Journal of Zankoy Sulaimani
Part-A- (Pure and Applied Sciences)
http://jzs.univsul.edu.iq

Numerical Treatment Solution of Volterra Integro-Fractional Differential Equation by Using Linear Spline Function

Karwan H.F. Jwamer [1] , Shazad Sh. Ahmed [1] and Diar Kh. Abdullah[1*]

1Department of Mathematics, College of Science, Sulaimani University, Sulaimani, Kurdistan Region, Iraq.
**Corresponding author:diar.khalid85@gmail.com*

Article info

Original: 5 August 2020
Revised: 1 September 2020
Accepted: 26 September 2020
Published online: 20 December 2020

Key Words: Integro-fractional differential equation, Caputo derivative, Linear spline, Extrapolation method, Clenshaw

Abstract

In this article, we propose two new approximate methods based totally on the use of normal linear spline function and second employed with the Richardson Extrapolation technique the usage of discrete collocation points for approximating the solution of the Volterra integro-fractional differential equations (VIFDEs). The fractional derivatives are used in the Caputo sense. Illustrative examples are included to demonstrate the validity and applicability of the technique. A new technique with the resource of MatLab program is written to treat numerically VIFDEs using spline function, as well as, follow the Clenshaw Curtis rule for calculating the required integrals for those equations.

1. Introduction

In this paper we will improve an approximation based on the linear spline to obtain the numerical solution of the following Volterra integro-fractional differential equation (VIFDE's) of the second kind of the form:

$$ {}_a^C D_t^{\alpha_n} u(t) + \sum_{i=1}^{n-1} \mathcal{P}_i(t) \; {}_a^C D_t^{\alpha_{(n-i)}} u(t) + \mathcal{P}_n(t)u(t) = f(t) + \sum_{\ell=0}^{m} \lambda_\ell \int_a^t \mathcal{K}_\ell(t,s) \; {}_a^C D_s^{\beta_{m-\ell}} u(s)ds, \quad t \in [a,b] \qquad (1) $$

Subject to

$$ [u(t)]_{t=a} = u_a \qquad (2) $$

Where

$$ \alpha_n > \alpha_{n-1} > \cdots > \alpha_1 > \alpha_0 = 0 \text{ and} \beta_m > \beta_{m-1} > \cdots > \beta_1 > \beta_0 = 0, \; 0 < \alpha_i \, , \beta_j \leq 1 $$

Connected with N-condition; $N = \max\{n_i, m_i \; for \; all \; i \; and \; j\}$, where $\alpha_i, \beta_j \in \mathbb{R}^+, n_i - 1 < \alpha_i \leq n_i$ and $m_j - 1 < \beta_j \leq m_j$, $n_i = \lceil \alpha_i \rceil$ and $m_j = \lceil \beta_j \rceil$ for all $i = 1,2,\dots,n$ and $j = 1,2,\dots,m, u(t)$ is the unknown function and $\mathcal{K}: S \times \mathbb{R} \to \mathbb{R}$ (with $S = \{(t,s): a \leq s \leq t \leq b\}$) denote a given functions,$f(t)$, $\mathcal{P}_i(t); (i = 0,1,2,\dots,n)$are given continuouse real valued function on I, and λ_ℓ is a scalar parameters.

The integer order derivative and integral concepts are widely known by the mathematicians. The concept fractional is a misnomer but is kept despite the prevalent usage. The fractional calculus may also be considered as an old yet novel topic [1]. The derivative $d^n y/dx^n$ modifies the changes of variable y with respect to variable x, and its physical background is significant. Generalizing n into a fraction, even a complex number is an existing issue now [2].

This problem (to generalize n into a fraction, even a complex number) remained for a long time unit when L'Hôpital sent a letter to Leibniz in 1695, in which it is asked what the derivative $d^n y/dx^n$ is when $n = 1/2$. Furthermore in the same year, Leibniz mentioned the derivative of general order in a letter which he sent to J. Bernoulli. The problem was also thought by Euler (1730), Lagrange (1849) and others, and gave some related information. In 1812, Laplace provided a definition of fractional derivative [3].

Based on history, the theory of fractional calculus, in which derivatives and integrals of fractional has a long history [4]. Fractional calculus (FC) has been a successful field of science and engineering research in recent years [5]. In truth, many scientific areas are currently paying attention to FC concepts and we can refer to their adoption in viscoelasticity and damping, diffusion and propagation of waves, electromagnetism, heat transfer, biology, traffic systems, electronics, signal processing, robotics, genetic algorithms, percolation, modeling and identification, telecommunications, chemistry, irreversibility, physics and control systems as well as economy [6]. Many researchers tried to put a definition of a fractional derivative. Most of them used an integral form for the fractional derivative. Two of which are the most popular ones [7].

1. Riemann–Liouville definition. Suppose that $\alpha > 0,\ t > a$ and $\alpha \in \mathbb{R}^+$ then we have [4]:

$$ {}_a^R D_t^\alpha u(t) = \begin{cases} \dfrac{1}{\Gamma(\mathrm{n}-\alpha)} \dfrac{d^n}{dt^n} \displaystyle\int_a^t \frac{u(s)}{(t-s)^{\alpha+1-n}} ds & n-1 < \alpha < n \in \mathbb{N} \\ \dfrac{d^n}{dt^n} u(t) & \alpha = n \in \mathbb{N} \end{cases} $$

2. Caputo definition. Suppose that $\alpha > 0,\ t > a$ and $\alpha \in \mathbb{R}^+$ then we have [4]:

$$ {}_a^C D_t^\alpha u(t) = \begin{cases} \dfrac{1}{\Gamma(\mathrm{n}-\alpha)} \displaystyle\int_a^t \frac{u^{(n)}(s)}{(t-s)^{\alpha+1-n}} ds & n-1 < \alpha < n \in \mathbb{N} \\ \dfrac{d^n}{dt^n} u(t) & \alpha = n \in \mathbb{N} \end{cases} $$

In Eqn. (1), ${}_a^C D_t^\alpha$ *and* ${}_a^C D_t^\beta$ denotes fractional differential operator where $n-1 < \alpha,\ \beta < n \in \mathbb{N}$ in the sense of Caputo.

Several methods have been introduced in the literature for the numerical approaches to IFDE's have been recently studied by numerous authors [8,9,10,11].

This study is organized as follows: we start by introduction then focus the fractional differential operator Caputo sense. Preliminaries and discussing numerical methods and linear spline function are defined and devoted to applying the integro fractional differential described in section two. Section three the proposed method is applied to two examples. Also a conclusion is given in section five.

2. Prelimiaries

In this section, we will introduce and study the concepts such that we divided into two subsections.

Definition 1. If $0 < \alpha \leq 1,\ t_r \in \mathbb{R}$ and $m \in \mathbb{R}^+$ and for any arbitrary $t_r \geq a = t_0$.Then, for all $a \leq t \leq b$ [12].

$$ {}_a^C D_t^\alpha (t - t_r)^m = \left[\sum_{i=0}^{m-1} \frac{(-1)^i \Gamma(m+1)(t-a)^{m-\alpha}}{i!\ \Gamma(m+1-i-\alpha)} \left(\frac{t_r - a}{t-a} \right)^i \right] $$

Definition 2. Let $\alpha \geq 0, n = \lceil \alpha \rceil$ and $u(t) = (t-a)^p$ for some $p \geq 0$ then [2, 4].

$$ {}_a^C D_t^\alpha u(t) = \begin{cases} 0 & \text{if} \quad p \in \{0,1,2,\dots,n-1\} \\ \dfrac{\Gamma(\mathrm{p}+1)}{\Gamma(p-\alpha+1)} (t-a)^{p-\alpha} & \text{if} \quad p \in \mathbb{N}\ and\ p \geq n\ or\ p \notin \mathbb{N}\ and\ p > n-1 \end{cases} $$

Where $\lceil \alpha \rceil$ denote the smallest integer greater than or equal to α. In the present paper.

Remark 1. Properties of the operator ${}_a^C D_t^\alpha$can be found in [1, 2, 3, 4], we mention the following:

1. Representation

Let $n-1 < \alpha < n,\ n \in \mathbb{N}, \alpha \in \mathbb{R}$ and $u(t)$be such that ${}_a^C D_t^\alpha u(t)$ exist Then ${}_a^C D_t^\alpha u(t) = {}_a J_t^{n-\alpha}\ D^n u(t)$.

This means that the Caputo fractional operator is equivalent to $(n-\alpha)$fold Integration after n^{th} order differentiation. While Riemann-Liouville fractional derivative is equivalent to the composition of same operator but it reverse order.

2. Linearity

Let $n-1 < \alpha < n,\ n \in \mathbb{N}\ and\ c_1, c_2 \in \mathbb{R}$ and functions $u(t)$ and $v(t)$ be such that both ${}_a^C D_t^\alpha u(t)$ and ${}_a^C D_t^\alpha v(t)$ exist. The Caputo fractional derivative is a linear operator.

$$ {}_a^C D_t^\alpha (c_1 u(t) + c_2 v(t)) = c_1 {}_a^C D_t^\alpha u(t) + c_2 {}_a^C D_t^\alpha v(t) $$

3. Non-commutation

Let $n-1 < \alpha < n,\ \ n, m \in \mathbb{N}\ and\ \alpha \in \mathbb{R}$ and functions $u(t)$ is such that ${}_a^C D_t^\alpha u(t)$ exist. Then in general

$$ {}_a^C D_t^\alpha\ D^m\ u(t) = {}_a^C D_t^{\alpha+m} u(t) \neq D^m {}_a^C D_t^\alpha u(t) $$

4. Fractional derivative of constant

Let $n-1 < \alpha < n,\ \ n \in \mathbb{N}\ and\ \alpha, A \in \mathbb{R}$ and functions $u(t) = A$ is constant function such that ${}_a^C D_t^\alpha u(t) = 0$

Definition 3. Quadrature method, the area under the integral function $f(x)$. In general can be approximated as:

$$ \int_a^b f(x)dx \cong \sum_{j=1}^{N} w_j f(x_j) = w_1 f_1 + w_2 f_2 + \cdots + w_N f_N. \qquad (3) $$

Where w_j are constants which are called quadrature weights and $f(x_j)$ are the values of the function at the points x_j in which the domain is divided [13].

Definition 4. Clenshaw and Curtis (1960) defined a procedure for evaluating a definite integral by expanding the integrand in the finite Chebyshev series and adding the terms in the series one by one the technique is very effective specially for integral equations as follows [14].

$$ \int_{-1}^{1} f(x)dx = \sum_{\substack{r=0 \\ r\ even}}^{N\ ''} \frac{2}{N} \sum_{k=0}^{N\ ''} \cos\left(\frac{rk\pi}{N}\right) f\left(\cos\left(\frac{k\pi}{N}\right)\right), \qquad k = 0,1,\dots,N. \qquad (4) $$

Remark 2.

i. The notation$\sum''$ means the first and last terms are to be halved before summing.

ii. The transformation $x = \frac{a+b}{2} + \frac{b-a}{2}y$, converting interval $[a, b]$ into $[-1,1]$.

Definition 5. Richardson's extrapolation is used to create high-accuracy result using low-order formulas.

$$T_i^{(m)} = T_{i+1}^{(m-1)} + \frac{T_{i+1}^{(m-1)} - T_i^{(m-1)}}{\left(\frac{h_i}{h_{i+m}}\right) - 1}, \qquad i = 1,2,\dots,m \quad \text{and } m = 1,2,\dots,k-1 \tag{5}$$

Where $T_i^{(m)}$ is an approximate value. Richardson extrapolation using step sizes of h_i, h_{i+1} ,…,h_{i+m} ,and $0 < m \leq k-1$ [15].

Remark 3.

The sequence $\{h_i\}$ usually is of the form $\left\{h_0, \frac{h_0}{2}, \frac{h_0}{4}, \frac{h_0}{8}, \frac{h_0}{16}, \dots\right\}$,$\left\{h_0, \frac{h_0}{2}, \frac{h_0}{3}, \frac{h_0}{4}, \frac{h_0}{5}, \dots\right\}$ and $\left\{h_0, \frac{h_0}{2}, \frac{h_0}{3}, \frac{h_0}{4}, \frac{h_0}{6}, \frac{h_0}{8}, \frac{h_0}{12} \dots\right\}$. In this paper we will consider the sequence $\{h_i\}$ as the form $\left\{h_0, \frac{h_0}{2}, \frac{h_0}{4}, \frac{h_0}{8}, \frac{h_0}{16}, \dots\right\}$.

2.1 Linear Classic Spline $L(t)$ [16]

A piecewise linear function for the spline of degree one can be written as:

$$L(t) = \begin{cases} L_0(t) & t \in [t_0, t_1] \\ L_1(t) & t \in [t_1, t_2] \\ \vdots & \vdots \\ L_{N-1}(t) & t \in [t_{N-1}, t_N] \end{cases} \qquad \text{Where } L_r(t) = a_r\, t + b_r,\ r = 0,1,\dots,N-1.$$

$L(t)$ satisfy the following conditions:

1. The domain of L is an interval $[a, b]$.
2. L is continuous on $[a, b]$.
3. There is a partitioning of the interval $a = t_0 < t_1 < \cdots < t_N = b$ such that L is a linear polynomial on each subinterval $[t_r, t_{r+1}]$ after some manipulation we obtain

$$L(t) = \left(\frac{t_{r+1}-t}{h}\right) L_r + \left(\frac{t-t_r}{h}\right) L_{r+1}, \quad \text{where } h = t_{r+1} - t_r,\ \forall\, r = 0,1,\dots,N-1. \tag{6}$$

2.2 Fractional Derivative of Spline Functions

In this section we discuss the way of obtaining fractional derivative for all the linear spline. So that, here we apply Caputo properties to accomplish.

Lemma 1. The fractional derivative of linear classic spline of order α with respect to t as:

$$ {}_a^C D_t^\alpha L(t) = \frac{(t-a)^{1-\alpha}}{h\,\Gamma(2-\alpha)}[L_{r+1} - L_r] \qquad where \ \ 0 < \alpha \leq 1.$$

Proof: Linear spline function $L(t)$ in the interval $[t_r, t_{r+1}]$ give the formula in Eqn.(6)

$$L(t) = A_r(t)L_r + B_r(t)L_{r+1} \tag{7}$$

Where:

$$A_r(t) = \frac{t_{r+1}-t}{h},\ B_r(t) = \frac{t-t_r}{h} \qquad \forall\, r = 0,1,\dots,N-1.$$

So, we have

$$ {}_a^C D_t^\alpha L(t) = {}_a^C D_t^\alpha [A_r(t)L_r + B_r(t)L_{r+1}] = L_r\, {}_a^C D_t^\alpha \left(\frac{t_{r+1}-t}{h}\right) + L_{r+1}\, {}_a^C D_t^\alpha \left(\frac{t-t_r}{h}\right)$$

Using the definition of the Caputo fractional derivative in the form (1-3), we have

$${}_a^C D_t^{\alpha} L(t) = L_r\left(\frac{-(t-a)^{1-\alpha}}{h\,\Gamma(2-\alpha)}\right) + L_{r+1}\left(\frac{(t-a)^{1-\alpha}}{h\,\Gamma(2-\alpha)}\right).$$

Through calculation, you can get

$${}_a^C D_t^{\alpha} L(t) = \frac{(t-a)^{1-\alpha}}{h\,\Gamma(2-\alpha)}[L_{r+1} - L_r]. \tag{8}$$

3. Description of the Methods

To find numerical solution of Eqn.(1), using linear spline function we can use the following two cases. The first one using the normal and the second using extrapolation method. Now we can drive each cases.

3.1 Normal linear spline function (NLS)

To develop the linear spline approximation method for solving Volterra integro- fractional differential equation from Eqns. (1) and (2) on the interval $[a,b]$. First divided the interval $[a,b]$ is into N-equal subintervals of length of $h = \frac{b-a}{N}$ with endpoints, the linear spline $L(t)$ interpolating the function $u(t)$ at the grid points is given by the equation:

$${}_a^C D_t^{\alpha_n} L(t) + \sum_{i=1}^{n-1} \mathcal{P}_i(t)\ {}_a^C D_t^{\alpha_{(n-i)}} L(t) + \mathcal{P}_n(t)L(t) = f(t) + \sum_{\ell=0}^{m} \lambda_\ell \int_a^t \mathcal{K}_\ell(t,s)\ {}_a^C D_s^{\beta_{m-\ell}} L(s)ds \tag{9}$$

Substituting $t = t_{r+1}$, when into Eqn.(9) then collocate Eqn. (9) at the uniform grid points after substituting Eqn. (8) into Eqn.(9), we obtain

$$\frac{((r+1)h)^{1-\alpha_n}}{h\,\Gamma(2-\alpha_n)}[L_{r+1} - L_r] + \sum_{i=1}^{n-1} \mathcal{P}_{ir+1}\frac{((r+1)h)^{1-\alpha_{(n-i)}}}{h\,\Gamma\big(2-\alpha_{(n-i)}\big)}[L_{r+1} - L_r] + \mathcal{P}_{nr+1}L_{r+1} = f_{r+1} + \sum_{\ell=0}^{m-1} \lambda_\ell \Bigg\{$$

$$\int_{t_j}^{t_{j+1}} k_\ell(t_{r+1},s)\ {}_a^C D_s^{\beta_{m-\ell}}\big(A_j(s)L_j + B_j(s)L_{j+1}\big)ds + \int_{t_r}^{t_{r+1}} \mathcal{K}_\ell(t_{r+1},s)\ {}_a^C D_s^{\beta_{m-\ell}}(A_r(s)L_r + B_r(s)L_{r+1})ds\Bigg\}$$

$$+\lambda_m\left[\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}} \mathcal{K}_m(t_{r+1},s)\,(A_j(s)L_j + B_j(s)L_{j+1})ds + \int_{t_r}^{t_{r+1}} \mathcal{K}_m(t_{r+1},s)\ (A_r(s)L_r + B_r(s)L_{r+1})ds\right]. \tag{10}$$

Let

$$\mathcal{W}_n^r(\mathrm{s}) = \mathcal{P}_{s(r+1)}\frac{((r+1)h)^{1-\alpha_{(n-s)}}}{h\,\Gamma\big(2-\alpha_{(n-s)}\big)} \text{ and } \mathcal{H}_n^r(\mathrm{s}) = \mathcal{P}_{n(r+1)} + \sum_{s=0}^{n-1} w_n^r(\mathrm{s}),\ \mathrm{s} = 0,1,\dots,n-1,\ r = 1,2,..,N-1. \tag{11}$$

The Eqn. (10) becomes

$$L_{r+1}\left[\mathcal{H}_n^r - \sum_{\ell=0}^{m-1}\lambda_\ell\int_{t_r}^{t_{r+1}} \mathcal{K}_\ell(t_{r+1},s)\ {}_a^C D_s^{\beta_{m-\ell}} B_r(s)ds - \lambda_m\int_{t_r}^{t_{r+1}} \mathcal{K}_m(t_{r+1},s)\,B_r(s)ds\right] = L_r\left[\sum_{s=0}^{n-1}\mathcal{W}_n^r(\mathrm{s}) + \sum_{\ell=0}^{m-1}\lambda_\ell\right.$$

$$\int_{t_r}^{t_{r+1}} \mathcal{K}_\ell(t_{r+1},s)\ {}_a^C D_s^{\beta_{m-\ell}} A_r(s)ds\ + \lambda_m\int_{t_r}^{t_{r+1}} \mathcal{K}_m(t_{r+1},s)\,A_r(s)ds] + f_{r+1} + \sum_{\ell=0}^{m-1}\lambda_\ell\left[\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}} \mathcal{K}_\ell(t_{r+1},s){}_a^C D_s^{\beta_{m-\ell}}\right.$$

$$\big(A_j(s)L_j + B_j(s)L_{j+1}\big)ds\ \Big] + \lambda_m\ \left[\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}} \mathcal{K}_m(t_{r+1},s)\ (A_j(s)L_j + B_j(s)L_{j+1})ds\right]. \tag{12}$$

Eqn. (12) is obtained by using the initial condition Eqn. (2) together to find the approximate solution L_{r+1}, $r = 0,1,..,N-1$.

3.2 Linear spline function using Extrapolation (ELS)

To Designing the linear spline approximation method for solving Volterra integro-fractional differential equation Eqns. (1) and (2) using extrapolation, the interval $[a,b]$. First divided the interval $[a,b]$ be into N-equal subintervals of length of $h=\frac{b-a}{N}$ with endpoints, the linear spline $L(t)$ interpolating the function with Extrapolation in the interval $t\in[t_r+ih_{M^*},t_r+(i+1)h_{M^*}]$ is given by the formula

$$L^r_{M^*,i}(t)=A^r_{M^*,i}(t)L^r_{M^*,i}+B^r_{M^*,i}(t)L^r_{M^*,i+1} \tag{13}$$

$$r=0,1,..,N-1,\qquad i=0,1,\dots,2^{M^*}-1,\quad M^*=0,1,\dots,M,\qquad h_{M^*}=\frac{h}{2^{M^*}}$$

Where

$$A^r_{M^*,i}(t)=\frac{t_r+(i+1)h_{M^*}-t}{h_{M^*}},\ B^r_{M^*,i}=\frac{t-t_r+ih_{M^*}}{h_{M^*}}$$

Thus

$${}^C_aD^\alpha_t L^r_{M^*,i}(t)|_{t=t_r+(i+1)h_{M^*}}=\frac{(t-a)^{1-\alpha}}{h_{M^*}\,\Gamma(2-\alpha)}[L^{r+1}_{M^*,i}-L^r_{M^*,i}]|_{t=t_r+(i+1)h_{M^*}}$$

$$=\frac{(t_r+(i+1)h_{M^*}-a)^{1-\alpha}}{h_{M^*}\,\Gamma(2-\alpha)}\left[L^{r+1}_{M^*,i}-L^r_{M^*,i}\right].$$

Since $t_r=a+rh\quad\rightarrow\quad t_r-a=rh$, we obtain

$${}^C_aD^\alpha_t L^r_{M^*,i}(t)|_{t=t_r+(i+1)h_{M^*}}=\frac{[rh+(i+1)h_{M^*}]^{1-\alpha}}{h_{M^*}\,\Gamma(2-\alpha)}\left[L^{r+1}_{M^*,i}-L^r_{M^*,i}\right]. \tag{14}$$

From Eqn.(1) putting $t=t_r+(i+1)h_{M^*}$, then collocate Eqn.(9) at the uniform grid points after substituting Eqn. (14) into Eqn. (9), we obtain

$$\frac{[rh+(i+1)h_{M^*}]^{1-\alpha_n}}{h_{M^*}\,\Gamma(2-\alpha_n)}\left[L^{r+1}_{M^*,i}-L^r_{M^*,i}\right]+\sum_{ii=1}^{n-1}\mathcal{P}_{ii}(t_r+(i+1)h_{M^*})\frac{[rh+(i+1)h_{M^*}]^{1-\alpha_{n-ii}}}{h_{M^*}\,\Gamma(2-\alpha_{n-ii})}\left[L^{r+1}_{M^*,i}-L^r_{M^*,i}\right]$$

$$+\mathcal{P}_n(t_r+(i+1)h_{M^*})L^{r+1}_{M^*,i}=f_{(t_r+(i+1)h_{M^*})}+\sum_{\ell=0}^{m-1}\lambda_\ell\left[\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}}\mathcal{K}_\ell(t_r+(i+1)h_{M^*},s)[{}^C_aD^{\beta_{m-\ell}}_s A_j(s)\,L^c_j+\right.$$

$${}^C_aD^{\beta_{m-\ell}}_s B_j(s)L^c_{j+1}]ds+\sum_{j=0}^{i-1}\int_{t_r+jh_{M^*}}^{t_r+(j+1)h_{M^*}}\mathcal{K}_\ell(t_r+(i+1)h_{M^*},s)[{}^C_aD^{\beta_{m-\ell}}_s A^r_{M^*,j}(s)L^r_{M^*,j}$$

$$+{}^C_aD^{\beta_{m-\ell}}_s B^r_{M^*,j}(s)L^r_{M^*,j+1}]ds$$

$$\left.+\int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}}\mathcal{K}_\ell(t_r+(i+1)h_{M^*},s)\left[{}^C_aD^{\beta_{m-\ell}}_s A^r_{M^*,i}(s)L^r_{M^*,i}+{}^C_aD^{\beta_{m-\ell}}_s B^r_{M^*,i}(s)L^{r+1}_{M^*,i}\right]ds\right]\quad+\quad\lambda_m\left[\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}}\right.$$

$$\mathcal{K}_m(t_r+(i+1)h_{M^*},s)[\,A_j(s)\,L^c_j+B_j(s)L^c_{j+1}]ds\;+\sum_{j=0}^{i-1}\int_{t_r+jh_{M^*}}^{t_r+(i+1)h_{M^*}}\mathcal{K}_m(t_r+(i+1)h_{M^*},s)\;[A^r_{M^*,j}(s)L^r_{M^*,j}+$$

$$\left.B^r_{M^*,j}(s)L^r_{M^*,j+1}\,]ds+\int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}}\mathcal{K}_m(t_r+(i+1)h_{M^*},s)[A^r_{M^*,i}(s)L^r_{M^*,i}+B^r_{M^*,i}(s)L^{r+1}_{M^*,i}]ds\;\right]. \tag{15}$$

Let

$$\mathcal{W}^{n,r}_{M^*,i}(s) = \mathcal{P}_n(t_r + (i+1)h_{M^*})\frac{[rh + (i+1)h_{M^*}]^{1-\alpha_{n-s}}}{h_{M^*}\,\Gamma(2-\alpha_{n-s})}, \mathcal{H}^{n,r}_{M^*,i}(s) = \mathcal{P}_n(t_r + (i+1)h_{M^*}) + \sum_{s=0}^{n-1}\mathcal{W}^{n,r}_{M^*,i}(s) \quad (16)$$

$\forall\, r = 0,1,\dots,N-1,\ \forall\, s = 0,1,\dots,n-1,\ \forall\, i = 0,1,\dots,2^{M^*}-1,\ M^* = 0,1,\dots,M,\ h_{M^*} = \frac{h}{2^{M^*}},\ \forall\, M \in \mathbb{Z}^+ \cup \{0\}$, then Eqn.(15) becomes

$$L^{r+1}_{M^*,i}\left\{\mathcal{H}^{n,r}_{M^*,i}(s) - \sum_{\ell=0}^{m-1}\lambda_\ell \int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}} \mathcal{K}_\ell(t_r+(i+1)h_{M^*},s)\,{}_a^cD_s^{\beta_{m-\ell}}B^r_{M^*,i}(s)ds - \lambda_m \int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}} \mathcal{K}_m(t_r+(i+1)h_{M^*},s)B^r_{M^*,i}(s)ds\right\} = L^r_{M^*,i}\left\{\sum_{s=0}^{n-1}W^{n,r}_{M^*,i}(s) + \sum_{\ell=0}^{m-1}\lambda_\ell \int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}} \mathcal{K}_\ell(t_r+(i+1)h_{M^*},s)\ {}_a^cD_s^{\beta_{m-\ell}}A^r_{M^*,i}(s)ds + \lambda_m \int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}} \mathcal{K}_m(t_r+(i+1)h_{M^*},s)A^r_{M^*,i}(s)ds\right\} + f_{(t_r+(i+1)h_{M^*})} + \sum_{\ell=0}^{m-1}\lambda_\ell\left\{\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}} \mathcal{K}_\ell(t_r+(i+1)h_{M^*},s)\left[{}_a^cD_s^{\beta_{m-\ell}}A_j(s)\,L_j^c + {}_a^cD_s^{\beta_{m-\ell}}B_j(s)L_{j+1}^c\right]ds + \sum_{j=0}^{i-1}\int_{t_r+jh_{M^*}}^{t_r+(j+1)h_{M^*}} \mathcal{K}_\ell(t_r+(i+1)h_{M^*},s)\left[{}_a^cD_s^{\beta_{m-\ell}}A^r_{M^*,j}(s)L^r_{M^*,j} + {}_a^cD_s^{\beta_{m-\ell}}B^r_{M^*,j}(s)L^r_{M^*,j+1}\right]ds\right\} + \lambda_m\left\{\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}} \mathcal{K}_m(t_r+(i+1)h_{M^*},s)\left[A_j(s)\,L_j^c + B_j(s)L_{j+1}^c\right]ds + \sum_{j=0}^{i-1}\int_{t_r+jh_{M^*}}^{t_r+(j+1)h_{M^*}} \mathcal{K}_m(t_r+(i+1)h_{M^*},s)\,[A^r_{M^*,j}(s)L^r_{M^*,j} + B^r_{M^*,j}(s)L^r_{M^*,j+1}]ds\right\}. \quad (17)$$

Eqn. (17) is obtained by using the initial condition Eqn. (2) together to find the approximate solution $L^{r+1}_{M^*,i}$, at end of each steps putting $L^{r+1}_{M^*,i} = L^r_{M^*,i+1}$ *and* $L^c_r = L^r_{M^*,0}$, $\forall\, r = 0,1,\dots,N-1,\ \forall\, i = 0,1,\dots,2^{M^*}-1,$ $M^* = 0,1,\dots,M,\ h_{M^*} = \frac{h}{2^{M^*}},\ \forall\, M \in \mathbb{Z}^+ \cup \{0\}$.

4. Numerical Scheme

In this section, to show that valid a proposed method is in this paper, as well as its applicability. We solve two examples, so we have compared the results of this method for normal linear spline and using Richardson Extrapolation method. Moreover, we use the least-square error to show the accuracy of approximation as:

Example 1. Consider the linear VIFDE on $0 \le t \le 1$:

$${}_0^cD_t^{0.7}u(t) + t\,{}_0^cD_t^{0.2}u(t) - 2u(t) = f(t) + \int_0^t\left[(t-2s^2)\,{}_0^cD_s^{0.3}u(s) + (t-s)\,{}_0^cD_s^{0.1}u(s) - (ts-1)u(s)\right]ds,$$

where

$$f(t) = -\frac{2}{3}t^4 + \frac{1}{2}t^3 + t^2 - t - \frac{2}{\Gamma(1.3)}t^{0.3} - \frac{2}{\Gamma(1.8)}t^{1.8} - 2(1-2t) + \frac{20}{629\,\Gamma(1.7)}t^{2.7}(37-34t).$$

With the initial condition: $u(0) = 1,$

Where the exact solution is given by $u(t) = 1 - 2t$.

Example 2. Consider the linear VIFDE on $0 \le t \le 1$:

$$ {}_0^C D_t^{2\alpha} u(t) - \frac{1}{2} {}_0^C D_t^{\alpha} u(t) + (1+t^2)u(t) = f(t) + \int_0^t \left[t s\, {}_0^C D_s^{2\alpha} u(s) + (t^2 - s) {}_0^C D_s^{\alpha} u(s) + e^{t+s} u(s) \right] ds, $$

where

$$ f(t) = t^5 + t^3 - t^2 - 1 - 7e^t - e^{2t}(t^3 - 3t^2 + 6t - 7) + \frac{6}{\Gamma(4-2\alpha)} \left(1 - \frac{1}{5-2\alpha} t^3 \right) t^{3-2\alpha} $$
$$ + \frac{3}{\Gamma(4-\alpha)} \left(\frac{2}{5-\alpha} t^2 - 1 \right) t^{3-\alpha} - \frac{6}{\Gamma(5-\alpha)} t^{6-\alpha} \ . $$

If $0 < \alpha < 0.5$ *let* $\alpha = 0.4$;

with the initial condition: $u(0) = -1$.

The exact solution of this problem is known $u(t) = t^3 - 1$.

Table 1. Contain all values of $\mathcal{W}_n^r(s)$ and $\mathcal{H}_n^r(s)$ from Eqn. (11) or each $t_r = 0.1(0.1)1$ for $r = 0,1,2,\dots,9$. In example 1.

r	0	1	2	3	4
$\boldsymbol{t_r}$	0.1	0.2	0.3	0.4	0.5
$\mathcal{W}_n^r(r)$	5.7546066338	7.4678040319	8.9939363353	10.527810530	12.13377254
$\mathcal{H}_n^r(r)$	3.7546066338	5.4678040319	6.9939363353	8.5278105304	10.13377254
r	5	6	7	8	9
$\boldsymbol{t_r}$	0.6	0.7	0.8	0.9	1.0
$\mathcal{W}_n^r(r)$	13.840261936	15.661721785	17.606043930	19.677681788	21.879137825
$\mathcal{H}_n^r(r)$	11.840261936	13.661721785	15.606043930	17.677681788	19.879137825

Table II. Shows all values of $\mathcal{W}_{M^*,i}^{n,r}(s)$and $\mathcal{H}_{M^*,i}^{n,r}(s)$ from Eqn. (16) for each $t_r = 0.1(0.1)1$ for $r = 0,1,2,\dots,9$. For example 1.

r	i	M^*	$N=10$			i	M^*	$N=10$	
			$\mathcal{W}_{M^*,i}^{n,r}$	$\mathcal{H}_{M^*,i}^{n,r}$				$\mathcal{W}_{M^*,i}^{n,r}$	$\mathcal{H}_{M^*,i}^{n,r}$
0	0	1	5.7546066338	3.7546066338	5	0	1	13.840261936	11.840261936
	1	2	9.16968587567	7.1696858756		1	2	25.946680275	23.9466802757
			11.5092132676	9.5092132676				27.680523872	25.6805238721
	2	4	14.7935623506	12.793562350		2	4	50.200994272	48.2009942721
			18.3393717513	16.339371751				51.893360551	49.8933605514
			20.8963086324	18.896308632				53.613114171	51.6131141716
			23.0184265352	21.018426535				55.361047744	53.3610477442
1	0	1	7.46780403193	5.4678040319	6	0	1	15.661721785	13.6617217850
	1	2	13.3196421198	11.319642119		1	2	29.472043468	27.4720434686
			14.9356080638	12.935608063				31.323443570	29.3234435701
	2	4	24.9014243654	22.901424365		2	4	57.137842761	55.1378427612
			26.6392842397	24.639284239				58.944086937	56.9440869373
			28.2849164030	26.284916403				60.780288362	58.7802883623
			29.8712161277	27.871216127				62.646887140	60.6468871403
2	0	1	8.99393633536	6.9939363353	7	0	1	17.606043930	15.6060439300
	1	2	16.4734152200	14.473415220		1	2	33.236376739	31.2363767394
			17.9878726707	15.987872670				35.212087860	33.2120878601
	2	4	31.4200801670	29.420080167		2	4	64.544265029	62.5442650291
			32.9468304400	30.946830440				66.472753478	64.4727534789
			34.4626046258	32.462604625				68.432640373	66.4326403737
			35.9757453414	33.975745341				70.424175720	68.4241757202

3	0	1	10.5278105304	8.5278105304	8	0	1	19.677681788	17.6776817882
	1	2	19.5091716475	17.509171647		1	2	37.251515317	35.2515153176
			21.0556210608	19.055621060				39.355363576	37.3553635764
	2	4	37.4926545036	35.492654503		2	4	72.447576470	70.4475764708
			39.0183432951	37.018343295				74.503030635	72.5030306352
			40.5567997242	38.556799724				76.590700799	74.5907007995
			42.1112421217	40.111242121				78.710727152	76.7107271529
4	0	1	12.1337725426	10.133772542	9	0	1	21.879137825	19.8791378257
	1	2	22.6390688726	20.639068872		1	2	41.524156262	39.5241562624
			24.2675450852	22.267545085				43.758275651	41.7582756515
	2	4	43.6842987069	41.684298706		2	4	80.863230100	78.8632301000
			45.2781377452	43.278137745				83.048312524	81.0483125248
			46.8945637994	44.894563799				85.266061759	83.2660617597
			48.5350901704	46.535090170				87.516551303	85.5165513031

Table III. Exact and numerical solution for example 1.

t	Exact	$N = 10$		
		NLS	ELS ($M = 2$)	ELS ($M = 4$)
0	1	1.0	1.0	1.0
0.1	0.8	0.800000026039	0.800000000073	0.800000000017
0.2	0.6	0.600000072374	0.6000000263600	0.600000025187
0.3	0.4	0.400000138014	0.4000000684211	0.400000065894
0.4	0.2	0.200000221852	0.2000001253451	0.200000121241
0.5	0.0	$3.222879e-007$	$1.960629e-007$	$1.9020049e-007$
0.6	-0.2	-0.19999956286	-0.19999972085	-0.19999972859
0.7	-0.4	-0.39999943638	-0.39999962723	-0.39999963691
0.8	-0.6	-0.59999930151	-0.59999952531	-0.59999953692
0.9	-0.8	-0.79999916185	-0.79999941769	-0.79999943112
1.0	-1	-0.99999902112	-0.99999930714	-0.99999932224
L.S.E		$2.8353638e-12$	$1.3208569e-12$	$1.2587463e-12$
R.Time/Sec		7.6504269	48.654911	380.45605

Table IV. Demonstrate the least-square errors and running times (elapsed time) for (**NLS** and **ELS**) with different values of steps size h. For Example 1.

h / Linear	0.1 ($N = 10$)		0.05 ($N = 20$)		0.033($N = 30$)	
	$L.S.E$	$R.Time$ /Sec	$L.S.E$	$R.Time$ /Sec	$L.S.E$	$R.Time$ /Sec
NLS	2.8353638 $e-12$	7.6504269	1.954149 $e-13$	24.71214	4.429182 $e-14$	51.722243
ELS ($M = 2$)	1.3208569 $e-12$	48.654911	1.390875 $e-13$	193.3859	3.5734055 $e-14$	386.18101
ELS ($M = 3$)	1.2791147 $e-12$	144.98434	1.3647152 $e-13$	437.03622	3.5255151 $e-14$	865.38053

Table V. Exact and numerical solution for example 2.

t	Exact	$N = 10$		
		NLS	ELS ($M = 2$)	ELS ($M = 4$)
0	-1.0	-1.0	-1.0	-1.0
0.1	-0.999	-0.99773091018803	-0.99898831541826	-0.999127896083035
0.2	-0.992	-0.98804739246910	-0.9924065253319	-0.99296051031453
0.3	-0.973	-0.965315740672717	-0.97456829113692	-0.975799391105796
0.4	-0.936	-0.92342991445436	-0.93938756126012	-0.941552287234379
0.5	-0.875	-0.855651126264768	-0.8803497409019	-0.883721543858047
0.6	-0.784	-0.75422609158972	-0.790317065020712	-0.79523255916152
0.7	-0.657	-0.60964402014635	-0.661100361716347	-0.668041293875645
0.8	$-0.488,$	-0.409209240765394	-0.4826446682049	-0.492384157159846
0.9	-0.271	-0.13416121909580	-0.24148436041945	-0.255363993533936
1.0	0	0.246588369298637	0.082340345914557	0.061842311856572
L.S.E		0.089476402	0.0077792236	0.0044519264
R.Time/Sec		7.8328779	57.136244	294.20931

Table VI. Comparison for different value of N. For Example 2.

h / Linear	0.05 ($N = 20$)		0.02 ($N = 50$)		0.01($N = 100$)	
	$L.S.E$	$R.Time$ $/Sec$	$L.S.E$	$R.Time$ $/Sec$	$L.S.E$	$R.Time$ $/Sec$
NLS	2.6453699 $e-2$	25.913053	8.562261 $e-3$	137.6159	8.562260 $e-3$	275.00894
ELS ($M = 2$)	4.9332381 $e-3$	193.70612	3.81623 $e-3$	1395.0517	3.5324915 $e-3$	3088.8683
ELS ($M = 3$)	4.0688479 $e-3$	368.6222	3.5446504 $e-3$	1770.8669	3.4081152 $e-3$	6036.8947

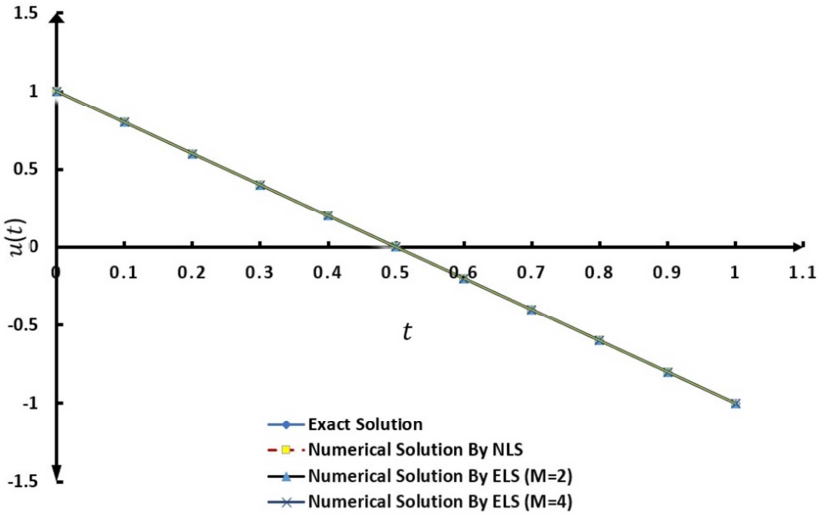

FIGURE 1. Compared numerical and the exact solution of the example 1, $h = 0.1$.

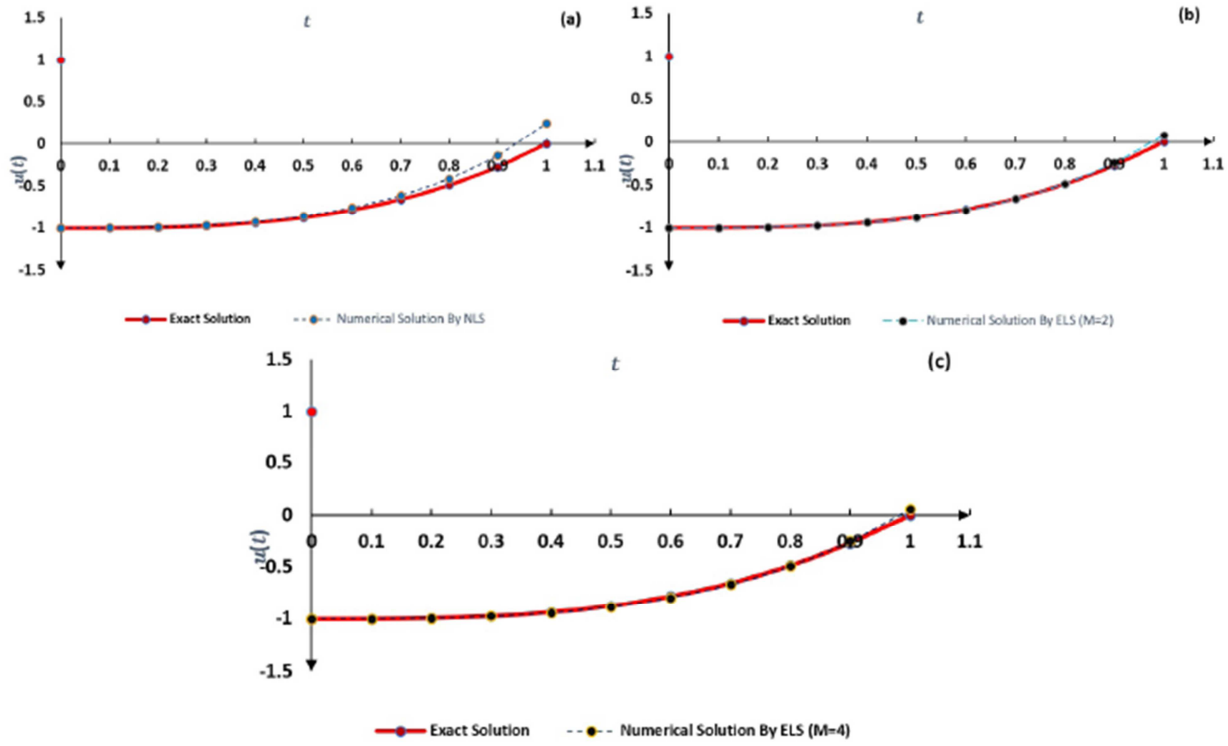

Figure 2. Three types numerical solution compared with the exact solutions of example 2; $h = 0.1$.

5. Conclusion

In this paper, the linear spline approximate solution is compared with other method (Extrapolation Method) for solving Volterra Fractional integro-differential equations. Examples are treated and good results were obtained, the comparison is made between two cases depending on least-square error which is calculated from the numerical against the exact solution. Tables (III, V) and figures (1, 2) shown comparison between normal linear spline and using Extrapolation method with different step sizes. The accuracy of the results depends on the Extrapolation method used as well as the number of M, tables (IV, VI) as shown.

In addition, interpolating linear spline for linear function is more accurate to the cubic function as shown in Table (III, V).

References

[1] J. Sabatier, O. P. Agrawal, J. A. Tenreiro Machado. "*Advances in Fractional Calculus*". Netherlands:Springer. (2007).

[2] B. Guo, X. Pu, and F. Huang. "*Fractional Partial Differential Equations and Their Numerical Solutions*". World Scientific Publishing Co. Pte, Ltd. (2015).

[3] R. Almeida. "*A Caputo fractional derivative of a function with respect to another function*". Communications in Nonlinear Science and Numerical Simulation. Vol. 44, pp. 460-481. (2017). DOI.org/10.1016/j.cnsns.2016.09.006.

[4] I. Podlubny. "*Fractional Differential Equations*". Academic Press, USA. (1999).

[5] K. S. Miller. "*An Introduction to Fractional Calculus and Fractional Differential Equations*". J. Wiley and Sons, New York. (1993).

[6] R. Khalil, M. Al Horani, A. Yousef, and M. Sababheh. *"A new definition of fractional derivative*". Journal of Computational and Applied Mathematics. Vol. 264, pp. 65-70. (2014).

[7] G. -H.Gaoa, Z. -Zh. Sun, and H. -W. Zhang. "*A new fractional numerical differentiation formula to approximate The Caputo fractional derivative and its applications*". Journal of Computational Physics. Vol. 259, pp. 33-50. (2014). DOI.org/10.1016 /j.jcp.2013.11.017.

[8] H. Mesgarani, H. Safdari, A. Ghasemian, and Y. Esmaeelzade. "*The Cubic B-spline Operational Matrix Based on Haar Scaling Functions for Solving Varieties of the Fractional Integro-differential Equations"*. Journal of Mathematics. Vol. 51, No. 8, pp. 45-65. (2019).

[9] Sh. Sh. Ahmed, and Sh. A. Hama Salh. "*Generalized taylor matrix method for solving linear integro-fractional Differential equations of volterra type"*. Applied Mathematical Sciences. Vol. 5, pp. 1765-1780. (2011).

[10] K. Maleknejad, M. N. Sahlan, and A. Ostadi. "*Numerical solution of fractional integro-differential equation by using cubic B-spline wavelets"*. Proceedings of the World Congress on Engineering. Vol. 1. (2013).

[11] Sh. Sh. Ahmed, Sh.A.Hama Salh, and M.R.Ahmad. "*Laplace adomian and laplace modified adomian decomposition methods for solving nonlinear integro-fractional differential equations of the volterra-hammerstein type"*. Iraqi Journal of Science. Vol. 60, pp. 2207-2222. (2019).

[12] Sh. Sh. Ahmed. "*On system of linear volterra integro-fractional differential equations"*. Ph.D.dissertation, University of Sulaimani, (2009).

[13] F. Tornabene, and N. Fantuzzi. "*Mechanics of laminated composite doubly-curved shell structures*". Bologna, Societa editrice Esculapio. (2014).

[14] R. B. Dash, and D. Das. "*A Mixed Quadrature Rule by Blending Clenshaw-Curtis and Gauss-Legendre Quadrature Rules for Approximation of Real Definite Integrals in Adaptive*". (2011).

[15] I. Faragó, Á. Havasi, and Z. Zlatev. "*Efficient implementation of stable Richardson Extrapolation algorithms*". Computers and Mathematics with Applications. Vol. 2010, pp. 2309-2325. (2010). DOI.org/10.1016/j.camwa.2010.08.025.60.

[16] M. Zamani. "*Three Simple Spline Method for Approximation and Interpolation of Data*". Contemporary Engineering Sciences. Vol. 2, No. 8, pp. 373-381. (2009).