DOI 10.31489/2021M1/50-64

UDC 519.642

K.H.F. Jwamer*, Sh.Sh. Ahmed, D.Kh. Abdullah

University of Sulaimani, Sulaimani, Kurdistan Region, Iraq
(E-mail: karwan.jwamer@univsul.edu.iq, Shazad.ahmed@univsul.edu.iq, Diar.khalid85@gmail.com)

Approximate Solution of Volterra Integro-Fractional Differential Equations Using Quadratic Spline Function

In this paper, we suggest two new methods for approximating the solution to the Volterra integro-fractional differential equation (VIFDEs), based on the normal quadratic spline function and the second method used the Richardson Extrapolation technique the usage of discrete collocation points. The fractional derivatives are regarded in the Caputo perception. A new theorem for the Richardson Extrapolation points for using the finite difference approximation of Caputo derivative is introduced with their proof. New techniques using the first derivative at the initial point such that obtained by follow two cases the first using trapezoidal rule and the second using the first step of linear spline function using the Richardson Extrapolation method. Specifically, the program is given in examples analysis in Matlab (R2018b). Numerical examples are available to illuminate the productivity and trustworthiness of the methods, as well as, follow the Clenshaw Curtis rule for calculating the required integrals for those equations.

Keywords: Integro-fractional differential equation, Caputo derivative, Quadratic spline, Extrapolation method, Clenshaw.

1 Introduction

In this research we will improve a proximity based on the quadratic spline to attain the numerical solution of the following Volterra integro- fractional differential equation (VIFDE's) of the second kind of the form:

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

Subject to

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

where $\alpha_n > \alpha_{n-1} > \cdots > \alpha_1 > \alpha_0 = 0$ and $\beta_m > \beta_{m-1} > \cdots > \beta_1 > \beta_0 = 0$

$$ 0 < \alpha_i,\ \beta_j \leq 1. $$

Connected with $N-$condition; $N = \max\{n_i, m_i \text{ for all } i \text{ and } j\}$, 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 $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$, $u(t)$ 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, $f(t)$, $\mathcal{P}_i(t)$; $(i = 0, 1, 2, \ldots, n)$ are given continuous real valued function on I, and λ_l is a scalar parameter.

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

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

*Corresponding author.
E-mail: karwan.jwamer@univsul.edu.iq

Properties of the operator ${}^C_aD^\alpha_t$ can be found in [1–5], we mention the following

i. $${}^C_aD^\alpha_t(t-a)^p = \begin{cases} 0 & \text{if } p \in \{0,1,2,\dots,n-1\} \\ \dfrac{\Gamma(p+1)}{\Gamma(p-\alpha+1)}(t-a)^{p-\alpha} & \text{if } p \in \mathbb{N} \text{ and } p \geq n \ \text{ or } p \notin \mathbb{N} \text{ and } p > n-1 \end{cases}$$

ii. 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 ${}^C_aD^\alpha_t u(t) = 0$

iii. 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 $a \leq t \leq b$ [6].

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

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

Several methods have been introduced in the literature for the numerical approaches to IFDE's have been recently studied by numerous authors [7–11].

This work is organized as follows: we start by an introduction then focus the fractional differential operator Caputo sense. Preliminaries and discussing numerical methods, quadratic spline function is defined ,devoted to applying the integro-fractional differential, and explains some of the theorems that are needed for this work described in section two. Section three the proposed method is applied to two examples. Also a conclusion is given in section five.

2 Preliminaries

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

Definition 1. 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 especially for integral equations [12] as follows.

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

Remark 1.

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

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

Definition 2. [13] Richardson's extrapolation is used to create high-accuracy result using low-order formulas.

$$T_i^{(m)} = T_{i+1}^{(m-1)} + \frac{T_{i+1}^{(m-1)} - T_i^{(m-1)}}{\left(\dfrac{h_i}{h_{i+m}}\right) - 1}.$$

$i = 1,2,\dots,m$ and $m = 1,2,\dots,k-1$. Where $T_i^{(m)}$ is an approximate value. Richardson extrapolation using step sizes of $h_i,\ h_{i+1},\ \dots,\ h_{i+m}$, and $0 < m \leq k-1$.

Remark 2. The sequence $\{h_i\}$ usually is of the form

$$\left\{h_0,\ \frac{h_0}{2},\ \frac{h_0}{4},\ \frac{h_0}{8},\ \frac{h_0}{16},\dots\right\},\ \left\{h_0,\ \frac{h_0}{2},\ \frac{h_0}{3},\ \frac{h_0}{4},\ \frac{h_0}{5},\dots\right\} \text{ and } \left\{h_0,\ \frac{h_0}{2},\ \frac{h_0}{3},\ \frac{h_0}{4},\ \frac{h_0}{6},\ \frac{h_0}{8},\ \frac{h_0}{12},\dots\right\}$$

In this paper we will consider the sequence $\{h_i\}$ as the form $\left\{h_0,\ \dfrac{h_0}{2},\ \dfrac{h_0}{4},\ \dfrac{h_0}{8},\ \dfrac{h_0}{16},\dots\right\}$.

2.1 Quadratic Classic Spline $Q(t)$: [14]

A function $Q(t)$ is a piecewise spline of degree two can be written as:

$$Q(t) = \begin{cases} Q_0(t) & t \in [t_0, t_1] \\ Q_1(t) & t \in [t_1, t_2] \\ \vdots & \vdots \\ Q_{N-1}(t) & t \in [t_{N-1}, t_N] \end{cases}$$

A quadratic spline consisting of N separate pieces of quadratic functions of the form

$$Q_r(t) = a_r t^2 + b_r t + c_r,\ \ t \in [t_r, t_{r+1}],\ \forall r = 0, 1, \ldots, N-1.$$

In addition $Q(t)$ satisfy the following conditions:
1 The domain of Q is an interval $[a, b]$.
2 Q and $Q^{'}$ are continuous on $[a, b]$.
3 There are points t_r such that $a = t_0 < t_1 < \cdots < t_N = b$ and Q is a polynomial of degree two on each subinterval $[t_r, t_{r+1}]$.
4 A quadratic spline is a continuously differentiable piecewise quadratic function.
5 A quadratic spline is a linear combination of basic functions $1, t, t^2$. The smoothness condition is stronger than that for the first-degree spline.
6 The interpolating condition $Q_r(t_r) = Q_r$ and $Q_r(t_{r+1}) = Q_{r+1}$.
We drive the equation for interpolating quadratic spline $Q(t)$, after some manipulation we obtain,

$$Q(t) = \left(1 - (\frac{t - t_r}{h})^2\right) Q_r + (\frac{t - t_r}{h})^2 Q_{r+1} + \frac{(t - t_r)(t_{r+1} - t)}{h} Q_r^{'}, \tag{3}$$

Where $Q_r(t_r) = Q_r,\ Q_{r+1} = Q_r(t_{r+1}),\ Q_r^{'}(t_r) = m_r,\ Q_r^{'} = Q_r^{'}(t_r)$ and $t_{r+1} - t_r = h$ for $r = 0, 1, \ldots, N-1$. We require from the continuously differentiable condition of quadratic spline function i.e. $Q_r^{'}(t_{r+1}) = Q_{r+1}^{'}$, differentiating Eqn.(3) with respect to t, we get

$$Q_r^{'}(t) = -\frac{2(t - t_r)}{h^2} Q_r(t_r) + \frac{2(t - t_r)}{h^2} Q_r(t_{r+1}) + \frac{(t_{r+1} - t) - (t - t_r)}{h} Q_r^{'}(t_r),$$

Putting $t = t_{r+1}$, we obtain $Q_r^{'}(t_{r+1}) = -Q_r^{'}(t_r) + 2\left(\frac{Q_r(t_{r+1}) - Q_r(t_r)}{h}\right)$

$$Q_{r+1}^{'} = -Q_r^{'} + 2\left(\frac{Q_{r+1} - Q_r}{h}\right),\ \ r = 1, 2, \ldots, N.$$

2.2 Fractional Derivative of Spline Functions

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

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

$$ {}_a^C D_t^{\alpha} Q(t) = \frac{(t-a)^{1-\alpha}}{h\Gamma(3-a)} \Bigg[\frac{-1}{h}(2(t-a) - 2(2-\alpha)(t_r - a))Q_r + \frac{1}{h}(2(t-a) - 2(2-\alpha)(t_r - a))Q_{r+1} $$
$$ + [(t_{r+1} + t_r - 2a)(2-\alpha) - 2(t-a)]Q_r^{'} \Bigg], \qquad \text{where } 0 < \alpha \leq 1.$$

Proof. Quadratic spline function Q(t) in the interval $[t_r, t_{r+1}]$ give the formula in Eqn.(3)

$$Q(t) = A_r(t) Q_r + B_r(t) Q_{r+1} + B_r(t) Q_r^{'},$$

where:

$$A_r(t) = 1 - \left(\frac{t - t_r}{h}\right)^2,\ B_r(t) = 1 - A_r(t) = \left(\frac{t - t_r}{h}\right)^2,$$

$$C_r(t) = \frac{(t-t_r)(t_{r+1}-t)}{h}, \ \forall r = 0, 1, \ldots, N-1.$$

$${}_a^C D_t^\alpha Q(t) = Q_r {}_a^C D_t^\alpha \left[1 - \left(\frac{t-t_r}{h}\right)^2\right] + Q_{r+1} {}_a^C D_t^\alpha \left(\frac{t-t_r}{h}\right)^2 + Q_r' {}_a^C D_t^\alpha \frac{(t-t_r)(t_{r+1}-t)}{h}$$

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

$${}_a^C D_t^\alpha Q(t) = \frac{-1}{h^2}\left[\frac{2(t-a)^{2-\alpha}}{\Gamma(3-\alpha)} - \frac{2(t-a)^{1-\alpha}}{\Gamma(2-\alpha)}(t_r-a)\right]Q_r + \frac{1}{h^2}\left[\frac{2(t-a)^{2-\alpha}}{\Gamma(3-\alpha)} - \frac{2(t-a)^{1-\alpha}}{\Gamma(2-\alpha)}(t_r-a)\right]Q_{r+1}$$
$$+ \frac{1}{h}\left[(t_{r+1}+t_r-2a)\frac{(t-a)^{1-\alpha}}{\Gamma(2-\alpha)} - \frac{2(t-a)^{2-\alpha}}{\Gamma(3-\alpha)}\right]Q_r'.$$

Through calculation, you can get

$$\begin{aligned} {}_a^C D_t^\alpha Q(t) \ &= \frac{(t-a)^{1-\alpha}}{h\Gamma(3-\alpha)}\Bigg[\frac{-1}{h}(2(t-a) - 2(2-\alpha)(t_r-a))Q_r + \frac{1}{h}(2(t-a) - 2(2-\alpha)(t_r-a))Q_{r+1} \\ &+[(t_{r+1}+t_r-2a)(2-\alpha) - 2(t-a)]Q_r'\Bigg], \qquad \text{where} \qquad 0<\alpha\leq 1. \end{aligned} \tag{4}$$

Theorem 1. [6] The finite difference approximation of Caputo derivative for $0 < \alpha \leq 1$ at define points $t = t_{r+1};\ r = 0, 1, \ldots, N-1$ and $h = (b-a)/N$, is formed as

$${}_a^C D_t^\alpha u(t_{r+1}) = \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{r} [u(t_{r-j+1}) - u(t_{r-j})] b_j^\alpha,$$

where $b_j^\alpha = (j+1)^{1-\alpha} - j^{1-\alpha}$.

Theorem 2.(new) The finite difference approximation of Caputo derivative for $0 < \alpha \leq 1$ of the subinterval $[t_r, t_{r+1}]$, define the point $t = t_r + (i+1)h_{M^*},\ h_{M^*} = \dfrac{h}{2^{M^*}}$, $h = (b-a)/N$, according the step size of Richardson Extrapolation $M^* = 0, 1, \ldots, M$, is formed as

$$\begin{aligned} {}_a^C D_t^\alpha u(t)|_{t=t_r+(i+1)h_{M^*}} &= \frac{h^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{r-1} [u(t_{r-j}) - u(t_{r-j-1})] C_{i,j}^{M^*,\alpha} \\ &+ \frac{h_{M^*}^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{i} [u(t_r + (i-j+1)h_{M^*}) - u(t_r + (i-j)h_{M^*})] b_j^\alpha, \end{aligned}$$

where $b_j^\alpha = (j+1)^{1-\alpha} - j^{1-\alpha}$, $C_{i,j}^{M^*,\alpha} = \left[\left((j+1) + \frac{i+1}{2^{M^*}}\right)^{1-\alpha} - \left(j + \frac{(i+1)}{2^{M^*}} h_{M^*}\right)^{1-\alpha}\right]$. $r = 0, 1, \ldots, N-1$, $i = 0, 1, \ldots, 2^{M^*} - 1,\ h = t_{r+1} - t_r$.

Proof. Recall the definition of Caputo fractional derivative for $0 < \alpha \leq 1$ and using first order forward difference approximation [15], to obtain

$${}_a^C D_t^\alpha u(t)|_{t=t_r+(i+1)h_{M^*}} = \frac{1}{\Gamma(1-\alpha)} \int_a^{t_r+(i+1)h_{M^*}} \frac{\partial u(s)/\partial s}{(t_r + (i+1)h_{M^*} - s)^\alpha} ds,$$

$$
\begin{aligned}
{}_a^C D_t^{\alpha} u(t)|_{t=t_r+(i+1)h_{M^*}} &= \frac{1}{\Gamma(1-\alpha)}\left\{\int_a^{t_r} \frac{\partial u(s)/\partial s}{(t_r+(i+1)h_{M^*}-s)^{\alpha}}ds + \int_{t_r}^{t_r+(i+1)h_{M^*}} \frac{\partial u(s)/\partial s}{(t_r+(i+1)h_{M^*}-s)^{\alpha}}ds\right\}\\
&= \frac{1}{\Gamma(1-\alpha)}\left\{\sum_{\ell=0}^{r-1}\int_{a+\ell h}^{a+(\ell+1)h} \frac{\partial u(s)/\partial s}{(t_r+(i+1)h_{M^*}-s)^{\alpha}}ds\right.\\
&\left.+\sum_{\ell=0}^{i}\int_{t_r+\ell h_{M^*}}^{t_r+(\ell+1)h_{M^*}} \frac{\partial u(s)/\partial s}{(t_r+(i+1)h_{M^*}-s)^{\alpha}}ds\right\}\\
&= \frac{1}{\Gamma(1-\alpha)}\left\{\sum_{\ell=0}^{r-1}\frac{u(t_{\ell+1})-u(t_\ell)}{h}\int_{a+\ell h}^{a+(\ell+1)h} \frac{ds}{(t_r+(i+1)h_{M^*}-s)^{\alpha}}\right.\\
&\left.+\sum_{\ell=0}^{i}\frac{u(t_r+(\ell+1)h_{M^*})-u(t_r+\ell h_{M^*})}{h_{M^*}}\int_{t_r+\ell h_{M^*}}^{t_r+(\ell+1)h_{M^*}} \frac{ds}{(t_r+(i+1)h_{M^*}-s)^{\alpha}}\right\}.
\end{aligned}
$$

By assumption Let $\xi = t_r + (i+1)h_{M^*} - s$ then $d\xi = -ds$, if $s = a + \ell h$ then $\xi = (r-\ell)h + (i+1)h_{M^*}$ $s = a+(\ell+1)h$ then $\xi = (r-\ell-1)h+(i+1)h_{M^*}$, if $s = t_r+\ell h_{M^*}$ then $\xi = (i-\ell+1)h_{M^*}$ $s = t_r+(\ell+1)h_{M^*}$ then $\xi = (i-\ell)h_{M^*}$, we obtain

$$
\begin{aligned}
{}_a^C D_t^{\alpha} u(t)|_{t=t_r+(i+1)h_{M^*}} &= \frac{1}{\Gamma(1-\alpha)}\left\{\sum_{\ell=0}^{r-1}\frac{u(t_{\ell+1})-u(t_\ell)}{h}\int_{(r-\ell-1)h+(i+1)h_{M^*}}^{(r-\ell)h+(i+1)h_{M^*}}\frac{d\xi}{\xi^{\alpha}}\right.\\
&\left.+\sum_{\ell=0}^{i}\frac{u(t_r+(\ell+1)h_{M^*})-u(t_r+\ell h_{M^*})}{h_{M^*}}\int_{(i-\ell)h_{M^*}}^{(i-\ell+1)h_{M^*}}\frac{d\xi}{\xi^{\alpha}}\right\}.
\end{aligned}
$$

For first sum; let $j = r-\ell-1$ and the second let $j = i-\ell$

$$
\begin{aligned}
&= \frac{1}{\Gamma(1-\alpha)}\left\{\sum_{\ell=0}^{r-1}\frac{u(t_{r-j})-u(t_r-j-1)}{h}\int_{jh+(i+1)h_{M^*}}^{(j+1)h+(i+1)h_{M^*}}\frac{d\xi}{\xi^{\alpha}}\right.\\
&\left.+\sum_{j=0}^{i}\frac{u(t_r+(i-j+1)h_{M^*})-u(t_r+(i-j)h_{M^*})}{h_{M^*}}\int_{jh_{M^*}}^{(i+1)h_{M^*}}\frac{d\xi}{\xi^{\alpha}}\right\}.
\end{aligned}
$$

After integrating then compute we can obtain,

$$
\begin{aligned}
{}_a^C D_t^{\alpha} u(t)|_{t=t_r+(i+1)h_{M^*}} &= \frac{h^{-\alpha}}{\Gamma(2-\alpha)}\textstyle\sum_{j=0}^{r-1}\left[u(t_{r-j})-u(t_{r-j-1})\right]C_{i,j}^{M^*,\alpha}\\
&+\frac{h_{M^*}^{-\alpha}}{\Gamma(2-\alpha)}\textstyle\sum_{j=0}^{i}\left[u(t_r+(i-j+1)h_{M^*})-u(t_r+(i-j)h_{M^*})b_j^{\alpha}\right],
\end{aligned}
$$

where $C_{i,j}^{M^*,\alpha} = \left[\left((j+1)+\frac{(i+1)}{2^{M^*}}\right)^{1-\alpha} - \left(j+\frac{(i+1)}{2^{M^*}}h_{M^*}\right)^{1-\alpha}\right]$, $b_j^{\alpha} = \left[(j+1)^{1-\alpha}+j^{1-\alpha}\right]$

3 Methods Analysis

To find numerical solution of Eqn.(1), using quadratic 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 Quadratic spline function (VFID's)

To progress the quadratic spline approximation method for solving Volterra integro-fractional differential equation from Eqns.(1) and (2) on the interval $[a,b]$. First divided the interval $[a,b]$ be into N-equal subintervals of length of $h = \dfrac{b-a}{N}$ with endpoints, the quadratic spline $Q(t)$ interpolating the function $u(t)$ at the grid points are specified by the equation.

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

Substituting $t = t_{r+1},\ r = 0,1,2,\ldots,N-1$, from Eqn.(5) then collocate Eqn.(5) at the uniform grid points after substituting Eqn.(4) into Eqn.(5) then we obtain

$$\begin{aligned}&\frac{((r+1)h)^{1-\alpha_n}}{\Gamma(3-\alpha_n)}\left\{\frac{2}{h}[(1-r)\alpha_n r](Q_{r+1}-Q_r)-[\alpha_n(2r+1)-2r]Q_r^{'}\right\}\\
&+\textstyle\sum_{i=1}^{n-1}p_i(t_{r+1})\frac{((r+1)h)^{1-\alpha_n-i}}{\Gamma(3-\alpha_{n-i})}\left\{\frac{2}{h}[(1-r)\alpha_{n-i}r](Q_{r+1}-Q_r)-[\alpha_{n-i}(2r+1)-2r]Q_r^{'}\right\}+p_{n(r+1)}Q_{r+1}\\
&=f_{r+1}+\textstyle\sum_{\ell=0}^{m-1}\lambda_\ell\Big\{\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}}\mathcal{K}_\ell(t_{r+1},s)[Q_j{}_a^C D_s^{\beta_{m-\ell}}A_j(s)+Q_{j+1}{}_a^C D_s^{\beta_{m-\ell}}B_j(s)+Q_j^{'}{}_a^C D_s^{\beta_{m-\ell}}C_j(s)]ds\\
&+\textstyle\int_{t_r}^{t_{r+1}}\mathcal{K}_\ell(t_{r+1},s)\left[Q_r{}_a^C D_s^{\beta_{m-\ell}}A_r(s)+Q_{r+1}{}_a^C D_s^{\beta_{m-\ell}}B_r(s)+Q_r^{'}\,{}_a^C D_s^{\beta_{m-\ell}}C_{r+1}(s)\right]ds\Big\}\\
&+\lambda_m\Big\{\textstyle\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}}\mathcal{K}_m(t_{r+1},s)\left[A_j(s)Q_j+B_j(s)Q_{j+1}+C_j(s)Q_j^{'}\right]ds\\
&+\textstyle\int_{t_r}^{t_{r+1}}\mathcal{K}_m(t_{r+1},s)\left[A_r(s)Q_r+B_r(s)Q_{r+1}+C_r(s)Q_r^{'}\right]ds\Big\}.\end{aligned}\tag{6}$$

Let $\mathcal{W}_n^r(s) = P_{s(r+1)}\frac{((r+1)h)^{1-\alpha_{n-s}}}{\Gamma(3-\alpha_{n-s})}\frac{2}{h}((1-r)+\alpha_{(n-s)r}),\ \mathcal{H}_n^r = P_{n(r+1)}+\sum_{s=0}^{n-1}\mathcal{W}_n^r(s)$ and

$$\mathcal{V}_n^r(s) = P_{s(r+1)}\frac{((r+1)h)^{1-\alpha_{n-s}}}{\Gamma(3-\alpha_{n-s})}(\alpha_{(n-s)}(2r+1)-2r),\ s=0,1,\ldots,n-1,\ r=0,1,2,\ldots,N-1.$$

The Eqn.(6) becomes,

$$\begin{aligned}&Q_{r+1}\left\{\mathcal{H}_n^r-\lambda_m\textstyle\int_{t_r}^{t_{r+1}}\mathcal{K}_m(t_{r+1},s)B_r(s)ds-\sum_{\ell=0}^{m-1}\lambda_\ell\int_{t_r}^{t_{r+1}}\mathcal{K}_\ell(t_{r+1},s){}_a^C D_s^{\beta_{m-\ell}}B_r(s)ds\right\}\\
&=Q_r\left\{\textstyle\sum_{s=0}^{n-1}\mathcal{W}_n^r(s)+\sum_{\ell=0}^{m-1}\lambda_\ell\int_{t_r}^{t_{r+1}}\mathcal{K}_\ell(t_{r+1},s){}_a^C D_s^{\beta_{m-\ell}}A_r(s)ds+\lambda_m\int_{t_r}^{t_{r+1}}\mathcal{K}_m(t_{r+1},s)A_r(s)ds\right\}\\
&+Q_r^{'}\left\{\textstyle\sum_{s=0}^{n-1}\mathcal{V}_n^r(s)+\sum_{\ell=0}^{m-1}\lambda_\ell\int_{t_r}^{t_{r+1}}\mathcal{K}_\ell(t_{r+1},s){}_a^C D_s^{\beta_{m-\ell}}C_{r+1}(s)ds+\lambda_m\int_{t_r}^{t_{r+1}}\mathcal{K}_m(t_{r+1},s)C_{r+1}(s)ds\right\}+f_{r+1}\\
&+\textstyle\sum_{\ell=0}^{m-1}\lambda_\ell\left\{\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}}\mathcal{K}_\ell(t_{r+1},s)\left[Q_j{}_a^C D_s^{\beta_{m-\ell}}A_j(s)+Q_{j+1}{}_a^C D_s^{\beta_{m-\ell}}B_j(s)+Q_j^{'}{}_a^C D_s^{\beta_{m-\ell}}C_j(s)\right]ds\right\}\\
&+\lambda_m\textstyle\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}}\mathcal{K}_m(t_{r+1},s)\left[A_j(s)Q_j+B_j(s)Q_{j+1}+C_j(s)Q_j^{'}\right]ds.\end{aligned}\tag{7}$$

To acquire Q_1, inserting $r = 0$ into Eqn.(7), we must enter Q_0 and $Q_0^{'}$ into Eqn.(7) but as we see we have the best Q_0 from the initial condition. To getting $Q_0^{'}$ from the forward formula, we ought to have $Q_1^{(p)}$. So for preparing $Q_1^{(p)}$ we've critical approaches: (I) Trapezoidal Method and (II) First step of the linear spline.

I- Using Trapezoidal Method

Calculate the Eqn.(1)on interval $[t_0,t_1]$ we obtain,

$$\begin{aligned}&{}_a^C D_t^{\alpha_n}u(t)|_{t=t_1}+\textstyle\sum_{i=1}^{n-1}p_i(t_1){}_a^C D_t^{\alpha_{n-i}}u(t)|_{t=t_1}+p_n(t_1)u_1\\
&=f(t_1)+\textstyle\sum_{\ell=0}^{m-1}\lambda_\ell\int_a^{t_1}\mathcal{K}_\ell(t_1,s){}_a^C D_s^{\beta_{m-\ell}}u(s)ds+\lambda_m\int_a^{t_1}\mathcal{K}_m(t_1,s)u(s)ds\end{aligned}\tag{8}$$

Recall the Theorem (1)

$${}_a^C D_t^{\alpha}u(t)|_{t=t_i}=\frac{h^{-\alpha}}{\Gamma(2-\alpha)}\sum_{j=0}^{i-1}\left[u(t_{i-j})-u(t_{i-j-1})\right]b_j^{\alpha}$$

where

$$b_j^{\alpha} = (j+1)^{1-\alpha} - j^{1-\alpha}.$$

First we can find ${}_a^C D_t^{\alpha} u(t)|_{t=t_1} = \dfrac{h^{-\alpha}}{\Gamma(2-\alpha)}[u_1 - u_0]$, $\int_a^{t_1} \mathcal{K}_\ell(t_1, s) {}_a^C D_s^{\beta_{m-\ell}} u(s) ds = \dfrac{h^{1-\beta_{m-\ell}}}{2\Gamma(2-\beta_{m-\ell})}\mathcal{K}_{11}^{\ell}[u_1 - u_0]$ and $\int_a^{t_1} \mathcal{K}_m(t_1, s) u(s) ds = \dfrac{h}{2}[\mathcal{K}_{10}^m u_0 + \mathcal{K}_{11}^m u_1]$, the Eqn.(8) becomes

$$\begin{aligned} &\frac{h^{-\alpha_n}}{\Gamma(2-\alpha_n)}[u_1 - u_0] + \textstyle\sum_{i=1}^{n-1} p_i(t_1) \dfrac{h^{-\alpha_n - i}}{\Gamma(2-\alpha_n - i)}[u_1 - u_0] + p_n(t_1) u_1 \\ &= f(t_1) + \textstyle\sum_{\ell=0}^{m-1} \lambda_\ell \dfrac{h^{1-\beta_{m-\ell}}}{2\Gamma(2-\beta_{m-\ell})} \mathcal{K}_{11}^{\ell}[u_1 - u_0] + \lambda_m \frac{h}{2}[\mathcal{K}_{10}^m u_0 + \mathcal{K}_{11}^m u_1] \end{aligned}$$

$$\begin{aligned} &u_1 \left\{ \frac{h^{-\alpha_n}}{\Gamma(2-\alpha_n)} + \textstyle\sum_{i=1}^{n-1} p_i(t_1) \dfrac{h^{-\alpha_n - i}}{\Gamma(2-\alpha_n - i)} + p_n(t_1) - \textstyle\sum_{\ell=0}^{m-1} \lambda_\ell \dfrac{h^{1-\beta_{m-\ell}}}{2\Gamma(2-\beta_{m-\ell})} \mathcal{K}_{11}^{\ell} - \lambda_m \frac{h}{2} \mathcal{K}_{11}^m \right\} \\ &= u_0 \left\{ \frac{h^{-\alpha_n}}{\Gamma(2-\alpha_n)} + \textstyle\sum_{i=1}^{n-1} p_i(t_1) \dfrac{h^{-\alpha_n - i}}{\Gamma(2-\alpha_n - i)} - \textstyle\sum_{\ell=0}^{m-1} \lambda_\ell \dfrac{h^{1-\beta_{m-\ell}}}{2\Gamma(2-\beta_{m-\ell})} \mathcal{K}_{11}^{\ell} + \lambda_m \frac{h}{2} \mathcal{K}_{10}^m \right\} + f(t_1) \end{aligned} \tag{9}$$

So $u_1 \approxeq Q_1^{(p)}$.

II- First step of linear spline(using R-Extrapolation Technique)

Calculate the Eqn.(1) on interval $[t_0, t_1]$, $t = t_0 + (i+1)h_{M^*}$, $i = 0, 1, \ldots, 2^{M^*} - 1$, $M^* = 0, 1, \ldots, M$. and $h_{M^*} = \dfrac{h}{2^{M^*}}$, is formed as

$$\begin{aligned} &{}_a^C D_t^{\alpha_n} u(t)|_{t=t_0+(i+1)h_{M^*}} + \textstyle\sum_{i=1}^{n-1} P_i(t_0 + (i+1)h_{M^*}) {}_a^C D_t^{\alpha_{n-i}} u(t)|_{t=t_0+(i+1)h_{M^*}} \\ &+p_n(t_0 + (i+1)h_{M^*}) u(t)|_{t=t_0+(i+1)h_{M^*}} = f(t_0 + (i+1)h_{M^*}) \\ &+ \textstyle\sum_{\ell=0}^{m-1} \lambda_\ell \int_a^{t_0+(i+1)h_{M^*}} k_\ell(t_0 + (i+1)h_{M^*}, s) {}_a^C D_s^{\beta_{m-\ell}} u(s) ds \\ &+\lambda_m \int_a^{t_0+(i+1)h_{M^*}} k_m(t_0 + (i+1)h_{M^*}, s) u(s) ds \end{aligned} \tag{10}$$

Since by finite difference approximation and recall the theorem (2)

$${}_a^C D_t^{\alpha} u(t)|_{t=t_0+(i+1)h_{M^*}} = \frac{h_{M^*}^{-\alpha}}{\Gamma(2-\alpha)} \sum_{j=0}^{i} \left[u(t_r + (i-j+1)h_{M^*}) - u(t_r + (i-j)h_{M^*}) \right] b_j^{\alpha}$$

if $M^* = 0$, $i = 0$, $t = t_0 + (0+1)h_{0^*}$ and $h_{0^*} = \dfrac{h}{2^{0^*}} = h$, we obtain
$= \dfrac{h_0^{-\alpha}}{\Gamma(2-\alpha)} \left[u_{0,1}^0 - u_{0,0}^0 \right] b_0^{\alpha} = \dfrac{h^{-\alpha}}{\Gamma(2-\alpha)} \left[u_1 - u_0 \right]$, the equation (10) becomes

$$\begin{aligned} &\frac{h_{M^*}^{-\alpha_n}}{\Gamma(2-\alpha_n)} \left\{ \left[u_{M^*,i+1}^0 - u_{M^*,i}^0 \right] + \textstyle\sum_{j=1}^{i} \left[u_{M^*,i-j+1}^0 - u_{M^*,i-j}^0 \right] b_j^{\alpha_n} \right\} \\ &+ \textstyle\sum_{ii=1}^{n-1} P_{ii}(t_0 + (i+1)h_{M^*}) \dfrac{h_{m^*}^{-\alpha_{n-ii}}}{\Gamma(2-\alpha_{n-ii})} \left\{ \left[u_{M^*,i+1}^0 - u_{M^*,i}^0 \right] \right. \\ &\left. + \textstyle\sum_{j=1}^{i} \left[u_{M^*,i-j+1}^0 - u_{M^*,i-j}^0 \right] b_j^{\alpha_{n-ii}} \right\} + p_n(t_0 + (i+1)h_{M^*}) u_{M^*,i+1}^0 \end{aligned}$$

$$= f(t_0+(i+1)h_{M^*}) + \sum_{\ell=0}^{m-1}\lambda_\ell \sum_{\ell=0}^{m-1}\lambda_\ell \frac{h_{M^*}}{2}$$
$$\Bigg\{2\sum_{d=0}^{i} k_\ell(t_0+(i+1)h_{M^*}, t_0+dh_{M^*})\frac{h_{M^*}^{-\beta_{m-\ell}}}{\Gamma(2-\beta_{m-\ell})}\sum_{j=0}^{d-1}\Big[u^0_{M^*,d-j}-u^0_{M^*,d-j-1}\Big]b_j^{\beta_{m-\ell}}$$
$$+k_\ell(t_0+(i+1)h_{M^*}, t_0+(i+1)h_{M^*})\frac{h_{M^*}^{-\beta_{m-\ell}}}{\Gamma(2-\beta_{m-\ell})}\Bigg[(u^0_{M^*,i+1}-u^0_{M^*,i})$$
$$+\sum_{j=1}^{i}(u^0_{M^*,i-j+1}-u^0_{M^*,i-j})b_j^{\beta_{m-\ell}}\Bigg]\Bigg\}$$
$$+\lambda_m\frac{h_{M^*}}{2}\Bigg[k_m(t_0+(i+1)h_{M^*}, t_0)u^0_{M^*,0}+2\sum_{d=1}^{i}k_m(t_0+(i+1)h_{M^*}, t_0+dh_{M^*})u^0_{M^*,d}$$
$$+k_m(t_0+(i+1)h_{M^*}, t_0(i+1)h_{M^*})u^0_{M^*,i+1}\Bigg].$$

Let $\mathcal{A}_k^{\sigma,M^*}(s)=\dfrac{h_{M^*}^{-\sigma_{k-s}}}{\Gamma(2-\sigma_{k-s})}$ and $\mathcal{H}_k^{\sigma,M^*}(i+1)p_k(t_0+(i+1)h_{M^*})+\sum_{s=0}^{n-1}P_s(t_0+(i+1)h_{M^*})\mathcal{A}_k^{\sigma,M^*}(s)$, $\forall k\in\mathbb{Z}^+,\ s=0,1,\dots,k-1,\ \ i=0,1,\dots,2^{M^*}-1$ with $\mathcal{A}_k^{\sigma,M^*}(k)=1,\ P_0(t_0+(i+1)h_{M^*})=1$

$$u^0_{M^*,i+1}\Bigg\{\mathcal{H}_n^{\alpha,M^*}(i+1)-\tfrac{h_{M^*}}{2}\sum_{\ell=0}^{m-1}\lambda_\ell k_{i+1,i+1}^{\ell,M^*,0}\mathcal{A}_m^{\beta,M^*}(\ell)-\frac{\lambda_m h_{M^*}}{2}k_{i+1,i+1}^{m,M^*,0}\Bigg\}$$
$$=f_{i+1}^{M^*,0}+\Bigg\{\sum_{s=0}^{n-1}p^0_{s,i+1}\mathcal{A}_n^{\alpha,M^*}(s)-\tfrac{h_{M^*}}{2}\sum_{\ell=0}^{m-1}\lambda_\ell k_{i+1,i+1}^{\ell,M^*,0}\mathcal{A}_m^{\beta,M^*}(\ell)\Bigg\}u^0_{M^*,i}$$
$$-\sum_{j=1}^{i}\Bigg[u^0_{M^*,i-j+1}-u^0_{M^*,i-j}\Bigg]\left(\sum_{ii=0}^{n-1}P^0_{ii,i+1}\mathcal{A}_n^{\alpha,M^*}(ii)b_j^{\alpha_n-ii}\right)$$
$$+h_{M^*}\sum_{d=1}^{i}\sum_{j=0}^{d=1}\Bigg[u^0_{M^*,d-j}-u^0_{M^*,d-j-1}\Bigg]\left(\sum_{\ell=0}^{m-1}\lambda_\ell k_{i+1,d}^{\ell,M^*,0}\mathcal{A}_m^{\beta,M^*}(\ell)b_j^{\beta_m-\ell}\right)$$
$$+\frac{h_{M^*}}{2}\sum_{j=1}^{i}\Bigg[u^0_{M^*,i-j+1}-u^0_{M^*,i-j}\Bigg]\left(\sum_{\ell=0}^{m-1}\lambda_\ell k_{i+1,i+1}^{\ell,M^*,0}\mathcal{A}_m^{\beta,M^*}(\ell)b_j^{\beta_m-\ell}\right)$$
$$+\frac{\lambda_m h_{M^*}}{2}k_{i+1,i}^{m,M^*,0}u^0_{M^*,i}+\lambda_m h_{M^*}\sum_{d=0}^{i}k_{i+1,d}^{m,M^*,0}u^0_{M^*,d}.$$

If $M^*=0,\ i=0$ and $h_0=\frac{h}{2^0}=h$, we obtain

$$u^0_{0,1}\Bigg\{\mathcal{H}_n^{\alpha,0}(1)-\tfrac{h_0}{2}\sum_{\ell=0}^{m-1}\lambda_\ell k_{1,1}^{\ell,0}\mathcal{A}_m^{\beta,0}(\ell)-\frac{\lambda_m h_0}{2}k_{1,1}^{m,0}\Bigg\}$$
$$=f_1^0+\Bigg\{\sum_{s=0}^{n-1}p^0_{s,1}\mathcal{A}_n^{\alpha,0}(s)-\tfrac{h_0}{2}\sum_{\ell=0}^{m-1}\lambda_\ell k_{1,1}^{\ell,0}\mathcal{A}_m^{\beta,0}(\ell)\Bigg\}u^0_{0,0}+\tfrac{h_0}{2}k_{1,0}^{m,0,0}u^0_{0,0} \tag{11}$$

So we can say $u^0_{0,1}\approxeq Q_1^{(p)}$, from this we come to the conclusion that the path normal quadratic spline function can be analyzed with two technique the first is using Trapezoid (**NQST**). And the second using First step of linear spline (using Richardson Extrapolation) (**NQSL**).

3.2 Quadratic spline function using Extrapolation (VFID's)

To sketching the quadratic spline proximity 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 $h=\dfrac{b-a}{N}$ with endpoints, the quadratic spline $Q(t)$ interpolating the function with Extrapolation in the interval $t\in[t_r+ih_{M^*}, t_r+(i+1)h_{M^*}]$ is specified by the formula

$$Q^r_{M^*,i}(t)=A^r_{M^*,i}(t)Q^r_{M^*,i}+B^r_{M^*,i}(t)Q^{r+1}_{M^*,i}+C^r_{M^*,i}(t)Q^{(')\ r}_{M^*,i}$$

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

Where

$$A^r_{M^*,i}(t) = 1 - \left(\frac{t-(t_r+ih_{M^*})}{h_{M^*}}\right)^2,\ B^r_{M^*,i}(t) = 1 - A^r_{M^*,i}(t) = \left(\frac{t-(t_r+ih_{M^*})}{h_{M^*}}\right)^2,$$
$$C^r_{M^*,i}(t) = \frac{(t-(t_r+ih_{M^*}))((t_r+(i+1)h_{M^*})-t)}{h_{M^*}},$$

Thus

$$\begin{aligned} {}^C_aD^\alpha_t Q^r_{M^*,i}(t) \quad &= \frac{-2}{h^2_{M^*}}\left[\frac{(t-a)^{2-\alpha}}{\Gamma(3-\alpha)} - \frac{(t-a)^{1-\alpha}}{\Gamma(2-\alpha)}(t_r+ih_{M^*}-a)\right]Q^r_{M^*,i} \\ &+\frac{2}{h^2_{M^*}}\left[\frac{(t-a)^{2-\alpha}}{\Gamma(3-\alpha)} - \frac{(t-a)^{1-\alpha}}{\Gamma(2-\alpha)}(t_r+ih_{M^*}-a)\right]Q^r_{M^*,i+1} \\ &-\frac{2}{h^2_{M^*}}\left[\frac{(t-a)^{2-\alpha}}{\Gamma(3-\alpha)} - \frac{(t-a)^{1-\alpha}}{\Gamma(2-\alpha)}(t_r+ih_{M^*}-a)\right]Q^r_{M^*,i+1} \\ &-\frac{2}{h^2_{M^*}}\left[\frac{(t-a)^{2-\alpha}}{\Gamma(3-\alpha)} - \left(t_r+(i+\frac{1}{2})h_{M^*}-a\right)\frac{(t-a)^{1-\alpha}}{\Gamma(2-\alpha)}\right]Q^{(')\,r}_{M^*,i} \end{aligned} \tag{12}$$

From Eqn.(1) putting $t = t_r + (i+1)h_{M^*}$, then collocate Eqn.(5) at the uniform grid points after substituting Eqn.(12) into Eqn.(5) then we obtain

$$\begin{aligned} &\frac{2h^{2-\alpha_n}_{M^*}(2^{M^*}r+i+1)^{1-\alpha_n}}{h^2_{M^*}\Gamma(3-\alpha_n)}\left[2^{M^*}r(\alpha_n-1)+i(\alpha_n-1)+1\right]\left[Q^r_{M^*,i+1}-Q^r_{M^*,i}\right] \\ &-\frac{2h^{2-\alpha_n}_{M^*}(2^{M^*}r+i+1)^{1-\alpha_n}}{h_{M^*}\Gamma(3-\alpha_n)}\left[2^{M^*}r(\alpha_n-1)+i(\alpha_n-1)+\tfrac{1}{2}\alpha_n\right]Q^{(')\,r}_{M^*,i} \\ &+\textstyle\sum_{ii=1}^{n-1}P_{ii}(t_r+(i+1)h_{M^*})\left\{\frac{2h^{2-\alpha_{n-ii}}_{M^*}(2^{M^*}r+i+1)^{1-\alpha_{n-ii}}}{h^2_{M^*}\Gamma(3-\alpha_{n-ii})}\right. \\ &\left.\left[2^{M^*}r(\alpha_{n-ii}-1)+i(\alpha_{n-ii}-1)+1\right]\left[Q^r_{M^*,i+1}-Q^r_{M^*,i}\right]\right\} - \frac{2h^{2-\alpha_{n-ii}}_{M^*}(2^{M^*}r+i+1)^{1-\alpha_{n-ii}}}{h_{M^*}\Gamma(3-\alpha_{n-ii})} \\ &\left[2^{M^*}r(\alpha_{n-ii}-1)+i(\alpha_{n-ii}-1)+\tfrac{1}{2}\alpha_{n-ii}\right]Q^{(')\,r}_{M^*,i} + p_n(t_r+(i+1)h_{M^*})Q^r_{M^*,i+1} \\ &= f(t_r+(i+1)h_{M^*}) + \textstyle\sum_{\ell=0}^{m-1}\lambda_\ell\left[\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}}k_\ell(t_r+(i+1)h_{M^*},s)[{}^C_aD^{\beta_{m-\ell}}_s A_j(s)Q^C_j\right. \\ &\left.+{}^C_aD^{\beta_{m-\ell}}_s B_j(s)Q^C_{j+1} + {}^C_aD^{\beta_{m-\ell}}_s C_j(s)Q^{('),C}_j]ds\right] \\ &+\textstyle\sum_{j=0}^{i-1}\int_{t_r+jh_{M^*}}^{t_r+(j+1)h_{M^*}}k_\ell(t_r+(i+1)h_{M^*},s)[{}^C_aD^{\beta_{m-\ell}}_s A^r_{M^*,j}(s)Q^r_{M^*,j} \\ &+{}^C_aD^{\beta_{m-\ell}}_s B^r_{M^*,j}(s)Q^r_{M^*,j+1} + {}^C_aD^{\beta_{m-\ell}}_s C^r_{M^*,j}(s)Q^{('),r}_{M^*,j}]ds \\ &+\textstyle\int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}}k_\ell(t_r+(i+1)h_{M^*},s) \\ &[{}^C_aD^{\beta_{m-\ell}}_s A^r_{M^*,i}(s)Q^r_{M^*,i} + {}^C_aD^{\beta_{m-\ell}}_s B^r_{M^*,i}(s)Q^r_{M^*,i+1} + {}^C_aD^{\beta_{m-\ell}}_s C^r_{M^*,i}(s)Q^{('),r}_{M^*,i}]ds \\ &\lambda_m\left[\textstyle\sum_{j=0}^{r-1}\int_{t_j}^{t_{j+1}}k_m(t_r+(i+1)h_{M^*},s)[A_j(s)Q^C_j + B_j(s)Q^C_{j+1} + C_j(s)Q^{('),C}_j]ds\right] \\ &+\textstyle\sum_{j=0}^{i-1}\int_{t_r+jh_{M^*}}^{t_r+(j+1)h_{M^*}}k_m(t_r+(i+1)h_{M^*},s)[A^r_{M^*,j}(s)Q^r_{M^*,j} + B^r_{M^*,j}(s)Q^r_{M^*,j+1} + C^r_{M^*,j}(s)Q^{('),r}_{M^*,j}]ds \\ &+\textstyle\int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}}k_m(t_r+(i+1)h_{M^*},s)[A^r_{M^*,i}(s)Q^r_{M^*,i} + B^r_{M^*,i}(s)Q^r_{M^*,i+1} + C^r_{M^*,i}(s)Q^{('),r}_{M^*,i}]ds \end{aligned} \tag{13}$$

Let

$$\mathcal{W}^{n,r}_{M^*,i}(s) = P_{s(t_r+(i+1)h_{M^*})}\frac{2h^{2-\alpha_{n-s}}_{M^*}(2^{M^*}r+i+1)^{1-\alpha_{(n-s)}}}{h^2_{M^*}\Gamma(3-\alpha_{(n-s)})}\left[2^{M^*}r(\alpha_{(n-s)}-1)+i(\alpha_{(n-s)}-1)+1\right]$$

$$\mathcal{V}^{n,r}_{M^*,i}(s) = P_{s(t_r+(i+1)h_{M^*})}\frac{2h^{2-\alpha_{n-s}}_{M^*}(2^{M^*}r+i+1)^{1-\alpha_{(n-s)}}}{h_{M^*}\Gamma(3-\alpha_{(n-s)})}\left[2^{M^*}r(\alpha_{(n-s)}-1)+i(\alpha_{(n-s)}-1)+\frac{1}{2}\alpha_{(n-s)}\right]$$

$$\mathcal{H}^{n,r}_{M^*,i}(s) = P_{s(t_r+(i+1)h_{M^*})} + \sum_{s=0}^{n-1}\mathcal{W}^{n,r}_{M^*,i}(s),\ r=0,1,\dots,N-1,\ s=0,1,\dots,n-1.$$

$i=0,1,\dots,2^{M^*}-1,\ M^*=0,1,\dots,M.\ h_{M^*}=\dfrac{h}{2^{M^*}},\ \forall M\in\mathbb{Z}^+\cup\{0\}$, then Eqn(13) becomes

$$\begin{aligned}
&Q^r_{M^*,i+1}\Bigg\{\mathcal{H}^{n,r}_{M^*,i}(s) - \textstyle\sum_{\ell=0}^{m-1}\lambda_\ell\int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}} k_\ell(t_r+(i+1)h_{M^*},s)\,{}^C_aD^{\beta_{m-\ell}}_s B^r_{M^*,j}(s)ds\\
&\quad -\lambda_m\textstyle\int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}} k_m(t_r+(i+1)h_{M^*},s)B^r_{M^*,j}(s)ds\Bigg\}\\
&= f(t_r+(i+1)h_{M^*}) + Q^r_{M^*,i}\Bigg\{\textstyle\sum_{s=0}^{n-1}\mathcal{W}^{n,r}_{M^*,i}(s)\\
&\quad +\textstyle\sum_{\ell=0}^{m-1}\lambda_\ell\int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}} k_\ell(t_r+(i+1)h_{M^*},s)\,{}^C_aD^{\beta_{m-\ell}}_s A^r_{M^*,i}(s)ds\\
&\quad +\lambda_m\textstyle\int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}} k_m(t_r+(i+1)h_{M^*},s)A^r_{M^*,j}(s)ds\Bigg\}\\
&+Q^{(')\,r}_{M^*,i}\Bigg\{\textstyle\sum_{s=0}^{n-1}\mathcal{V}^{n,r}_{M^*,i}(s) + \sum_{\ell=0}^{m-1}\lambda_\ell\int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}} k_\ell(t_r+(i+1)h_{M^*},s)\,{}^C_aD^{\beta_{m-\ell}}_s C^r_{M^*,i}(s)ds\\
&\quad +\lambda_m\textstyle\int_{t_r+ih_{M^*}}^{t_r+(i+1)h_{M^*}} k_m(t_r+(i+1)h_{M^*},s)C^r_{M^*,i}(s)ds\Bigg\}\\
&+\textstyle\sum_{\ell=0}^{m-1}\lambda_\ell\Bigg[\sum_{j=0}^{n-1}\int_{t_j}^{t_{j+1}} k_\ell(t_r+(i+1)h_{M^*},s)[{}^C_aD^{\beta_{m-\ell}}_s A_j(s)Q^c_j\\
&\quad +{}^C_aD^{\beta_{m-\ell}}_s B_j(s)Q^C_{j+1}(s) + {}^C_aD^{\beta_{m-\ell}}_s C_j(s)Q^{('),C}_j]ds\\
&\textstyle\sum_{j=0}^{i-1}\int_{t_r+jh_{M^*}}^{t_r+(j+1)h_{M^*}} k_\ell(t_r+(i+1)h_{M^*},s)[{}^C_aD^{\beta_{m-\ell}}_s A^r_{M^*,j}(s)Q^r_{M^*,j}\\
&\quad +{}^C_aD^{\beta_{m-\ell}}_s B^r_{M^*,j}(s)Q^r_{M^*,j+1}(s) + {}^C_aD^{\beta_{m-\ell}}_s C^r_{M^*,j}(s)Q^{('),r}_{M^*,j}]ds\Bigg]\\
&\quad +\lambda_m\Bigg[\textstyle\sum_{j=0}^{n-1}\int_{t_j}^{t_{j+1}} k_m(t_r+(i+1)h_{M^*},s)[A_j(s)Q^c_j\\
&+B_j(s)Q^C_{j+1} + C_j(s)Q^{('),C}_j]ds + \textstyle\sum_{j=0}^{i-1}\int_{t_r+jh_{M^*}}^{t_r+(j+1)h_{M^*}} k_m(t_r+(i+1)h_{M^*},s)\\
&[A^r_{M^*,j}(s)Q^r_{M^*,j} + B^r_{M^*,j}(s)Q^r_{M^*,j+1}(s) + C^r_{M^*,j}(s)Q^{('),r}_{M^*,j}]ds\Bigg]
\end{aligned} \tag{14}$$

Eqn.(14) is obtained to find the approximate solution $Q^r_{M^*,i+1}$, $\forall r=0,1,\dots,N-1$, $\forall i=0,1,\dots,2^{M^*}-1$, $M^*=0,1,\dots,M,\ h_{M^*}=\dfrac{h}{2^{M^*}}$, $\forall M\in\mathbb{Z}^+\cup\{0\}$. For finding $Q_0^{'}$ by forward formula $Q_0^{'}=\dfrac{Q_1^{(p)}-Q_0}{h}$, Q_0 is the initial condition Eqn.(2), using for next iterations $Q_r^{'}+Q_{r-1}^{'}=\dfrac{2(Q_r-Q_{r-1})}{h}$, $r=1,2,\dots,N$, in this technique we need to find $Q_1^{(p)}$ so in the same way such as how through from Eqn.(9) or Eqn.(11), from this we come to the conclusion that the path Extrapolation quadratic spline function can be analyzed with two technique the first is using trapezoid (**EQST**). And the second using first step of linear spline (using Richardson Extrapolation) (**EQSL**).

4 Numerical scheme

In this section, we present two examples in which their numerical results. In examples (1) and (2), we have compared the results of this method for (VIFDE's) of the second kind with collocation normal spline and Richard Extrapolation method the result shown in tables (1,3) an almost large interval to show capability of the method shown in table (4) for example 2. Finally, demonstrates figures (1,2) to compared numerical and the exact solution of the examples (1) and (2) where $h = 0.1$.

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

$$ {}_0^C D_t^{0.7} u(t) + t {}_0^C D_t^{0.2} - 2u(t) = $$

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

where

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

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 $0 \le t \le 1$:

$$ {}_0^C D_t^{2\beta} u(t) - \frac{1}{2} {}_0^C D_t^{\beta} + (1 + t^3) u(t) = $$

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

where

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

With the initial condition: $u(0) = -1$, where the exact solution of this problem is known $u(t) = t^2 - 1$, and $\beta = 0.5$.

Table 1

Exact and numerical solution of example 1

t	Exact	$N = 10$			
		NQST	NQSL	EQST ($m = 2$)	EQSL ($m = 2$)
0	1	1.0	1.0	1.0	1.0
0.1	0.8	0.8000004930653	0.8	0.80000000707939	0.8
0.2	0.6	0.6000007031481	0.6	0.60000071148317	0.6
0.3	0.4	0.4000010136467	0.4	0.40000222633954	0.4
0.4	0.2	0.2000013769366	0.2	0.20000484834216	0.2
0.5	0.0	0.0000017667637	2.496349159 e^{-17}	0.0000084819165	7.57181976631 e^{-19}
0.6	-0.2	-0.1999978313918	-0.2	-0.19998695732248	-0.2
0.7	-0.4	-0.39999742490777	-0.4	-0.39998156309772	-0.4
0.8	-0.6	-0.59999701665405	-0.6	-0.59997541684731	-0.6
0.9	-0.8	-0.79999660662054	-0.8	-0.79996858034163	-0.8
1.0	-1	-0.99999619296744	-1.0	-0.99996109404	-1.0
L.S.E		5.3025254 $e-11$	6.2317591 $e-34$	3.7161423 $e-9$	5.7332455 $e-37$
R.Time/Sec		40.994059	51.198122	302.24999	419.6619

The result in Table (2) shows R-Extrapolation technique for solving quadratic spline methods using trapezoidal and first step of linear spline for $h = 0.05$ ($N = 20$). For Example 1.

Table 2

The comparison of the solution R-Extrapolation technique

t	Exact	$N = 20$			
		EQST($m = 2$)	EQSL ($m = 2$)	EQST ($m = 3$)	EQSL ($m = 3$)
0	1	1.0	1.0	1.0	1.0
0.1	0.8	0.800000008905707	0.8	0.800000006359992	0.8
0.2	0.6	0.600000050701418	0.6	0.600000045813416	0.6
0.3	0.4	0.400000151375986	0.4	0.400000137044427	0.4
0.4	0.2	0.20000031526505	0.2	0.200000284526254	0.2
0.5	0.0	0.00000053995763	1372334369 e^{-18}	0.0000004860751345	$-7.73860979575\ e^{-19}$
0.6	-0.2	-0.199999179770634	-0.2	-0.199999262960361	-0.2
0.7	-0.4	-0.399998849511254	-0.4	-0.399998967571431	-0.4
0.8	-0.6	-0.599998474114693	-0.6	-0.599998632098907	-0.6
0.9	-0.8	-0.799998057255924	-0.8	-0.799998259837467	-0.8
1.0	-1	-0.999997601330418	-1.0	-0.999997852936052	-1.0
$L.S.E$		1.4269108 $e{-11}$	4.5677668 $e{-36}$	1.1456484 $e{-11}$	5.9886082 $e{-37}$
$R.Time/Sec$		855.47371	2374.498	2413.3623	2398.2578

Table 3

Exact and numerical solution of example 2

t	Exact	$N = 10$			
		NQST	NQSL	EQST ($m = 2$)	EQSL ($m = 2$)
0	1	1.0	1.0	1.0	1.0
0	-1.0	-1.0	-1.0	-1.0	-1.0
0.1	-0.99	-0.984097358703339	-0.98385921585012	-0.991476290775673	-0.991479158909235
0.2	-0.96	-0.956638239698859	-0.95627586881473	-0.965258118725224	-0.965263505489059
0.3	-0.91	-0.914517584731656	-0.914089026848564	-0.921013271619213	-0.921018780914492
0.4	-0.84	-0.856459372872957	-0.855994041063225	-0.859259280306947	-0.859264546925167
0.5	-0.75	-0.782066581154528	-0.781577394923225	-0.780434507191968	-0.780439517960847
0.6	-0.64	-0.691478975785645	-0.690968031530877	-0.685006055226451	-0.685010914123675
0.7	-0.51	-0.585278870476111	-0.584740180555245	-0.573535077158801	-0.57353994630581
0.8	-0.36	-0.464504501114389	-0.463925265789229	-0.446739340155018	-0.446744421203363
0.9	-0.19	-0.330716434084407	-0.330077470104528	-0.305564188908158	-0.305569718588994
1.0	0	-0.186096022057057	-0.185371604481513	-0.151269656763997	-0.151275909797876
$L.S.E$		0.075036756	0.074289865	0.051271853	0.051277655
$R.Time/Sec$		46.939625	53.091482	355.20677	297.93127

The result in table (4) shows the least-square errors and running times (elapsed time) for quadratic spline methods with different values of steps size h. For Example 2.

Table 4

Comparision for different value of N

h	0.1($N = 10$)		0.05($N = 20$)		0.033($N = 30$)	
Quadratic	L.S.E	R.Time /Sec	L.S.E	R.Time /Sec	L.S.E	R.Time /Sec
NQSL	0.074289865	53.091482	0.066459532	136.90037	0.059164974	308.13296
EQSL ($m = 2$)	0.051277655	297.93127	0.046061105	1115.3817	0.044419219	2788.2102
EQSL ($m = 3$)	0.045270078	1908.428	0.043314536	3855.422	0.04268915	8619.3941

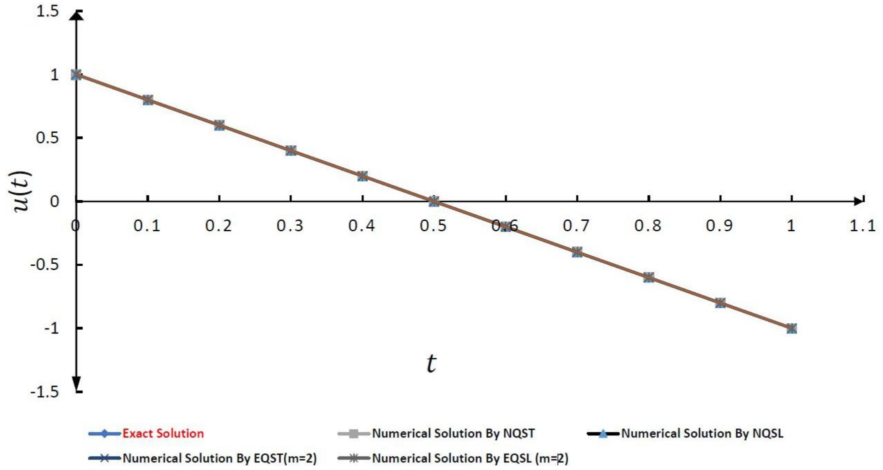

Figure 1. Compared numerical and the exact solution of the example 1, $h = 0.1$.

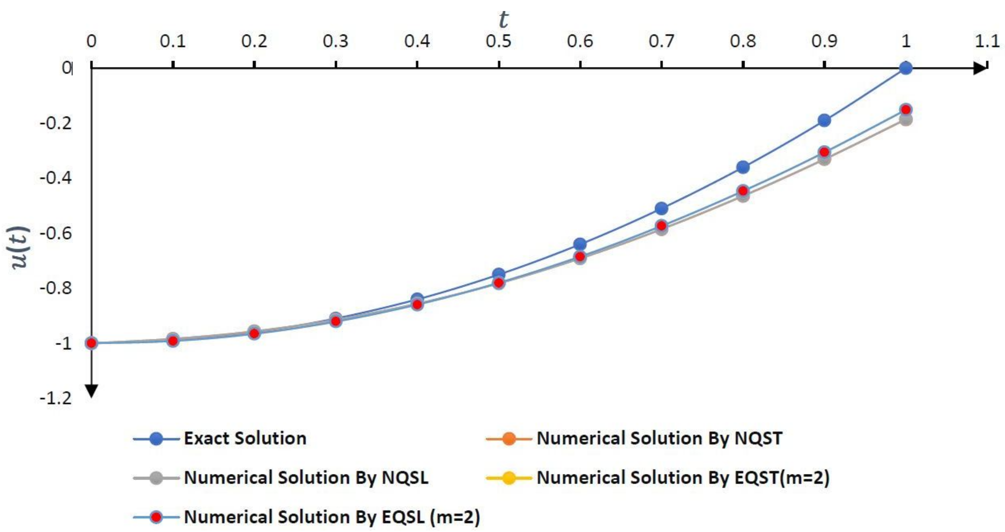

Figure 2. Compared numerical and the exact solution of the example 2, $h = 0.1$.

5. Conclusion

In this work, we have fully attempted to find the numerical solution of the Volterra integro-fractional differential equations (VIFDE's) by using quadratic spline approximate. The numerical procedure and methodology are done in a very straightforward and effective manner. Through the numerical calculation, we confirmed that the Richardson Extrapolation method has the highest degree of accuracy. On the basis of this work, tables (2) and (4) displayed comparison between normal quadratic spline and using Extrapolation method with different step sizes. Furthermore, interpolating quadratic spline for linear function is closer to the quadratic function as displayed in tables (1) and (3). Likewise, new techniques using the first derivative at the initial point by the First step of linear spline(using R-Extrapolation technique), then our solution would be better and the ration of mistake would be fewer in comparison of the method of Trapezoidal at finding first derivative at the initial point. Figures (1) and (2) represents which one is the best technique for solving (VIFDE's).

References

1 Ricardo, A. (2017). A Caputo fractional derivative of a function with respect to another function. *Communications in Nonlinear Science and Numerical Simulation, 44*, 460–481. DOI.org/10.1016/j.cnsns.2016.09.006.

2 Tenreiro, J., Silva, M., Barbosa, R. , Jesus, I., Reis, C., Marcos, M., & Galhano, A. (2010). Some Applications of Fractional Calculus in Engineering. *Mathematical Problems in Engineering*, 1–34.

3 Guo, B., Pu, X., & Huang, F. (2015). *Fractional Partial Differential Equations and Their Numerical Solutions.* World Scientific Publishing Co: Pte. Ltd.

4 Guang-hua, G., Zhi-Zhong, S., & Hong-Wei, Zh. (2014). A new fractional numerical differentiation formula to approximate the Caputo fractional derivative and its applications. *Journal of Computa-tional Physics, 259*, 33–50. DOI.org/10.1016/j.jcp.2013.11.017.

5 Mustafa, A.D., Ercan C., Hasan, B., & Haci, M. (2018). Cancer treatment model with the Caputo-Fabrizio fractional Derivative. *The European physical journal plus, 133*, 92.
DOI 10.1140/epjp/i2018-11950-y.

6 Shazad, Sh.A. (2009). *On system of linear volterra integro-fractional differential equations.* Ph.D. dissertation, University of Sulaimani.

7 Hamid, M., Hamid, S., Abolfazl, Gh., & Yones, E. (2019). The Cubic B-spline Operational Matrix Based on Haar Scaling Functions for Solving Varieties of the Fractional Integro-differential Equations. *Journal of Mathematics, 51*(8), 45–65.

8 Shazad, Sh.A., & Shokhan, A.H. (2011). Generalized taylor matrix method for solving linear integro-fractional differential equations of volterra type. *Applied Mathematical Sciences, 5*, 1765–1780.

9 Khosrow, M., Reza, M., & Paria, M. (2013). Numerical Solution of Volterra Functional Integral Equation by Using Cubic B-Spline Scaling Functions. *Numerical Methods for Partial Differential Equations, 30*(2), 699–722. DOI: 10.1002/num.21837.

10 Shazad, Sh.A., Shokhan, A.H., & Mariwan, R.A. (2019). Laplace adomian and laplace modified adomian decomposition methods for solving nonlinear integro-fractional differential equations of the volterra-hammerstein type. *Iraqi Journal of Science, 60*, 2207–2222.

11 Shazad, Sh.A., & Bzhween, A.S. (2019). Generalized Taylor Matrix Method for Solving Multi-Higher Nonlinear Integro-Fractional Differential Equations of Fredholm Type. *ZANCO Journal of Pure and Applied Sciences*, 2218-0230. DOI: http://dx.doi.org/10.21271/zjpas.

12 Rajani, B.D., & Debasish, D. (2011). A Mixed Quadrature Rule by Blending Clenshaw-Curtis and Gauss-Legendre Quadrature Rules for Approximation of Real Definite Integrals in Adaptive Environment. *Proceedings of the International MultiConference of Engineers and Computer Scientists, 1*, 16–18.

13 Istvan, F., Agnes, H., & Zahari, Z. (2010). Efficient implementation of stable Richardson Extrapolation algorithms. *Computers and Mathematics with Applications, 60*, 2309–2325. DOI.org/10.1016/j.camwa.2010.08.025.

14 Michael, F., Tom, L., Marie, M., Knut, M., Larry, L., & Schumaker. (2012). Mathematical Methods for Curves and Surfaces. *8th International Conference*, Oslo, Norway.

15 Burden, R., & Faires, J. (2011). *Numerical Analysis.* 9th ed., USA: Brooks/Cole. (34).

К.Х.Ф. Джвамер, Ш.Ш. Ахмед, Д.Х. Абдулла

Квадраттық сплайн-функциясын қолдана отырып, Вольтеррдің интегро-бөлшекті дифференциалдық теңдеулерінің жуық шешімі

Мақалада қалыпты квадраттық сплайн-функциясына негізделген Вольтеррдің интегро-бөлшекті-дифференциалдық теңдеудің шешімін жуықтаудың екі жаңа әдісі ұсынылған, ал екінші әдісте дискретті коллокация нүктелерін қолдана отырып, Ричардсонның экстраполяция әдісі қолданылған. Бөлшекті туындылары Капуто түсінігінде қарастырылды. Олардың дәлелдерімен қатар, Ричардсонның экстраполяция нүктелеріне Капуто туындысының ақырлы айырмашылығын қолдану үшін жаңа теорема енгізілді. Бастапқы нүктеде бірінші туындыны қолданатын жаңа әдістер негізінде келесі екі жағдай алынған: біріншісінде трапеция ережесі, екіншісінде Ричардсонның экстраполяция әдісі негізінде сызықтық сплайн-функциясының бірінші қадамы қолданылған. Атап айтқанда, бағдарлама Matlab (R2018b) талдау мысалдарында келтірілген. Әдістердің өнімділігі мен сенімділігін көрсететін сандық мысалдар бар, сонымен қатар, осы теңдеулер үшін қажетті интегралдарды есептеу үшін Кертис Кленшоу ережесі қолданылды.

Кілт сөздер: интегро-бөлшекті дифференциалдық теңдеу, Капуто туындысы, квадраттық сплайн, экстраполяция әдісі, Кленшоу.

К.Х.Ф. Джвамер, Ш.Ш. Ахмед, Д.Х. Абдулла

Приближенное решение интегро-дробных дифференциальных уравнений Вольтерра с использованием квадратичной сплайн-функции

В статье предложены два новых метода аппроксимации решения интегро-дробно-дифференциального уравнения Вольтерра (VIFDE), основанные на нормальной квадратичной сплайн-функции, а в основе второго метода лежит метод экстраполяции Ричардсона с использованием дискретных точек коллокации. Дробные производные рассмотрены в восприятии Капуто. Вместе с их доказательством введена новая теорема для точек экстраполяции Ричардсона для использования конечно-разностной аппроксимации производной Капуто. Новые методы с использованием первой производной в начальной точке таковы, что получены следующие два случая: первый с использованием правила трапеции, а второй — с учетом первого шага линейной сплайн-функции методом экстраполяции Ричардсона. В частности, программа приведена в примерах анализа в Matlab (R2018b). Имеются числовые примеры, чтобы продемонстрировать продуктивность и надежность методов, а также следовать правилу Кленшоу Кертиса для вычисления требуемых интегралов для этих уравнений.

Ключевые слова: интегро-дробное дифференциальное уравнение, производная Капуто, квадратичный сплайн, метод экстраполяции, Кленшоу.