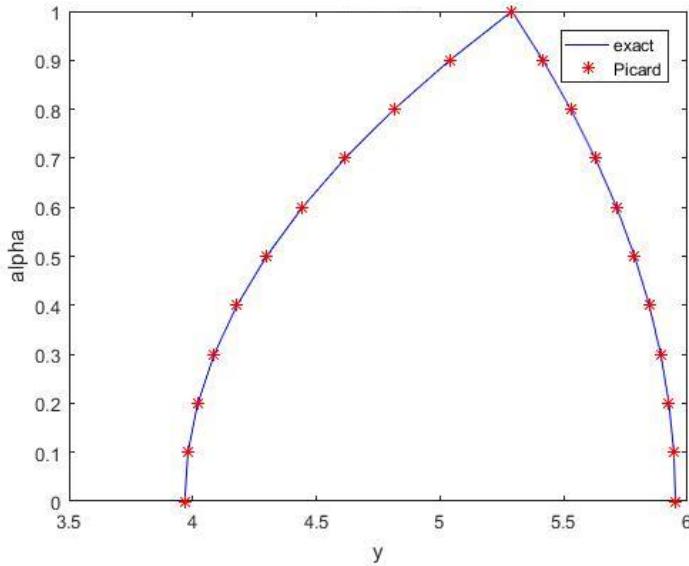
$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.038711371344284	3.058067057016426	2.038711371344284	3.058067057016426
0.1	2.045507075915432	3.054669204730852	2.045507075915432	3.054669204730852
0.2	2.065894189628875	3.044475647874132	2.065894189628875	3.044475647874131
0.3	2.099872712484613	3.027486386446262	2.099872712484613	3.027486386446262
0.4	2.147442644482646	3.003701420447245	2.147442644482646	3.003701420447245
0.5	2.208603985622974	2.973120749877081	2.208603985622974	2.973120749877081
0.6	2.283356735905599	2.935744374735769	2.283356735905598	2.935744374735769
0.7	2.371700895330517	2.891572295023310	2.371700895330517	2.891572295023309
0.8	2.473636463897732	2.840604510739702	2.473636463897731	2.840604510739702
0.9	2.589163441607241	2.782841021884948	2.589163441607241	2.782841021884948
1	2.718281828459046	2.718281828459046	2.718281828459045	2.718281828459045



**Figure 4.1.4:** Exact and Picard solutions for  $t = 1$

**Table 4.1.8:** Numerical values for the exact and approximate solutions (Picard) for  $t = 1.5$

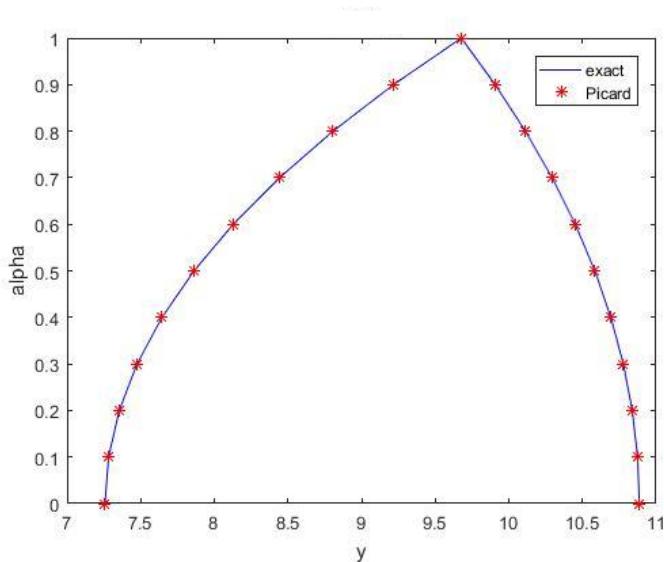
$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	3.967666294227795	5.951499441341692	3.967666294227794	5.951499441341692
0.1	3.980891848541887	5.944886664184645	3.980891848541888	5.944886664184645
0.2	4.020568511484165	5.925048332713508	4.020568511484165	5.925048332713507
0.3	4.086696283054629	5.891984446928276	4.086696283054629	5.891984446928275
0.4	4.179275163253278	5.845695006828951	4.179275163253277	5.845695006828951
0.5	4.298305152080111	5.786180012415534	4.298305152080110	5.786180012415533
0.6	4.443786249535131	5.713439463688025	4.443786249535130	5.713439463688024
0.7	4.615718455618335	5.627473360646422	4.615718455618334	5.627473360646421
0.8	4.814101770329725	5.528281703290727	4.814101770329724	5.528281703290726
0.9	5.038936193669300	5.415864491620940	5.038936193669300	5.415864491620940
1	5.290221725637060	5.290221725637060	5.290221725637058	5.290221725637058



**Figure 4.1.5:** Exact and Picard solutions for  $t = 1.5$

**Table 4.1.9:** Numerical values for the exact and approximate solutions (Picard) for  $t = 2$

$\alpha$	Exact		Picard	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	7.257731754268338	10.88659763140251	7.257731754268336	10.88659763140251
0.1	7.281924193449231	10.87450141181206	7.281924193449234	10.87450141181206
0.2	7.354501510991915	10.83821275304072	7.354501510991916	10.83821275304072
0.3	7.475463706896387	10.77773165508848	7.475463706896390	10.77773165508848
0.4	7.644810781162650	10.69305811795535	7.644810781162649	10.69305811795535
0.5	7.862542733790699	10.58419214164133	7.862542733790698	10.58419214164132
0.6	8.128659564780540	10.45113372614641	8.128659564780538	10.45113372614641
0.7	8.443161274132168	10.29388287147059	8.443161274132166	10.29388287147059
0.8	8.806047861845585	10.11243957761388	8.806047861845583	10.11243957761388
0.9	9.217319327920789	9.906803844576279	9.217319327920791	9.906803844576283
1	9.676975672357784	9.676975672357784	9.676975672357781	9.676975672357781



**Figure 4.1.6:** Exact and Picard solutions for  $t = 2$

**Table 4.1.10:** The absolute errors of the Picard method for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$10^{-16}$	$10^{-16}$
0.1	$10^{-16}$	$10^{-16}$
0.2	$10^{-16}$	$8.8818 \times 10^{-16}$
0.3	$10^{-16}$	$10^{-16}$
0.4	$10^{-16}$	$10^{-16}$
0.5	$10^{-16}$	$10^{-16}$
0.6	$8.8818 \times 10^{-16}$	$10^{-16}$
0.7	$10^{-16}$	$8.8818 \times 10^{-16}$
0.8	$1.3323 \times 10^{-15}$	$10^{-16}$
0.9	$10^{-16}$	$10^{-16}$
1	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$

**Table 4.1.11:** The absolute errors of the Picard method for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$8.8818 \times 10^{-16}$	$10^{-16}$
0.1	$8.8818 \times 10^{-16}$	$10^{-16}$
0.2	$10^{-16}$	$8.8818 \times 10^{-16}$
0.3	$10^{-16}$	$8.8818 \times 10^{-16}$
0.4	$8.8818 \times 10^{-16}$	$10^{-16}$
0.5	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.6	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.7	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.8	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.9	$10^{-16}$	$10^{-16}$
1	$1.7764 \times 10^{-15}$	$1.7764 \times 10^{-15}$

**Table 4.1.12:** The absolute errors of the Picard method for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.7764 \times 10^{-15}$	$10^{-16}$
0.1	$2.6645 \times 10^{-15}$	$10^{-16}$
0.2	$8.8818 \times 10^{-16}$	$10^{-16}$
0.3	$2.6645 \times 10^{-15}$	$10^{-16}$
0.4	$8.8818 \times 10^{-16}$	$10^{-16}$
0.5	$8.8818 \times 10^{-16}$	$8.8818 \times 10^{-16}$
0.6	$1.7764 \times 10^{-15}$	$10^{-16}$
0.7	$1.7764 \times 10^{-15}$	$10^{-16}$
0.8	$1.7764 \times 10^{-15}$	$10^{-16}$
0.9	$1.7764 \times 10^{-15}$	$3.5527 \times 10^{-15}$
1	$3.5527 \times 10^{-15}$	$3.5527 \times 10^{-15}$

As shown in Tables 4.1.7-12 , the Picard method with triangular shaped fuzzy number as initial condition gave high accurate results when used with high number of iteration.

## 4.2 Runge-Kutta of Order Five

We will apply the Runge-Kutta of order five for hybrid fuzzy differential equations (29). Let  $N = 100$  When  $t \in [0,2]$

a- Trapezoidal Fuzzy Number

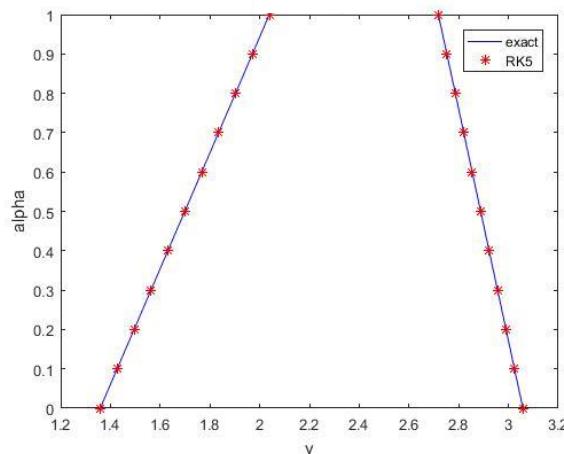
$$\text{let } y(0) = (0.5, 0.75, 1, 1.125)$$

$$y(0, \alpha) = [0.5 + 0.25\alpha, 1.125 - 0.125\alpha], \quad 0 \leq \alpha \leq 1$$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 4.2.1-3 and Figs 4.2.1-3 respectively, and the absolute errors of the approximate results in Tables 4.2.4-6.

**Table 4.2.1:** Numerical values for the exact and approximate solutions (Runge-Kutta) for  $t = 1$

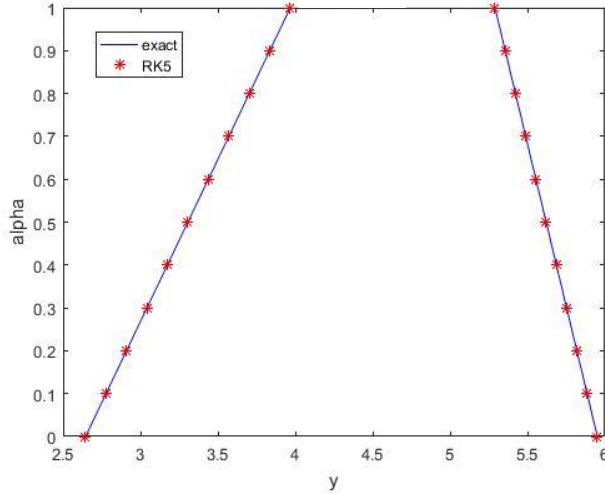
$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	1.359140914229523	3.058067057016426	1.359140913927534	3.058067056336951
0.1	1.427097959940999	3.024088534160689	1.427097959623911	3.024088533488763
0.2	1.495055005652475	2.990110011304950	1.495055005320287	2.990110010640575
0.3	1.563012051363951	2.956131488449212	1.563012051016664	2.956131487792387
0.4	1.630969097075427	2.922152965593474	1.630969096713041	2.922152964944198
0.5	1.698926142786903	2.888174442737736	1.698926142409418	2.888174442096010
0.6	1.766883188498380	2.854195919881998	1.766883188105794	2.854195919247821
0.7	1.834840234209856	2.820217397026260	1.834840233802171	2.820217396399633
0.8	1.902797279921332	2.786238874170521	1.902797279498548	2.786238873551445
0.9	1.970754325632808	2.752260351314784	1.970754325194924	2.752260350703256
1	2.038711371344284	2.718281828459046	2.038711370891301	2.718281827855068



**Figure 4.2.1:** Exact and RK5 solutions for  $t = 1$

**Table 4.2.2:** Numerical values for the exact and approximate solutions (Runge-Kutta) for  $t = 1.5$

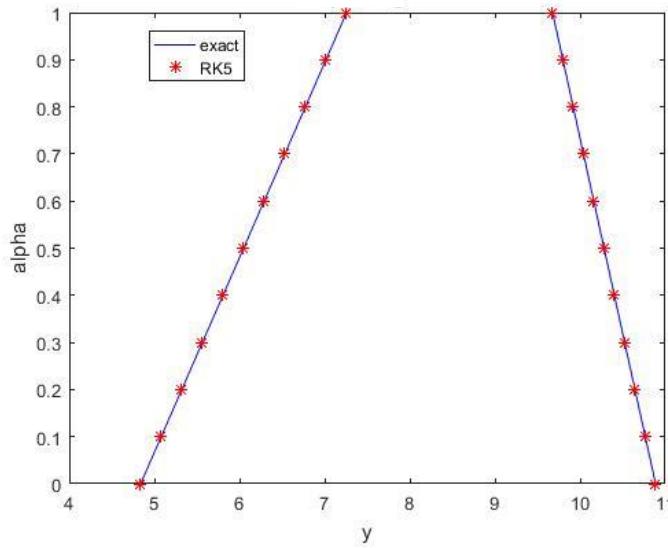
$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.645110862818530	5.951499441341692	2.645110861483967	5.951499438338925
0.1	2.777366405959457	5.885371669771230	2.777366404558166	5.885371666801826
0.2	2.909621949100383	5.819243898200766	2.909621947632363	5.819243895264729
0.3	3.041877492241309	5.753116126630302	3.041877490706562	5.753116123727630
0.4	3.174133035382236	5.686988355059839	3.174133033780761	5.686988352190530
0.5	3.306388578523162	5.620860583489376	3.306388576854960	5.620860580653431
0.6	3.438644121664089	5.554732811918913	3.438644119929157	5.554732809116330
0.7	3.570899664805015	5.488605040348450	3.570899663003356	5.488605037579232
0.8	3.703155207945942	5.422477268777985	3.703155206077555	5.422477266042134
0.9	3.835410751086868	5.356349497207523	3.835410749151752	5.356349494505033
1	3.967666294227795	5.290221725637060	3.967666292225951	5.290221722967934



**Figure 4.2.2:** Exact and RK5 solutions for

**Table 4.2.3:** Numerical values for the exact and approximate solutions (Runge-Kutta) for  $t = 2$

$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	4.838487836178892	10.88659763140251	4.838487833636942	10.88659762568312
0.1	5.080412227987837	10.76563543549804	5.080412225318790	10.76563542984220
0.2	5.322336619796782	10.64467323959356	5.322336617000635	10.64467323400128
0.3	5.564261011605725	10.52371104368909	5.564261008682483	10.52371103816035
0.4	5.806185403414671	10.40274884778462	5.806185400364332	10.40274884231943
0.5	6.048109795223614	10.28178665188015	6.048109792046180	10.28178664647850
0.6	6.290034187032560	10.16082445597567	6.290034183728024	10.16082445063758
0.7	6.531958578841504	10.03986226007120	6.531958575409873	10.03986225479666
0.8	6.773882970650448	9.918900064166728	6.773882967091722	9.918900058955734
0.9	7.015807362459393	9.797937868262256	7.015807358773566	9.797937863114807
1	7.257731754268338	9.676975672357784	7.257731750455414	9.676975667273885



**Figure 4.2.3:** Exact and RK5 solutions for  $t = 2$

**Table 4.2.4:** The absolute errors of the Runge-kutta method for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$3.0199 \times 10^{-10}$	$6.7948 \times 10^{-10}$
0.1	$3.1709 \times 10^{-10}$	$6.7193 \times 10^{-10}$
0.2	$3.3219 \times 10^{-10}$	$6.6437 \times 10^{-10}$
0.3	$3.4729 \times 10^{-10}$	$6.5682 \times 10^{-10}$
0.4	$3.6239 \times 10^{-10}$	$6.4928 \times 10^{-10}$
0.5	$3.7748 \times 10^{-10}$	$6.4173 \times 10^{-10}$
0.6	$3.9259 \times 10^{-10}$	$6.3418 \times 10^{-10}$
0.7	$4.0768 \times 10^{-10}$	$6.2663 \times 10^{-10}$
0.8	$4.2278 \times 10^{-10}$	$6.1908 \times 10^{-10}$
0.9	$4.3788 \times 10^{-10}$	$6.1153 \times 10^{-10}$
1	$4.5298 \times 10^{-10}$	$6.0398 \times 10^{-10}$

**Table 4.2.5:** The absolute errors of the Runge-kutta method for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.3346 \times 10^{-9}$	$3.0028 \times 10^{-9}$
0.1	$1.4013 \times 10^{-9}$	$2.9694 \times 10^{-9}$
0.2	$1.4680 \times 10^{-9}$	$2.9360 \times 10^{-9}$
0.3	$1.5347 \times 10^{-9}$	$2.9027 \times 10^{-9}$
0.4	$1.6015 \times 10^{-9}$	$2.8693 \times 10^{-9}$
0.5	$1.6682 \times 10^{-9}$	$2.8359 \times 10^{-9}$
0.6	$1.7349 \times 10^{-9}$	$2.8026 \times 10^{-9}$
0.7	$1.8017 \times 10^{-9}$	$2.7692 \times 10^{-9}$
0.8	$1.8684 \times 10^{-9}$	$2.7359 \times 10^{-9}$
0.9	$1.9351 \times 10^{-9}$	$2.7025 \times 10^{-9}$
1	$2.0018 \times 10^{-9}$	$2.6691 \times 10^{-9}$

**Table 4.2.6:** The absolute errors of the Runge-kutta method for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$2.5419 \times 10^{-9}$	$5.7194 \times 10^{-9}$
0.1	$2.6690 \times 10^{-9}$	$5.6558 \times 10^{-9}$
0.2	$2.7961 \times 10^{-9}$	$5.5923 \times 10^{-9}$
0.3	$2.9232 \times 10^{-9}$	$5.5287 \times 10^{-9}$
0.4	$3.0503 \times 10^{-9}$	$5.4652 \times 10^{-9}$
0.5	$3.1774 \times 10^{-9}$	$5.4017 \times 10^{-9}$
0.6	$3.3045 \times 10^{-9}$	$5.3381 \times 10^{-9}$
0.7	$3.4316 \times 10^{-9}$	$5.2745 \times 10^{-9}$
0.8	$3.5587 \times 10^{-9}$	$5.2110 \times 10^{-9}$
0.9	$3.6858 \times 10^{-9}$	$5.1474 \times 10^{-9}$
1	$3.8129 \times 10^{-9}$	$5.0839 \times 10^{-9}$

As shown in Tables 4.2.1-6 , the Runge-Kutta of order five method with trapezoidal fuzzy number as initial condition gave accurate results with small  $h$ .

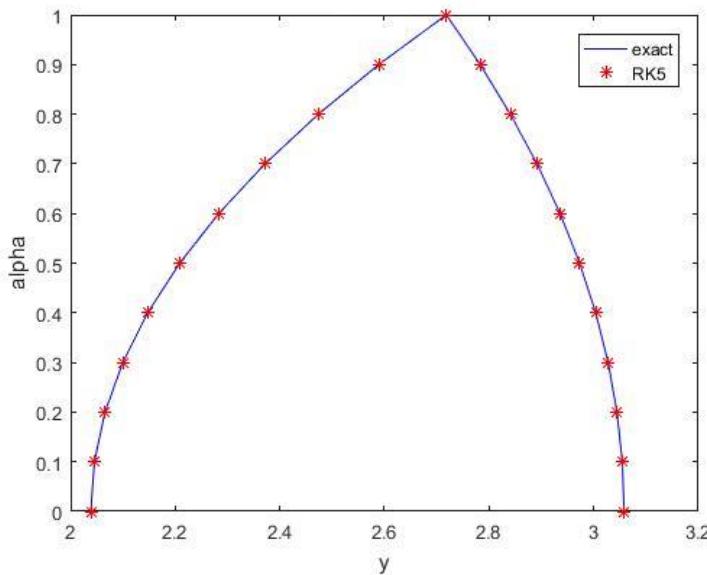
### b- Triangular Shaped Fuzzy Number

$$\text{let } y(0, \alpha) = [0.75 + 0.25\alpha^2, 1.125 - 0.125\alpha^2], 0 \leq \alpha \leq 1$$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 4.2.7-9 and Figs 4.2.4-6 respectively, and the absolute errors of the approximate results in Tables 4.2.10-12.

**Table 4.2.7:** Numerical values for the exact and approximate solutions (Runge-Kutta) for  $t = 1$

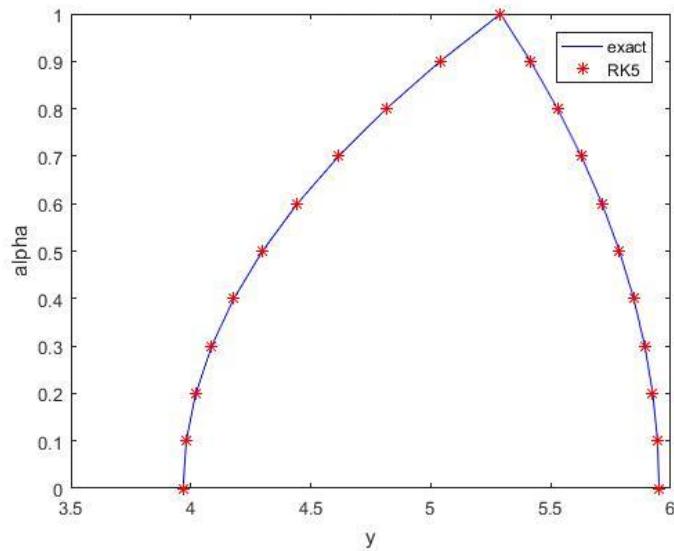
$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.038711371344284	3.058067057016426	2.038711370891301	3.058067056336951
0.1	2.045507075915432	3.054669204730852	2.045507075460939	3.054669204052133
0.2	2.065894189628875	3.044475647874132	2.065894189169852	3.044475647197676
0.3	2.099872712484613	3.027486386446262	2.099872712018040	3.027486385773582
0.4	2.147442644482646	3.003701420447245	2.147442644005503	3.003701419779850
0.5	2.208603985622974	2.973120749877081	2.208603985132243	2.973120749216481
0.6	2.283356735905599	2.935744374735769	2.283356735398257	2.935744374083473
0.7	2.371700895330517	2.891572295023310	2.371700894803547	2.891572294380829
0.8	2.473636463897732	2.840604510739702	2.473636463348112	2.840604510108546
0.9	2.589163441607241	2.782841021884948	2.589163441031952	2.782841021266626
1	2.718281828459046	2.718281828459046	2.718281827855068	2.718281827855068



**Figure 4.2.4:** Exact and RK5 solutions for  $t = 1$

**Table 4.2.8:** Numerical values for the exact and approximate solutions (Runge-Kutta) for  $t = 1.5$

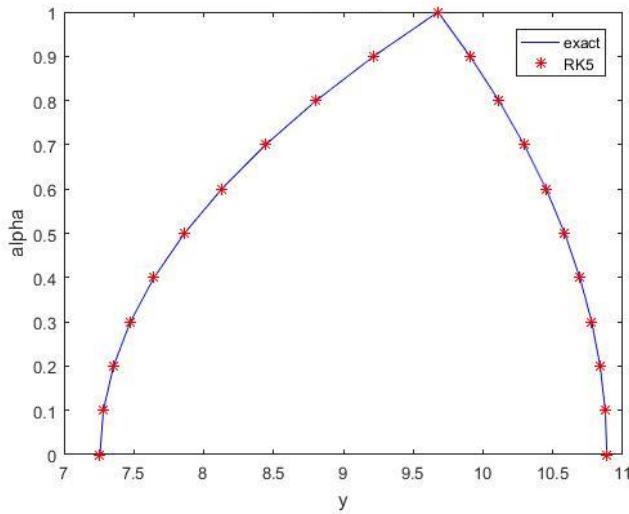
$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	3.967666294227795	5.951499441341692	3.967666292225951	5.951499438338925
0.1	3.980891848541887	5.944886664184645	3.980891846533372	5.944886661185217
0.2	4.020568511484165	5.925048332713508	4.020568509455631	5.925048329724087
0.3	4.086696283054629	5.891984446928276	4.086696280992729	5.891984443955537
0.4	4.179275163253278	5.845695006828951	4.179275161144667	5.845695003879568
0.5	4.298305152080111	5.786180012415534	4.298305149911448	5.786180009496180
0.6	4.443786249535131	5.713439463688025	4.443786247293065	5.713439460805369
0.7	4.615718455618335	5.627473360646422	4.615718453289524	5.627473357807141
0.8	4.814101770329725	5.528281703290727	4.814101767900820	5.528281700501491
0.9	5.038936193669300	5.415864491620940	5.038936191126958	5.415864488888424
1	5.290221725637060	5.290221725637060	5.290221722967934	5.290221722967934



**Figure 4.2.5:** Exact and RK5 solutions for  $t = 1.5$

**Table 4.2.9:** Numerical values for the exact and approximate solutions (Runge-Kutta) for  $t = 2$

$\alpha$	Exact		Runge-kutta	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	7.257731754268338	10.88659763140251	7.257731750455414	10.88659762568312
0.1	7.281924193449231	10.87450141181206	7.281924189623601	10.87450140609903
0.2	7.354501510991915	10.83821275304072	7.354501507128155	10.83821274734675
0.3	7.475463706896387	10.77773165508848	7.475463702969076	10.77773164942629
0.4	7.644810781162650	10.69305811795535	7.644810777146367	10.69305811233764
0.5	7.862542733790699	10.58419214164133	7.862542729660033	10.58419213608081
0.6	8.128659564780540	10.45113372614641	8.128659560510064	10.45113372065580
0.7	8.443161274132168	10.29388287147059	8.443161269696468	10.29388286606260
0.8	8.806047861845585	10.11243957761388	8.806047857219236	10.11243957230121
0.9	9.217319327920789	9.906803844576279	9.217319323078376	9.906803839371642
1	9.676975672357784	9.676975672357784	9.676975667273885	9.676975667273885



**Figure 4.2.6:** Exact and RK5 solutions for  $t = 2$

**Table 4.2.10:** The absolute errors of the Runge-kutta method for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$4.5298 \times 10^{-10}$	$6.7948 \times 10^{-10}$
0.1	$4.5449 \times 10^{-10}$	$6.7872 \times 10^{-10}$
0.2	$4.5902 \times 10^{-10}$	$6.7646 \times 10^{-10}$
0.3	$4.6657 \times 10^{-10}$	$6.7268 \times 10^{-10}$
0.4	$4.7714 \times 10^{-10}$	$6.6740 \times 10^{-10}$
0.5	$4.9073 \times 10^{-10}$	$6.6060 \times 10^{-10}$
0.6	$5.0734 \times 10^{-10}$	$6.5230 \times 10^{-10}$
0.7	$5.2697 \times 10^{-10}$	$6.4248 \times 10^{-10}$
0.8	$5.4962 \times 10^{-10}$	$6.3116 \times 10^{-10}$
0.9	$5.7529 \times 10^{-10}$	$6.1832 \times 10^{-10}$
1	$6.0398 \times 10^{-10}$	$6.0398 \times 10^{-10}$

**Table 4.2.11** :The absolute errors of the Runge-kutta method for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$2.0018 \times 10^{-9}$	$3.0028 \times 10^{-9}$
0.1	$2.0085 \times 10^{-9}$	$2.9994 \times 10^{-9}$
0.2	$2.0285 \times 10^{-9}$	$2.9894 \times 10^{-9}$
0.3	$2.0619 \times 10^{-9}$	$2.9727 \times 10^{-9}$
0.4	$2.1086 \times 10^{-9}$	$2.9494 \times 10^{-9}$
0.5	$2.1687 \times 10^{-9}$	$2.9194 \times 10^{-9}$
0.6	$2.2421 \times 10^{-9}$	$2.8827 \times 10^{-9}$
0.7	$2.3288 \times 10^{-9}$	$2.8393 \times 10^{-9}$
0.8	$2.4289 \times 10^{-9}$	$2.7892 \times 10^{-9}$
0.9	$2.5423 \times 10^{-9}$	$2.7325 \times 10^{-9}$
1	$2.6691 \times 10^{-9}$	$2.6691 \times 10^{-9}$

**Table 4.2.12:** The absolute errors of the Runge-kutta method for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$3.8129 \times 10^{-9}$	$5.7194 \times 10^{-9}$
0.1	$3.8256 \times 10^{-9}$	$5.7130 \times 10^{-9}$
0.2	$3.8638 \times 10^{-9}$	$5.6940 \times 10^{-9}$
0.3	$3.9273 \times 10^{-9}$	$5.6622 \times 10^{-9}$
0.4	$4.0163 \times 10^{-9}$	$5.6177 \times 10^{-9}$
0.5	$4.1307 \times 10^{-9}$	$5.5605 \times 10^{-9}$
0.6	$4.2705 \times 10^{-9}$	$5.4906 \times 10^{-9}$
0.7	$4.4357 \times 10^{-9}$	$5.4080 \times 10^{-9}$
0.8	$4.6263 \times 10^{-9}$	$5.3127 \times 10^{-9}$
0.9	$4.8424 \times 10^{-9}$	$5.2046 \times 10^{-9}$
1	$5.0839 \times 10^{-9}$	$5.0839 \times 10^{-9}$

As shown in Tables 4.2.7-12 , the Runge-Kutta of order five method with triangular shaped fuzzy number as initial condition gave accurate results with small  $h$ .

### 4.3 General Linear Methods (GLM)

We will apply the GLM for hybrid fuzzy differential equations (29). Let  $N = 100$  When  $t \in [0,2]$

When  $K = 4$

a- Trapezoidal Fuzzy Number

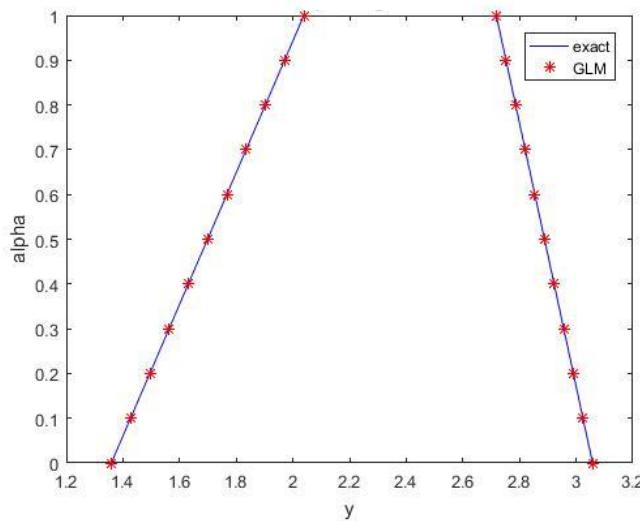
$$\text{let } y(0) = (0.5, 0.75, 1, 1.125)$$

$$y(0, \alpha) = [0.5 + 0.25\alpha, 1.125 - 0.125\alpha], \quad 0 \leq \alpha \leq 1$$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 4.3.1-3 and Figs 4.3.1-3 respectively, and the absolute errors of the approximate results in Tables 4.3.4-6.

**Table 4.3.1:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 1$

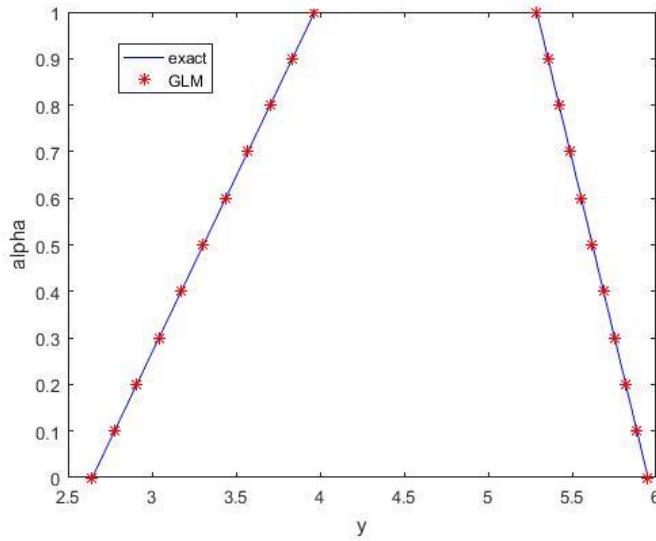
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	1.359140914229523	3.058067057016426	1.359140845041163	3.058066901342616
0.1	1.427097959940999	3.024088534160689	1.427097887293220	3.024088380216586
0.2	1.495055005652475	2.990110011304950	1.495054929545279	2.990109859090558
0.3	1.563012051363951	2.956131488449212	1.563011971797337	2.956131337964530
0.4	1.630969097075427	2.922152965593474	1.630969014049394	2.922152816838501
0.5	1.698926142786903	2.888174442737736	1.698926056301453	2.888174295712471
0.6	1.766883188498380	2.854195919881998	1.766883098553512	2.854195774586440
0.7	1.834840234209856	2.820217397026260	1.834840140805570	2.820217253460412
0.8	1.902797279921332	2.786238874170521	1.902797183057628	2.786238732334387
0.9	1.970754325632808	2.752260351314784	1.970754225309687	2.752260211208354
1	2.038711371344284	2.718281828459046	2.038711267561744	2.718281690082326



**Figure 4.3.1:** Exact and FGLM solutions for  $t = 1$

**Table 4.3.2:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 1.5$

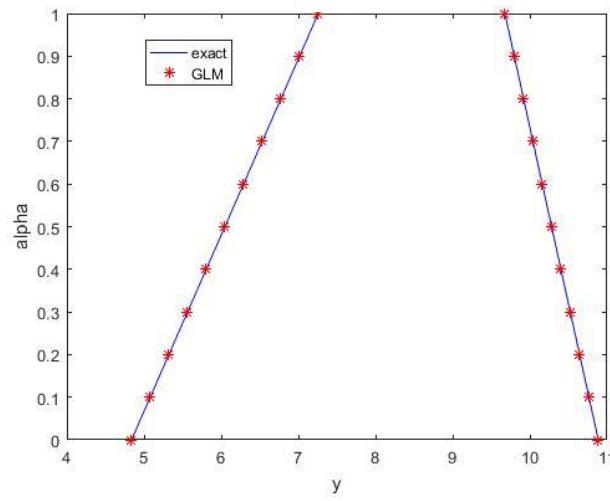
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.645110862818530	5.951499441341692	2.645110567705002	5.951498777336254
0.1	2.777366405959457	5.885371669771230	2.777366096090250	5.885371013143627
0.2	2.909621949100383	5.819243898200766	2.909621624475501	5.819243248951001
0.3	3.041877492241309	5.753116126630302	3.041877152860752	5.753115484758379
0.4	3.174133035382236	5.686988355059839	3.174132681245999	5.686987720565753
0.5	3.306388578523162	5.620860583489376	3.306388209631251	5.620859956373129
0.6	3.438644121664089	5.554732811918913	3.438643738016503	5.554732192180500
0.7	3.570899664805015	5.488605040348450	3.570899266401753	5.488604427987878
0.8	3.703155207945942	5.422477268777985	3.703154794787003	5.422476663795258
0.9	3.835410751086868	5.356349497207523	3.835410323172253	5.356348899602628
1	3.967666294227795	5.290221725637060	3.967665851557501	5.290221135410005



**Figure 4.3.2:** Exact and GLM solutions for  $t = 1.5$

**Table 4.3.3:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 2$

$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	4.838487836178892	10.88659763140251	4.838487274706883	10.88659636809048
0.1	5.080412227987837	10.76563543549804	5.080411638442222	10.76563418622281
0.2	5.322336619796782	10.64467323959356	5.322336002177567	10.64467200435513
0.3	5.564261011605725	10.52371104368909	5.564260365912914	10.52370982248747
0.4	5.806185403414671	10.40274884778462	5.806184729648250	10.40274764061979
0.5	6.048109795223614	10.28178665188015	6.048109093383600	10.28178545875212
0.6	6.290034187032560	10.16082445597567	6.290033457118945	10.16082327688444
0.7	6.531958578841504	10.03986226007120	6.531957820854288	10.03986109501678
0.8	6.773882970650448	9.918900064166728	6.773882184589633	9.918898913149114
0.9	7.015807362459393	9.797937868262256	7.015806548324976	9.797936731281430
1	7.257731754268338	9.676975672357784	7.257730912060317	9.676974549413766



**Figure 4.3.3:** Exact and GLM solutions for  $t = 2$

**Table 4.3.4:** The absolute errors of the FGLM for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$6.9188 \times 10^{-8}$	$1.5567 \times 10^{-7}$
0.1	$7.2648 \times 10^{-8}$	$1.5394 \times 10^{-7}$
0.2	$7.6107 \times 10^{-8}$	$1.5221 \times 10^{-7}$
0.3	$7.9567 \times 10^{-8}$	$1.5048 \times 10^{-7}$
0.4	$8.3026 \times 10^{-8}$	$1.4875 \times 10^{-7}$
0.5	$8.6485 \times 10^{-8}$	$1.4703 \times 10^{-7}$
0.6	$8.9945 \times 10^{-8}$	$1.4530 \times 10^{-7}$
0.7	$9.3404 \times 10^{-8}$	$1.4357 \times 10^{-7}$
0.8	$9.6864 \times 10^{-8}$	$1.4184 \times 10^{-7}$
0.9	$1.0032 \times 10^{-7}$	$1.4011 \times 10^{-7}$
1	$1.0378 \times 10^{-7}$	$1.3838 \times 10^{-7}$

**Table 4.3.5:** The absolute errors of the FGLM for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$2.9511 \times 10^{-7}$	$6.6401 \times 10^{-7}$
0.1	$3.0987 \times 10^{-7}$	$6.5663 \times 10^{-7}$
0.2	$3.2462 \times 10^{-7}$	$6.4925 \times 10^{-7}$
0.3	$3.3938 \times 10^{-7}$	$6.4187 \times 10^{-7}$
0.4	$3.5414 \times 10^{-7}$	$6.3449 \times 10^{-7}$
0.5	$3.6889 \times 10^{-7}$	$6.2712 \times 10^{-7}$
0.6	$3.8365 \times 10^{-7}$	$6.1974 \times 10^{-7}$
0.7	$3.9840 \times 10^{-7}$	$6.1236 \times 10^{-7}$
0.8	$4.1316 \times 10^{-7}$	$6.0498 \times 10^{-7}$
0.9	$4.2791 \times 10^{-7}$	$5.9760 \times 10^{-7}$
1	$4.4267 \times 10^{-7}$	$5.9023 \times 10^{-7}$

**Table 4.3.6:** The absolute errors of the FGLM for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$5.6147 \times 10^{-7}$	$1.2633 \times 10^{-6}$
0.1	$5.8955 \times 10^{-7}$	$1.2493 \times 10^{-6}$
0.2	$6.1762 \times 10^{-7}$	$1.2352 \times 10^{-6}$
0.3	$6.4569 \times 10^{-7}$	$1.2212 \times 10^{-6}$
0.4	$6.7377 \times 10^{-7}$	$1.2072 \times 10^{-6}$
0.5	$7.0184 \times 10^{-7}$	$1.1931 \times 10^{-6}$
0.6	$7.2991 \times 10^{-7}$	$1.1791 \times 10^{-6}$
0.7	$7.5799 \times 10^{-7}$	$1.1651 \times 10^{-6}$
0.8	$7.8606 \times 10^{-7}$	$1.1510 \times 10^{-6}$
0.9	$8.1413 \times 10^{-7}$	$1.1370 \times 10^{-6}$
1	$8.4221 \times 10^{-7}$	$1.1229 \times 10^{-6}$

As shown in Tables 4.3.1-6, the fuzzy general linear method with trapezoidal fuzzy number as initial condition gave less accurate than Runge-Kutta but it needed less number of steps.

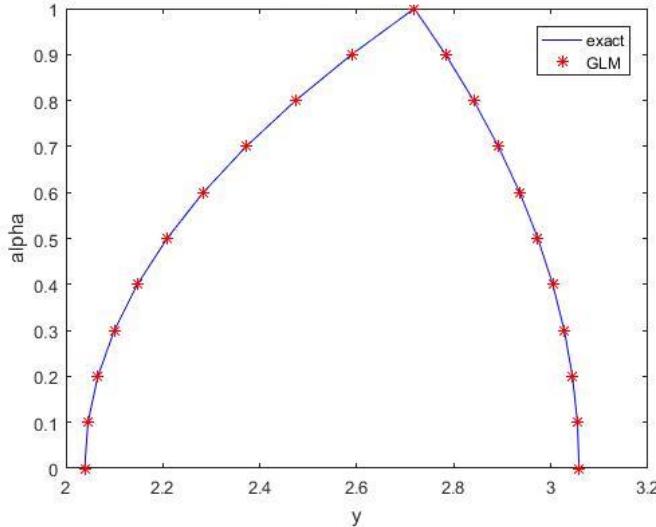
### b- Triangular Shaped Fuzzy Number

$$\text{let } y(0, \alpha) = [0.75 + 0.25\alpha^2, 1.125 - 0.125\alpha^2], 0 \leq \alpha \leq 1$$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 4.3.7-9 and Figs 4.3.4-6 respectively, and the absolute errors of the approximate results in Tables 4.3.10-12.

**Table 4.3.7:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 1$

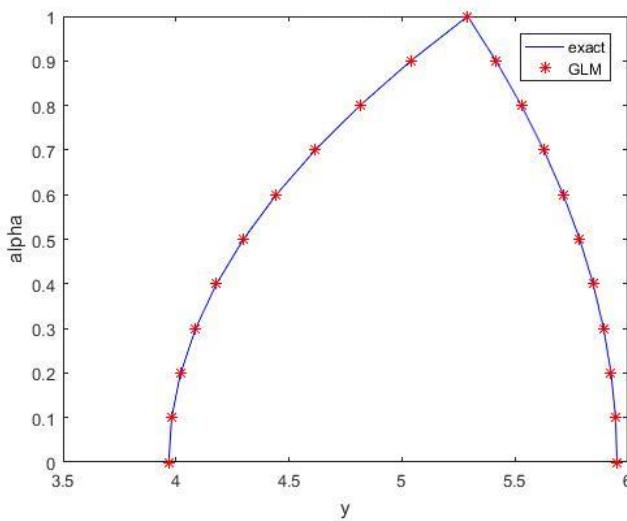
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.038711371344284	3.058067057016426	2.038711267561744	3.058066901342616
0.1	2.045507075915432	3.054669204730852	2.045506971786950	3.054669049230014
0.2	2.065894189628875	3.044475647874132	2.065894084462567	3.044475492892205
0.3	2.099872712484613	3.027486386446262	2.099872605588597	3.027486232329192
0.4	2.147442644482646	3.003701420447245	2.147442535165037	3.003701267540970
0.5	2.208603985622974	2.973120749877081	2.208603873191889	2.973120598527546
0.6	2.283356735905599	2.935744374735769	2.283356619669154	2.935744225288912
0.7	2.371700895330517	2.891572295023310	2.371700774596830	2.891572147825074
0.8	2.473636463897732	2.840604510739702	2.473636337974917	2.840604366136030
0.9	2.589163441607241	2.782841021884948	2.589163309803415	2.782840880221781
1	2.718281828459046	2.718281828459046	2.718281690082326	2.718281690082326



**Figure 4.3.4:** Exact and GLM solutions for  $t = 1$

**Table 4.3.8:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 1.5$

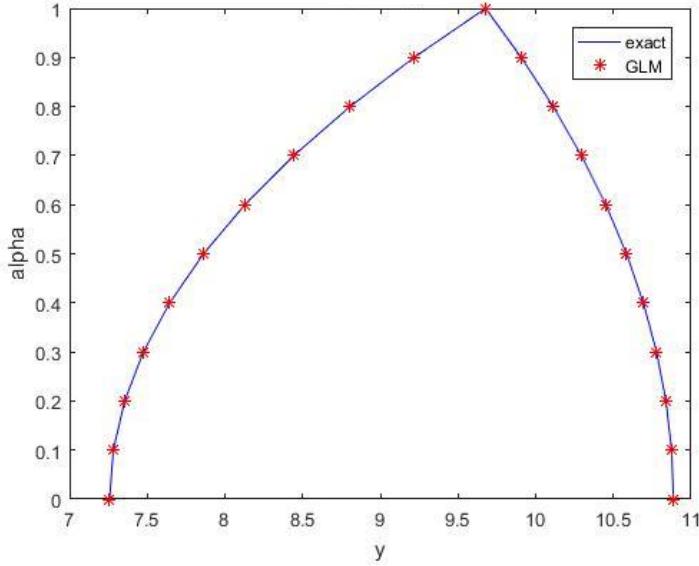
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	3.967666294227795	5.951499441341692	3.967665851557501	5.951498777336254
0.1	3.980891848541887	5.944886664184645	3.980891404396027	5.944886000916993
0.2	4.020568511484165	5.925048332713508	4.020568062911601	5.925047671659204
0.3	4.086696283054629	5.891984446928276	4.086695827104228	5.891983789562895
0.4	4.179275163253278	5.845695006828951	4.179274696973901	5.845694354628052
0.5	4.298305152080111	5.786180012415534	4.298304672520626	5.786179366854693
0.6	4.443786249535131	5.713439463688025	4.443785753744404	5.713438826242805
0.7	4.615718455618335	5.627473360646422	4.615717940645231	5.627472732792390
0.8	4.814101770329725	5.528281703290727	4.814101233223103	5.528281086503453
0.9	5.038936193669300	5.415864491620940	5.038935631478029	5.415863887375990
1	5.290221725637060	5.290221725637060	5.290221135410005	5.290221135410005



**Figure 4.3.5:** Exact and GLM solutions for  $t = 1.5$

**Table 4.3.9:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 2$

$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	7.257731754268338	10.88659763140251	7.257730912060317	10.88659636809048
0.1	7.281924193449231	10.87450141181206	7.281923348433852	10.87450014990372
0.2	7.354501510991915	10.83821275304072	7.354500657554453	10.83821149534341
0.3	7.475463706896387	10.77773165508848	7.475462839422129	10.77773040440958
0.4	7.644810781162650	10.69305811795535	7.644809894036866	10.69305687710221
0.5	7.862542733790699	10.58419214164133	7.862541821398675	10.58419091342130
0.6	8.128659564780540	10.45113372614641	8.128658621507558	10.45113251336687
0.7	8.443161274132168	10.29388287147059	8.443160294363510	10.29388167693889
0.8	8.806047861845585	10.11243957761388	8.806046839966523	10.11243840413738
0.9	9.217319327920789	9.906803844576279	9.217318258316608	9.906802694962339
1	9.676975672357784	9.676975672357784	9.676974549413766	9.676974549413766



**Figure 4.3.6:** Exact and GLM solutions for  $t = 2$

**Table 4.3.10:** The absolute errors of the FGLM for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - y  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.0378 \times 10^{-7}$	$1.5567 \times 10^{-7}$
0.1	$1.0413 \times 10^{-7}$	$1.5550 \times 10^{-7}$
0.2	$1.0517 \times 10^{-7}$	$1.5498 \times 10^{-7}$
0.3	$1.0690 \times 10^{-7}$	$1.5412 \times 10^{-7}$
0.4	$1.0932 \times 10^{-7}$	$1.5291 \times 10^{-7}$
0.5	$1.1243 \times 10^{-7}$	$1.5135 \times 10^{-7}$
0.6	$1.1624 \times 10^{-7}$	$1.4945 \times 10^{-7}$
0.7	$1.2073 \times 10^{-7}$	$1.4720 \times 10^{-7}$
0.8	$1.2592 \times 10^{-7}$	$1.4460 \times 10^{-7}$
0.9	$1.3180 \times 10^{-7}$	$1.4166 \times 10^{-7}$
1	$1.3838 \times 10^{-7}$	$1.3838 \times 10^{-7}$

**Table 4.3.11:** The absolute errors of the FGML for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$4.4267 \times 10^{-7}$	$6.6401 \times 10^{-7}$
0.1	$4.4415 \times 10^{-7}$	$6.6327 \times 10^{-7}$
0.2	$4.4857 \times 10^{-7}$	$6.6105 \times 10^{-7}$
0.3	$4.5595 \times 10^{-7}$	$6.5737 \times 10^{-7}$
0.4	$4.6628 \times 10^{-7}$	$6.5220 \times 10^{-7}$
0.5	$4.7956 \times 10^{-7}$	$6.4556 \times 10^{-7}$
0.6	$4.9579 \times 10^{-7}$	$6.3745 \times 10^{-7}$
0.7	$5.1497 \times 10^{-7}$	$6.2785 \times 10^{-7}$
0.8	$5.3711 \times 10^{-7}$	$6.1679 \times 10^{-7}$
0.9	$5.6219 \times 10^{-7}$	$6.0424 \times 10^{-7}$
1	$5.9023 \times 10^{-7}$	$5.9023 \times 10^{-7}$

**Table 4.3.12:** The absolute errors of the FGML for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$8.4502 \times 10^{-7}$	$1.2633 \times 10^{-6}$
0.1	$8.5344 \times 10^{-7}$	$1.2619 \times 10^{-6}$
0.2	$8.6747 \times 10^{-7}$	$1.2577 \times 10^{-6}$
0.3	$8.8713 \times 10^{-7}$	$1.2507 \times 10^{-6}$
0.4	$9.1239 \times 10^{-7}$	$1.2409 \times 10^{-6}$
0.5	$9.4327 \times 10^{-7}$	$1.2282 \times 10^{-6}$
0.6	$9.7977 \times 10^{-7}$	$1.2128 \times 10^{-6}$
0.7	$1.0219 \times 10^{-6}$	$1.1945 \times 10^{-6}$
0.8	$1.0696 \times 10^{-6}$	$1.1735 \times 10^{-6}$
0.9	$1.1229 \times 10^{-6}$	$1.1496 \times 10^{-6}$
1	$8.4502 \times 10^{-7}$	$1.1229 \times 10^{-6}$

As shown in Tables 4.3.7-12, the fuzzy general linear method with triangular shaped fuzzy number as initial condition gave less accurate than Runge-Kutta but it needed less number of steps.

When  $K = 5$

a- Trapezoidal Fuzzy Number

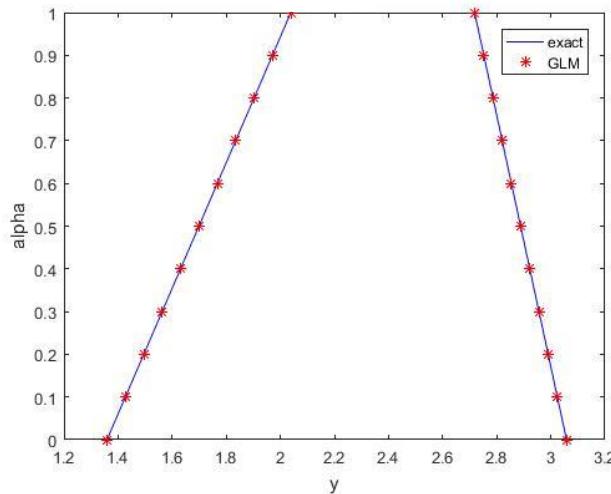
$$\text{let } y(0) = (0.5, 0.75, 1, 1.125)$$

$$y(0, \alpha) = [0.5 + 0.25\alpha, 1.125 - 0.125\alpha], \quad 0 \leq \alpha \leq 1$$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 4.3.13-15 and Figs 4.3.7-9 respectively, and the absolute errors of the approximate results in Tables 4.3.16-18.

**Table 4.3.13:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 1$

$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	1.359140914229523	3.058067057016426	1.359140912938296	3.058067054111163
0.1	1.427097959940999	3.024088534160689	1.427097958585210	3.024088531287706
0.2	1.495055005652475	2.990110011304950	1.495055004232123	2.990110008464247
0.3	1.563012051363951	2.956131488449212	1.563012049879039	2.956131485640793
0.4	1.630969097075427	2.922152965593474	1.630969095525954	2.922152962817334
0.5	1.698926142786903	2.888174442737736	1.698926141172869	2.888174439993877
0.6	1.766883188498380	2.854195919881998	1.766883186819783	2.854195917170421
0.7	1.834840234209856	2.820217397026260	1.834840232466698	2.820217394346960
0.8	1.902797279921332	2.786238874170521	1.902797278113612	2.786238871523505
0.9	1.970754325632808	2.752260351314784	1.970754323760527	2.752260348700048
1	2.038711371344284	2.718281828459046	2.038711369407443	2.718281825876591



**Figure 4.3.7:** Exact and GLM solutions for  $t = 1$

**Table 4.3.14:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 1.5$

$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$y$	$\bar{y}$
0	2.645110862818530	5.951499441341692	2.645110857325112	5.951499428981495
0.1	2.777366405959457	5.885371669771230	2.777366400191366	5.885371657548369
0.2	2.909621949100383	5.819243898200766	2.909621943057618	5.819243886115239
0.3	3.041877492241309	5.753116126630302	3.041877485923876	5.753116114682116
0.4	3.174133035382236	5.686988355059839	3.174133028790132	5.686988343248985
0.5	3.306388578523162	5.620860583489376	3.306388571656389	5.620860571815860
0.6	3.438644121664089	5.554732811918913	3.438644114522641	5.554732800382736
0.7	3.570899664805015	5.488605040348450	3.570899657388897	5.488605028949600

0.8	3.703155207945942	5.422477268777985	3.703155200255151	5.422477257516476
0.9	3.835410751086868	5.356349497207523	3.835410743121407	5.356349486083349
1	3.967666294227795	5.290221725637060	3.967666285987665	5.290221714650222

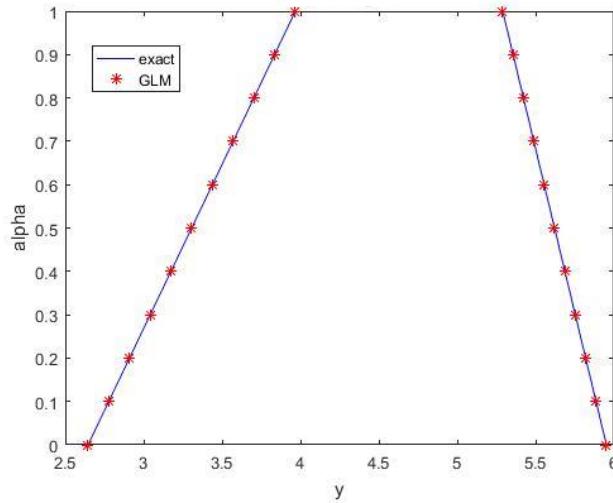


Figure 4.3.8: Exact and GLM solutions for  $t = 1.5$

Table 4.3.15: Numerical values for the exact and approximate solutions (FGLM) for  $t = 2$

$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	4.838487836178892	10.88659763140251	4.838487825728191	10.88659760788841
0.1	5.080412227987837	10.76563543549804	5.080412217014596	10.76563541224522
0.2	5.322336619796782	10.64467323959356	5.322336608300999	10.64467321660200
0.3	5.564261011605725	10.52371104368909	5.564260999587413	10.52371102095881
0.4	5.806185403414671	10.40274884778462	5.806185390873822	10.40274882531560
0.5	6.048109795223614	10.28178665188015	6.048109782160234	10.28178662967240
0.6	6.290034187032560	10.16082445597567	6.290034173446638	10.16082443402920
0.7	6.531958578841504	10.03986226007120	6.531958564733049	10.03986223838597
0.8	6.773882970650448	9.918900064166728	6.773882956019454	9.918900042742784
0.9	7.015807362459393	9.797937868262256	7.015807347305866	9.797937847099576
1	7.257731754268338	9.676975672357784	7.257731738592279	9.676975651456381

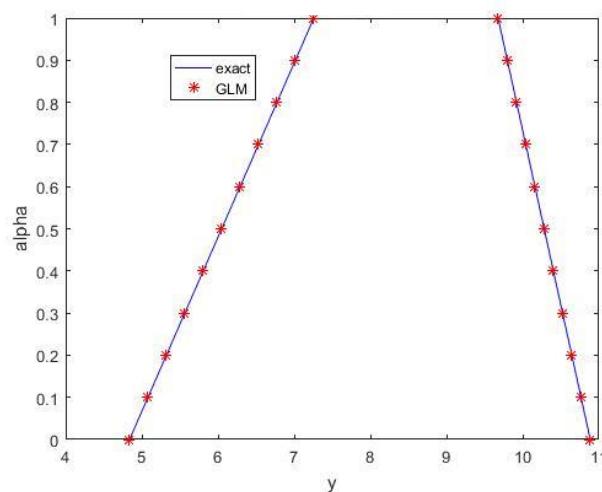


Figure 4.3.9: Exact and GLM solutions for  $t = 2$

**Table 4.3.16:** The absolute errors of the FGLM for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.2912 \times 10^{-9}$	$2.9053 \times 10^{-9}$
0.1	$1.3558 \times 10^{-9}$	$2.8730 \times 10^{-9}$
0.2	$1.4204 \times 10^{-9}$	$2.8407 \times 10^{-9}$
0.3	$1.4849 \times 10^{-9}$	$2.8084 \times 10^{-9}$
0.4	$1.5495 \times 10^{-9}$	$2.7761 \times 10^{-9}$
0.5	$1.6140 \times 10^{-9}$	$2.7439 \times 10^{-9}$
0.6	$1.6786 \times 10^{-9}$	$2.7116 \times 10^{-9}$
0.7	$1.7432 \times 10^{-9}$	$2.6793 \times 10^{-9}$
0.8	$1.8077 \times 10^{-9}$	$2.6470 \times 10^{-9}$
0.9	$1.8723 \times 10^{-9}$	$2.6147 \times 10^{-9}$
1	$1.9368 \times 10^{-9}$	$2.5825 \times 10^{-9}$

**Table 4.3.17:** The absolute errors of the FGLM for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$5.4934 \times 10^{-9}$	$1.2360 \times 10^{-8}$
0.1	$5.7681 \times 10^{-9}$	$1.2223 \times 10^{-8}$
0.2	$6.0428 \times 10^{-9}$	$1.2086 \times 10^{-8}$
0.3	$6.3174 \times 10^{-9}$	$1.1948 \times 10^{-8}$
0.4	$6.5921 \times 10^{-9}$	$1.1811 \times 10^{-8}$
0.5	$6.8668 \times 10^{-9}$	$1.1674 \times 10^{-8}$
0.6	$7.1414 \times 10^{-9}$	$1.1536 \times 10^{-8}$
0.7	$7.4161 \times 10^{-9}$	$1.1399 \times 10^{-8}$
0.8	$7.6908 \times 10^{-9}$	$1.1262 \times 10^{-8}$
0.9	$7.9655 \times 10^{-9}$	$1.1124 \times 10^{-8}$
1	$8.2401 \times 10^{-9}$	$1.0987 \times 10^{-8}$

**Table 4.3.18 :** The absolute errors of the FGLM for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.0451 \times 10^{-8}$	$2.3514 \times 10^{-8}$
0.1	$1.0973 \times 10^{-8}$	$2.3253 \times 10^{-8}$
0.2	$1.1496 \times 10^{-8}$	$2.2992 \times 10^{-8}$
0.3	$1.2018 \times 10^{-8}$	$2.2730 \times 10^{-8}$
0.4	$1.2541 \times 10^{-8}$	$2.2469 \times 10^{-8}$
0.5	$1.3063 \times 10^{-8}$	$2.2208 \times 10^{-8}$
0.6	$1.3586 \times 10^{-8}$	$2.1946 \times 10^{-8}$
0.7	$1.4108 \times 10^{-8}$	$2.1685 \times 10^{-8}$
0.8	$1.4631 \times 10^{-8}$	$2.1424 \times 10^{-8}$
0.9	$1.5154 \times 10^{-8}$	$2.1163 \times 10^{-8}$
1	$1.5676 \times 10^{-8}$	$2.0901 \times 10^{-8}$

As shown in Tables 4.3.13-18, the fuzzy general linear method with trapezoidal fuzzy number as initial condition gave less accurate than Runge-Kutta but it needed less number of steps.

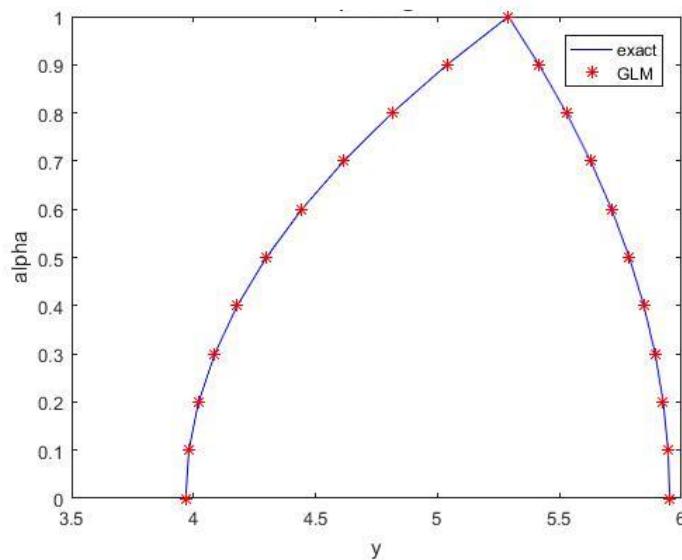
### b- Triangular Shaped Fuzzy Number

$$\text{let } y(0, \alpha) = [0.75 + 0.25\alpha^2, 1.125 - 0.125\alpha^2], 0 \leq \alpha \leq 1$$

We solve by Matlab software the exact solutions with approximate results of this example are presented in Tables 4.3.19-21 and Figs 4.3.10-12 respectively, and the absolute errors of the approximate results in Tables 4.3.22-24.

**Table 4.3.19:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 1$

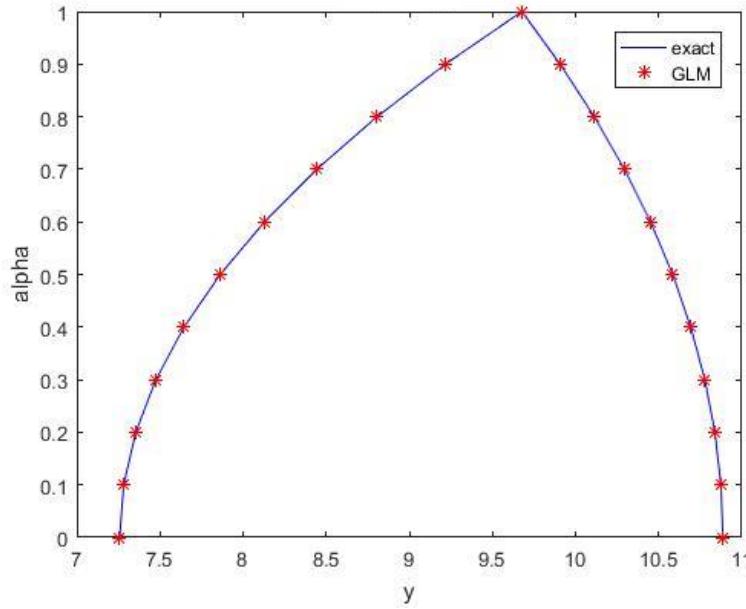
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	2.038711371344284	3.058067057016426	2.038711369407443	3.058067054111163
0.1	2.045507075915432	3.054669204730852	2.045507073972134	3.054669201828817
0.2	2.065894189628875	3.044475647874132	2.065894187666208	3.044475644981782
0.3	2.099872712484613	3.027486386446262	2.099872710489666	3.027486383570052
0.4	2.147442644482646	3.003701420447245	2.147442642442506	3.003701417593631
0.5	2.208603985622974	2.973120749877081	2.208603983524728	2.973120747052521
0.6	2.283356735905599	2.935744374735769	2.283356733736335	2.935744371946717
0.7	2.371700895330517	2.891572295023310	2.371700893077324	2.891572292276222
0.8	2.473636463897732	2.840604510739702	2.473636461547696	2.840604508041035
0.9	2.589163441607241	2.782841021884948	2.589163439147451	2.782841019241159
1	2.718281828459046	2.718281828459046	2.718281825876591	2.718281825876591



**Figure 4.3.10:** Exact and GLM solutions for  $t = 1$

**Table 4.3.20:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 1.5$

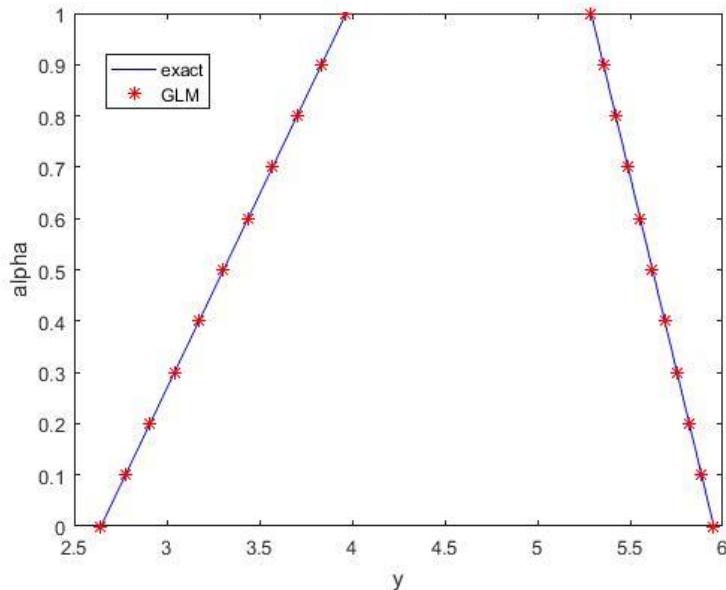
$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	3.967666294227795	5.951499441341692	3.967666285987665	5.951499428981495
0.1	3.980891848541887	5.944886664184645	3.980891840274289	5.944886651838181
0.2	4.020568511484165	5.925048332713508	4.020568503134166	5.925048320408249
0.3	4.086696283054629	5.891984446928276	4.086696274567295	5.891984434691680
0.4	4.179275163253278	5.845695006828951	4.179275154573673	5.845694994688491
0.5	4.298305152080111	5.786180012415534	4.298305143153301	5.786180000398678
0.6	4.443786249535131	5.713439463688025	4.443786240306182	5.713439451822237
0.7	4.615718455618335	5.627473360646422	4.615718446032315	5.627473348959170
0.8	4.814101770329725	5.528281703290727	4.814101760331699	5.528281691809477
0.9	5.038936193669300	5.415864491620940	5.038936183204331	5.415864480373159
1	5.290221725637060	5.290221725637060	5.290221714650222	5.290221714650222



**Figure 4.3.11:** Exact and GLM solutions for  $t = 1.5$

**Table 4.3.21:** Numerical values for the exact and approximate solutions (FGLM) for  $t = 2$

$\alpha$	Exact		FGLM	
	$\underline{Y}$	$\bar{Y}$	$\underline{y}$	$\bar{y}$
0	7.257731754268338	10.88659763140251	7.257731738592279	10.88659760788841
0.1	7.281924193449231	10.87450141181206	7.281924177720919	10.87450138832409
0.2	7.354501510991915	10.83821275304072	7.354501495106844	10.83821272963114
0.3	7.475463706896387	10.77773165508848	7.475463690750045	10.77773163180953
0.4	7.644810781162650	10.69305811795535	7.644810764650527	10.69305809485929
0.5	7.862542733790699	10.58419214164133	7.862542716808297	10.58419211878040
0.6	8.128659564780540	10.45113372614641	8.128659547223347	10.45113370357288
0.7	8.443161274132168	10.29388287147059	8.443161255895680	10.29388284923672
0.8	8.806047861845585	10.11243957761388	8.806047842825295	10.11243955577191
0.9	9.217319327920789	9.906803844576279	9.217319308012186	9.906803823178459
1	9.676975672357784	9.676975672357784	9.676975651456381	9.676975651456381



**Figure 4.3.12:** Exact and GLM solutions for  $t = 2$

**Table 4.3.22:** The absolute errors of the FGLM for  $t = 1$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.9368 \times 10^{-9}$	$2.9053 \times 10^{-9}$
0.1	$1.9433 \times 10^{-9}$	$2.9020 \times 10^{-9}$
0.2	$1.9627 \times 10^{-9}$	$2.8923 \times 10^{-9}$
0.3	$1.9949 \times 10^{-9}$	$2.8762 \times 10^{-9}$
0.4	$2.0401 \times 10^{-9}$	$2.8536 \times 10^{-9}$
0.5	$2.0982 \times 10^{-9}$	$2.8246 \times 10^{-9}$
0.6	$2.1693 \times 10^{-9}$	$2.7891 \times 10^{-9}$
0.7	$2.2532 \times 10^{-9}$	$2.7471 \times 10^{-9}$
0.8	$2.3500 \times 10^{-9}$	$2.6987 \times 10^{-9}$
0.9	$2.4598 \times 10^{-9}$	$2.6438 \times 10^{-9}$
1	$2.5825 \times 10^{-9}$	$2.5825 \times 10^{-9}$

**Table 4.3.23:** The absolute errors of the FGLM for  $t = 1.5$

$\alpha$	Absolute error	
	$ \underline{Y} - \underline{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$8.2401 \times 10^{-9}$	$1.2360 \times 10^{-8}$
0.1	$8.2676 \times 10^{-9}$	$1.2346 \times 10^{-8}$
0.2	$8.3500 \times 10^{-9}$	$1.2305 \times 10^{-8}$
0.3	$8.4873 \times 10^{-9}$	$1.2237 \times 10^{-8}$
0.4	$8.6796 \times 10^{-9}$	$1.2140 \times 10^{-8}$
0.5	$8.9268 \times 10^{-9}$	$1.2017 \times 10^{-8}$
0.6	$9.2289 \times 10^{-9}$	$1.1866 \times 10^{-8}$
0.7	$9.5860 \times 10^{-9}$	$1.1687 \times 10^{-8}$
0.8	$9.9980 \times 10^{-9}$	$1.1481 \times 10^{-8}$
0.9	$1.0465 \times 10^{-8}$	$1.1248 \times 10^{-8}$
1	$1.0987 \times 10^{-8}$	$1.0987 \times 10^{-8}$

**Table 4.3.24:** The absolute errors of the FGLM for  $t = 2$

$\alpha$	Absolute error	
	$ \underline{y} - \bar{y}  \approx$	$ \bar{Y} - \bar{y}  \approx$
0	$1.5676 \times 10^{-8}$	$2.3514 \times 10^{-8}$
0.1	$1.5728 \times 10^{-8}$	$2.3488 \times 10^{-8}$
0.2	$1.5885 \times 10^{-8}$	$2.3410 \times 10^{-8}$
0.3	$1.6146 \times 10^{-8}$	$2.3279 \times 10^{-8}$
0.4	$1.6512 \times 10^{-8}$	$2.3096 \times 10^{-8}$
0.5	$1.6982 \times 10^{-8}$	$2.2861 \times 10^{-8}$
0.6	$1.7557 \times 10^{-8}$	$2.2574 \times 10^{-8}$
0.7	$1.8236 \times 10^{-8}$	$2.2234 \times 10^{-8}$
0.8	$1.9020 \times 10^{-8}$	$2.1842 \times 10^{-8}$
0.9	$1.9909 \times 10^{-8}$	$2.1398 \times 10^{-8}$
1	$2.0901 \times 10^{-8}$	$2.0901 \times 10^{-8}$

As shown in Tables 4.3.19-24, the fuzzy general linear method with triangular shaped fuzzy number as initial condition gave less accurate than Runge-Kutta but it needed less number of steps.

## 4.4 Variational Iteration Method (VIM)

The Variational Iteration Method will be applied for hybrid fuzzy differential equations (29).

a- Trapezoidal Fuzzy Number

$$\text{let } y(0) = (0.5, 0.75, 1, 1.125)$$

$$y(0, \alpha) = [0.5 + 0.25\alpha, 1.125 - 0.125\alpha], \quad 0 \leq \alpha \leq 1$$

when  $t \in [0,1]$  the iteration formulas (43) can be used.

Starting with initial approximations

$$\underline{y}(0, \alpha) = (0.5 + 0.25\alpha), \quad \bar{y}(0, \alpha) = (1.125 - 0.125\alpha), \quad 0 \leq \alpha \leq 1$$

by iteration formulas, one can obtain

$$\underline{y}_1(t, \alpha) = e^t(0.5 + 0.25\alpha), \quad \bar{y}_1(t, \alpha) = e^t(1.125 - 0.125\alpha)$$

$$\underline{y}_2(t, \alpha) = e^t(0.5 + 0.25\alpha), \quad \bar{y}_2(t, \alpha) = e^t(1.125 - 0.125\alpha)$$

:

these are exactly the same as the components of equations (31). Therefore, by only one iteration, the exact solution is obtained.

For the case where  $t \in [1,1.5]$  the iteration formulas (43) can be used.

Starting with initial approximations

$$\underline{y}(1, \alpha) = (0.5 + 0.25\alpha)e, \quad \bar{y}(1, \alpha) = (1.125 - 0.125\alpha)e, \quad 0 \leq \alpha \leq 1$$

by iteration formulas, the following is obtained:

$$\underline{y}_1(t, \alpha) = \underline{y}(1, \alpha)(3e^{t-1} - 2t), \quad \bar{y}_1(t, \alpha) = \bar{y}(1, \alpha)(3e^{t-1} - 2t)$$

$$\underline{y}_2(t, \alpha) = \underline{y}(1, \alpha)(3e^{t-1} - 2t), \quad \bar{y}_2(t, \alpha) = \bar{y}(1, \alpha)(3e^{t-1} - 2t)$$

⋮

they are exactly the same as the components of equations (31). Therefore, by only one iteration, the exact solution is obtained.

when  $t \in [1.5, 2]$  by using iteration formulas (43):

starting with initial approximations

$$\underline{y}(1.5, \alpha) = \underline{y}(1, \alpha)(3e^{0.5} - 3), \quad \bar{y}(1, \alpha) = \bar{y}(1, \alpha)(3e^{0.5} - 3), \quad 0 \leq \alpha \leq 1$$

by iteration formulas, the following is obtained:

$$\begin{aligned} \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t) \\ \bar{y}_1(t, \alpha) &= \bar{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t) \end{aligned}$$

$$\begin{aligned} \underline{y}_2(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t) \\ \bar{y}_2(t, \alpha) &= \bar{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t) \end{aligned}$$

⋮

they are exactly the same as the components of equations (31). Therefore, by only one iteration, the exact solution is obtained.

### b- Triangular Shaped Fuzzy Number

$$\text{let } y(0, \alpha) = [0.75 + 0.25\alpha^2, 1.125 - 0.125\alpha^2], 0 \leq \alpha \leq 1$$

When  $t \in [0, 1]$ ,

$$y(0, \alpha) = [0.75 + 0.25\alpha^2, 1.125 - 0.125\alpha^2], \quad 0 \leq \alpha \leq 1$$

by iteration formulas (43), the following is obtained:

$$\begin{aligned} \underline{y}_1(t, \alpha) &= e^t(0.75 + 0.25\alpha^2), & \bar{y}_1(t, \alpha) &= e^t(1.125 - 0.125\alpha^2) \\ \underline{y}_2(t, \alpha) &= e^t(0.75 + 0.25\alpha^2), & \bar{y}_2(t, \alpha) &= e^t(1.125 - 0.125\alpha^2) \\ &\vdots \end{aligned}$$

they are exactly the same as the components of equations (31). Therefore, by only one iteration, the exact solution is obtained.

When  $t \in [1, 1.5]$ , starting with initial approximations

$$\underline{y}(1, \alpha) = (0.75 + 0.25\alpha^2)e, \quad \bar{y}(1, \alpha) = (1.125 - 0.125\alpha^2)e, \quad 0 \leq \alpha \leq 1$$

by iteration formulas(43), the following is obtained:

$$\begin{aligned} \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 2t), & \bar{y}_1(t, \alpha) &= \bar{y}(1, \alpha)(3e^{t-1} - 2t) \\ \underline{y}_2(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 2t), & \bar{y}_2(t, \alpha) &= \bar{y}(1, \alpha)(3e^{t-1} - 2t) \\ &\vdots \end{aligned}$$

they are exactly the same as the components of equations (31). Therefore, by only one iteration, the exact solution is obtained.

when  $t \in [1.5, 2]$ , starting with initial approximations

$$\underline{y}(1.5, \alpha) = \underline{y}(1, \alpha)(3e^{0.5} - 3), \quad \bar{y}(1, \alpha) = \bar{y}(1, \alpha)(3e^{0.5} - 3), \quad 0 \leq \alpha \leq 1$$

by iteration formulas (43) , the following is obtained:

$$\begin{aligned} \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t), \\ \bar{y}_1(t, \alpha) &= \bar{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t) \\ \underline{y}_2(t, \alpha) &= \underline{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t), \\ \bar{y}_2(t, \alpha) &= \bar{y}(1, \alpha)(3e^{t-1} - 4e^{t-1.5} - 2 + 2t) \\ &\vdots \end{aligned}$$

they are exactly the same as the components of Equations (31). Therefore, by only one iteration the exact solution is obtained.

## 4.5 Adomian Decomposition Method (ADM)

The Adomian decomposition method will be applied for hybrid fuzzy differential equations (29).

a- Let  $y(0) = (0.5, 0.75, 1, 1.125)$   
 $y(0, \alpha) = [0.5 + 0.25\alpha, 1.125 - 0.125\alpha], \quad 0 \leq \alpha \leq 1$

According to Equations (49), when  $t \in [0,1]$  we have

$$\bar{y}_0(t, \alpha) = 1.125 - 0.125\alpha, \quad \underline{y}_0(t, \alpha) = 0.5 + 0.25\alpha$$

Approximating  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\Phi}_6(t)$  and  $\bar{\Phi}_6(t)$ , respectively, as follows:

$$\begin{aligned} \underline{\Phi}_6(t) &= \sum_{k=0}^6 \underline{y}_k(t, \alpha), & \bar{\Phi}_6(t) &= \sum_{k=0}^6 \bar{y}_k(t, \alpha) \\ \underline{y}_1(t, \alpha) &= \int_0^t \underline{y}_0(s, \alpha) ds, & \bar{y}_1(t, \alpha) &= \int_0^t \bar{y}_0(s, \alpha) ds \\ \underline{y}_1(t, \alpha) &= (0.5 + 0.25\alpha)t, & \bar{y}_1(t, \alpha) &= (1.125 - 0.125\alpha)t \\ \underline{y}_2(t, \alpha) &= (0.5 + 0.25\alpha) \frac{t^2}{2}, & \bar{y}_2(t, \alpha) &= (1.125 - 0.125\alpha) \frac{t^2}{2} \\ &\vdots \\ \underline{y}_6(t, \alpha) &= (0.5 + 0.25\alpha) \frac{t^6}{720}, & \bar{y}_6(t, \alpha) &= (1.125 - 0.125\alpha) \frac{t^6}{720} \\ \underline{\Phi}_6(t) &= \sum_{k=0}^6 \underline{y}_0(t, \alpha) \frac{t^k}{k!}, & \bar{\Phi}_6(t) &= \sum_{k=0}^6 \bar{y}_0(t, \alpha) \frac{t^k}{k!} \end{aligned}$$

To find exact solution

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{\Phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\Phi}_n(t)$$

$$\underline{y}(t, \alpha) = \underline{y}_0(t, \alpha)e^t, \quad \bar{y}(t, \alpha) = \bar{y}_0(t, \alpha)e^t$$

when  $t \in [1, 1.5]$  we have

$$y_0(\alpha) = y(1, \alpha) = [(0.5 + 0.25\alpha)e^t, (1.125 - 0.125\alpha)e^t]$$

$$\begin{aligned}\underline{y}_0(t, \alpha) &= \underline{y}_0(\alpha) + \int_1^t 2 \underline{y}(1, \alpha)(s-1) ds \\ \bar{y}_0(t, \alpha) &= \bar{y}_0(\alpha) + \int_1^t 2 \bar{y}(1, \alpha)(s-1) ds\end{aligned}$$

$$\underline{y}_0(t, \alpha) = \underline{y}(1, \alpha)((t-1)^2 + 1), \quad \bar{y}_0(t, \alpha) = \bar{y}(1, \alpha)((t-1)^2 + 1)$$

Approximating  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\underline{\phi}_6(t) = \sum_{k=0}^6 \underline{y}_k(t, \alpha), \quad \bar{\phi}_6(t) = \sum_{k=0}^6 \bar{y}_k(t, \alpha)$$

$$\underline{y}_1(t, \alpha) = \int_1^t \underline{y}_0(s, \alpha) ds, \quad \bar{y}_1(t, \alpha) = \int_1^t \bar{y}_0(s, \alpha) ds$$

$$\underline{y}_1(t, \alpha) = \underline{y}(1, \alpha) \left( \frac{(t-1)^3}{3} + t - 1 \right), \quad \bar{y}_1(t, \alpha) = \bar{y}(1, \alpha) \left( \frac{(t-1)^3}{3} + t - 1 \right)$$

$$\underline{y}_2(t, \alpha) = \underline{y}(1, \alpha) \left( \frac{(t-1)^4}{12} + \frac{(t-1)^2}{2} \right), \quad \bar{y}_2(t, \alpha) = \bar{y}(1, \alpha) \left( \frac{(t-1)^4}{12} + \frac{(t-1)^2}{2} \right)$$

⋮

$$\underline{y}_6(t, \alpha) = \underline{y}(1, \alpha) \left( \frac{(t-1)^8}{20160} + \frac{(t-1)^6}{720} \right), \quad \bar{y}_6(t, \alpha) = \bar{y}(1, \alpha) \left( \frac{(t-1)^8}{20160} + \frac{(t-1)^6}{720} \right)$$

Then

$$\begin{aligned}\underline{\phi}_6(t) &= \sum_{k=0}^6 \underline{y}(1, \alpha) \left( \frac{2(t-1)^{k+2}}{(k+2)!} + \frac{(t-1)^k}{(k)!} \right) \\ \bar{\phi}_6(t) &= \sum_{k=0}^6 \bar{y}(1, \alpha) \left( \frac{2(t-1)^{k+2}}{(k+2)!} + \frac{(t-1)^k}{(k)!} \right)\end{aligned}$$

To find exact solution

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{\phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\phi}_n(t)$$

$$\underline{y}(t, \alpha) = \underline{y}(1, \alpha)(3e^{t-1} - 2t), \quad \bar{y}(t, \alpha) = \bar{y}(1, \alpha)(3e^{t-1} - 2t)$$

when  $t \in [1.5, 2]$  we have

$$\underline{y}_0(\alpha) = y(1.5, \alpha) = [\underline{y}(1, \alpha)(3e^{0.5} - 3), \bar{y}(1, \alpha)(3e^{0.5} - 3)]$$

$$\begin{aligned}\underline{y}_0(t, \alpha) &= \underline{y}_0(\alpha) + \int_{1.5}^t 2\underline{y}(1, \alpha)(2-s)ds, & \bar{y}_0(t, \alpha) &= \bar{y}_0(\alpha) + \int_{1.5}^t 2\bar{y}(1, \alpha)(2-s)ds \\ \underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) + 2\underline{y}(1, \alpha)\left(2s - \frac{s^2}{2}|_{1.5}^t\right), \\ \bar{y}_0(t, \alpha) &= \bar{y}(1.5, \alpha) + 2\bar{y}(1, \alpha)\left(2s - \frac{s^2}{2}|_{1.5}^t\right)\end{aligned}$$

$$\begin{aligned}\underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) - \underline{y}(1, \alpha)((t-1.5)^2 - (t-1.5)), \\ \bar{y}_0(t, \alpha) &= \bar{y}(1.5, \alpha) - \bar{y}(1, \alpha)((t-1.5)^2 - (t-1.5))\end{aligned}$$

Approximating  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\begin{aligned}\underline{\phi}_6(t) &= \sum_{k=0}^6 \underline{y}_k(t, \alpha), & \bar{\phi}_6(t) &= \sum_{k=0}^6 \bar{y}_k(t, \alpha) \\ \underline{y}_1(t, \alpha) &= \int_{1.5}^t \underline{y}_0(s, \alpha)ds, & \bar{y}_1(t, \alpha) &= \int_{1.5}^t \bar{y}_0(s, \alpha)ds \\ \underline{y}_1(t, \alpha) &= \int_{1.5}^t \left(\underline{y}(1.5, \alpha) - \underline{y}(1, \alpha)((s-1.5)^2 - (s-1.5))\right)ds \\ \underline{y}_1(t, \alpha) &= \underline{y}(1.5, \alpha)(t-1.5) - \underline{y}(1, \alpha)\left(\frac{(t-1.5)^3}{3} - \frac{(t-1.5)^2}{2}\right) \\ \bar{y}_1(t, \alpha) &= \bar{y}(1.5, \alpha)(t-1.5) - \bar{y}(1, \alpha)\left(\frac{(t-1.5)^3}{3} - \frac{(t-1.5)^2}{2}\right) \\ \underline{y}_2(t, \alpha) &= \underline{y}(1.5, \alpha)\frac{(t-1.5)^2}{2} - \underline{y}(1, \alpha)\left(\frac{(t-1.5)^4}{12} - \frac{(t-1.5)^3}{6}\right) \\ \bar{y}_2(t, \alpha) &= \bar{y}(1.5, \alpha)\frac{(t-1.5)^2}{2} - \bar{y}(1, \alpha)\left(\frac{(t-1.5)^4}{12} - \frac{(t-1.5)^3}{6}\right) \\ &\vdots \\ \underline{y}_6(t, \alpha) &= \underline{y}(1.5, \alpha)\frac{(t-1.5)^6}{720} - \underline{y}(1, \alpha)\left(\frac{(t-1.5)^8}{20160} - \frac{(t-1.5)^7}{5040}\right)\end{aligned}$$

$$\bar{y}_6(t, \alpha) = \bar{y}(1.5, \alpha) \frac{(t - 1.5)^6}{720} - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^8}{20160} - \frac{(t - 1.5)^7}{5040} \right)$$

Then

$$\underline{\phi}_6(t) = \sum_{k=0}^6 \underline{y}(1.5, \alpha) \frac{(t - 1.5)^k}{k!} - \underline{y}(1, \alpha) \left( \frac{2(t - 1.5)^{k+2}}{(k+2)!} - \frac{(t - 1.5)^{k+1}}{(k+1)!} \right)$$

$$\bar{\phi}_6(t) = \sum_{k=0}^6 \bar{y}(1.5, \alpha) \frac{(t - 1.5)^k}{k!} - \bar{y}(1, \alpha) \left( \frac{2(t - 1.5)^{k+2}}{(k+2)!} - \frac{(t - 1.5)^{k+1}}{(k+1)!} \right)$$

To find exact solution

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{\phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\phi}_n(t)$$

$$\underline{y}(t, \alpha) = \bar{y}(1, \alpha)(2t - 2 + 3e^{t-1} - 4e^{t-1.5})$$

$$\bar{y}(t, \alpha) = \bar{y}(1, \alpha)(2t - 2 + 3e^{t-1} - 4e^{t-1.5})$$

b- Triangular Shaped Fuzzy Number

$$y(0, \alpha) = [0.75 + 0.25\alpha^2, 1.125 - 0.125\alpha^2], \quad 0 \leq \alpha \leq 1$$

According to Equations (49), when  $t \in [0,1]$ :

$$\begin{aligned} \underline{y}_0(t, \alpha) &= y_0(\alpha), & \bar{y}_0(t, \alpha) &= \bar{y}_0(\alpha) \\ \underline{y}_0(t, \alpha) &= 0.5 + 0.25\alpha^2, & \bar{y}_0(t, \alpha) &= 1.125 - 0.125\alpha^2 \end{aligned}$$

Approximating  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\underline{\phi}_6(t) = \sum_{k=0}^6 \underline{y}_k(t, \alpha), \quad \bar{\phi}_6(t) = \sum_{k=0}^6 \bar{y}_k(t, \alpha)$$

$$\underline{y}_1(t, \alpha) = \int_0^t \underline{y}_0(s, \alpha) ds, \quad \bar{y}_1(t, \alpha) = \int_0^t \bar{y}_0(s, \alpha) ds$$

$$\underline{y}_1(t, \alpha) = (0.5 + 0.25\alpha^2)t, \quad \bar{y}_1(t, \alpha) = (1.125 - 0.125\alpha^2)t$$

$$\underline{y}_2(t, \alpha) = (0.5 + 0.25\alpha^2) \frac{t^2}{2}, \quad \bar{y}_2(t, \alpha) = (1.125 - 0.125\alpha^2) \frac{t^2}{2}$$

⋮

$$\underline{y}_6(t, \alpha) = (0.5 + 0.25\alpha^2) \frac{t^6}{720}, \quad \bar{y}_6(t, \alpha) = (1.125 - 0.125\alpha^2) \frac{t^6}{720}$$

$$\underline{\phi}_6(t) = \sum_{k=0}^6 \underline{y}_0(t, \alpha) \frac{t^k}{k!}, \quad \bar{\phi}_6(t) = \sum_{k=0}^6 \bar{y}_0(t, \alpha) \frac{t^k}{k!}$$

To find exact solution

$$\begin{aligned}\underline{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \underline{\phi}_n(t), & \bar{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \bar{\phi}_n(t) \\ \underline{y}(t, \alpha) &= \underline{y}_0(t, \alpha) e^t, & \bar{y}(t, \alpha) &= \bar{y}_0(t, \alpha) e^t\end{aligned}$$

when  $t \in [1, 1.5]$  we have

$$\begin{aligned}y_0(\alpha) &= y(1, \alpha) = [(0.5 + 0.25\alpha^2)e^t, (1.125 - 0.125\alpha^2)e^t] \\ y_0(t, \alpha) &= \underline{y}_0(\alpha) + \int_1^t 2\underline{y}(1, \alpha)(s-1)ds, \quad \bar{y}_0(t, \alpha) = \bar{y}_0(\alpha) + \int_1^t 2\bar{y}(1, \alpha)(s-1)ds \\ \underline{y}_0(t, \alpha) &= \underline{y}(1, \alpha)((t-1)^2 + 1), \quad \bar{y}_0(t, \alpha) = \bar{y}(1, \alpha)((t-1)^2 + 1)\end{aligned}$$

Approximating  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\begin{aligned}\underline{\phi}_6(t) &= \sum_{k=0}^6 \underline{y}_k(t, \alpha), & \bar{\phi}_6(t) &= \sum_{k=0}^6 \bar{y}_k(t, \alpha) \\ \underline{y}_1(t, \alpha) &= \int_1^t \underline{y}_0(s, \alpha) ds, & \bar{y}_1(t, \alpha) &= \int_1^t \bar{y}_0(s, \alpha) ds \\ \underline{y}_1(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^3}{3} + t - 1 \right), & \bar{y}_1(t, \alpha) &= \bar{y}(1, \alpha) \left( \frac{(t-1)^3}{3} + t - 1 \right) \\ \underline{y}_2(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^4}{12} + \frac{(t-1)^2}{2} \right), & \bar{y}_2(t, \alpha) &= \bar{y}(1, \alpha) \left( \frac{(t-1)^4}{12} + \frac{(t-1)^2}{2} \right) \\ &\vdots \\ \underline{y}_6(t, \alpha) &= \underline{y}(1, \alpha) \left( \frac{(t-1)^8}{20160} + \frac{(t-1)^6}{720} \right), & \bar{y}_6(t, \alpha) &= \bar{y}(1, \alpha) \left( \frac{(t-1)^8}{20160} + \frac{(t-1)^6}{720} \right)\end{aligned}$$

Then

$$\underline{\phi}_6(t) = \sum_{k=0}^6 \underline{y}(1, \alpha) \left( \frac{2(t-1)^{k+2}}{(k+2)!} + \frac{(t-1)^k}{(k)!} \right)$$

$$\bar{\Phi}_6(t) = \sum_{k=0}^6 \bar{y}(1, \alpha) \left( \frac{2(t-1)^{k+2}}{(k+2)!} + \frac{(t-1)^k}{(k)!} \right)$$

To find exact solution

$$\underline{y}(t, \alpha) = \lim_{n \rightarrow \infty} \underline{\phi}_n(t), \quad \bar{y}(t, \alpha) = \lim_{n \rightarrow \infty} \bar{\phi}_n(t)$$

$$\underline{y}(t, \alpha) = \underline{y}(1, \alpha)(3e^{t-1} - 2t)$$

$$\bar{y}(t, \alpha) = \bar{y}(1, \alpha)(3e^{t-1} - 2t)$$

when  $t \in [1.5, 2]$  we have

$$y_0(\alpha) = y(1.5, \alpha) = [\underline{y}(1, \alpha)(3e^{0.5} - 3), \bar{y}(1, \alpha)(3e^{0.5} - 3)]$$

$$\underline{y}_0(t, \alpha) = \underline{y}_0(\alpha) + \int_{1.5}^t 2\underline{y}(1, \alpha)(2-s)ds, \quad s\bar{y}_0(t, \alpha) = \bar{y}_0(\alpha) + \int_{1.5}^t 2\bar{y}(1, \alpha)(2-s)ds$$

$$\begin{aligned} \underline{y}_0(t, \alpha) &= \underline{y}(1.5, \alpha) - \underline{y}(1, \alpha)((t-1.5)^2 - (t-1.5)) \\ \bar{y}_0(t, \alpha) &= \bar{y}(1.5, \alpha) - \bar{y}(1, \alpha)((t-1.5)^2 - (t-1.5)) \end{aligned}$$

Approximating  $\underline{y}(t, \alpha)$  and  $\bar{y}(t, \alpha)$ , with  $\underline{\phi}_6(t)$  and  $\bar{\phi}_6(t)$ , respectively, as follows:

$$\underline{\phi}_6(t) = \sum_{k=0}^6 \underline{y}_k(t, \alpha), \quad \bar{\phi}_6(t) = \sum_{k=0}^6 \bar{y}_k(t, \alpha)$$

$$\underline{y}_1(t, \alpha) = \int_{1.5}^t \underline{y}_0(s, \alpha)ds, \quad \bar{y}_1(t, \alpha) = \int_{1.5}^t \bar{y}_0(s, \alpha)ds$$

$$\underline{y}_1(t, \alpha) = \underline{y}(1.5, \alpha)(t-1.5) - \underline{y}(1, \alpha) \left( \frac{(t-1.5)^3}{3} - \frac{(t-1.5)^2}{2} \right)$$

$$\bar{y}_1(t, \alpha) = \bar{y}(1.5, \alpha)(t-1.5) - \bar{y}(1, \alpha) \left( \frac{(t-1.5)^3}{3} - \frac{(t-1.5)^2}{2} \right)$$

$$\underline{y}_2(t, \alpha) = \underline{y}(1.5, \alpha) \frac{(t-1.5)^2}{2} - \underline{y}(1, \alpha) \left( \frac{(t-1.5)^4}{12} - \frac{(t-1.5)^3}{6} \right)$$

$$\bar{y}_2(t, \alpha) = \bar{y}(1.5, \alpha) \frac{(t-1.5)^2}{2} - \bar{y}(1, \alpha) \left( \frac{(t-1.5)^4}{12} - \frac{(t-1.5)^3}{6} \right)$$

$$\begin{aligned} \vdots \\ \underline{y}_6(t, \alpha) &= \underline{y}(1.5, \alpha) \frac{(t - 1.5)^6}{720} - \underline{y}(1, \alpha) \left( \frac{(t - 1.5)^8}{20160} - \frac{(t - 1.5)^7}{5040} \right) \\ \bar{y}_6(t, \alpha) &= \bar{y}(1.5, \alpha) \frac{(t - 1.5)^6}{720} - \bar{y}(1, \alpha) \left( \frac{(t - 1.5)^8}{20160} - \frac{(t - 1.5)^7}{5040} \right) \end{aligned}$$

Then

$$\begin{aligned} \underline{\Phi}_6(t) &= \sum_{k=0}^6 \underline{y}(1.5, \alpha) \frac{(t - 1.5)^k}{k!} - \underline{y}(1, \alpha) \left( \frac{2(t - 1.5)^{k+2}}{(k+2)!} - \frac{(t - 1.5)^{k+1}}{(k+1)!} \right) \\ \bar{\Phi}_6(t) &= \sum_{k=0}^6 \bar{y}(1.5, \alpha) \frac{(t - 1.5)^k}{k!} - \bar{y}(1, \alpha) \left( \frac{2(t - 1.5)^{k+2}}{(k+2)!} - \frac{(t - 1.5)^{k+1}}{(k+1)!} \right) \end{aligned}$$

To find exact solution

$$\begin{aligned} \underline{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \underline{\Phi}_n(t), & \bar{y}(t, \alpha) &= \lim_{n \rightarrow \infty} \bar{\Phi}_n(t) \\ \underline{y}(t, \alpha) &= \underline{y}(1, \alpha)(2t - 2 + 3e^{t-1} - 4e^{t-1.5}) \\ \bar{y}(t, \alpha) &= \bar{y}(1, \alpha)(2t - 2 + 3e^{t-1} - 4e^{t-1.5}) \end{aligned}$$

## 4.8 Summary

An example of the HFDEs was solved by several numerical methods and it is the first time the trapezoidal and triangular shaped fuzzy number were used as initial conditions. A Matlab code was constructed for each numerical method. Then exact and approximate solutions were compared under Hukuhara derivative. Finally, the results were compared for the used numerical methods.

## Chapter Five

### Conclusion and Comments

In this work, we solved a hybrid fuzzy differential equation by several numerical methods (Picard method, Runge-Kutta of order five, General linear method (GLM), Variational iteration method (VIM), Adomian decomposition method (ADM), Predictor-Corrector method (PCM) and Improved Predictor-Corrector (IPC) method) and different cases of initial conditions as triangular, trapezoidal and triangular shaped fuzzy numbers. Also, we solved HFDEs with generalized Hukuhara derivative.

Now to discuss the results and give some interpretation, we found out that. Picard method gives high accurate results for each initial conditions. However, these accurate results are obtained with high number of iterations which require more CPU time by Matlab software.

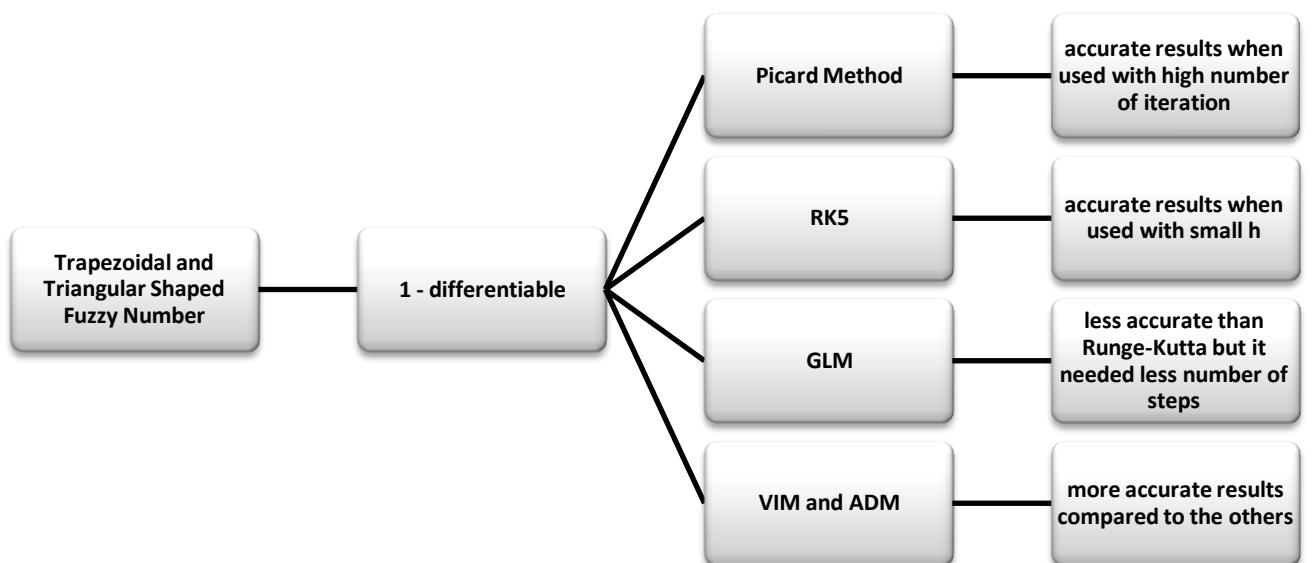
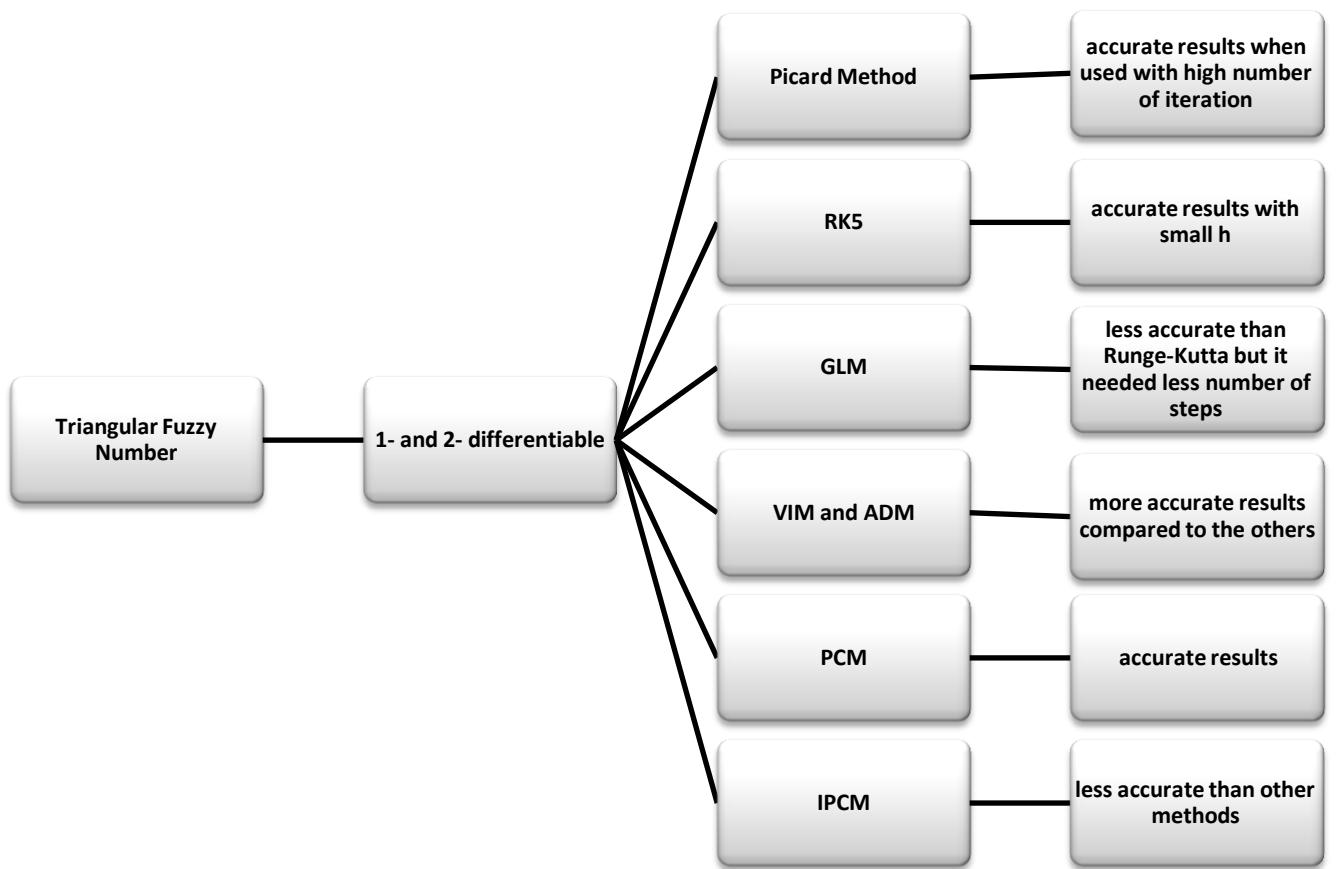
Although Runge-Kutta of order five is rarely used by researchers, however, it gives accurate results with small  $h$ . One can't compare between Picard method and Runge-Kutta method because the first depends on iteration but the second depends on the value of  $h$ .

The general linear method was applied using Runge-Kutta and multistep type methods. The results for this method were less accurate than classical 4<sup>th</sup> order Runge-Kutta but it needed less number of steps compared with Runge-Kutta 4<sup>th</sup> order.

Variational iteration and Adomian decomposition methods give exact solutions under generalized Hukuhara derivative (1 –derivative, 2 –derivative).

We applied predictor-corrector method using Milne's 4<sup>th</sup> order and Adams-Bashforth 4<sup>th</sup> order and obtained accurate results. To find initial conditions for the method, we utilized the 4<sup>th</sup> order Runge-Kutta method.

Applying the improved predictor-corrector method to fuzzy equations, the obtained results were less accurate than previous methods.



## References

- [1] S. Abbasbandy and T. Allahviranloo (2004), Numerical solution of Fuzzy differential equation by Runge-Kutta method, Nonlinear Studies 11 , 117–129.
- [2] S. Abbasbandy and T. Allahviranloo (2002), Numerical solution of fuzzy differential equation by Talor method, Journal of Computational Methods in Applied Mathematics 2 , 113–124.
- [3] S. Abbasbandy, T. Allahviranloo, O. Lopez-Pouso and J.J. Nieto (2004), Numerical methods for fuzzy differential inclusions, Computer and Mathematics With Applications 48, 1633-1641.
- [4] T. Allahviranlo and S. Gholami (2012) , Note on "Generalized Hukuhara differentiability of interval-valued functions and interval differential equations".
- [5] T. Allahviranloo and L. Jamshidi (2009), Solution of fuzzy differential equations under generalized differentiability by Adomian decomposition method, Iranian Journal of Optimization 1, 57-75.
- [6] A. Q. Ansari(1998), The Basics of Fuzzy Logic: A Tutorial Review, Article in Computer Education - Stafford - Computer Education Group.
- [7] H. Bandemer and S. Gottwald (1994), Fuzzy Sets, Fuzzy Logic,Fuzzy Methods with Applications, Freiberg/Leipzig, 1-55.
- [8] B. Bede and S.G. Gal(2005), Generalizations of the differentiability of fuzzy number-valued functions with applications to fuzzy differential equations, Fuzzy Sets Syst. 151, 581-599.
- [9] B. Bede and L. Stefanini ( 2012), Generalized Differentiability of Fuzzy-valued Functions, Working Papers Series in Economics, Mathematics and Statistics, ISSN 1974-4110.
- [10] T. A. Bidgoli and O. S. Fard (2011), The Nystrom method for hybrid fuzzy differential equation IVPs, Journal of King Saud University, 371-379.
- [11]J.J. Buckley and L.J. Jowers (2005), Studies in Fuzziness and Soft Computing, Systems Research Institute, Vol 188.
- [12] R. L. Burden and J. D. Faires (2011), Numerical Analysis, Ninth Edition.

- [13] J. Butcher (2008), Numerical Methods for Ordinary Differential Equations, Copyright John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, Second Edition.
- [14] J. Butcher (2006) , General linear methods for ordinary differential equations, The University of Auckland, New Zealand, 1-46.
- [15] J. Butcher (2005) , General linear methods, The University of Auckland, New Zealand, Stellenbosch, SANUM.
- [16] Y. C. Cano and H. R. Flores (2006), On new solutions of fuzzy differential equations, Chaos Solitons Fractals 38 , 112-119.
- [17] S.L. Chang and L.A. Zadeh (1972), On fuzzy mapping and control, IEEE Trans. Syst Man Cybern., 2, 30–34 .
- [18] M. Clement Joe Anand and J. Bharatraj (2017), Theory Triangular Fuzzy Number, Proceedings of NCATM, ISBN: 978 - 93 - 85126 - 14 – 7, 80-83
- [19] B. C. Dhage and N. S. Jadhav (2013), Basic Results in the Theory of Hybrid Differential Equations with Linear Perturbations of Second Type, TAMKANG JOURNAL OF MATHEMATICS, Vol 44, Number 2, 171-186
- [20] J.G. Dijkman, H. Van Haeringen and S.J.De Lange(1983), Fuzzy Numbers, Journal of Mathematical analysis and applications 92, 301-341.
- [21] D. Dubois and H. Prade (1982) Towards fuzzy differential. Fuzzy Sets and Systems 8, 1-17.
- [22] J. Farzi, A. Moradi (2018), Fuzzy General Linear Methods, Sahand University Of Technology, Tabriz, iran.
- [23] M. C. Floreno , G. Novelli (1996), Implementing Fuzzy Polynomial interpolation (FPI) and Fuzzy Linear Regression, Le Matematiche .
- [24] M. Ghanbari (2012), Solution of the first order linear fuzzy differential equations by some reliable methods, Aliabad Katoul Branch, Islamic Azad University, Vol 2012, Article ID jfsva-00126.
- [25] L. T. Gomes, L. C.de Barros, B. Bede (2015), Fuzzy Differential Equations in Various Approaches.

- [26] W.P.M.H. Heemels, D. Lehmann, J. Lunze, and B. De Schutter (2009), “Introduction to hybrid systems,” Chapter 1 in Handbook of Hybrid Systems Control – Theory, Tools• Applications (J. Lunze and F. Lamnabhi-Lagarrigue, eds.), Cambridge, UK: Cambridge University Press, ISBN 978-0-521-76505-3, 3–30.
- [27] K. Ivaz, A. Khastan, and J. J. Nieto (2012) , A Numerical Method for Fuzzy Differential Equations and Hybrid Fuzzy Differential Equations, Abstract and Applied Analysis, Article ID 735128.
- [28] T. Jayakumar (2012), Numerical Solution for Hybrid Fuzzy System by Runge-Kutta Method of Order Five, Applied Mathematical Sciences, Vol. 6, no. 72, 3591-3606.
- [29] T. Jayakumar, K. Kanagarajan and T. Muthukumar (2014), Numerical Solution for Hybrid Fuzzy System by Adams fourth order predictor-corrector method, Department of Mathematics, Sri Ramakrishna Mission Vidyalaya College of Arts and Science, Coimbatore-641 020, Tamilnadu, India.
- [30] T. Jayakumar, D. Maheskumar and K. Kanagarajan (2012), Numerical Solution of Fuzzy Differential Equations by Runge Kutta Method of Order Five, Vol. 6, no. 60, 2989 - 3002.
- [31] T. Jayakumar ,T. Muthukumar and K. Kanagarajan (2015), Numerical Solution for Hybrid Fuzzy System by Adams fourth order predictor-corrector method.
- [32] O.Kaleva (1987) Fuzzy Differential Equations. Fuzzy Sets and Systems 24, 301-317
- [33] O. Kaleva(1990), The Cauchy problem for fuzzy differential equations, Fuzzy Sets and Systems 35, 389-396.
- [34] K. Kanagarajan, S. Indrakumar and S. Muthukumar (2015), Numerical Solution of Hybrid Fuzzy differential equations by Runge-Kutta Method of Order Five and the dependency problem, Annals of Fuzzy Mathematics and Informatics.
- [35] K. Kanagarajan and M.Sambath (2012), Numerical Solution of Hybrid Fuzzy differential equations by improved predictor-corrector method, Nonlinear studies,Vol. 19,No.2, 225-238.

- [36] K. Kanagarajan and R.Suresh (2013) , Numerical solution of fuzzy differential equations under generalized differentiability by Modified Euler method, Vol. 2, Issue 11.
- [37] K. Kanagarajan and D. Vivek (2016), Fifth order Runge-Kutta method for solving first-order fully fuzzy differential equations under strongly generalized H-differentiability, Vol. 2016, Issue 1.
- [38] A. Kandel and W. Byatt (1980). Fuzzy Processes, Fuzzy Sets and Systems, 117-152.
- [39] K.H. Lee (2004), First Course on Fuzzy Theory and Applications, Library of Congress Control Number: 2004112297, 1-150.
- [40] M. Ma, M. Friedman and A. Kandel (1999), Numerical solutions of fuzzy differential equations, *Fuzzy Sets and Systems* 105 133–138.
- [41] S. Mathankumar and S. Narayananamoorthy (2017), Hybrid Fuzzy Differential Equations-Picard's Method, *International Journal of Pure and Applied Mathematics*, Vol.113, No.12, 133-141.
- [42] S. Narayananamoorthy , S.P. Sathiyapriya , M. Santhiya (2016), Solving hybrid fuzzy differential equations using Taylor series method, Vol. 106, No. 5.
- [43] S. Natrsan, N. Venkatachalem and S.C. Pandian (2013), New Approach to Solve Hybrid Fuzzy Differential Equation, *Journal of Mathematics and statistic* 9 (3), 175-179.
- [44] G. Nhawu, P. Mafuta and J. Mushanyu (2016), The Adomian Decomposition Method for Numerical Solution of First-Order Differential Equations, *J. Math. Comput. Sci.* 6, No. 3, 307-314.
- [45] Dr. S.C. Pandian and N. Saveetha (2012) , Numerical Solution of Fuzzy Hybrid Differential Equations by Third order Runge Kutta Nystrom Method, *Mathematical Theory and Modeling*, Vol.2, No.4.
- [46] S. Pederson and M. Sambandham (2007), Numerical solution to hybrid fuzzy systems, *Mathematical and Computer Modelling* 45, 1133–1144.
- [47] S. Pederson and M. Sambandham (2008), The Runge-Kutta for Hybrid Fuzzy Differential Equations, *Nonlinear Analysis Hybrid Systems* 2, 626–634.

- [48] M. L. Puri, and D. A. Ralescu (1983) Differentials of fuzzy functions. J. of Math. Analysis and App. 91, 552-558.
- [49] T. J. Ross (2004), Fuzzy Logic With Engineering Applications, Third Edition, University of New Mexico, USA.
- [50] S. Salahshour (2011), Nth-order fuzzy differential equations under generalized differentiability, Journal of Fuzzy Set Valued Analysis, Vol. 2011, 1-14.
- [51] A.V.D. Schaft and H. Schumacher (1999), An Introduction to Hybrid Dynamical Systems, Enschede /Amsterdam.
- [52] S. Seikkala (1987) On the fuzzy initial value problem. Fuzzy Sets and Systems 24, 319-330.
- [53] H. Singh, M. M. Gupta, T, Meitzler, Z.G. Hou, K,Garg, A. M. G. Solo, and L. A. Zadeh (2013), Real-Life Applications of Fuzzy Logic , Advances in Fuzzy Systems.
- [54] A. Wazwaz (2014), The Variational iteration method for solving linear and nonlinear ODEs and scientific models with variable coefficients, Cent. Eur. J. Eng. 4(1) , 64-71.
- [55] L .A . Zadeh (1965), "Fuzzy sets", Information and Control 8 (3): 338–353.
- [56] H.-J. Zimmermann, Fuzzy Set Theory and Its Applications, Springer Science+Business Media, LLC, Fourth Edition, 11-21.

# Appendix

## a- 1 – differentiable

### Picard Method

```
+ syms t w v z r
format long
w0=zeros(1,11)
v0= zeros(1,11)
for i=1:11
w0(1,i)=0.75+0.25*(i-1)*0.1
v0(1,i)=1.125-0.125*(i-1)*0.1
end
t0=0
n=50
for k=1:11
for j=1:n
if j==1
w(k,j)=w0(1,k)+int(w0(1,k),t0,t)
v(k,j)=v0(1,k)+int(v0(1,k),t0,t)
else
w(k,j)=w0(1,k)+int(w(k,j-1),t0,t)
v(k,j)=v0(1,k)+int(v(k,j-1),t0,t)
end
z(t)=w(k,j)
r(t)=v(k,j)
end
z(1)
r(1)
end
+
syms t w v z r
format long
w0=[2.038711371344284,2.106668417055760,2.174625462767236,2.242582508
478713,2.310539554190189,2.378496599901665,2.446453645613141,2.514410
691324617,2.582367737036093,2.650324782747569,2.718281828459045]
v0=[3.058067057016426,3.024088534160688,2.990110011304950,2.956131488
449211,2.922152965593474,2.888174442737736,2.854195919881997,2.820217
397026260,2.786238874170521,2.752260351314783,2.718281828459045]
t0=1
n=50
for k=1:11
for j=1:n
if j==1
w(k,j)=w0(1,k)+int(w0(1,k)+2*w0(1,k)*(t-1),t0,t)
v(k,j)=v0(1,k)+int(v0(1,k)+2*v0(1,k)*(t-1),t0,t)
else
w(k,j)=w0(1,k)+int(w(k,j-1)+2*w0(1,k)*(t-1),t0,t)
v(k,j)=v0(1,k)+int(v(k,j-1)+2*v0(1,k)*(t-1),t0,t)
end
z(t)=w(k,j)
r(t)=v(k,j)
end
z(1.5)
r(1.5)
end
```

```

 $\ddot{w} = \frac{1}{2} [c_0(2-t) + w_0(1, k) + v_0(1, k)]$ 
 $\ddot{v} = \frac{1}{2} [d_0(2-t) + v_0(1, k) + w_0(1, k)]$ 
 $w(t) = w_0(1, k) + \int_{t_0}^t [\dot{w}(s) + 2 \cdot c_0(1, k) \cdot (2-s)] ds$ 
 $v(t) = v_0(1, k) + \int_{t_0}^t [\dot{v}(s) + 2 \cdot d_0(1, k) \cdot (2-s)] ds$ 

```

for k=1:n  
 for j=1:n  
 if j==1  
 w(k,j)=w0(1,k)+int(w0(1,k)+2\*c0(1,k)\*(2-t),t0,t)  
 v(k,j)=v0(1,k)+int(v0(1,k)+2\*d0(1,k)\*(2-t),t0,t)  
 else  
 w(k,j)=w0(1,k)+int(w(k,j-1)+2\*w0(1,k)\*(t-1),t0,t)  
 v(k,j)=v0(1,k)+int(v(k,j-1)+2\*v0(1,k)\*(t-1),t0,t)  
 end  
 z(t)=w(k,j)  
 r(t)=v(k,j)  
 end  
 z(2)  
 r(2)  
end

## Runge-Kutta Method

```

 $\ddot{w} = \frac{1}{2} [f_1(t_0, w_0) + f_2(t_0, v_0)]$ 
 $\ddot{v} = \frac{1}{2} [f_2(t_0, v_0) + f_1(t_0, w_0)]$ 
 $w(t) = w_0 + \int_{t_0}^t [\dot{w}(s) + f_1(s, w(s))] ds$ 
 $v(t) = v_0 + \int_{t_0}^t [\dot{v}(s) + f_2(s, v(s))] ds$ 

```

for i=1:n  
 w0(1,i)=0.75+0.25\*(i-1)\*0.1  
 z0(1,i)=1.125-0.125\*(i-1)\*0.1  
 end  
 h=0.02  
 t0=0  
 n=50  
 f1(t,w)=w  
 f2(t,v)=v  
 for i=1:n  
 t(i)=t0+i\*h  
 end  
 for j=1:n  
 if j==1  
 for k=1:11  
 k1=h\*f1(t0,w0(1,k))  
 k11=h\*f2(t0,v0(1,k))  
 k2=h\*f1(t0+(h/3),w0(1,k)+(k1/3))  
 k22=h\*f2(t0+(h/3),v0(1,k)+(k11/3))  
 end  
 end  
 end
end

```

k3=h*f1(t0+(h/3),w0(1,k)+(k1/6)+(k2/6))
k33=h*f2(t0+(h/3),v0(1,k)+(k11/6)+(k22/6))
k4=h*f1(t0+(h/2),w0(1,k)+(k1/8)+(3*k3/8))
k44=h*f2(t0+(h/2),v0(1,k)+(k11/8)+(3*k33/8))
k5=h*f1(t0+h,w0(1,k)+(k1/2)-(3*k3/2)+(2*k4))
k55=h*f2(t0+h,v0(1,k)+(k11/2)-(3*k33/2)+(2*k44))
w(j,k)=w0(1,k)+(1/6)*(k1+4*k4+k5)
v(j,k)=v0(1,k)+(1/6)*(k11+4*k44+k55)
end
else
for k=1:11
k1=h*f1(t(j-1),w(j-1,k))
k11=h*f2(t(j-1),v(j-1,k))
k2=h*f1(t(j-1)+(h/3),w(j-1,k)+(k1/3))
k22=h*f2(t(j-1)+(h/3),v(j-1,k)+(k11/3))
k3=h*f1(t(j-1)+(h/3),w(j-1,k)+(k1/6)+(k2/6))
k33=h*f2(t(j-1)+(h/3),v(j-1,k)+(k11/6)+(k22/6))
k4=h*f1(t(j-1)+(h/2),w(j-1,k)+(k1/8)+(3*k3/8))
k44=h*f2(t(j-1)+(h/2),v(j-1,k)+(k11/8)+(3*k33/8))
k5=h*f1(t(j-1)+h,w(j-1,k)+(k1/2)-(3*k3/2)+(2*k4))
k55=h*f2(t(j-1)+h,v(j-1,k)+(k11/2)-(3*k33/2)+(2*k44))
w(j,k)=w(j-1,k)+(1/6)*(k1+4*k4+k5)
v(j,k)=v(j-1,k)+(1/6)*(k11+4*k44+k55)
end
end
end
ww=double(w)
vv=double(v)

```



```

syms t w v
format long
w0=[2.038711370891301,2.106668416587678,2.174625462284054,2.242582507
980431,2.310539553676808,2.378496599373185,2.446453645069561,2.514410
690765938,2.582367736462314,2.650324782158691,2.718281827855068]
v0=[3.058067056336951,3.024088533488763,2.990110010640575,2.956131487
792387,2.922152964944198,2.888174442096010,2.854195919247821,2.820217
396399633,2.786238873551445,2.752260350703256,2.718281827855068]
h=0.02
t0=1
n=25
f1(t,w)=w+2*w0*(t-1)
f2(t,v)=v+2*v0*(t-1)
for i=1:n
t(i)=t0+i*h
end
for j=1:n
if j==1
for k=1:11
k1=h*f1(t0,w0(1,k))
k11=h*f2(t0,v0(1,k))
k2=h*f1(t0+(h/3),w0(1,k)+(k1(1,k)/3))
k22=h*f2(t0+(h/3),v0(1,k)+(k11(1,k)/3))
k3=h*f1(t0+(h/3),w0(1,k)+(k1(1,k)/6)+(k2(1,k)/6))
k33=h*f2(t0+(h/3),v0(1,k)+(k11(1,k)/6)+(k22(1,k)/6))
k4=h*f1(t0+(h/2),w0(1,k)+(k1(1,k)/8)+(3*k3(1,k)/8))
k44=h*f2(t0+(h/2),v0(1,k)+(k11(1,k)/8)+(3*k33(1,k)/8))

k5=h*f1(t0+h,w0(1,k)+(k1(1,k)/2)-(3*k3(1,k)/2)+(2*k4(1,k)))
k55=h*f2(t0+h,v0(1,k)+(k11(1,k)/2)-(3*k33(1,k)/2)+(2*k44(1,k)))
w(j,k)=w0(1,k)+(1/6)*(k1(1,k)+4*k4(1,k)+k5(1,k))
v(j,k)=v0(1,k)+(1/6)*(k11(1,k)+4*k44(1,k)+k55(1,k))

```

```

        end
    else
        for k=1:11
            k1=h*f1(t(j-1),w(j-1,k))
            k11=h*f2(t(j-1),v(j-1,k))
            k2=h*f1(t(j-1)+(h/3),w(j-1,k)+(k1(1,k)/3))
            k22=h*f2(t(j-1)+(h/3),v(j-1,k)+(k11(1,k)/3))
            k3=h*f1(t(j-1)+(h/3),w(j-1,k)+(k1(1,k)/6)+(k2(1,k)/6))
            k33=h*f2(t(j-1)+(h/3),v(j-1,k)+(k11(1,k)/6)+(k22(1,k)/6))
            k4=h*f1(t(j-1)+(h/2),w(j-1,k)+(k1(1,k)/8)+(3*k3(1,k)/8))
            k44=h*f2(t(j-1)+(h/2),v(j-1,k)+(k11(1,k)/8)+(3*k33(1,k)/8))
            k5=h*f1(t(j-1)+h,w(j-1,k)+(k1(1,k)/2)-(3*k3(1,k)/2)+(2*k4(1,k)))
            k55=h*f2(t(j-1)+h,v(j-1,k)+(k11(1,k)/2)-(3*k33(1,k)/2)+(2*k44(1,k)))
            w(j,k)=w(j-1,k)+(1/6)*(k1(1,k)+4*k4(1,k)+k5(1,k))
            v(j,k)=v(j-1,k)+(1/6)*(k11(1,k)+4*k44(1,k)+k55(1,k))
        end
    end
ww=double(w)
vv=double(v)

#  

syms t w v  

format long  

c0=[2.038711370891301,2.106668416587678,2.174625462284054,2.242582507  

980431,2.310539553676808,2.378496599373185,2.446453645069561,2.514410  

690765938,2.582367736462314,2.650324782158691,2.718281827855068]  

d0=[3.058067056336951,3.024088533488763,2.990110010640575,2.956131487  

792387,2.922152964944198,2.888174442096010,2.854195919247821,2.820217  

396399633,2.786238873551445,2.752260350703256,2.718281827855068]  

w0=[3.967666292225951,4.099921835300150,4.232177378374347,4.364432921  

448546,4.496688464522745,4.628944007596944,4.761199550671140,4.893455  

093745340,5.025710636819537,5.157966179893736,5.290221722967934]  

v0=[5.951499438338925,5.885371666801826,5.819243895264729,5.753116123  

727630,5.686988352190530,5.620860580653431,5.554732809116330,5.488605  

037579232,5.422477266042134,5.356349494505033,5.290221722967934]  

h=0.02  

t0=1.5  

n=25  

f1(t,w)=w+2*w0*(2-t)  

f2(t,v)=v+2*v0*(2-t)  

for i=1:n  

    t(i)=t0+i*h  

end  

for j=1:n  

if j==1  

    for k=1:11
        k1=h*f1(t0,w0(1,k))
        k11=h*f2(t0,v0(1,k))
        k2=h*f1(t0+(h/3),w0(1,k)+(k1(1,k)/3))
        k22=h*f2(t0+(h/3),v0(1,k)+(k11(1,k)/3))
        k3=h*f1(t0+(h/3),w0(1,k)+(k1(1,k)/6)+(k2(1,k)/6))
        k33=h*f2(t0+(h/3),v0(1,k)+(k11(1,k)/6)+(k22(1,k)/6))
        k4=h*f1(t0+(h/2),w0(1,k)+(k1(1,k)/8)+(3*k3(1,k)/8))
        k44=h*f2(t0+(h/2),v0(1,k)+(k11(1,k)/8)+(3*k33(1,k)/8))

        k5=h*f1(t0+h,w0(1,k)+(k1(1,k)/2)-(3*k3(1,k)/2)+(2*k4(1,k)))
        k55=h*f2(t0+h,v0(1,k)+(k11(1,k)/2)-(3*k33(1,k)/2)+(2*k44(1,k)))
        w(j,k)=w0(1,k)+(1/6)*(k1(1,k)+4*k4(1,k)+k5(1,k))
    end
end

```

```

v(j,k)=v0(1,k)+(1/6)*(k11(1,k)+4*k44(1,k)+k55(1,k))

    end
else
for k=1:11
k1=h*f1(t(j-1),w(j-1,k))
k11=h*f2(t(j-1),v(j-1,k))
k2=h*f1(t(j-1)+(h/3),w(j-1,k)+(k1(1,k)/3))
k22=h*f2(t(j-1)+(h/3),v(j-1,k)+(k11(1,k)/3))
k3=h*f1(t(j-1)+(h/3),w(j-1,k)+(k1(1,k)/6)+(k2(1,k)/6))
k33=h*f2(t(j-1)+(h/3),v(j-1,k)+(k11(1,k)/6)+(k22(1,k)/6))
k4=h*f1(t(j-1)+(h/2),w(j-1,k)+(k1(1,k)/8)+(3*k3(1,k)/8))
k44=h*f2(t(j-1)+(h/2),v(j-1,k)+(k11(1,k)/8)+(3*k33(1,k)/8))
k5=h*f1(t(j-1)+h,w(j-1,k)+(k1(1,k)/2)-(3*k3(1,k)/2)+(2*k4(1,k)))
k55=h*f2(t(j-1)+h,v(j-1,k)+(k11(1,k)/2)-
(3*k33(1,k)/2)+(2*k44(1,k)))
w(j,k)=w(j-1,k)+(1/6)*(k1(1,k)+4*k4(1,k)+k5(1,k))
v(j,k)=v(j-1,k)+(1/6)*(k11(1,k)+4*k44(1,k)+k55(1,k))
end
end
end
ww=double(w)
vv=double(v)

```

## General Linear Method

**K = 4**

```

syms t w v
format long
f1(t,w)=w
f2(t,v)=v
a=0
b=1
N=50
h=(b-a)/N
t(1)=a
for j=1:11
w0(1,j)=0.75+(0.25*(j-1)*0.1)
v0(1,j)=1.125-(0.125*(j-1)*0.1)
t(1)=a
for i=1:3
t(i+1)=t(i)+h
if i==1
k1(j,i)=h*f1(t(i),w0(1,j))
k11(j,i)=h*f2(t(i),v0(1,j))
k2(j,i)=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(j,i))
k22(j,i)=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(j,i))
k3(j,i)=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(j,i))
k33(j,i)=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(j,i))
k4(j,i)=h*f1(t(i+1),w0(1,j)+k3(j,i))
k44(j,i)=h*f2(t(i+1),v0(1,j)+k33(j,i))
w(j,i)=w0(1,j)+(k1(j,i)+2*(k2(j,i)+k3(j,i))+k4(j,i))/6
v(j,i)=v0(1,j)+(k11(j,i)+2*(k22(j,i)+k33(j,i))+k44(j,i))/6
else
k1(j,i)=h*f1(t(i),w(j,i-1))
k11(j,i)=h*f2(t(i),v(j,i-1))
k2(j,i)=h*f1(t(i)+0.5*h,w(j,i-1)+0.5*k1(j,i))

```

```

k22(j,i)=h*f2(t(i)+0.5*h,v(j,i-1)+0.5*k11(j,i))
k3(j,i)=h*f1(t(i)+0.5*h,w(j,i-1)+0.5*k2(j,i))
k33(j,i)=h*f2(t(i)+0.5*h,v(j,i-1)+0.5*k22(j,i))

k4(j,i)=h*f(t(i+1),w(j,i-1)+k3(j,i))
k44(j,i)=h*f(t(i+1),v(j,i-1)+k33(j,i))
w(j,i)=w(j,i-1)+(k1(j,i)+2*(k2(j,i)+k3(j,i))+k4(j,i))/6
v(j,i)=v(j,i-1)+(k11(j,i)+2*(k22(j,i)+k33(j,i))+k44(j,i))/6
end
end
for i=4:N
t0=a+i*h
p1(1,j)=55*f1(t(4),w(j,3))-59*f1(t(3),w(j,2))+37*f1(t(2),w(j,1))-
9*f1(t(1),w0(1,j))
d1(1,j)=55*f2(t(4),v(j,3))-59*f2(t(3),v(j,2))+37*f2(t(2),v(j,1))-
9*f2(t(1),v0(1,j))
z0(1,j)=w(j,3)+h*(p1(1,j))/24
q0(1,j)=v(j,3)+h*(d1(1,j))/24
for e=1:3
t(e)=t(e+1)
if e==1
w0(1,j)=w(j,1)
v0(1,j)=v(j,1)
else
w(j,e-1)=w(j,e)
v(j,e-1)=v(j,e)
end
end
t(4)=t0
w(j,3)=z0(1,j)
v(j,3)=q0(1,j)
end
ww=double(w)
vv=double(v)
end

```

```

syms t w v
format long
c0=[2.038711267561744,2.106668309813803,2.174625352065861,2.2425823943179,9
,2.310539436569978,2.378496478822036,2.446453521074094,2.514410563326151,2.
582367605578210,2.650324647830267,2.718281690082326]
f0=[3.058066901342616,3.024088380216586,2.990109859090558,2.956131337964530
,2.922152816838501,2.888174295712471,2.854195774586440,2.820217253460412,2.
786238732334387,2.752260211208354,2.718281690082326]
f1(t,w)=y+2*c0*(t-1)
f2(t,v)=y+2*f0*(t-1)
a=1
b=1.5
N=25
h=(b-a)/N
t(1)=a
for j=1:11
w0=[2.038711267561744,2.106668309813803,2.174625352065861,2.2425823943179,9
,2.310539436569978,2.378496478822036,2.446453521074094,2.514410563326151,2.
582367605578210,2.650324647830267,2.718281690082326]
v0=[3.058066901342616,3.024088380216586,2.990109859090558,2.956131337964530
,2.922152816838501,2.888174295712471,2.854195774586440,2.820217253460412,2.
786238732334387,2.752260211208354,2.718281690082326]
t(1)=a

```

```

for i=1:3
    t(i+1)=t(i)+h
    if i==1
        k1=h*f1(t(i),w0(1,j))
        k11=h*f2(t(i),v0(1,j))
        k2=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
        k22=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
        k3=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
        k33=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
        k4=h*f1(t(i+1),w0(1,j)+k3(1,j))
        k44=h*f2(t(i+1),v0(1,j)+k33(1,j))
        w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
        v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

    else
        k1=h*f1(t(i),w(1,i-1))
        k11=h*f2(t(i),v(1,i-1))
        k2=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
        k22=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
        k3=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
        k33=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
        k4=h*f1(t(i+1),w(1,i-1)+k3(1,j))
        k44=h*f2(t(i+1),v(1,i-1)+k33(1,j))
        w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
        v(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

    end
end
for i=4:N
    t0=a+i*h
    p1=55*f1(t(4),w(1,3))-59*f1(t(3),w(1,2))+37*f1(t(2),w(1,1))-
    9*f1(t(1),w0(1,j))
    d1=55*f2(t(4),v(1,3))-59*f2(t(3),v(1,2))+37*f2(t(2),v(1,1))-
    9*f2(t(1),v0(1,j))
    z0(1,j)=w(1,3)+h*(p1(1,j))/24
    q0(1,j)=v(1,3)+h*(d1(1,j))/24
    for e=1:3
        t(e)=t(e+1)
        if z==1
            w0(1,j)=w(1,1)
            v0(1,j)=v(1,1)
        else
            w(1,e-1)=w(1,e)
            v(1,e-1)=v(1,e)
        end
    end
    t(4)=t0
    w(1,3)=z0(1,j)
    v(1,3)=q0(1,j)
end
ww=double(w)
vv=double(v)
end

```

```

syms t w v
format long
c0=[2.038711267561744,2.106668309813803,2.174625352065861,2.242582394
3179,9,2.310539436569978,2.378496478822036,2.446453521074094,2.514410
563326151,2.582367605578210,2.650324647830267,2.718281690082326]
f0=[3.058066901342616,3.024088380216586,2.990109859090558,2.956131337
964530,2.922152816838501,2.888174295712471,2.854195774586440,2.820217
253460412,2.786238732334387,2.752260211208354,2.718281690082326]
f1(t,w)=y+2*c0*(2-t)
f2(t,v)=y+2*f0*(2-t)
a=1.5
b=2
N=25
h=(b-a)/N
t(1)=a
for j=1:11
w0=[3.967665851557501,4.099921379942754,4.232176908328005,4.364432436
713252,4.496687965098505,4.628943493483756,4.761199021869006,4.893454
550254253,5.025710078639504,5.157965607024751,5.290221135410005]
v0=[5.951498777336254,5.885371013143627,5.819243248951001,5.753115484
758379,5.686987720565753,5.620859956373129,5.554732192180500,5.488604
427987878,5.422476663795258,5.356348899602628,5.290221135410005]
t(1)=a
for i=1:3
t(i+1)=t(i)+h
if i==1
k1=h*f1(t(i),w0(1,j))
k11=h*f2(t(i),v0(1,j))
k2=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
k22=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
k3=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
k33=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
k4=h*f1(t(i+1),w0(1,j)+k3(1,j))
k44=h*f2(t(i+1),v0(1,j)+k33(1,j))
w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
else
k1=h*f1(t(i),w(1,i-1))
k11=h*f2(t(i),v(1,i-1))
k2=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
k22=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
k3=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
k33=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
k4=h*f1(t(i+1),w(1,i-1)+k3(1,j))
k44=h*f2(t(i+1),v(1,i-1)+k33(1,j))
w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
w(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
end
end
for i=4:N
t0=a+i*h
p1=55*f1(t(4),w(1,3))-59*f1(t(3),w(1,2))+37*f1(t(2),w(1,1))-
9*f1(t(1),w0(1,j))
d1=55*f2(t(4),v(1,3))-59*f2(t(3),v(1,2))+37*f2(t(2),v(1,1))-
9*f2(t(1),v0(1,j))
z0(1,j)=w(1,3)+h*(p1(1,j))/24

```

```

q0(1,j)=v(1,3)+h*(d1(1,j))/24

for e=1:3
    t(e)=t(e+1)
    if z==1
        w0(1,j)=w(1,1)
        v0(1,j)=v(1,1)
    else
        w(1,e-1)=w(1,e)
        v(1,e-1)=v(1,e)
    end
end
t(4)=t0
w(1,3)=z0(1,j)
v(1,3)=q0(1,j)
end
ww=double(w)
vv=double(v)
end

```

## **K = 5**

```

syms t w v
format long
f1(t,w)=w
f2(t,v)=v
a=0
b=1
N=50
h=(b-a)/N
t(1)=a
for j=1:11
    w0(1,j)=0.75+(0.25*(j-1)*0.1)
    v0(1,j)=1.125-(0.125*(j-1)*0.1)
    t(1)=a
for i=1:4
    t(i+1)=t(i)+h
    if i==1
        k1(j,i)=h*f1(t(i),w0(1,j))
        k11(j,i)=h*f2(t(i),v0(1,j))
        k2(j,i)=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(j,i))
        k22(j,i)=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(j,i))
        k3(j,i)=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(j,i))
        k33(j,i)=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(j,i))
        k4(j,i)=h*f1(t(i+1),w0(1,j)+k3(j,i))
        k44(j,i)=h*f2(t(i+1),v0(1,j)+k33(j,i))
        w(j,i)=w0(1,j)+(k1(j,i)+2*(k2(j,i)+k3(j,i))+k4(j,i))/6
        v(j,i)=v0(1,j)+(k11(j,i)+2*(k22(j,i)+k33(j,i))+k44(j,i))/6
    else
        k1(j,i)=h*f1(t(i),w(j,i-1))
        k11(j,i)=h*f2(t(i),v(j,i-1))
        k2(j,i)=h*f1(t(i)+0.5*h,w(j,i-1)+0.5*k1(j,i))
        k22(j,i)=h*f2(t(i)+0.5*h,v(j,i-1)+0.5*k11(j,i))
        k3(j,i)=h*f1(t(i)+0.5*h,w(j,i-1)+0.5*k2(j,i))
        k33(j,i)=h*f2(t(i)+0.5*h,v(j,i-1)+0.5*k22(j,i))

        k4(j,i)=h*f1(t(i+1),w(j,i-1)+k3(j,i))
        k44(j,i)=h*f2(t(i+1),v(j,i-1)+k33(j,i))
        w(j,i)=w(j,i-1)+(k1(j,i)+2*(k2(j,i)+k3(j,i))+k4(j,i))/6
    end
end

```

```

v(j,i)=v(j,i-1)+(k11(j,i)+2*(k22(j,i)+k33(j,i))+k44(j,i))/6
end
end
for i=5:N
t0=a+i*h
p1(1,j)=1901*f1(t(5),w(j,4))-2774*f1(t(4),w(j,3))+2616*f1(t(3),w(j,2))-
1274*f1(t(2),w(j,1))+251*f1(t(1),w0(1,j))
d1(1,j)= 1901*f2(t(5),v(j,4))-2774*f2(t(4),v(j,3))+2616*f2(t(3),v(j,2))-
1274*f2(t(2),v(j,1))+251*f2(t(1),v0(1,j))
z0(1,j)=w(j,4)+h*(p1(1,j))/720
q0(1,j)=v(j,4)+h*(d1(1,j))/720
for e=1:4
t(e)=t(e+1)
if e==1
w0(1,j)=w(j,1)
v0(1,j)=v(j,1)
else
w(j,e-1)=w(j,e)
v(j,e-1)=v(j,e)
end
end
t(5)=t0
w(j,4)=z0(1,j)
v(j,4)=q0(1,j)
end
ww=double(w)
vv=double(v)
end

```



```

syms t w v
format long
c0=[2.038711369407443,2.106668415054357,2.174625460701272,2.242582506
348186,2.310539551995102,2.378496597642015,2.446453643288931,2.514410
688935846,2.582367734582760,2.650324780229675,2.718281825876591]
f0=[3.058067054111163,3.024088531287706,2.990110008464247,2.956131485
640793,2.922152962817334,2.888174439993877,2.854195917170421,2.820217
394346960,2.786238871523505,2.752260348700048,2.718281825876591]
f1(t,w)=y+2*c0*(t-1)
f2(t,v)=y+2*f0*(t-1)
a=1
b=1.5
N=25
h=(b-a)/N
t(1)=a
for j=1:11
w0=[2.038711369407443,2.106668415054357,2.174625460701272,2.242582506
348186,2.310539551995102,2.378496597642015,2.446453643288931,2.514410
688935846,2.582367734582760,2.650324780229675,2.718281825876591]
v0=[3.058067054111163,3.024088531287706,2.990110008464247,2.956131485
640793,2.922152962817334,2.888174439993877,2.854195917170421,2.820217
394346960,2.786238871523505,2.752260348700048,2.718281825876591]
t(1)=a
for i=1:4
t(i+1)=t(i)+h
if i==1
k1=h*f1(t(i),w0(1,j))
k11=h*f2(t(i),v0(1,j))
k2=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
k22=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))

```

```

k3=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
k33=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
k4=h*f1(t(i+1),w0(1,j)+k3(1,j))
k44=h*f2(t(i+1),v0(1,j)+k33(1,j))
w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

else
k1=h*f1(t(i),w(1,i-1))
k11=h*f2(t(i),v(1,i-1))
k2=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
k22=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
k3=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
k33=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
k4=h*f1(t(i+1),w(1,i-1)+k3(1,j))
k44=h*f2(t(i+1),v(1,i-1)+k33(1,j))
w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
v(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

end
end
for i=5:N
t0=a+i*h
p1=1901*f1(t(5),w(1,4))-2774*f1(t(4),w(1,3))+2616*f1(t(3),w(1,2))-
1274*f1(t(2),w(1,1))+251*f1(t(1),w0(1,j))
d1=1901*f2(t(5),v(1,4))-2774*f2(t(4),v(1,3))+2616*f2(t(3),v(1,2))-
1274*f2(t(2),v(1,1))+251*f2(t(1),v0(1,j))
z0(1,j)=w(1,4)+h*(p1(1,j))/720
q0(1,j)=v(1,4)+h*(d1(1,j))/720
for e=1:4
t(e)=t(e+1)
if z==1
w0(1,j)=w(1,1)
v0(1,j)=v(1,1)
else
w(1,e-1)=w(1,e)
v(1,e-1)=v(1,e)
end
end
t(5)=t0
w(1,4)=z0(1,j)
v(1,4)=q0(1,j)
end
ww=double(w)
vv=double(v)
end

```

```

syms t w v
format long
c0=[2.038711369407443,2.106668415054357,2.174625460701272,2.242582506
348186,2.310539551995102,2.378496597642015,2.446453643288931,2.514410
688935846,2.582367734582760,2.650324780229675,2.718281825876591]
f0=[3.058067054111163,3.024088531287706,2.990110008464247,2.956131485
640793,2.922152962817334,2.888174439993877,2.854195917170421,2.820217
394346960,2.786238871523505,2.752260348700048,2.718281825876591]
f1(t,w)=y+2*c0*(2-t)
f2(t,v)=y+2*f0*(2-t)
a=1.5
b=2

```

```

N=25
h=(b-a)/N
t(1)=a
for j=1:11
w0=[3.967665851557501,4.099921379942754,4.232176908328005,4.364432436
713252,4.496687965098505,4.628943493483756,4.761199021869006,4.893454
550254253,5.025710078639504,5.157965607024751,5.290221135410005]
v0=[5.951498777336254,5.885371013143627,5.819243248951001,5.753115484
758379,5.686987720565753,5.620859956373129,5.554732192180500,5.488604
427987878,5.422476663795258,5.356348899602628,5.290221135410005]
t(1)=a
for i=1:4
t(i+1)=t(i)+h
if i==1
k1=h*f1(t(i),w0(1,j))
k11=h*f2(t(i),v0(1,j))
k2=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
k22=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
k3=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
k33=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
k4=h*f1(t(i+1),w0(1,j)+k3(1,j))
k44=h*f2(t(i+1),v0(1,j)+k33(1,j))
w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

else
k1=h*f1(t(i),w(1,i-1))
k11=h*f2(t(i),v(1,i-1))
k2=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
k22=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
k3=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
k33=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
k4=h*f1(t(i+1),w(1,i-1)+k3(1,j))
k44=h*f2(t(i+1),v(1,i-1)+k33(1,j))
w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
w(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

end
end
for i=5:N
t0=a+i*h
p1=1901*f1(t(5),w(1,4))-2774*f1(t(4),w(1,3))+2616*f1(t(3),w(1,2))-
1274*f1(t(2),w(1,1))+251*f1(t(1),w0(1,j))
d1=1901*f2(t(5),v(1,4))-2774*f2(t(4),v(1,3))+2616*f2(t(3),v(1,2))-
1274*f2(t(2),v(1,1))+251*f2(t(1),v0(1,j))
z0(1,j)=w(1,4)+h*(p1(1,j))/720
q0(1,j)=v(1,4)+h*(d1(1,j))/720

for e=1:4
t(e)=t(e+1)
if z==1
w0(1,j)=w(1,1)
v0(1,j)=v(1,1)
else
w(1,e-1)=w(1,e)
v(1,e-1)=v(1,e)
end
end
t(5)=t0
w(1,4)=z0(1,j)
v(1,4)=q0(1,j)

```

```

end
ww=double(w)
vv=double(v)
end

```

## Variational Iteration Method (VIM)

```

 $\ddot{w} = \frac{d^2w}{dt^2}$ 
syms t w v h q z r
format long
w0=zeros(1,11)
a=1
b=0
for i=1:11
w0(1,i)=0.75+0.25*(i-1)*0.1
v0(1,i)=1.125-0.125*(i-1)*0.1
end
t0=0
z(r)=int(1,r,t)
exp(z)
for k=1:11
w(k)=w0(1,k)-int(exp(z(r))*(-a*w0(1,k)-b),r,t0,t)
v(k)=v0(1,k)-int(exp(z(r))*(-a*v0(1,k)-b),r,t0,t)
h(t)=w(k)
q(t)=v(k)
h(1)
q(1)
end

```

```

 $\ddot{w} = \frac{d^2w}{dt^2}$ 
syms t w v h q z r
format long
for i=1:11
w0(1,i)=(0.75+0.25*(i-1)*0.1)*exp(1)
v0(1,i)=(1.125-0.125*(i-1)*0.1)*exp(1)
end
a=1
for k=1:11
b1(1,k)=2*w0(1,k)*(r-1)
b11(1,k)= 2*v0(1,k)*(r-1)
end
t0=1
z(r)=int(a,r,t)
for k=1:11
w(k)=w0(1,k)-int(exp(z(r))*(-a*w0(1,k)-b1(1,k)),r,t0,t)
v(k)=v0(1,k)-int(exp(z(r))*(-a*v0(1,k)-b11(1,k)),r,t0,t)
h(t)=w(k)
q(t)=v(k)
h(1.5)
q(1.5)
end

```

```

 $\ddot{w} = \frac{d^2w}{dt^2}$ 
syms t w v h q z r
format long
for i=1:11
c0(1,i)=(0.75+0.25*(i-1)*0.1)*exp(1)
d0(1,i)=(1.125-0.125*(i-1)*0.1)*exp(1)
end

```

```

w0=[3.967666294227795,4.099921837368721,4.232177380509648,4.364432923
650575,4.496688466791501,4.628944009932427,4.761199553073354,4.893455
096214280,5.025710639355206,5.157966182496133,5.290221725637060]
v0=[5.951499441341692,5.885371669771230,5.819243898200766,5.753116126
630302,5.686988355059839,5.620860583489376,5.554732811918913,5.488605
040348450,5.422477268777985,5.356349497207523,5.290221725637060]
a=1
for k=1:11
b1(1,k)=2*c0(1,k)*(2-r)
b11(1,k)= 2*d0(1,k)*(2-r)
end
t0=1.5
z(r)=int(a,r,t)
for k=1:11
w(k)=w0(1,k)-int(exp(z(r))*(-a*w0(1,k)-b1(1,k)),r,t0,t)
v(k)=v0(1,k)-int(exp(z(r))*(-a*v0(1,k)-b11(1,k)),r,t0,t)
h(t)=w(k)
q(t)=v(k)
h(2)
q(2)
end

```

## Adomian Decomposition Method (ADM)

```

syms t w v z h s
format long
t0=0
a=1
b=0
y0=zeros(1,11)
q0=zeros(1,11)
w0=zeros(1,11)
v0=zeros(1,11)
for i=1:11
y0(1,i)=0.75+0.25*(i-1)*0.1
q0(1,i)=1.125-0.125*(i-1)*0.1
w0(1,i)=y0(1,i)+int(b,s,t0,t)
v0(1,i)=q0(1,i)+int(b,s,t0,t)
end
n=6
for k=1:11
z(t)=w0(1,k)*t^0
h(t)=v0(1,k)*t^0
for j=1:n
if j==1
w(k,j)=int(a*w0(1,k),s,t0,t)
v(k,j)=int(a*v0(1,k),s,t0,t)
else
w(k,j)=int(a*w(k,j-1),t0,t)
v(k,j)=int(a*v(k,j-1),t0,t)
end
z(t)=z(t)+w(k,j)
h(t)=h(t)+v(k,j)
z(1)
h(1)
end
end

```

```

syms t w v z h s
format long
for i=1:11
y0(1,i)=(0.75+0.25*(i-1)*0.1)*exp(1)
q0(1,i)=(1.125-0.125*(i-1)*0.1)*exp(1)
end
t0=1
a=1
for i=1:11
b1(1,i)=2*y0(1,i)*(s-1)
b2(1,i)=2*q0(1,i)*(s-1)
c1(1,i)=int(b1(1,i),t0,s)
c2(1,i)=int(b2(1,i),t0,s)
w0(1,i)=y0(1,i)+c1(1,i)
v0(1,i)=q0(1,i)+c2(1,i)
end
for i=1:11
w0(1,i)=y0(1,i)+c1(1,i)
v0(1,i)=q0(1,i)+c2(1,i)
end
n=6
for k=1:11
z(s)=w0(1,k)*s^0
h(s)=v0(1,k)*s^0
for j=1:n
if j==1
w(k,j)=int(a*w0(1,k),t0,s)
v(k,j)=int(a*v0(1,k),t0,s)
else
w(k,j)=int(a*w(k,j-1),t0,s)
v(k,j)=int(a*v(k,j-1),t0,s)
end
z(s)=z(s)+w(k,j)
h(s)=h(s)+v(k,j)
z(1.5)
h(1.5)
end
end

```

```

syms t w v z h s
format long
for i=1:11
c0(1,i)=(0.75+0.25*(i-1)*0.1)*exp(1)
d0(1,i)=(1.125-0.125*(i-1)*0.1)*exp(1)
end
y0=[3.967666294227795,4.099921837368721,4.232177380509648,4.364432923
650575,4.496688466791501,4.628944009932427,4.761199553073354,4.893455
096214280,5.025710639355206,5.157966182496133,5.290221725637060]
q0=[5.951499441341692,5.885371669771230,5.819243898200766,5.753116126
630302,5.686988355059839,5.620860583489376,5.554732811918913,5.488605
040348450,5.422477268777985,5.356349497207523,5.290221725637060]
t0=1.5
a=1
for i=1:11
b1(1,i)=2*y0(1,i)*(2-s)
b2(1,i)=2*q0(1,i)*(2-s)
c1(1,i)=int(b1(1,i),t0,s)
c2(1,i)=int(b2(1,i),t0,s)

w0(1,i)=y0(1,i)+c1(1,i)

```

```

v0(1,i)=q0(1,i)+c2(1,i)
end
for i=1:11
w0(1,i)=y0(1,i)+c(1,i)
v0(1,i)=q0(1,i)+c2(1,i)
end
n=6
for k=1:11
z(s)=w0(1,k)*s^0
h(s)=v0(1,k)*s^0
for j=1:n
if j==1
w(k,j)=int(a*w0(1,k),t0,s)
v(k,j)=int(a*v0(1,k),t0,s)
else
w(k,j)=int(a*w(k,j-1),t0,s)
v(k,j)=int(a*v(k,j-1),t0,s)
end
z(s)=z(s)+w(k,j)
h(s)=h(s)+v(k,j)
z(2)
h(2)
end
end

```

## Predictor-Corrector Method

### 1-Milne's four step method

```

syms t w v
format long
for j=1:11
w0(1,j)=0.75+0.25*(j-1)*0.1
w1(1,j)=(0.75+0.25*(j-1)*0.1)*exp(0.02)
w2(1,j)=(0.75+0.25*(j-1)*0.1)*exp(0.04)
w3(1,j)=(0.75+0.25*(j-1)*0.1)*exp(0.06)
v0(1,j)=1.125-1.125*(j-1)*0.1
v1(1,j)=(1.125-1.125*(j-1)*0.1)*exp(0.02)
v2(1,j)=(1.125-1.125*(j-1)*0.1)*exp(0.04)
v3(1,j)=(1.125-1.125*(j-1)*0.1)*exp(0.06)
end
f1(t,w)=w
f2(t,v)=v
t0=0
h=0.02
for i=1:50
t(i)=t0+i*h
end
for j=1:11
for i=1:47
if i==1
k1(i+3,j)=w0(1,j)+(4*h/3)*(2*f1(t(i+2),w3(1,j))-
f1(t(i+1),w2(1,j))+2*f1(t(i),w1(1,j)))
k2(i+3,j)=v0(1,j)+(4*h/3)*(2*f2(t(i+2),v3(1,j))-
f2(t(i+1),v2(1,j))+2*f2(t(i),v1(1,j)))
w(i+3,j)=w2(1,j)+(h/3)*(f1(t(i+3),k1(i+3,j))+4*f1(t(i+2),w3(1,j))+f1(t(i+1),
w2(1,j)))

```

```

v(i+3,j)=v2(1,j)+(h/3)*(f2(t(i+3),k2(i+3,j))+4*f2(t(i+2),v3(1,j))+f2(t(i+1)
,v2(1,j)))

    elseif i==2
        k1(i+3,j)=w1(1,j)+(4*h/3)*(2*f1(t(i+2),w(i+2,j))-f1(t(i+1),w3(1,j))+2*f1(t(i),w2(1,j)))
        k2(i+3,j)=v1(1,j)+(4*h/3)*(2*f2(t(i+2),v(i+2,j))-f2(t(i+1),v3(1,j))+2*f2(t(i),v2(1,j)))
        w(i+3,j)=w3(1,j)+(h/3)*(f1(t(i+3),k1(i+3,j))+4*f1(t(i+2),w(i+2,j))+f1(t(i+1),
),w3(1,j)))
        v(i+3,j)=v3(1,j)+(h/3)*(f2(t(i+3),k2(i+3,j))+4*f2(t(i+2),v(i+2,j))+f2(t(i+1
),v3(1,j)))
    elseif i==3
        k1(i+3,j)=w2(1,j)+(4*h/3)*(2*f1(t(i+2),w(i+2,j))-f1(t(i+1),w(i+1,j))+2*f1(t(i),w3(1,j)))
        k2(i+3,j)=v2(1,j)+(4*h/3)*(2*f2(t(i+2),v(i+2,j))-f2(t(i+1),v(i+1,j))+2*f2(t(i),v3(1,j)))
        w(i+3,j)=w(i+1,j)+(h/3)*(f1(t(i+3),k1(i+3,j))+4*f1(t(i+2),w(i+2,j))+f1(t(i+
1),w(i+1,j)))
        v(i+3,j)=v(i+1,j)+(h/3)*(f2(t(i+3),k2(i+3,j))+4*f2(t(i+2),v(i+2,j))+f2(t(i+
1),v(i+1,j)))
    elseif i==4
        k1(i+3,j)=w3(1,j)+(4*h/3)*(2*f1(t(i+2),w(i+2,j))-f1(t(i+1),w(i+1,j))+2*f1(t(i),w(i,j)))
        k2(i+3,j)=v3(1,j)+(4*h/3)*(2*f2(t(i+2),v(i+2,j))-f2(t(i+1),v(i+1,j))+2*f2(t(i),v(i,j)))
        w(i+3,j)=w(i+1,j)+(h/3)*(f1(t(i+3),k1(i+3,j))+4*f1(t(i+2),w(i+2,j))+f1(t(i+
1),w(i+1,j)))
        v(i+3,j)=v(i+1,j)+(h/3)*(f2(t(i+3),k2(i+3,j))+4*f2(t(i+2),v(i+2,j))+f2(t(i+
1),v(i+1,j)))
    else
        k1(i+3,j)=w(i-1,j)+(4*h/3)*(2*f1(t(i+2),w(i+2,j))-f1(t(i+1),w(i+1,j))+2*f1(t(i),w(i,j)))
        k2(i+3,j)=v(i-1,j)+(4*h/3)*(2*f2(t(i+2),v(i+2,j))-f2(t(i+1),v(i+1,j))+2*f2(t(i),v(i,j)))
        w(i+3,j)=w(i+1,j)+(h/3)*(f1(t(i+3),k1(i+3,j))+4*f1(t(i+2),w(i+2,j))+f1(t(i+
1),w(i+1,j)))
        v(i+3,j)=v(i+1,j)+(h/3)*(f2(t(i+3),k2(i+3,j))+4*f2(t(i+2),v(i+2,j))+f2(t(i+
1),v(i+1,j)))
    end

end
ww=double(w(i+3,j))
vv=double(v(i+3,j))

end

```

```

+ syms t w v
format long
w0=[2.038711372736861,2.106668418494757,2.174625464252652,2.242582510
010547,2.310539555768443,2.378496601526338,2.446453647284233,2.514410
693042129,2.582367738800024,2.650324784557919,2.718281830315815]
v0=[3.058067059105292,3.024088536226344,2.990110013347397,2.956131490
468449,2.922152967589501,2.888174444710553,2.854195921831606,2.820217
398952658,2.786238876073711,2.752260353194763,2.718281830315815]
f1(t,w)=w+2*w0*(t-1)
f2(t,v)=v+2*v0*(t-1)
a=1
h=0.02

```

```

        t(1)=a
n=25
for i=1:n
t(i+1)=t(i)+h
end

for j=1:1
for i=1:3
if i==1
k1=h*f1(t(i),w0(1,j))
k2=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
k4=h*f1(t(i+1),w0(1,j)+k3(1,j))
w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
k11=h*f2(t(i),v0(1,j))
k22=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
k33=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
k44=h*f2(t(i+1),v0(1,j)+k33(1,j))
v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

else
k1=h*f1(t(i),w(1,i-1))
k2=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
k4=h*f1(t(i+1),w(1,i-1)+k3(1,j))
w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
k11=h*f2(t(i),v(1,i-1))
k22=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
k33=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
k44=h*f2(t(i+1),v(1,i-1)+k33(1,j))
v(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

end
end
for i=1:22
if i==1
k1=w0(1,j)+(4*h/3)*(2*f1(t(i+3),w(1,3))-f1(t(i+2),w(1,2))+2*f1(t(i+1),w(1,1)))
k11=(h/3)*(f1(t(i+4),k1(1,j))+4*f1(t(i+3),w(1,3))+f1(t(i+2),w(1,2)))
w(1,i+3)=w(1,2)+k11(1,j)
k2=v0(1,j)+(4*h/3)*(2*f2(t(i+3),v(1,3))-f2(t(i+2),v(1,2))+2*f2(t(i+1),v(1,1)))
k22=(h/3)*(f2(t(i+4),k2(1,j))+4*f2(t(i+3),v(1,3))+f2(t(i+2),v(1,2)))
v(1,i+3)=v(1,2)+k22(1,j)
elseif i==2
k1=w(1,1)+(4*h/3)*(2*f1(t(i+3),w(1,i+2))-f1(t(i+2),w(1,3))+2*f1(t(i+1),w(1,2)))
k11=(h/3)*(f1(t(i+4),k1(1,j))+4*f1(t(i+3),w(1,i+2))+f1(t(i+2),w(1,3)))
w(1,i+3)=w(1,3)+k11(1,j)
k2=v(1,1)+(4*h/3)*(2*f2(t(i+3),v(1,i+2))-f2(t(i+2),v(1,3))+2*f2(t(i+1),v(1,2)))
k22=(h/3)*(f2(t(i+4),k2(1,j))+4*f2(t(i+3),v(1,i+2))+f2(t(i+2),v(1,3)))
v(1,i+3)=v(1,3)+k22(1,j)
elseif i==3
k1=w(1,2)+(4*h/3)*(2*f1(t(i+3),w(1,i+2))-f1(t(i+2),w(1,i+1))+2*f1(t(i+1),w(1,3)))
k11=(h/3)*(f1(t(i+4),k1(1,j))+4*f1(t(i+3),w(1,i+2))+f1(t(i+2),w(1,i+1)))
w(1,i+3)=w(1,i+1)+k11(1,j)
k2=v(1,2)+(4*h/3)*(2*f2(t(i+3),v(1,i+2))-f2(t(i+2),v(1,i+1))+2*f2(t(i+1),v(1,3)))
k22=(h/3)*(f2(t(i+4),k2(1,j))+4*f2(t(i+3),v(1,i+2))+f2(t(i+2),v(1,i+1)))
v(1,i+3)=v(1,i+1)+k22(1,j)

```

```

v(1,i+3)=v(1,i+1)+k22(1,j)

    elseif i==4
k1=w(1,3)+(4*h/3)*(2*f1(t(i+3),w(1,i+2))-f1(t(i+2),w(1,i+1))+2*f1(t(i+1),w(1,i)))
k11=(h/3)*(f1(t(i+4),k1(1,j))+4*f1(t(i+3),w(1,i+2))+f1(t(i+2),w(1,i+1)))
w(1,i+3)=w(1,i+1)+k11(1,j)
k2=v(1,3)+(4*h/3)*(2*f2(t(i+3),v(1,i+2))-f2(t(i+2),v(1,i+1))+2*f2(t(i+1),v(1,i)))
k22=(h/3)*(f2(t(i+4),k2(1,j))+4*f2(t(i+3),v(1,i+2))+f2(t(i+2),v(1,i+1)))
v(1,i+3)=v(1,i+1)+k22(1,j)
    else
k1=w(1,i-1)+(4*h/3)*(2*f1(t(i+3),w(1,i+2))-f1(t(i+2),w(1,i+1))+2*f1(t(i+1),w(1,i)))
k11=(h/3)*(f1(t(i+4),k1(1,j))+4*f1(t(i+3),w(1,i+2))+f1(t(i+2),w(1,i+1)))
w(1,i+3)=w(1,i+1)+k11(1,j)
k2=v(1,i-1)+(4*h/3)*(2*f2(t(i+3),v(1,i+2))-f2(t(i+2),v(1,i+1))+2*f2(t(i+1),v(1,i)))
k22=(h/3)*(f2(t(i+4),k2(1,j))+4*f2(t(i+3),v(1,i+2))+f2(t(i+2),v(1,i+1)))
v(1,i+3)=v(1,i+1)+k22(1,j)
end

end
ww=double(w(1,i+3))
vv=double(v(1,i+3))
end

```

```

#+ syms t w v
format long
c0=[2.038711372736861,2.106668418494757,2.174625464252652,2.242582510
010547,2.310539555768443,2.378496601526338,2.446453647284233,2.514410
693042129,2.582367738800024,2.650324784557919,2.718281830315815]
d0=[3.058067059105292,3.024088536226344,2.990110013347397,2.956131490
468449,2.922152967589501,2.888174444710553,2.854195921831606,2.820217
398952658,2.786238876073711,2.752260353194763,2.718281830315815]
w0=[3.967666299420739,4.099921842734765,4.232177386048789,4.364432929
362812,4.496688472676839,4.628944015990862,4.761199559304886,4.893455
102618911,5.025710645932936,5.157966189246960,5.290221732560985]
v0=[5.951499449131109,5.885371677474096,5.819243905817085,5.753116134
160072,5.686988362503059,5.620860590846046,5.554732819189035,5.488605
047532023,5.422477275875011,5.356349504217998,5.290221732560985]
f1(t,w)=w+2*c0*(2-t)
f2(t,v)=v+2*d0*(2-t)
a=1.5
h=0.02
t(1)=a
n=25
for i=1:n
t(i+1)=t(i)+h
end

for j=1:11
for i=1:3
if i==1
k1=h*f1(t(i),w0(1,j))
k2=h*f1(t(i)+0.5*h,w0(1,j))+0.5*k1(1,j)
k3=h*f1(t(i)+0.5*h,w0(1,j))+0.5*k2(1,j)
k4=h*f1(t(i+1),w0(1,j)+k3(1,j))

```

```

w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
k11=h*f2(t(i),v0(1,j))
k22=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
k33=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
k44=h*f2(t(i+1),v0(1,j)+k33(1,j))
v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

else
k1=h*f1(t(i),w(1,i-1))
k2=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
k4=h*f1(t(i+1),w(1,i-1)+k3(1,j))
w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
k11=h*f2(t(i),v(1,i-1))
k22=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
k33=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
k44=h*f2(t(i+1),v(1,i-1)+k33(1,j))
v(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

end
for i=1:22
if i==1
k1=w0(1,j)+(4*h/3)*(2*f1(t(i+3),w(1,3))-f1(t(i+2),w(1,2))+2*f1(t(i+1),w(1,1)))
k11=(h/3)*(f1(t(i+4),k1(1,j))+4*f1(t(i+3),w(1,3))+f1(t(i+2),w(1,2)))
w(1,i+3)=w(1,2)+k11(1,j)
k2=v0(1,j)+(4*h/3)*(2*f2(t(i+3),v(1,3))-f2(t(i+2),v(1,2))+2*f2(t(i+1),v(1,1)))
k22=(h/3)*(f2(t(i+4),k2(1,j))+4*f2(t(i+3),v(1,3))+f2(t(i+2),v(1,2)))
v(1,i+3)=v(1,2)+k22(1,j)
elseif i==2
k1=w(1,1)+(4*h/3)*(2*f1(t(i+3),w(1,i+2))-f1(t(i+2),w(1,3))+2*f1(t(i+1),w(1,2)))
k11=(h/3)*(f1(t(i+4),k1(1,j))+4*f1(t(i+3),w(1,i+2))+f1(t(i+2),w(1,3)))
w(1,i+3)=w(1,3)+k11(1,j)
k2=v(1,1)+(4*h/3)*(2*f2(t(i+3),v(1,i+2))-f2(t(i+2),v(1,3))+2*f2(t(i+1),v(1,2)))
k22=(h/3)*(f2(t(i+4),k2(1,j))+4*f2(t(i+3),v(1,i+2))+f2(t(i+2),v(1,3)))
v(1,i+3)=v(1,3)+k22(1,j)
elseif i==3
k1=w(1,2)+(4*h/3)*(2*f1(t(i+3),w(1,i+2))-f1(t(i+2),w(1,i+1))+2*f1(t(i+1),w(1,3)))
k11=(h/3)*(f1(t(i+4),k1(1,j))+4*f1(t(i+3),w(1,i+2))+f1(t(i+2),w(1,i+1)))
w(1,i+3)=w(1,i+1)+k11(1,j)
k2=v(1,2)+(4*h/3)*(2*f2(t(i+3),v(1,i+2))-f2(t(i+2),v(1,i+1))+2*f2(t(i+1),v(1,3)))
k22=(h/3)*(f2(t(i+4),k2(1,j))+4*f2(t(i+3),v(1,i+2))+f2(t(i+2),v(1,i+1)))
v(1,i+3)=v(1,i+1)+k22(1,j)

elseif i==4
k1=w(1,3)+(4*h/3)*(2*f1(t(i+3),w(1,i+2))-f1(t(i+2),w(1,i+1))+2*f1(t(i+1),w(1,i)))
k11=(h/3)*(f1(t(i+4),k1(1,j))+4*f1(t(i+3),w(1,i+2))+f1(t(i+2),w(1,i+1)))
w(1,i+3)=w(1,i+1)+k11(1,j)
k2=v(1,3)+(4*h/3)*(2*f2(t(i+3),v(1,i+2))-f2(t(i+2),v(1,i+1))+2*f2(t(i+1),v(1,i)))
k22=(h/3)*(f2(t(i+4),k2(1,j))+4*f2(t(i+3),v(1,i+2))+f2(t(i+2),v(1,i+1)))
v(1,i+3)=v(1,i+1)+k22(1,j)
else

```

```

k1=w(1,i-1)+(4*h/3)*(2*f1(t(i+3),w(1,i+2))-  

f1(t(i+2),w(1,i+1))+2*f1(t(i+1),w(1,i)))  

k11=(h/3)*(f1(t(i+4),k1(1,j))+4*f1(t(i+3),w(1,i+2))+f1(t(i+2),w(1,i+1)))  

w(1,i+3)=w(1,i+1)+k11(1,j)  

k2=v(1,i-1)+(4*h/3)*(2*f2(t(i+3),v(1,i+2))-  

f2(t(i+2),v(1,i+1))+2*f2(t(i+1),v(1,i)))  

k22=(h/3)*(f2(t(i+4),k2(1,j))+4*f2(t(i+3),v(1,i+2))+f2(t(i+2),v(1,i+1)))  

v(1,i+3)=v(1,i+1)+k22(1,j)

end

end
ww=double(w(1,i+3))
vv=double(v(1,i+3))
end

```

## Adams-Bashforth four step method

```

syms t w v
format long
for j=1:11
w0(1,j)=0.75+0.25*(j-1)*0.1
w1(1,j)=(0.75+0.25*(j-1)*0.1)*exp(0.02)
w2(1,j)=(0.75+0.25*(j-1)*0.1)*exp(0.04)
w3(1,j)=(0.75+0.25*(j-1)*0.1)*exp(0.06)
v0(1,j)=1.125-0.125*(j-1)*0.1
v1(1,j)=(1.125-0.125*(j-1)*0.1)*exp(0.02)
v2(1,j)=(1.125-0.125*(j-1)*0.1)*exp(0.04)
v3(1,j)=(1.125-0.125*(j-1)*0.1)*exp(0.06)

end
f1(t,w)=w
f2(t,v)=v
t0=0
h=0.02
for i=1:50
t(i)=t0+i*h
end
for j=1:11
for i=1:47
if i==1
k1(i+3,j)=w3(1,j)+(h/24)*(55*f1(t(i+2),w3(1,j))-  

59*f1(t(i+1),w2(1,j))+37*f1(t(i),w1(1,j))-9*f1(t0,w0(1,j)))  

w(i+3,j)=w3(1,j)+(h/24)*(9*f1(t(i+3),k1(i+3,j))+19*f1(t(i+2),w3(1,j))-  

5*f1(t(i+1),w2(1,j))+f1(t(i),w1(1,j)))  

k2(i+3,j)=v3(1,j)+(h/24)*(55*f2(t(i+2),v3(1,j))-  

59*f2(t(i+1),v2(1,j))+37*f2(t(i),v1(1,j))-9*f2(t0,v0(1,j)))  

v(i+3,j)=v3(1,j)+(h/24)*(9*f2(t(i+3),k2(i+3,j))+19*f2(t(i+2),v3(1,j))-  

5*f2(t(i+1),v2(1,j))+f2(t(i),v1(1,j)))
elseif i==2
k1(i+3,j)=w(i+2,j)+(h/24)*(55*f1(t(i+2),w(i+2,j))-  

59*f1(t(i+1),w3(1,j))+37*f1(t(i),w2(1,j))-9*f1(t(i-1),w1(1,j)))  

w(i+3,j)=w(i+2,j)+(h/24)*(9*f1(t(i+3),k1(i+3,j))+19*f1(t(i+2),w(i+2,j))-  

5*f1(t(i+1),w3(1,j))+f1(t(i),w2(1,j)))  

k2(i+3,j)=v(i+2,j)+(h/24)*(55*f2(t(i+2),v(i+2,j))-  

59*f2(t(i+1),v3(1,j))+37*f2(t(i),v2(1,j))-9*f2(t(i-1),v1(1,j)))
end
end

```

```

v(i+3,j)=v(i+2,j)+(h/24)*(9*f2(t(i+3),k2(i+3,j))+19*f2(t(i+2),v(i+2,j))-5*f2(t(i+1),v3(1,j))+f2(t(i),v2(1,j)))
    elseif i==3
k1(i+3,j)=w(i+2,j)+(h/24)*(55*f1(t(i+2),w(i+2,j))-59*f1(t(i+1),w(i+1,j))+37*f1(t(i),w3(1,j))-9*f1(t(i-1),w2(1,j)))
w(i+3,j)=w(i+2,j)+(h/24)*(9*f1(t(i+3),k1(i+3,j))+19*f1(t(i+2),w(i+2,j))-5*f1(t(i+1),w(i+1,j))+f1(t(i),w3(1,j)))
k2(i+3,j)=v(i+2,j)+(h/24)*(55*f2(t(i+2),v(i+2,j))-59*f2(t(i+1),v(i+1,j))+37*f2(t(i),v3(1,j))-9*f2(t(i-1),v2(1,j)))
v(i+3,j)=v(i+2,j)+(h/24)*(9*f2(t(i+3),k2(i+3,j))+19*f2(t(i+2),v(i+2,j))-5*f2(t(i+1),v(i+1,j))+f2(t(i),v3(1,j)))
    elseif i==4
k1(i+3,j)=w(i+2,j)+(h/24)*(55*f1(t(i+2),w(i+2,j))-59*f1(t(i+1),w(i+1,j))+37*f1(t(i),w(i,j))-9*f1(t(i-1),w3(1,j)))
w(i+3,j)=w(i+2,j)+(h/24)*(9*f1(t(i+3),k1(i+3,j))+19*f1(t(i+2),w(i+2,j))-5*f1(t(i+1),w(i+1,j))+f1(t(i),w(i,j)))
k2(i+3,j)=v(i+2,j)+(h/24)*(55*f2(t(i+2),v(i+2,j))-59*f2(t(i+1),v(i+1,j))+37*f2(t(i),v(i,j))-9*f2(t(i-1),v3(1,j)))
v(i+3,j)=v(i+2,j)+(h/24)*(9*f2(t(i+3),k2(i+3,j))+19*f2(t(i+2),v(i+2,j))-5*f2(t(i+1),v(i+1,j))+f2(t(i),v(i,j)))
    else
k1(i+3,j)=w(i+2,j)+(h/24)*(55*f1(t(i+2),w(i+2,j))-59*f1(t(i+1),w(i+1,j))+37*f1(t(i),w(i,j))-9*f1(t(i-1),w(i-1,j)))
w(i+3,j)=w(i+2,j)+(h/24)*(9*f1(t(i+3),k1(i+3,j))+19*f1(t(i+2),w(i+2,j))-5*f1(t(i+1),w(i+1,j))+f1(t(i),w(i,j)))
k2(i+3,j)=v(i+2,j)+(h/24)*(55*f2(t(i+2),v(i+2,j))-59*f2(t(i+1),v(i+1,j))+37*f2(t(i),v(i,j))-9*f2(t(i-1),v(i-1,j)))
v(i+3,j)=v(i+2,j)+(h/24)*(9*f2(t(i+3),k2(i+3,j))+19*f2(t(i+2),v(i+2,j))-5*f2(t(i+1),v(i+1,j))+f2(t(i),v(i,j)))
end

end
ww=double(w(i+3,j))
vv=double(v(i+3,j))

end

```

```

syms t w v
format long
w0=[2.038711378478180,2.106668424427452,2.174625470376725,2.242582516
325997,2.310539562275270,2.378496608224543,2.446453654173816,2.514410
700123088,2.582367746072360,2.650324792021633,2.718281837970906]
v0=[3.058067067717269,3.024088544742633,2.990110021767997,2.956131498
793360,2.922152975818724,2.888174452844087,2.854195929869452,2.820217
406894816,2.786238883920178,2.752260360945542,2.718281837970906]
f1(t,w)=w+2*w0*(t-1)
f2(t,v)=v+2*v0*(t-1)
a=1
h=0.02
t(1)=a
n=25
for i=1:n
    t(i+1)=t(i)+h
end

for j=1:11
    for i=1:3
        if i==1

```

```

k1=h*f1(t(i),w0(1,j))
k2=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
k4=h*f1(t(i+1),w0(1,j)+k3(1,j))
w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
k11=h*f2(t(i),v0(1,j))
k22=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
k33=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
k44=h*f2(t(i+1),v0(1,j)+k33(1,j))
v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
else
k1=h*f1(t(i),w(1,i-1))
k2=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
k4=h*f1(t(i+1),w(1,i-1)+k3(1,j))
w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
k11=h*f2(t(i),v(1,i-1))
k22=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
k33=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
k44=h*f2(t(i+1),v(1,i-1)+k33(1,j))
v(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

end
end
for i=1:22
if i==1
k1=w(1,3)+(h/24)*(55*f1(t(i+3),w(1,3))-59*f1(t(i+2),w(1,2))+37*f1(t(i+1),w(1,1))-9*f1(t(i),w0(1,j)))
k11=(h/24)*(9*f1(t(i+4),k1(1,j))+19*f1(t(i+3),w(1,3))-5*f1(t(i+2),w(1,2))+f1(t(i+1),w(1,1)))
w(1,i+3)=w(1,3)+k11(1,j)
k2=v(1,3)+(h/24)*(55*f2(t(i+3),v(1,3))-59*f2(t(i+2),v(1,2))+37*f2(t(i+1),v(1,1))-9*f2(t(i),v0(1,j)))
k22=(h/24)*(9*f2(t(i+4),k2(1,j))+19*f2(t(i+3),v(1,3))-5*f2(t(i+2),v(1,2))+f2(t(i+1),v(1,1)))
v(1,i+3)=v(1,3)+k22(1,j)
elseif i==2
k1=w(1,i+2)+(h/24)*(55*f1(t(i+3),w(1,i+2))-59*f1(t(i+2),w(1,3))+37*f1(t(i+1),w(1,2))-9*f1(t(i),w(1,1)))
k11=(h/24)*(9*f1(t(i+4),k1(1,j))+19*f1(t(i+3),w(1,i+2))-5*f1(t(i+2),w(1,3))+f1(t(i+1),w(1,2)))
w(1,i+3)=w(1,i+2)+k11(1,j)
k2=v(1,i+2)+(h/24)*(55*f2(t(i+3),v(1,i+2))-59*f2(t(i+2),v(1,3))+37*f2(t(i+1),v(1,2))-9*f2(t(i),v(1,1)))
k22=(h/24)*(9*f2(t(i+4),k2(1,j))+19*f2(t(i+3),v(1,i+2))-5*f2(t(i+2),v(1,3))+f2(t(i+1),v(1,2)))
v(1,i+3)=v(1,i+2)+k22(1,j)
elseif i==3
k1=w(1,i+2)+(h/24)*(55*f1(t(i+3),w(1,i+2))-59*f1(t(i+2),w(1,i+1))+37*f1(t(i+1),w(1,3))-9*f1(t(i),w(1,2)))
k11=(h/24)*(9*f1(t(i+4),k1(1,j))+19*f1(t(i+3),w(1,i+2))-5*f1(t(i+2),w(1,i+1))+f1(t(i+1),w(1,3)))
w(1,i+3)=w(1,i+2)+k11(1,j)
k2=v(1,i+2)+(h/24)*(55*f2(t(i+3),v(1,i+2))-59*f2(t(i+2),v(1,i+1))+37*f2(t(i+1),v(1,3))-9*f2(t(i),v(1,2)))
k22=(h/24)*(9*f2(t(i+4),k2(1,j))+19*f2(t(i+3),v(1,i+2))-5*f2(t(i+2),v(1,i+1))+f2(t(i+1),v(1,3)))
v(1,i+3)=v(1,i+2)+k22(1,j)
elseif i==4
k1=w(1,i+2)+(h/24)*(55*f1(t(i+3),w(1,i+2))-59*f1(t(i+2),w(1,i+1))+37*f1(t(i+1),w(1,i))-9*f1(t(i),w(1,3)))

```

```

k11=(h/24)*(9*f1(t(i+4),k1(1,j))+19*f1(t(i+3),w(1,i+2))-  

5*f1(t(i+2),w(1,i+1))+f1(t(i+1),w(1,i)))  

w(1,i+3)=w(1,i+2)+k11(1,j)  

k2=v(1,i+2)+(h/24)*(55*f2(t(i+3),v(1,i+2))-  

59*f2(t(i+2),v(1,i+1))+37*f2(t(i+1),v(1,i))-9*f2(t(i),v(1,3)))  

k22=(h/24)*(9*f2(t(i+4),k2(1,j))+19*f2(t(i+3),v(1,i+2))-  

5*f2(t(i+2),v(1,i+1))+f2(t(i+1),v(1,i)))  

v(1,i+3)=v(1,i+2)+k22(1,j)  

    else  

k1=w(1,i+2)+(h/24)*(55*f1(t(i+3),w(1,i+2))-  

59*f1(t(i+2),w(1,i+1))+37*f1(t(i+1),w(1,i))-9*f1(t(i),w(1,i-1)))  

k11= (h/24)*(9*f1(t(i+4),k1(1,j))+19*f1(t(i+3),w(1,i+2))-  

5*f1(t(i+2),w(1,i+1))+f1(t(i+1),w(1,i)))  

w(1,i+3)=w(1,i+2)+k11(1,j)  

k2=v(1,i+2)+(h/24)*(55*f2(t(i+3),v(1,i+2))-  

59*f2(t(i+2),v(1,i+1))+37*f2(t(i+1),v(1,i))-9*f2(t(i),v(1,i-1)))  

k22=(h/24)*(9*f2(t(i+4),k2(1,j))+19*f2(t(i+3),v(1,i+2))-  

5*f2(t(i+2),v(1,i+1))+f2(t(i+1),v(1,i)))  

v(1,i+3)=v(1,i+2)+k22(1,j)
    end  

end  

ww=double(w(1,i+3))  

vv=double(v(1,i+3))
```

end

```

syms t w v
format long
c0=[2.038711378478180,2.106668424427452,2.174625470376725,2.242582516  

325997,2.310539562275270,2.378496608224543,2.446453654173816,2.514410  

700123088,2.582367746072360,2.650324792021633,2.718281837970906]  

d0=[3.058067067717269,3.024088544742633,2.990110021767997,2.956131498  

793360,2.922152975818724,2.888174452844087,2.854195929869452,2.820217  

406894816,2.786238883920178,2.752260360945542,2.718281837970906]  

w0=[3.967666323833623,4.099921867961409,4.232177412089197,4.364432956  

216984,4.496688500344771,4.628944044472560,4.761199588600348,4.893455  

132728134,5.025710676855920,5.157966220983708,5.290221765111497]  

v0=[5.951499485750433,5.885371713686539,5.819243941622646,5.753116169  

558751,5.686988397494859,5.620860625430963,5.554732853367072,5.488605  

081303179,5.422477309239282,5.356349537175389,5.290221765111497  

f1(t,w)=w+2*c0*(2-t)  

f2(t,v)=v+2*d0*(2-t)  

a=1.5  

h=0.02  

t(1)=a  

n=25
for i=1:n
t(i+1)=t(i)+h
end

for j=1:11
for i=1:3
if i==1
k1=h*f1(t(i),w0(1,j))
k2=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
k4=h*f1(t(i+1),w0(1,j)+k3(1,j))
w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
end

```

```

k11=h*f2(t(i),v0(1,j))
k22=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
k33=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
k44=h*f2(t(i+1),v0(1,j)+k33(1,j))
v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
else
k1=h*f1(t(i),w(1,i-1))
k2=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
k4=h*f1(t(i+1),w(1,i-1)+k3(1,j))
w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
k11=h*f2(t(i),v(1,i-1))
k22=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
k33=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
k44=h*f2(t(i+1),v(1,i-1)+k33(1,j))
v(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6

end
for i=1:22
if i==1
k1=w(1,3)+(h/24)*(55*f1(t(i+3),w(1,3))-
59*f1(t(i+2),w(1,2))+37*f1(t(i+1),w(1,1))-9*f1(t(i),w0(1,j)))
k11=(h/24)*(9*f1(t(i+4),k1(1,j))+19*f1(t(i+3),w(1,3))-5*f1(t(i+2),w(1,2))+f1(t(i+1),w(1,1)))
w(1,i+3)=w(1,3)+k11(1,j)
k2=v(1,3)+(h/24)*(55*f2(t(i+3),v(1,3))-
59*f2(t(i+2),v(1,2))+37*f2(t(i+1),v(1,1))-9*f2(t(i),v0(1,j)))
k22=(h/24)*(9*f2(t(i+4),k2(1,j))+19*f2(t(i+3),v(1,3))-5*f2(t(i+2),v(1,2))+f2(t(i+1),v(1,1)))
v(1,i+3)=v(1,3)+k22(1,j)
elseif i==2
k1=w(1,i+2)+(h/24)*(55*f1(t(i+3),w(1,i+2))-
59*f1(t(i+2),w(1,3))+37*f1(t(i+1),w(1,2))-9*f1(t(i),w(1,1)))
k11=(h/24)*(9*f1(t(i+4),k1(1,j))+19*f1(t(i+3),w(1,i+2))-5*f1(t(i+2),w(1,3))+f1(t(i+1),w(1,2)))
w(1,i+3)=w(1,i+2)+k11(1,j)
k2=v(1,i+2)+(h/24)*(55*f2(t(i+3),v(1,i+2))-
59*f2(t(i+2),v(1,3))+37*f2(t(i+1),v(1,2))-9*f2(t(i),v(1,1)))
k22=(h/24)*(9*f2(t(i+4),k2(1,j))+19*f2(t(i+3),v(1,i+2))-5*f2(t(i+2),v(1,3))+f2(t(i+1),v(1,2)))
v(1,i+3)=v(1,i+2)+k22(1,j)
elseif i==3
k1=w(1,i+2)+(h/24)*(55*f1(t(i+3),w(1,i+2))-
59*f1(t(i+2),w(1,i+1))+37*f1(t(i+1),w(1,3))-9*f1(t(i),w(1,2)))
k11=(h/24)*(9*f1(t(i+4),k1(1,j))+19*f1(t(i+3),w(1,i+2))-5*f1(t(i+2),w(1,i+1))+f1(t(i+1),w(1,3)))
w(1,i+3)=w(1,i+2)+k11(1,j)
k2=v(1,i+2)+(h/24)*(55*f2(t(i+3),v(1,i+2))-
59*f2(t(i+2),v(1,i+1))+37*f2(t(i+1),v(1,3))-9*f2(t(i),v(1,2)))
k22=(h/24)*(9*f2(t(i+4),k2(1,j))+19*f2(t(i+3),v(1,i+2))-5*f2(t(i+2),v(1,i+1))+f2(t(i+1),v(1,3)))
v(1,i+3)=v(1,i+2)+k22(1,j)
elseif i==4
k1=w(1,i+2)+(h/24)*(55*f1(t(i+3),w(1,i+2))-
59*f1(t(i+2),w(1,i+1))+37*f1(t(i+1),w(1,i))-9*f1(t(i),w(1,3)))
k11=(h/24)*(9*f1(t(i+4),k1(1,j))+19*f1(t(i+3),w(1,i+2))-5*f1(t(i+2),w(1,i+1))+f1(t(i+1),w(1,i)))
w(1,i+3)=w(1,i+2)+k11(1,j)
k2=v(1,i+2)+(h/24)*(55*f2(t(i+3),v(1,i+2))-
59*f2(t(i+2),v(1,i+1))+37*f2(t(i+1),v(1,i))-9*f2(t(i),v(1,3)))

```

```

k22=(h/24)*(9*f2(t(i+4),k2(1,j))+19*f2(t(i+3),v(1,i+2))-  

5*f2(t(i+2),v(1,i+1))+f2(t(i+1),v(1,i)))  

v(1,i+3)=v(1,i+2)+k22(1,j)  

    else  

k1=w(1,i+2)+(h/24)*(55*f1(t(i+3),w(1,i+2))-  

59*f1(t(i+2),w(1,i+1))+37*f1(t(i+1),w(1,i))-9*f1(t(i),w(1,i-1)))  

k11=(h/24)*(9*f1(t(i+4),k1(1,j))+19*f1(t(i+3),w(1,i+2))-  

5*f1(t(i+2),w(1,i+1))+f1(t(i+1),w(1,i)))  

w(1,i+3)=w(1,i+2)+k11(1,j)  

k2=v(1,i+2)+(h/24)*(55*f2(t(i+3),v(1,i+2))-  

59*f2(t(i+2),v(1,i+1))+37*f2(t(i+1),v(1,i))-9*f2(t(i),v(1,i-1)))  

k22=(h/24)*(9*f2(t(i+4),k2(1,j))+19*f2(t(i+3),v(1,i+2))-  

5*f2(t(i+2),v(1,i+1))+f2(t(i+1),v(1,i)))  

v(1,i+3)=v(1,i+2)+k22(1,j)  

    end  

end  

ww=double(w(1,i+3))  

vv=double(v(1,i+3))  

end

```

## Improved Predictor-Corrector (IPC) Method

```

syms t w v
format long
for j=1:11
w0(1,j)=0.75+0.25*(j-1)*0.1
w1(1,j)=(0.75+0.25*(j-1)*0.1)*exp(0.02)
w2(1,j)=(0.75+0.25*(j-1)*0.1)*exp(0.04)
v0(1,j)=1.125-0.125*(j-1)*0.1
v1(1,j)=(1.125-0.125*(j-1)*0.1)*exp(0.02)
v2(1,j)=(1.125-0.125*(j-1)*0.1)*exp(0.04)
end
f1(t,w)=w
f2(t,v)=v
t(1)=0
h=0.02
for i=1:50
t(i+1)=t(i)+h
end
for j=1:11
for i=1:48
if i==1
k1=w0(1,j)+(h/2)*(f1(t(i),w0(1,j))+f1(t(i+1),w1(1,j))+4*f1(t(i+2),w2(1,j)))
k11=(h/2)*f1(t(i+1),w1(1,j))+h*f1(t(i+2),w2(1,j))+(h/2)*f1(t(i+3),k1)
w(1,i+2)=w1(1,j)+k11
k2=v0(1,j)+(h/2)*(f2(t(i),v0(1,j))+f2(t(i+1),v1(1,j))+4*f2(t(i+2),v2(1,j)))
k22=(h/2)*f2(t(i+1),v1(1,j))+h*f2(t(i+2),v2(1,j))+(h/2)*f2(t(i+3),k2)
v(1,i+2)=v1(1,j)+k22
elseif i==2
k1=w1(1,j)+(h/2)*(f1(t(i),w1(1,j))+f1(t(i+1),w2(1,j))+4*f1(t(i+2),w(1,i+1)))
k11=(h/2)*f1(t(i+1),w2(1,j))+h*f1(t(i+2),w(1,i+1))+(h/2)*f1(t(i+3),k1)
w(1,i+2)=w2(1,j)+k11
k2=v1(1,j)+(h/2)*(f2(t(i),v1(1,j))+f2(t(i+1),v2(1,j))+4*f2(t(i+2),v(1,i+1)))
k22=(h/2)*f2(t(i+1),v2(1,j))+h*f2(t(i+2),v(1,i+1))+(h/2)*f2(t(i+3),k2)
end

```

```

v(1,i+2)=v2(1,j)+k22

    elseif i==3
k1=w2(1,j)+(h/2)*(f1(t(i),w2(1,j))+f1(t(i+1),w(1,i))+4*f1(t(i+2),w(1,i+1)))
k11=(h/2)*f1(t(i+1),w(1,i))+h*f1(t(i+2),w(1,i+1))+(h/2)*f1(t(i+3),k1)
w(1,i+2)=w(1,i)+k11
k2=v2(1,j)+(h/2)*(f2(t(i),v2(1,j))+f2(t(i+1),v(1,i))+4*f2(t(i+2),v(1,i+1)))
k22=(h/2)*f2(t(i+1),v(1,i))+h*f2(t(i+2),v(1,i+1))+(h/2)*f2(t(i+3),k2)
v(1,i+2)=v(1,i)+k22
    else
k1=w(1,i-1)+(h/2)*(f1(t(i),w(1,i-
1))+f1(t(i+1),w(1,i))+4*f1(t(i+2),w(1,i+1)))
k11=(h/2)*f1(t(i+1),w(1,i))+h*f1(t(i+2),w(1,i+1))+(h/2)*f1(t(i+3),k1)
w(1,i+2)=w(1,i)+k11
k2=v(1,i-1)+(h/2)*(f2(t(i),v(1,i-
1))+f2(t(i+1),v(1,i))+4*f2(t(i+2),v(1,i+1)))
k22=(h/2)*f2(t(i+1),v(1,i))+h*f2(t(i+2),v(1,i+1))+(h/2)*f2(t(i+3),k2)
v(1,i+2)=v(1,i)+k22
    end
end
ww=double(w(1,i+2))
vv=double(v(1,i+2))

end

```



```

syms t w v
format long
w0=[2.038775193568358,2.106734366687303,2.174693539806248,2.242652712
925193,2.310611886044139,2.378571059163084,2.446530232282029,2.514489
405400975,2.582448578519919,2.650407751638865,2.718366924757810]
v0=[3.058162790352537,3.024183203793064,2.990203617233592,2.956224030
674118,2.922244444114646,2.888264857555173,2.854285270995701,2.820305
684436228,2.786326097876755,2.752346511317283,2.718366924757810]
f1(t,w)=w+2*w0*(t-1)
f2(t,v)=v+2*v0*(t-1)
t(1)=1
h=0.02
for i=1:25
t(i+1)=t(i)+h
end
for j=1:11
    for i=1:2
if i==1
k1=h*f1(t(i),w0(1,j))
k2=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
k4=h*f1(t(i+1),w0(1,j)+k3(1,j))
w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
k11=h*f2(t(i),v0(1,j))
k22=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
k33=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
k44=h*f2(t(i+1),v0(1,j)+k33(1,j))
v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
else
k1=h*f1(t(i),w(1,i-1))
k2=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
k4=h*f1(t(i+1),w(1,i-1)+k3(1,j))

```

```

w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
k11=h*f2(t(i),v(1,i-1))
k22=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
k33=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
k44=h*f2(t(i+1),v(1,i-1)+k33(1,j))
v(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
end
for i=1:23
if i==1
k1=w0(1,j)+(h/2)*(f1(t(i),w0(1,j))+f1(t(i+1),w(1,1))+4*f1(t(i+2),w(1,2)))
k11=(h/2)*f1(t(i+1),w(1,1))+h*f1(t(i+2),w(1,2))+(h/2)*f1(t(i+3),k1(1,j))
w(1,i+2)=w(1,1)+k11(1,j)
k2=v0(1,j)+(h/2)*(f2(t(i),v0(1,j))+f2(t(i+1),v(1,1))+4*f2(t(i+2),v(1,2)))
k22=(h/2)*f2(t(i+1),v(1,1))+h*f2(t(i+2),v(1,2))+(h/2)*f2(t(i+3),k2(1,j))
v(1,i+2)=v(1,1)+k22(1,j)
elseif i==2
k1=w(1,1)+(h/2)*(f1(t(i),w(1,1))+f1(t(i+1),w(1,2))+4*f1(t(i+2),w(1,i+1)))
k11=(h/2)*f1(t(i+1),w(1,2))+h*f1(t(i+2),w(1,i+1))+(h/2)*f1(t(i+3),k1(1,j))
w(1,i+2)=w(1,2)+k11(1,j)
k2=v(1,1)+(h/2)*(f2(t(i),v(1,1))+f2(t(i+1),v(1,2))+4*f2(t(i+2),v(1,i+1)))
k22=(h/2)*f2(t(i+1),v(1,2))+h*f2(t(i+2),v(1,i+1))+(h/2)*f2(t(i+3),k2(1,j))
v(1,i+2)=v(1,2)+k22(1,j)
elseif i==3
k1=w(1,2)+(h/2)*(f1(t(i),w(1,2))+f1(t(i+1),w(1,i))+4*f1(t(i+2),w(1,i+1)))
k11=(h/2)*f1(t(i+1),w(1,i))+h*f1(t(i+2),w(1,i+1))+(h/2)*f1(t(i+3),k1(1,j))
w(1,i+2)=w(1,i)+k11(1,j)
k2=v(1,2)+(h/2)*(f2(t(i),v(1,2))+f2(t(i+1),v(1,i))+4*f2(t(i+2),v(1,i+1)))
k22=(h/2)*f2(t(i+1),v(1,2))+h*f2(t(i+2),v(1,i+1))+(h/2)*f2(t(i+3),k2(1,j))
v(1,i+2)=v(1,i)+k22(1,j)
else
k1=w(1,i-1)+(h/2)*(f1(t(i),w(1,i-1))+f1(t(i+1),w(1,i))+4*f1(t(i+2),w(1,i+1)))
k11=(h/2)*f1(t(i+1),w(1,i))+h*f1(t(i+2),w(1,i+1))+(h/2)*f1(t(i+3),k1(1,j))
w(1,i+2)=w(1,i)+k11(1,j)
k2=v(1,i-1)+(h/2)*(f2(t(i),v(1,i-1))+f2(t(i+1),v(1,i))+4*f2(t(i+2),v(1,i+1)))
k22=(h/2)*f2(t(i+1),v(1,i))+h*f2(t(i+2),v(1,i+1))+(h/2)*f2(t(i+3),k2(1,j))
v(1,i+2)=v(1,i)+k22(1,j)
end
end
ww=double(w(1,i+2))
vv=double(v(1,i+2))
end

```

```

syms t w v
format long
c0=[2.038775193568358,2.106734366687303,2.174693539806248,2.242652712
925193,2.310611886044139,2.378571059163084,2.446530232282029,2.514489
405400975,2.582448578519919,2.650407751638865,2.718366924757810]
d0=[3.058162790352537,3.024183203793064,2.990203617233592,2.956224030
674118,2.92224444114646,2.888264857555173,2.854285270995701,2.820305
684436228,2.786326097876755,2.752346511317283,2.718366924757810]
w0=[3.967945475290978,4.100210324467344,4.232475173643709,4.364740022
820075,4.497004871996443,4.629269721172808,4.761534570349173,4.893799
419525541,5.026064268701903,5.158329117878271,5.290593967054637]

```

```
v0=[5.951918212936468,5.885785788348284,5.819653363760102,5.753520939
171917,5.687388514583735,5.621256089995551,5.555123665407369,5.488991
240819185,5.422858816231002,5.356726391642821,5.290593967054637]
```

```
f1(t,w)=w+2*c0*(2-t)
f2(t,v)=v+2*d0*(2-t)
t(1)=1.5
h=0.02
for i=1:25
    t(i+1)=t(i)+h
end
for j=1:11
    for i=1:2
        if i==1
            k1=h*f1(t(i),w0(1,j))
            k2=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
            k3=h*f1(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
            k4=h*f1(t(i+1),w0(1,j)+k3(1,j))
            w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
            k11=h*f2(t(i),v0(1,j))
            k22=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
            k33=h*f2(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
            k44=h*f2(t(i+1),v0(1,j)+k33(1,j))
            v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
        else
            k1=h*f1(t(i),w(1,i-1))
            k2=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
            k3=h*f1(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
            k4=h*f1(t(i+1),w(1,i-1)+k3(1,j))
            w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
            k11=h*f2(t(i),v(1,i-1))
            k22=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
            k33=h*f2(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
            k44=h*f2(t(i+1),v(1,i-1)+k33(1,j))
            v(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
        end
    end
    for i=1:23
        if i==1
            k1=w0(1,j)+(h/2)*(f1(t(i),w0(1,j))+f1(t(i+1),w(1,1))+4*f1(t(i+2),w(1,2)))
            k11=(h/2)*f1(t(i+1),w(1,1))+h*f1(t(i+2),w(1,2))+(h/2)*f1(t(i+3),k1(1,j))
            w(1,i+2)=w(1,1)+k11(1,j)
            k2=v0(1,j)+(h/2)*(f2(t(i),v0(1,j))+f2(t(i+1),v(1,1))+4*f2(t(i+2),v(1,2)))
            k22=(h/2)*f2(t(i+1),v(1,1))+h*f2(t(i+2),v(1,2))+(h/2)*f2(t(i+3),k2(1,j))
            v(1,i+2)=v(1,1)+k22(1,j)
        elseif i==2
            k1=w(1,1)+(h/2)*(f1(t(i),w(1,1))+f1(t(i+1),w(1,2))+4*f1(t(i+2),w(1,i+1)))
            k11=(h/2)*f1(t(i+1),w(1,2))+h*f1(t(i+2),w(1,i+1))+(h/2)*f1(t(i+3),k1(1,j))
            w(1,i+2)=w(1,2)+k11(1,j)
            k2=v(1,1)+(h/2)*(f2(t(i),v(1,1))+f2(t(i+1),v(1,2))+4*f2(t(i+2),v(1,i+1)))
            k22=(h/2)*f2(t(i+1),v(1,2))+h*f2(t(i+2),v(1,i+1))+(h/2)*f2(t(i+3),k2(1,j))
            v(1,i+2)=v(1,2)+k22(1,j)
        elseif i==3
            k1=w(1,2)+(h/2)*(f1(t(i),w(1,2))+f1(t(i+1),w(1,i))+4*f1(t(i+2),w(1,i+1)))
        end
    end
end
```

```

k11=(h/2)*f1(t(i+1),w(1,i))+h*f1(t(i+2),w(1,i+1))+(h/2)*f1(t(i+3),k1(1,j))
w(1,i+2)=w(1,i)+k11(1,j)
k2=v(1,2)+(h/2)*(f2(t(i),v(1,2))+f2(t(i+1),v(1,i))+4*f2(t(i+2),v(1,i+1)))
k22=(h/2)*f2(t(i+1),v(1,i))+h*f2(t(i+2),v(1,i+1))+(h/2)*f2(t(i+3),k2(1,j))
v(1,i+2)=v(1,i)+k22(1,j)
    else
k1=w(1,i-1)+(h/2)*(f1(t(i),w(1,i-
1))+f1(t(i+1),w(1,i))+4*f1(t(i+2),w(1,i+1)))
k11=(h/2)*f1(t(i+1),w(1,i))+h*f1(t(i+2),w(1,i+1))+(h/2)*f1(t(i+3),k1(1,j))
w(1,i+2)=w(1,i)+k11(1,j)
k2=v(1,i-1)+(h/2)*(f2(t(i),v(1,i-
1))+f2(t(i+1),v(1,i))+4*f2(t(i+2),v(1,i+1)))
k22=(h/2)*f2(t(i+1),v(1,i))+h*f2(t(i+2),v(1,i+1))+(h/2)*f2(t(i+3),k2(1,j))
v(1,i+2)=v(1,i)+k22(1,j)
    end
end
ww=double(w(1,i+2))
vv=double(v(1,i+2))

end

```

## b- 2 – differentiable

### Picard Method

```

 syms t w v z q
format long e
w0=zeros(1,11)
v0=zeros(1,11)
for i=1:11
w0(1,i)=0.75+0.25*(i-1)*0.1
v0(1,i)=1.125-0.125*(i-1)*0.1
end
t0=0
n=50
for k=1:11
    for j=1:n
        if j==1
w(k,j)=w0(1,k)+int(v0(1,k),t0,t)
v(k,j)=w0(1,k)+int(w0(1,k),t0,t)
else
w(k,j)=w0(1,k)+int(v(k,j-1),t0,t)
v(k,j)=w0(1,k)+int(w(k,j-1),t0,t)
end
z(t)=w(k,j)
q(t)=v(k,j)
end
z(1)
q(1)
end

```

```

 $\text{syms } t \text{ w z v q}$ 
 $\text{format long}$ 
 $w_0 = [2.479411818960710e+00, 2.503298819910544e+00, 2.527185820860376e+00$ 
 $, 2.551072821810210e+00, 2.574959822760043e+00, 2.598846823709878e+00, 2.$ 
 $622733824659711e+00, 2.646620825609544e+00, 2.670507826559379e+00$ 
 $, 2.694394827509211e+00, 2.718281828459045e+00]$ 
 $v_0 = [2.617366609400000e+00, 2.627458131305905e+00, 2.637549653211810e+00$ 
 $, 2.647641175117714e+00, 2.657732697023618e+00, 2.667824218929523e+00, 2.$ 
 $677915740835428e+00, 2.68007262741332e+00, 2.698098784647236e+00, 2.708$ 
 $190306553141e+00, 2.718281828459045e+00]$ 
 $t_0 = 1$ 
 $n = 50$ 
 $\text{for } k=1:11$ 
 $\text{for } j=1:n$ 
 $\text{if } j==1$ 
 $\quad w(k,j) = w_0(1,k) + \int (v_0(1,k) + 2*v_0(1,k)*(t-1), t_0, t)$ 
 $\quad v(k,j) = v_0(1,k) + \int (w_0(1,k) + 2*w_0(1,k)*(t-1), t_0, t)$ 
 $\text{else}$ 
 $\quad w(k,j) = w_0(1,k) + \int (v(k,j-1) + 2*v_0(1,k)*(t-1), t_0, t)$ 
 $\quad v(k,j) = v_0(1,k) + \int (w(k,j-1) + 2*w_0(1,k)*(t-1), t_0, t)$ 
 $\text{end}$ 
 $z(t) = w(k,j)$ 
 $q(t) = v(k,j)$ 
 $\text{end}$ 
 $z(1.5)$ 
 $q(1.5)$ 
 $\text{end}$ 

```

```

 $\text{syms } t \text{ w v q z}$ 
 $\text{format long e}$ 
 $c_0 = [2.479411818960710e+00, 2.503298819910544e+00, 2.527185820860376e+00$ 
 $, 2.551072821810210e+00, 2.574959822760043e+00, 2.598846823709878e+00, 2.$ 
 $622733824659711e+00, 2.646620825609544e+00, 2.670507826559379e+00$ 
 $, 2.694394827509211e+00, 2.718281828459045e+00]$ 
 $d_0 = [2.617366609400000e+00, 2.627458131305905e+00, 2.637549653211810e+00$ 
 $, 2.647641175117714e+00, 2.657732697023618e+00, 2.667824218929523e+00, 2.$ 
 $677915740835428e+00, 2.68007262741332e+00, 2.698098784647236e+00, 2.708$ 
 $190306553141e+00, 2.718281828459045e+00]$ 
 $w_0 = [4.932442377592928e+00, 4.968220312397342e+00, 5.003998247201754e+00$ 
 $, 5.039776182006166e+00, 5.075554116810579e+00, 5.111332051614995e+00$ 
 $, 5.147109986419407e+00, 5.182887921223819e+00, 5.218665856028233e+00,$ 
 $5.254443790832645e+00, 5.290221725637058e+00]$ 
 $v_0 = [4.986723357976557e+00, 5.017073194742609e+00, 5.047423031508657e+00$ 
 $, 5.077772868274708e+00, 5.108122705040757e+00, 5.138472541806809e+00, 5.$ 
 $168822378572860e+00, 5.199172215338908e+00, 5.229522052104960e+00, 5.259$ 
 $871888871008e+00, 5.290221725637058e+00]$ 
 $t_0 = 1.5$ 
 $n = 50$ 
 $\text{for } k=1:11$ 
 $\text{for } j=1:n$ 
 $\text{if } j==1$ 
 $\quad w(k,j) = w_0(1,k) + \int (v_0(1,k) + 2*d_0(1,k)*(2-t), t_0, t)$ 
 $\quad v(k,j) = v_0(1,k) + \int (w_0(1,k) + 2*c_0(1,k)*(2-t), t_0, t)$ 
 $\text{else}$ 
 $\quad w(k,j) = w_0(1,k) + \int (v(k,j-1) + 2*d_0(1,k)*(2-t), t_0, t)$ 
 $\quad v(k,j) = v_0(1,k) + \int (w(k,j-1) + 2*c_0(1,k)*(2-t), t_0, t)$ 

```

```

end
z(t)=w(k,j)
q(t)=v(k,j)
end
z(2)
q(2)
end

```

## Runge-Kutta Method

```

 $\ddot{}$  syms t w v
format long
w0=zeros(1,11)
v0=zeros(1,11)
for i=1:11
w0(1,i)=0.75+0.25*(i-1)*0.1
v0(1,i)=1.125-0.125*(i-1)*0.1
end
h=0.02
t0=0
n=50
f1(t,v)=v
f2(t,w)=w
for i=1:n
t(i)=t0+i*h
end
for j=1:n
if j==1
for k=1:11
k1=h*f1(t0,v0(1,k))
k11=h*f2(t0,w0(1,k))
k2=h*f1(t0+(h/3),v0(1,k)+(k11/3))
k22=h*f2(t0+(h/3),w0(1,k)+(k11/3))
k3=h*f1(t0+(h/3),v0(1,k)+(k11/6)+(k22/6))
k33=h*f2(t0+(h/3),w0(1,k)+(k11/6)+(k2/6))
k4=h*f1(t0+(h/2),v0(1,k)+(k11/8)+(3*k33/8))
k44=h*f2(t0+(h/2),w0(1,k)+(k1/8)+(3*k3/8))
k5=h*f1(t0+h,v0(1,k)+(k11/2)-(3*k33/2)+(2*k44))
k55=h*f2(t0+h,w0(1,k)+(k1/2)-(3*k3/2)+(2*k4))
w(j,k)=w0(1,k)+(1/6)*(k1+4*k4+k5)
v(j,k)=v0(1,k)+(1/6)*(k11+4*k44+k55)
end
else
for k=1:11
k1=h*f1(t(j-1),v(j-1,k))
k11=h*f2(t(j-1),w(j-1,k))
k2=h*f1(t(j-1)+(h/3),v(j-1,k)+(k11/3))
k22=h*f2(t(j-1)+(h/3),w(j-1,k)+(k11/3))
k3=h*f1(t(j-1)+(h/3),v(j-1,k)+(k11/6)+(k22/6))
k33=h*f2(t(j-1)+(h/3),w(j-1,k)+(k11/6)+(k2/6))
k4=h*f1(t(j-1)+(h/2),v(j-1,k)+(k11/8)+(3*k33/8))
k44=h*f2(t(j-1)+(h/2),w(j-1,k)+(k1/8)+(3*k3/8))
k5=h*f1(t(j-1)+h,v(j-1,k)+(k11/2)-(3*k33/2)+(2*k44))
k55=h*f2(t(j-1)+h,w(j-1,k)+(k1/2)-(3*k3/2)+(2*k4))
w(j,k)=w(j-1,k)+(1/6)*(k1+4*k4+k5)
v(j,k)=v(j-1,k)+(1/6)*(k11+4*k44+k55);
end
end

```

```

end
ww=double(w)
vv=double(v)

# syms t w z
format long e
w0=[2.479411818379155,2.503298819326746,2.527185820274337,2.551072821221929
,2.574959822169520,2.598846823117111,2.622733824064703,2.646620825012294,2.
670507825959885,2.694394826907477,2.718281827855068]
v0=[2.617366608849098,2.627458130749695,2.637549652650292,2.647641174550889
,2.657732696451486,2.667824218352083,2.677915740252680,2.688007262153277,2.
698098784053874,2.708190305954471,2.718281827855068]
h=0.02
t0=1
n=25
f1(t,v)=v+2*v0*(t-1)
f2(t,w)=w+2*w0*(t-1)
for i=1:n
    t(i)=t0+i*h
end
for j=1:n
    if j==1
        for k=1:11
            k1=h*f1(t0,v0(1,k))
            k11=h*f2(t0,w0(1,k))
            k2=h*f1(t0+(h/3),v0(1,k)+(k11(1,k)/3))
            k22=h*f2(t0+(h/3),w0(1,k)+(k1(1,k)/3))
            k3=h*f1(t0+(h/3),v0(1,k)+(k11(1,k)/6)+(k22(1,k)/6))
            k33=h*f2(t0+(h/3),w0(1,k)+(k1(1,k)/6)+(k2(1,k)/6))
            k4=h*f1(t0+(h/2),v0(1,k)+(k11(1,k)/8)+(3*k33(1,k)/8))
            k44=h*f2(t0+(h/2),w0(1,k)+(k1(1,k)/8)+(3*k3(1,k)/8))
            k5=h*f1(t0+h,v0(1,k)+(k11(1,k)/2)-(3*k33(1,k)/2)+(2*k44(1,k)))
            k55=h*f2(t0+h,w0(1,k)+(k1(1,k)/2)-(3*k3(1,k)/2)+(2*k4(1,k)))
            w(j,k)=w0(1,k)+(1/6)*(k1(1,k)+4*k4(1,k)+k5(1,k))
            v(j,k)=v0(1,k)+(1/6)*(k11(1,k)+4*k44(1,k)+k55(1,k))
        end
    else
        for k=1:11
            k1=h*f1(t(j-1),v(j-1,k))
            k11=h*f2(t(j-1),w(j-1,k))
            k2=h*f1(t(j-1)+(h/3),v(j-1,k)+(k11(1,k)/3))
            k22=h*f2(t(j-1)+(h/3),w(j-1,k)+(k1(1,k)/3))
            k3=h*f1(t(j-1)+(h/3),v(j-1,k)+(k11(1,k)/6)+(k22(1,k)/6))
            k33=h*f2(t(j-1)+(h/3),w(j-1,k)+(k1(1,k)/6)+(k2(1,k)/6))
            k4=h*f1(t(j-1)+(h/2),v(j-1,k)+(k11(1,k)/8)+(3*k33(1,k)/8))
            k44=h*f2(t(j-1)+(h/2),w(j-1,k)+(k1(1,k)/8)+(3*k3(1,k)/8))
            k5=h*f1(t(j-1)+h,v(j-1,k)+(k11(1,k)/2)-(3*k33(1,k)/2)+(2*k44(1,k)))
            k55=h*f2(t(j-1)+h,w(j-1,k)+(k1(1,k)/2)-(3*k3(1,k)/2)+(2*k4(1,k)))
            w(j,k)=w(j-1,k)+(1/6)*(k1(1,k)+4*k4(1,k)+k5(1,k))
            v(j,k)=v(j-1,k)+(1/6)*(k11(1,k)+4*k44(1,k)+k55(1,k))
        end
    end
end
ww=double(w)
vv=double(v)

```

```

syms t w v
format long
c0=[2.479411818379155,2.503298819326746,2.527185820274337,2.551072821221929
,2.574959822169520,2.598846823117111,2.622733824064703,2.646620825012294,2.
670507825959885,2.694394826907477,2.718281827855068]
d0=[2.617366608849098,2.627458130749695,2.637549652650292,2.647641174550889
,2.657732696451486,2.667824218352083,2.677915740252680,2.688007262153277,2.
698098784053874,2.708190305954471,2.718281827855068]
w0=[4.932442375089241,4.968220309877111,5.003998244664980,5.039776179452850
,5.075554114240719,5.111332049028587,5.147109983816458,5.182887918604327,5.
21866585392196,5.254443788180066,5.290221722967934]
v0=[4.986723355475636,5.017073192224866,5.047423028974096,5.077772865723326
,5.108122702472556,5.138472539221786,5.168822375971016,5.199172212720246,5.
229522049469475,5.259871886218705,5.290221722967934]
h=0.02
t0=1.5
n=25
f1(t,v)=v+2*d0*(2-t)
f2(t,w)=w+2*c0*(2-t)
for i=1:n
    t(i)=t0+i*h
end
for j=1:n
    if j==1
        for k=1:11
            k1=h*f1(t0,v0(1,k))
            k11=h*f2(t0,w0(1,k))
            k2=h*f1(t0+(h/3),v0(1,k)+(k11(1,k)/3))
            k22=h*f2(t0+(h/3),w0(1,k)+(k1(1,k)/3))
            k3=h*f1(t0+(h/3),v0(1,k)+(k11(1,k)/6)+(k22(1,k)/6))
            k33=h*f2(t0+(h/3),w0(1,k)+(k1(1,k)/6)+(k2(1,k)/6))
            k4=h*f1(t0+(h/2),v0(1,k)+(k11(1,k)/8)+(3*k33(1,k)/8))
            k44=h*f2(t0+(h/2),w0(1,k)+(k1(1,k)/8)+(3*k3(1,k)/8))
            k5=h*f1(t0+h,v0(1,k)+(k11(1,k)/2)-(3*k33(1,k)/2)+(2*k44(1,k)))
            k55=h*f2(t0+h,w0(1,k)+(k1(1,k)/2)-(3*k3(1,k)/2)+(2*k4(1,k)))
            w(j,k)=w0(1,k)+(1/6)*(k1(1,k)+4*k4(1,k)+k5(1,k))
            v(j,k)=z0(1,k)+(1/6)*(k11(1,k)+4*k44(1,k)+k55(1,k))
        end
    else
        for k=1:11
            k1=h*f1(t(j-1),v(j-1,k))
            k11=h*f2(t(j-1),w(j-1,k))
            k2=h*f1(t(j-1)+(h/3),v(j-1,k)+(k11(1,k)/3))
            k22=h*f2(t(j-1)+(h/3),w(j-1,k)+(k1(1,k)/3))
            k3=h*f1(t(j-1)+(h/3),v(j-1,k)+(k11(1,k)/6)+(k22(1,k)/6))
            k33=h*f2(t(j-1)+(h/3),w(j-1,k)+(k1(1,k)/6)+(k2(1,k)/6))
            k4=h*f1(t(j-1)+(h/2),v(j-1,k)+(k11(1,k)/8)+(3*k33(1,k)/8))
            k44=h*f2(t(j-1)+(h/2),w(j-1,k)+(k1(1,k)/8)+(3*k3(1,k)/8))
            k5=h*f1(t(j-1)+h,v(j-1,k)+(k11(1,k)/2)-(3*k33(1,k)/2)+(2*k44(1,k)))
            k55=h*f2(t(j-1)+h,w(j-1,k)+(k1(1,k)/2)-(3*k3(1,k)/2)+(2*k4(1,k)))
            w(j,k)=w(j-1,k)+(1/6)*(k1(1,k)+4*k4(1,k)+k5(1,k))
            v(j,k)=v(j-1,k)+(1/6)*(k11(1,k)+4*k44(1,k)+k55(1,k))
        end
    end
end
ww=double(w)
vv=double(v)

```

## General Linear Method

**K = 4**

```

syms t w v
format long
f1(t,v)=v
f2(t,w)=w
a=0
b=1
N=50
h=(b-a)/N
t(1)=a
for j=1:11
    w0(1,j)=0.75+(0.25*(j-1)*0.1)
    v0(1,j)=1.125-(0.125*(j-1)*0.1)
    t(1)=a
for i=1:3
    t(i+1)=t(i)+h
    if i==1
        k1(j,i)=h*f1(t(i),v0(1,j))
        k11(j,i)=h*f2(t(i),w0(1,j))
        k2(j,i)=h*f1(t(i)+0.5*h,v0(1,j)+0.5*k1(j,i))
        k22(j,i)=h*f2(t(i)+0.5*h,w0(1,j)+0.5*k1(j,i))
        k3(j,i)=h*f1(t(i)+0.5*h,v0(1,j)+0.5*k22(j,i))
        k33(j,i)=h*f2(t(i)+0.5*h,w0(1,j)+0.5*k2(j,i))
        k4(j,i)=h*f1(t(i+1),v0(1,j)+k33(j,i))
        k44(j,i)=h*f2(t(i+1),w0(1,j)+k3(j,i))
        w(j,i)=w0(1,j)+(k1(j,i)+2*(k2(j,i)+k3(j,i))+k4(j,i))/6
        v(j,i)=v0(1,j)+(k11(j,i)+2*(k22(j,i)+k33(j,i))+k44(j,i))/6
    else
        k1(j,i)=h*f1(t(i),v(j,i-1))
        k11(j,i)=h*f2(t(i),w(j,i-1))
        k2(j,i)=h*f1(t(i)+0.5*h,v(j,i-1)+0.5*k11(j,i))
        k22(j,i)=h*f2(t(i)+0.5*h,w(j,i-1)+0.5*k1(j,i))
        k3(j,i)=h*f1(t(i)+0.5*h,v(j,i-1)+0.5*k22(j,i))
        k33(j,i)=h*f2(t(i)+0.5*h,w(j,i-1)+0.5*k2(j,i))
        k4(j,i)=h*f1(t(i+1),v(j,i-1)+k33(j,i))
        k44(j,i)=h*f2(t(i+1),w(j,i-1)+k3(j,i))
        w(j,i)=w(j,i-1)+(k1(j,i)+2*(k2(j,i)+k3(j,i))+k4(j,i))/6
        v(j,i)=v(j,i-1)+(k11(j,i)+2*(k22(j,i)+k33(j,i))+k44(j,i))/6
    end
end
for i=4:N
    t0=a+i*h
    p1(1,j)=55*f1(t(4),v(j,3))-59*f1(t(3),v(j,2))+37*f1(t(2),v(j,1))-
    9*f1(t(1),v0(1,j))
    z0(1,j)=w(j,3)+h*(p1(1,j))/24
    d1(1,j)=55*f2(t(4),w(j,3))-59*f2(t(3),w(j,2))+37*f2(t(2),w(j,1))-
    9*f2(t(1),w0(1,j))
    q0(1,j)=v(j,3)+h*(d1(1,j))/24
    for e=1:3
        t(e)=t(e+1)
        if e==1
            w0(1,j)=w(j,1)
            v0(1,j)=v(j,1)
        else
            w(j,e-1)=w(j,e)
        end
    end
end

```

```

v(j,e-1)=v(j,e)
    end
end
t(4)=t0
w(j,3)=z0(1,j)
v(j,3)=q0(1,j)

end
ww=double(w)
vv=double(v)
end

syms t w v
format long

c0=[2.479411685495919,2.503298685954558,2.527185686413198,2.551072686871839
,2.574959687330481,2.598846687789121,2.622733688247763,2.646620688706403,2.
670507689165044,2.694394689623684,2.718281690082326]
d0=[2.617366483408444,2.627458004075831,2.637549524743220,2.647641045410608
,2.657732566077996,2.667824086745385,2.677915607412773,2.688007128080161,2.
698098648747549,2.708190169414938,2.718281690082326]
f1(t,v)=v+2*d0*(t-1)
f2(t,w)=w+2*c0*(t-1)
a=1
b=1.5
N=25
h=(b-a)/N
t(1)=a
for j=11:11
w0=[2.479411685495919,2.503298685954558,2.527185686413198,2.551072686871839
,2.574959687330481,2.598846687789121,2.622733688247763,2.646620688706403,2.
670507689165044,2.694394689623684,2.718281690082326]
v0=[2.617366483408444,2.627458004075831,2.637549524743220,2.647641045410608
,2.657732566077996,2.667824086745385,2.677915607412773,2.688007128080161,2.
698098648747549,2.708190169414938,2.718281690082326]

t(1)=a
for i=1:3
    t(i+1)=t(i)+h
    if i==1
        k1=h*f1(t(i),v0(1,j))
        k11=h*f2(t(i),w0(1,j))
        k2=h*f1(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
        k22=h*f2(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
        k3=h*f1(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
        k33=h*f2(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
        k4=h*f1(t(i+1),v0(1,j)+k33(1,j))
        k44=h*f2(t(i+1),w0(1,j)+k3(1,j))
        w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
        v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
    else
        k1=h*f1(t(i),v(1,i-1))
        k11=h*f2(t(i),w(1,i-1))
        k2=h*f1(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
        k22=h*f2(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
        k3=h*f1(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
        k33=h*f2(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
        k4=h*f1(t(i+1),v(1,i-1)+k33(1,j))
        k44=h*f2(t(i+1),w(1,i-1)+k3(1,j))
    end
end

```

```

w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
v(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
end
end
for i=4:N
t0=a+i*h
p1=55*f1(t(4),v(1,3))-59*f1(t(3),v(1,2))+37*f1(t(2),v(1,1))-
9*f1(t(1),v0(1,j))
z0(1,j)=w(1,3)+h*(p1(1,j))/24
d1=55*f2(t(4),w(1,3))-59*f2(t(3),w(1,2))+37*f2(t(2),w(1,1))-
9*f2(t(1),w0(1,j))
q0(1,j)=v(1,3)+h*(d1(1,j))/24
for e=1:3
t(e)=t(e+1)
if e==1
w0(1,j)=w(1,1)
v0(1,j)=v(1,1)
else
w(1,e-1)=w(1,e)
v(1,e-1)=v(1,e)
end
end
t(4)=t0
w(1,3)=z0(1,j)
v(1,3)=q0(1,j)
end
ww=double(w)
vv=double(v)
end

```

```

syms t w v
format long

c0=[2.479411685495919,2.503298685954558,2.527185686413198,2.551072686871839
,2.574959687330481,2.598846687789121,2.622733688247763,2.646620688706403,2.
670507689165044,2.694394689623684,2.718281690082326]
d0=[2.617366483408444,2.627458004075831,2.637549524743220,2.647641045410608
,2.657732566077996,2.667824086745385,2.677915607412773,2.688007128080161,2.
698098648747549,2.708190169414938,2.718281690082326]

```

```

f1(t,v)=v+2*d0*(2-t)
f2(t,w)=w+2*c0*(2-t)

```

```

a=1.5
b=2
N=25
h=(b-a)/N

t(1)=a
for j=11:11
w0=[4.932441823847529,4.968219755003775,5.003997686160021,5.039775617316268
,5.075553548472518,5.111331479628763,5.147109410785012,5.182887341941259,5.
218665273097508,5.254443204253756,5.290221135410005]
v0=[4.986722805046231,5.017072638082606,5.047422471118983,5.077772304155361
,5.108122137191739,5.138471970228117,5.168821803264494,5.199171636300870,5.
229521469337248,5.259871302373626,5.290221135410004]
t(1)=a
for i=1:3

```

```

t(i+1)=t(i)+h
if i==1
k1=h*f1(t(i),v0(1,j))
k11=h*f2(t(i),w0(1,j))
k2=h*f1(t(i)+0.5*h,v0(1,j)+0.5*k11(1,j))
k22=h*f2(t(i)+0.5*h,w0(1,j)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,v0(1,j)+0.5*k22(1,j))
k33=h*f2(t(i)+0.5*h,w0(1,j)+0.5*k2(1,j))
k4=h*f1(t(i+1),v0(1,j)+k33(1,j))
k44=h*f2(t(i+1),w0(1,j)+k3(1,j))
w(1,i)=w0(1,j)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
v(1,i)=v0(1,j)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
else
k1=h*f1(t(i),v(1,i-1))
k11=h*f2(t(i),w(1,i-1))
k2=h*f1(t(i)+0.5*h,v(1,i-1)+0.5*k11(1,j))
k22=h*f2(t(i)+0.5*h,w(1,i-1)+0.5*k1(1,j))
k3=h*f1(t(i)+0.5*h,v(1,i-1)+0.5*k22(1,j))
k33=h*f2(t(i)+0.5*h,w(1,i-1)+0.5*k2(1,j))
k4=h*f1(t(i+1),v(1,i-1)+k33(1,j))
k44=h*f2(t(i+1),w(1,i-1)+k3(1,j))
w(1,i)=w(1,i-1)+(k1(1,j)+2*(k2(1,j)+k3(1,j))+k4(1,j))/6
v(1,i)=v(1,i-1)+(k11(1,j)+2*(k22(1,j)+k33(1,j))+k44(1,j))/6
end
end
for i=4:N
t0=a+i*h
p1=55*f1(t(4),v(1,3))-59*f1(t(3),v(1,2))+37*f1(t(2),v(1,1))-
9*f1(t(1),v0(1,j))
z0(1,j)=w(1,3)+h*(p1(1,j))/24
d1=55*f2(t(4),w(1,3))-59*f2(t(3),w(1,2))+37*f2(t(2),w(1,1))-
9*f2(t(1),w0(1,j))
q0(1,j)=v(1,3)+h*(d1(1,j))/24
for e=1:3
t(e)=t(e+1)
if e==1
w0(1,j)=w(1,1)
v0(1,j)=v(1,1)
else
w(1,e-1)=w(1,e)
v(1,e-1)=v(1,e)
end
end
t(4)=t0
w(1,3)=z0(1,j)
v(1,3)=q0(1,j)
end
ww=double(w)
vv=double(v)
end

```

## **K = 5**

```

 $\text{syms } t \text{ w v}$ 
 $\text{format long}$ 
 $f1(t,v)=v$ 
 $f2(t,w)=w$ 
 $a=0$ 
 $b=1$ 
 $N=50$ 

```

```

h=(b-a)/N
t(1)=a
for j=1:11
    w0(1,j)=0.75+(0.25*(j-1)*0.1)
    v0(1,j)=1.125-(0.125*(j-1)*0.1)
    t(1)=a
for i=1:4
    t(i+1)=t(i)+h
if i==1
    k1(j,i)=h*f1(t(i),v0(1,j))
    k11(j,i)=h*f2(t(i),w0(1,j))
    k2(j,i)=h*f1(t(i)+(h/3),v0(1,j)+(k1(j,i)/3))
    k22(j,i)=h*f2(t(i)+(h/3),w0(1,j)+(k1(j,i)/3))
    k3(j,i)=h*f1(t(i)+(h/3),v0(1,j)+(k11(j,i)/6)+(k22(j,i)/6))
    k33(j,i)=h*f2(t(i)+(h/3),w0(1,j)+(k1(j,i)/6)+(k2(j,i)/6))
    k4(j,i)=h*f1(t(i)+(h/2),v0(1,j)+(k11(j,i)/8)+(3*k33(j,i)/8))
    k44(j,i)=h*f2(t(i)+(h/2),w0(1,j)+(k1(j,i)/8)+(3*k3(j,i)/8))
    k5(j,i)=h*f1(t(i+1),v0(1,j)+(k11(j,i)/2)-(3*k33(j,i)/2)+(2*k44(j,i)))
    k55(j,i)=h*f2(t(i+1),w0(1,j)+(k1(j,i)/2)-(3*k3(j,i)/2)+(2*k4(j,i)))
    w(j,i)=w0(1,j)+(k1(j,i)+4*k4(j,i)+k5(j,i))/6
    v(j,i)=v0(1,j)+(k11(j,i)+4*k44(j,i)+k55(j,i))/6
else
    k1(j,i)=h*f1(t(i),v(j,i-1))
    k11(j,i)=h*f2(t(i),w(j,i-1))
    k2(j,i)=h*f1(t(i)+(h/3),v(j,i-1)+(k11(j,i)/3))
    k22(j,i)=h*f2(t(i)+(h/3),w(j,i-1)+(k1(j,i)/3))
    k3(j,i)=h*f1(t(i)+(h/3),v(j,i-1)+(k11(j,i)/6)+(k22(j,i)/6))
    k33(j,i)=h*f2(t(i)+(h/3),w(j,i-1)+(k1(j,i)/6)+(k2(j,i)/6))
    k4(j,i)=h*f1(t(i)+(h/2),v(j,i-1)+(k11(j,i)/8)+(3*k33(j,i)/8))
    k44(j,i)=h*f2(t(i)+(h/2),w(j,i-1)+(k1(j,i)/8)+(3*k3(j,i)/8))
    k5(j,i)=h*f1(t(i+1),v(j,i-1)+(k11(j,i)/2)-(3*k33(j,i)/2)+(2*k44(j,i)))
    k55(j,i)=h*f2(t(i+1),w(j,i-1)+(k1(j,i)/2)-(3*k3(j,i)/2)+(2*k4(j,i)))
    w(j,i)=w(j,i-1)+(k1(j,i)+4*k4(j,i)+k5(j,i))/6
    v(j,i)=v(j,i-1)+(k11(j,i)+4*k44(j,i)+k55(j,i))/6
end
end
for i=5:N
    t0=a+i*h
    p1(1,j)=1901*f1(t(5),v(j,4))-2774*f1(t(4),v(j,3))+2616*f1(t(3),v(j,2))-
    1274*f1(t(2),v(j,1))+251*f1(t(1),v0(1,j))
    z0(1,j)=w(j,4)+h*(p1(1,j))/720
    d1(1,j)=1901*f2(t(5),w(j,4))-2774*f2(t(4),w(j,3))+2616*f2(t(3),w(j,2))-
    1274*f2(t(2),w(j,1))+251*f2(t(1),w0(1,j))
    q0(1,j)=v(j,4)+h*(d1(1,j))/720
    for e=1:4
        t(e)=t(e+1)
        if e==1
            w0(1,j)=w(j,1)
            v0(1,j)=v(j,1)
        else
            w(j,e-1)=w(j,e)
            v(j,e-1)=v(j,e)
        end
    end
    t(5)=t0
    w(j,4)=z0(1,j)
    v(j,4)=q0(1,j)
end
ww=double(w)
vv=double(v)
end

```

```

syms t w v
format long

c0=[2.479411816608218,2.503298817535054,2.527185818461892,2.551072819388729
,2.574959820315566,2.598846821242404,2.622733822169241,2.646620823096077,2.
670507824022915,2.694394824949753,2.718281825876591]
d0=[2.617366606910388,2.627458128807009,2.637549650703629,2.647641172600248
,2.657732694496869,2.667824216393489,2.677915738290109,2.688007260186729,2.
698098782083350,2.708190303979970,2.718281825876590]
f1(t,v)=v+2*d0*(t-1)
f2(t,w)=w+2*c0*(t-1)
a=1
b=1.5
N=25
h=(b-a)/N
t(1)=a
for j=8:8
w0=[2.479411816608218,2.503298817535054,2.527185818461892,2.551072819388729
,2.574959820315566,2.598846821242404,2.622733822169241,2.646620823096077,2.
670507824022915,2.694394824949753,2.718281825876591]
v0=[2.617366606910388,2.627458128807009,2.637549650703629,2.647641172600248
,2.657732694496869,2.667824216393489,2.677915738290109,2.688007260186729,2.
698098782083350,2.708190303979970,2.718281825876590]
    t(1)=a
for i=1:4
    t(i+1)=t(i)+h
    if i==1
        k1=h*f1(t(i),v0(1,j))
        k11=h*f2(t(i),w0(1,j))
        k2=h*f1(t(i)+(h/3),v0(1,j)+(k11(1,j)/3))
        k22=h*f2(t(i)+(h/3),w0(1,j)+(k1(1,j)/3))
        k3=h*f1(t(i)+(h/3),v0(1,j)+(k11(1,j)/6)+(k22(1,j)/6))
        k33=h*f2(t(i)+(h/3),w0(1,j)+(k1(1,j)/6)+(k2(1,j)/6))
        k4=h*f1(t(i)+(h/2),v0(1,j)+(k11(1,j)/8)+(3*k33(1,j)/8))
        k44=h*f2(t(i)+(h/2),w0(1,j)+(k1(1,j)/8)+(3*k3(1,j)/8))
        k5=h*f1(t(i+1),v0(1,j)+(k11(1,j)/2)-(3*k33(1,j)/2)+(2*k44(1,j)))
        k55=h*f2(t(i+1),w0(1,j)+(k1(1,j)/2)-(3*k3(1,j)/2)+(2*k4(1,j)))
        w(1,i)=w0(1,j)+(k1(1,j)+4*k4(1,j)+k5(1,j))/6
        v(1,i)=v0(1,j)+(k11(1,j)+4*k44(1,j)+k55(1,j))/6

    else
        k1=h*f1(t(i),v(1,i-1))
        k11=h*f2(t(i),w(1,i-1))
        k2=h*f1(t(i)+(h/3),v(1,i-1)+(k11(1,j)/3))
        k22=h*f2(t(i)+(h/3),w(1,i-1)+(k1(1,j)/3))
        k3=h*f1(t(i)+(h/3),v(1,i-1)+(k11(1,j)/6)+(k22(1,j)/6))
        k33=h*f2(t(i)+(h/3),w(1,i-1)+(k1(1,j)/6)+(k2(1,j)/6))
        k4=h*f1(t(i)+(h/2),v(1,i-1)+(k11(1,j)/8)+(3*k33(1,j)/8))
        k44=h*f2(t(i)+(h/2),w(1,i-1)+(k1(1,j)/8)+(3*k3(1,j)/8))
        k5=h*f1(t(i+1),v(1,i-1)+(k11(1,j)/2)-(3*k33(1,j)/2)+(2*k44(1,j)))
        k55=h*f2(t(i+1),w(1,i-1)+(k1(1,j)/2)-(3*k3(1,j)/2)+(2*k4(1,j)))
        w(1,i)=w(1,i-1)+(k1(1,j)+4*k4(1,j)+k5(1,j))/6
        v(1,i)=v(1,i-1)+(k11(1,j)+4*k44(1,j)+k55(1,j))/6
    end
end
for i=5:N
    t0=a+i*h
    p1=1901*f1(t(5),v(1,4))-2774*f1(t(4),v(1,3))+2616*f1(t(3),v(1,2))-
1274*f1(t(2),v(1,1))+251*f1(t(1),v0(1,j))
    z0(1,j)=w(1,4)+h*(p1(1,j))/720

```

```

d1=1901*f2(t(5),w(1,4))-2774*f2(t(4),w(1,3))+2616*f2(t(3),w(1,2))-
1274*f2(t(2),w(1,1))+251*f2(t(1),w0(1,j))
q0(1,j)=v(1,4)+h*(d1(1,j))/720
for e=1:4
    t(e)=t(e+1)
    if e==1
        w0(1,j)=w(1,1)
        v0(1,j)=v(1,1)
    else
        w(1,e-1)=w(1,e)
        v(1,e-1)=v(1,e)
    end
end
t(5)=t0
w(1,4)=z0(1,j)
v(1,4)=q0(1,j)
end
ww=double(w)
vv=double(v)
end

```

```

syms t w v
format long
c0=[2.479411816608218,2.503298817535054,2.527185818461892,2.551072819
388729,2.574959820315566,2.598846821242404,2.622733822169241,2.646620
823096077,2.670507824022915,2.694394824949753,2.718281825876591]
d0=[2.617366606910388,2.627458128807009,2.637549650703629,2.647641172
600248,2.657732694496869,2.667824216393489,2.677915738290109,2.688007
260186729,2.698098782083350,2.708190303979970,2.718281825876590]
f1(t,v)=v+2*d0*(2-t)
f2(t,w)=w+2*c0*(2-t)
a=1.5
b=2
N=25
h=(b-a)/N
t(1)=a
for j=1:11
w0=[4.932442367301161,4.968220302036066,5.003998236770974,5.039776171
505876,5.075554106240784,5.111332040975690,5.147109975710596,5.182887
910445476,5.218665845180407,5.254443779915314,5.290221714650222]
v0=[4.986723347668000,5.017073184366222,5.047423021064444,5.077772857
762664,5.108122694460887,5.138472531159110,5.168822367857331,5.199172
204555550,5.229522041253775,5.2598718779519985.290221714650220]
t(1)=a
for i=1:4
t(i+1)=t(i)+h
if i==1
k1=h*f1(t(i),v0(1,j))
k11=h*f2(t(i),w0(1,j))
k2=h*f1(t(i)+(h/3),v0(1,j)+(k11(1,j)/3))
k22=h*f2(t(i)+(h/3),w0(1,j)+(k1(1,j)/3))
k3=h*f1(t(i)+(h/3),v0(1,j)+(k11(1,j)/6)+(k22(1,j)/6))
k33=h*f2(t(i)+(h/3),w0(1,j)+(k1(1,j)/6)+(k2(1,j)/6))
k4=h*f1(t(i)+(h/2),v0(1,j)+(k11(1,j)/8)+(3*k33(1,j)/8))
k44=h*f2(t(i)+(h/2),w0(1,j)+(k1(1,j)/8)+(3*k3(1,j)/8))
k5=h*f1(t(i+1),v0(1,j)+(k11(1,j)/2)-(3*k33(1,j)/2)+(2*k44(1,j)))
k55=h*f2(t(i+1),w0(1,j)+(k1(1,j)/2)-(3*k3(1,j)/2)+(2*k4(1,j)))

```

```

w(1,i)=w0(1,j)+(k1(1,j)+4*k4(1,j)+k5(1,j))/6
z(1,i)=v0(1,j)+(k11(1,j)+4*k44(1,j)+k55(1,j))/6

else
k1=h*f1(t(i),v(1,i-1))
k11=h*f2(t(i),w(1,i-1))
k2=h*f1(t(i)+(h/3),v(1,i-1)+(k11(1,j)/3))
k22=h*f2(t(i)+(h/3),w(1,i-1)+(k1(1,j)/3))
k3=h*f1(t(i)+(h/3),v(1,i-1)+(k11(1,j)/6)+(k22(1,j)/6))
k33=h*f2(t(i)+(h/3),w(1,i-1)+(k1(1,j)/6)+(k2(1,j)/6))
k4=h*f1(t(i)+(h/2),v(1,i-1)+(k11(1,j)/8)+(3*k33(1,j)/8))
k44=h*f2(t(i)+(h/2),w(1,i-1)+(k1(1,j)/8)+(3*k3(1,j)/8))
k5=h*f1(t(i+1),v(1,i-1)+(k11(1,j)/2)-(3*k33(1,j)/2)+(2*k44(1,j)))
k55=h*f2(t(i+1),w(1,i-1)+(k1(1,j)/2)-(3*k3(1,j)/2)+(2*k4(1,j)))
w(1,i)=w(1,i-1)+(k1(1,j)+4*k4(1,j)+k5(1,j))/6
v(1,i)=v(1,i-1)+(k11(1,j)+4*k44(1,j)+k55(1,j))/6
end
end
for i=5:N
t0=a+i*h
p1=1901*f1(t(5),v(1,4))-2774*f1(t(4),v(1,3))+2616*f1(t(3),v(1,2))-
1274*f1(t(2),v(1,1))+251*f1(t(1),v0(1,j))
z0(1,j)=w(1,4)+h*(p1(1,j))/720
d1=1901*f2(t(5),w(1,4))-2774*f2(t(4),w(1,3))+2616*f2(t(3),w(1,2))-
1274*f2(t(2),w(1,1))+251*f2(t(1),w0(1,j))
q0(1,j)=v(1,4)+h*(d1(1,j))/720
for e=1:4
t(e)=t(e+1)
if e==1
w0(1,j)=w(1,1)
v0(1,j)=v(1,1)
else
w(1,e-1)=w(1,e)
v(1,e-1)=v(1,e)

end
end
t(5)=t0
w(1,4)=z0(1,j)
v(1,4)=q0(1,j)
end
ww=double(w)
vv=double(v)
end

```

## Variational Iteration Method (VIM)

```

# syms t w v h z
format long
w0=zeros(1,11)
v0=zeros(1,11)
a=1
b=0
for i=1:11
w0(1,i)=0.75+0.25*(i-1)*0.1
v0(1,i)=1.125-0.125*(i-1)*0.1
end
t0=0
n=18

```

```

for k=2:2
for i=1:n
if i==1
w(1,i)=w0(1,k)-int(-a*v0(1,k)-b,t0,t)
v(1,i)=v0(1,k)-int(-a*w0(1,k)-b,t0,t)
else
w(1,i)=w0(1,k)-int(-a*v(1,i-1)-b,t0,t)
v(1,i)=v0(1,k)-int(-a*w(1,i-1)-b,t0,t)
end
end
h(t)= w(1,i)
h(1)
z(t)=v(1,i)
z(1)
end

```



```

syms t w v h z
format long e
w0=[2.479411818960710,2.503298819910544,2.527185820860377,2.551072821
810211,2.574959822760044,2.598846823709878,2.622733824659711,
2.646620825609545,2.670507826559379,2.694394827509212,
2.718281828459046]
v0=[2.617366609400000,2.627458131305905,2.637549653211810,
2.647641175117714,2.657732697023619,2.667824218929523,2.6779157408354
28,2.688007262741332,2.698098784647236,2.708190306553141,
2.718281828459046]
a=1
for i=1:11
b1=2*w0*(t-1)
b2=2*v0*(t-1)
end
t0=1
n=18
for k=1:1
for i=1:n
if i==1
w(1,i)=w0(1,k)-int(-a*v0(1,k)-b2(1,k),t0,t)
v(1,i)=v0(1,k)-int(-a*w0(1,k)-b1(1,k),t0,t)
else
w(1,i)=w0(1,k)-int(-a*v(1,i-1)-b2(1,k),t0,t)
v(1,i)=v0(1,k)-int(-a*w(1,i-1)-b1(1,k),t0,t)
end
end
h(t)= w(1,i)
h(1.5)
z(t)=v(1,i)
z(1.5)
end

```



```

syms t w v h z
format long
c0=[2.479411818960710,2.503298819910544,2.527185820860377,2.551072821
810211,2.574959822760044,2.598846823709878,2.622733824659711,2.646620
825609545,2.670507826559379,2.694394827509212,2.718281828459046]
d0=[2.617366609400000,2.627458131305905,2.637549653211810,2.647641175
117714,2.657732697023619,2.667824218929523,2.677915740835428,2.688007
262741332,2.698098784647236,2.708190306553141,2.718281828459046]
w0=[4.932442377592928,4.968220312397341,5.003998247201754,5.039776182

```

```

006168,5.075554116810579,5.111332051614995,5.147109986419408,5.182887
921223820,5.218665856028234,5.254443790832648,5.290221725637062]
v0=[4.986723357976558,5.017073194742609,5.047423031508658,5.077772868
274710,5.108122705040760,5.138472541806809,5.168822378572860,5.199172
215338908,5.229522052104959,5.259871888871011,5.290221725637062]
a=1
for i=1:11
b1=2*c0*(2-t)
b2=2*d0*(2-t)
end
t0=1.5
n=18
for k=1:1
for i=1:n
if i==1
w(1,i)=w0(1,k)-int(-a*v0(1,k)-b2(1,k),t0,t)
v(1,i)=v0(1,k)-int(-a*w0(1,k)-b1(1,k),t0,t)
else
w(1,i)=w0(1,k)-int(-a*v(1,i-1)-b2(1,k),t0,t)
v(1,i)=v0(1,k)-int(-a*w(1,i-1)-b1(1,k),t0,t)
end
end
h(t)= w(1,i)
h(2)
z(t)=v(1,i)
z(2)
end

```

### Adomian Decomposition Method (ADM)

```

# syms t w v z h s
format long
t0=0
a=1
b=0
y0=zeros(1,11)
q0=zeros(1,11)
w0=zeros(1,11)
v0=zeros(1,11)
for i=1:11
y0(1,i)=0.75+0.25*(i-1)*0.1
q0(1,i)=1.125-0.125*(i-1)*0.1
w0(1,i)=y0(1,i)+int(b,s,t0,t)
v0(1,i)=q0(1,i)+int(b,s,t0,t)
end
n=6
for k=1:11
z(t)=w0(1,k)*t^0
h(t)=v0(1,k)*t^0
for j=1:n
if j==1
w(k,j)=int(a*v0(1,k),s,t0,t)
v(k,j)=int(a*w0(1,k),s,t0,t)
else
w(k,j)=int(a*v(k,j-1),t0,t)
v(k,j)=int(a*w(k,j-1),t0,t)
end
z(t)=z(t)+w(k,j)
h(t)=h(t)+v(k,j)
z(1)

```

```

h(1)
end
end

```

```

 $\text{syms } t \text{ w v z h s}$ 
 $\text{format long}$ 
y0=[2.479411818960710,2.503298819910544,2.527185820860377,2.551072821
810211,2.574959822760044,2.598846823709878,2.622733824659711,2.646620
825609545,2.670507826559379,2.694394827509212,2.718281828459046]
q0=[2.617366609400000,2.627458131305905,2.637549653211810,2.647641175
117714,2.657732697023619,2.667824218929523,2.677915740835428,2.688007
262741332,2.698098784647236,2.708190306553141,2.718281828459046]
t0=1
a=1
for i=1:11
b1(1,i)=2*y0(1,i)*(s-1)
b2(1,i)=2*q0(1,i)*(s-1)
c1(1,i)=int(b1(1,i),t0,s)
c2(1,i)=int(b2(1,i),t0,s)
w0(1,i)=y0(1,i)+c2(1,i)
v0(1,i)=q0(1,i)+c1(1,i)
end
for i=1:11
w0(1,i)=y0(1,i)+c2(1,i)
v0(1,i)=q0(1,i)+c1(1,i)
end
n=6
for k=1:11
z(s)=w0(1,k)*s^0
h(s)=v0(1,k)*s^0
for j=1:n
if j==1
w(k,j)=int(a*v0(1,k),t0,s)
v(k,j)=int(a*w0(1,k),t0,s)
else
w(k,j)=int(a*v(k,j-1),t0,s)
v(k,j)=int(a*w(k,j-1),t0,s)
end
z(s)=z(s)+w(k,j)
h(s)=h(s)+v(k,j)
z(1.5)
h(1.5)
end
end

```

```

 $\text{syms } t \text{ w v z h s}$ 
 $\text{format long}$ 
c0=[2.479411818960710,2.503298819910544,2.527185820860377,2.551072821
810211,2.574959822760044,2.598846823709878,2.622733824659711,2.646620
825609545,2.670507826559379,2.694394827509212,2.718281828459046]
d0=[2.617366609400000,2.627458131305905,2.637549653211810,2.647641175
117714,2.657732697023619,2.667824218929523,2.677915740835428,2.688007
262741332,2.698098784647236,2.708190306553141,2.718281828459046]
y0=[4.932442377592928,4.968220312397341,5.003998247201754,5.039776182
006168,5.075554116810579,5.111332051614995,5.147109986419408,5.182887
921223820,5.218665856028234,5.254443790832648,5.290221725637062]
q0=[4.986723357976558,5.017073194742609,5.047423031508658,5.077772868
274710,5.108122705040760,5.138472541806809,5.168822378572860,5.199172
215338908,5.229522052104959,5.259871888871011,5.290221725637062]

```

```

t0=1.5
a=1
for i=1:11
b1(1,i)=2*c0(1,i)*(2-s)
b2(1,i)=2*d0(1,i)*(2-s)
c1(1,i)=int(b1(1,i),t0,s)
c2(1,i)=int(b2(1,i),t0,s)
w0(1,i)=y0(1,i)+c2(1,i)
v0(1,i)=q0(1,i)+c1(1,i)
end
for i=1:11
w0(1,i)=y0(1,i)+c2(1,i)
v0(1,i)=q0(1,i)+c1(1,i)
end
n=6
for k=1:11
z(s)=w0(1,k)*s^0
h(s)=v0(1,k)*s^0
for j=1:n
if j==1
w(k,j)=int(a*v0(1,k),t0,s)
v(k,j)=int(a*w0(1,k),t0,s)
else
w(k,j)=int(a*v(k,j-1),t0,s)
v(k,j)=int(a*w(k,j-1),t0,s)
end
z(s)=z(s)+w(k,j)
h(s)=h(s)+v(k,j)
z(2)
h(2)
end
end

```