Journal of Basrah Researches (Sciences) **51**(2), 1 (2025)
DOI: https://doi.org/10.56714/bjrs.51.2.1

Finite Difference Approximation with the Quadrature Method for Solving Fredholm Integro-Differential Equations of Fractional Order

Dashne Chapuk Zahir[1,*], Shazad Shawki Ahmed[2]

[1]Department of Mathematics, Faculty of Science and Health, Koya University, Koya 46017, Iraq.
[2]Department of Mathematics, College of Science, Sulaimani University, Sulaymaniyah 46001, Iraq.

ARTICLE INFO

Received 12 May 2025
Revised 19 July 2025
Accepted 28 July 2025
Published 31 December 2025

Keywords:

Fractional Calculus, Caputo-Fractional Derivative, Integral-Differential Equation, Newton-Cotes Quadrature Technique, Trapezoidal Method, Simpson's Method, Forward Difference Approximation.

Citation: D. C. Zahir, S. S. Ahmed , J. Basrah Res. (Sci.) **50**(2), 1 (2025). DOI:https://doi.org/10.56714/bjrs.51.2.1

ABSTRACT

In this article, effective techniques are described to solve numerically the Fredholm integro-differential equations of multi-fractional order that lie in (0,1] in the Caputo sense (FIFDEs). The approach uses finite difference approximation to Caputo derivative utilizing collocation points and is based on the quadrature rule, Trapezoidal, and Simpson process. Our method simplifies the evaluation of treatments by transforming the FIFDEs into algebraic equations with operational matrices. After calculating the Caputo derivative at a specific point using the finite difference method, we use the quadrature method, which includes the trapezoidal and Simpson rules, to create a finite difference formula for our fractional equation. Additionally, numerical examples are provided to demonstrate the validity and use of the approach as well as comparisons with earlier findings. The aforementioned procedure has been used to construct algorithms for treating FIFDEs. A MATLAB program is created to express these solutions. Furthermore, some numerical tests are provided to demonstrate the method's accuracy.

1. Introduction

The Fredholm integro-differential equations (FIFDEs) of multi-fractional orders with variable coefficients that lie in the interval (0,1] in the Caputo sense are the aim of this work by using a suitable finite difference method. Their general form is as follows:

$$\begin{aligned}{}_{a}^{C}D_{t}^{\alpha_n}u(t)+\sum_{i=1}^{n-1}\mathcal{P}_i(t)\,{}_{a}^{C}D_{t}^{\alpha_{n-i}}u(t)+\mathcal{P}_n(t)u(t)\\ =f(t)+\lambda\int_a^b\sum_{j=0}^{m}\mathcal{K}_j(t,s)\,{}_{a}^{C}D_{s}^{\beta_{m-j}}u(s)ds\,,\ a\le t\le b \qquad (1)\end{aligned}$$

subject to the boundary condition:

$$g_{11}u(a)+h_{11}u(b)=C_1 \qquad (2)$$

where g_{11}, h_{11} and $C_1\in\mathbb{R}$. The fractional orders $0=\alpha_0<\alpha_1<\cdots<\alpha_n\le 1$, and $0=\beta_0<\beta_1<\cdots<\beta_m\le 1$ such that $\mu=\max\{\lceil\alpha_n\rceil,\lceil\beta_m\rceil\}=1$. The unknown function to be found in equation (1) is u. Additionally, the functions $f,\mathcal{P}_i\in C([a,b],\mathbb{R})$, $\mathcal{K}_j\in C(\Theta,\mathbb{R})$, and the $\Theta=\{(t,s):a\le t<s\le b\}$ represented the known continuous function for all $i=1,2,\dots,n$ and $j=0,1,\dots,m$. with scalar parameter. Since $\rho=\{\alpha_{n-i},\beta_{m-j}\}\in(0,1]$ for all $i=0,1,\dots,n$ and $j=$

*Corresponding author email: dashne.zahir@koyauniversity.org

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq

$0,1,\ldots,m$, Where n,m are nonnegative integers, the ${}_a^CD_t^\rho$ represents the ρ-Caputo fractional differential operator of the real-valued function $u(t)$ on the closed bounded interval $[a,b]$.

The purpose of this research is to solve Fredholm integro-fractional differential equations (1) with boundary conditions (2) using the quadrature rule. This numerical method for estimating a solution at a point can be used to approximate the value of an unknown function at that given point. The quadrature techniques are the basis of every numerical method for finding solutions of integral parts in functional equations: Abdullo, Samandar, and Bobomurod [1] used quadrature methods to solve the first kind Abel integral equation; Al-Nasir [2] applied it to solve Volterra integral equations of the second kind; and J. Saberi-Nadjafi with M. Heidari [3] used the modified trapezoid quadrature method for solving Fredholm integral equations of the second kind, although S. Rahbar and E. Hashemizadeh [4]. While Emamzadeh and Kajani [5] used the quadrature technique for the second kind of nonlinear Fredholm integral equation. Moreover, S. A. Isaacson and R. M. Kirby [6] applied it to solve singular Volterra types. Furthermore, Saadati with Shakeri [7] and M. Al-jawary [8] are solving linear integro-differential equations applying quadrature techniques. Also, Ahmed with Hamasalih [9] used it to numerically treat the solution of the most general linear Volterra integro-fractional differential equations.

The structure of the paper is as follows: The necessary definitions and fundamental introduction to fractional calculus are provided in Section 2. Section 3 provides a fundamental review of the formulation of Quadrature-Midpoint techniques. Numerical techniques are derived for FIFDEs in detail in Section 4. Additionally, this section's algorithm explains the scheme's primary phases. The numerical results are shown in Section 5, and Section 6 offers the conclusions. The purpose of this study is to use quadrature techniques for Caputo derivative terms that depend on collocation points and convert to an algebraic system using the finite difference approximation. Finally, it evaluates the multi-order linear FIFDEs' approximate solution.

2. Fundamental Definitions of Fractional Derivatives:

This section outlines the fundamental definitions, features, and attributes of fractional derivatives. In addition, a number of basic ideas and lemmas that are used in this study were described.

Definition 1 ([10]). If there is a real number $\rho > \eta$ such that $u(t) = (t-a)^\rho u_0(t)$, with $u_0 \in C[a,b]$, then a real function u, defined on $[a,b]$, belongs to the space $C_\eta[a,b], \eta \in \mathbb{R}$. It is also said to belong to the space $C_\eta^m[a,b]$ if and only if it's m-th derivative$u^{(m)}(t)$ also belongs to $C_\eta[a,b], m \in \mathbb{Z}^+$.

Definition 2 ([11,12]). For a function $u \in C_\eta[a,b], \eta \geq -1$, the left-sided Riemann-Liouville fractional integral of order $\rho > 0$ is defined as

$${}_aJ_t^\rho u(t) = \frac{1}{\Gamma(\rho)} \int_a^t (t-\xi)^{\rho-1} u(\xi) d\xi, \qquad \rho \in \mathbb{R}^+, \qquad a \leq t \leq b$$

Here, $\Gamma(.)$ denotes the gamma function, and for $\rho = 0$, we have the Riemann-Liouville identity operator, ${}_aJ_t^0 u(t) = u(t)$.

Definition 3 ([11,12]). The operator ${}_a^RD_t^\rho u(t)$, for a function $u \in C_{-1}^{\lceil\rho\rceil}[a,b]$ of order $\rho \geq 0$ and $t > a$, defined as

$${}_a^RD_t^\rho u(t) = D_t^{\lceil\rho\rceil} {}_aJ_t^{\lceil\rho\rceil-\rho} u(t)$$

is called the Riemann-Liouville fractional derivative of order ρ. Where $\lceil . \rceil$denotes the ceiling function, and for $\rho = 0$, we have the Riemann-Liouville identity derivative operator, ${}_a^RD_t^0 u(t) = u(t)$.

Definition 4 ([11,12]). The operator ${}_a^CD_t^\rho u(t)$, for a function $u \in C_{-1}^{\lceil\rho\rceil}[a,b]$ of order $\rho \geq 0$ and $t > a$, defined as

$${}_a^CD_t^\rho u(t) = {}_aJ_t^{\lceil\rho\rceil-\rho} D_t^{\lceil\rho\rceil} u(t) = \frac{1}{\Gamma(\lceil\rho\rceil-\rho)} \int_a^t (t-\xi)^{\lceil\rho\rceil-\rho-1} \frac{d^{\lceil\rho\rceil} u(\xi)}{d\xi^{\lceil\rho\rceil}} d\xi$$

is called the Caputo fractional differential operator of order ρ. In the Caputo manner derivative for $\rho = 0$, we have the Caputo identity derivative operator, ${}_a^C D_t^0 u(t) = u(t)$. The following properties hold:

- ${}_aJ_t^{\rho_1}\,{}_aJ_t^{\rho_2}u(t) = {}_aJ_t^{\rho_1+\rho_2}u(t) = {}_aJ_t^{\rho_2}\,{}_aJ_t^{\rho_1}u(t)$ for all $\rho_1, \rho_2 \geq 0$.
- ${}_a^R D_t^\rho \mathcal{A} = \mathcal{A}\,\frac{(t-a)^{-\rho}}{\Gamma(1-\rho)}$ and ${}_a^C D_t^\rho \mathcal{A} = 0$; $\mathcal{A}$ is any constant;($\rho \geq 0, \rho \notin \mathbb{N}$).
- ${}_a^C D_t^\rho\ {}_aJ_t^\rho u(t) = u(t)$, for $\lceil\rho\rceil - 1 < \rho \leq \lceil\rho\rceil,\ a \leq t \leq b$.
- ${}_aJ_t^\rho\,{}_a^C D_t^\rho u(t) = u(t) - \sum_{k=0}^{\lceil\rho\rceil-1}\frac{u^{(k)}(a)}{k!}(t-a)^k$, for $\lceil\rho\rceil - 1 < \rho \leq \lceil\rho\rceil$.
- ${}_a^C D_t^\rho u(t) = {}_a^R D_t^\rho\left[u(t) - T_{\lceil\rho\rceil-1}[u;a]\right]$, and $T_{\lceil\rho\rceil-1}[u;a]$ denotes the Taylor polynomial of degree $\lceil\rho\rceil - 1$ for the function u, centered at a.

Lemma 1. ([13]) If $t > a$ and for $u(t) = (t-a)^\gamma$, for $\gamma > -1$ and $\rho \in \mathbb{R}^+$ then the following statement hold:

$$ {}_aJ_t^\rho u(t) = \frac{\Gamma(\gamma+1)}{\Gamma(\gamma+\rho+1)}(t-a)^{\gamma+\rho}. $$

Lemma 2. ([13]) The function $u(t) = (x-a)^\gamma$, for $\gamma \geq 0$, has a Caputo derivative of order $\rho \geq 0$, which is formed as: For $\gamma \in \{0,1,2,\cdots,\lceil\rho\rceil - 1\}$: ${}_a^C D_t^\rho u(t) = 0$ and for $\gamma \in \mathbb{N}$ and $\gamma \geq \lceil\rho\rceil$ or $\gamma \notin \mathbb{N}$ and $\gamma > \lceil\rho\rceil - 1$:

$$ {}_a^C D_t^\rho u(t) = \frac{\Gamma(\gamma+1)}{\Gamma(\gamma-\rho+1)}(t-a)^{\gamma-\rho}. $$

Lemma 3. ([14]) For fractional order $0 < \rho \leq 1$ at specified points $t = t_{r+1}; r = 0,1,\ldots,N-1$ and $h = (b-a)/N$, the Caputo derivative finite difference approximation is created

$$ {}_a^C D_t^\rho\, u(t_{r+1}) = \frac{h^{-\rho}}{\Gamma(2-\rho)}\sum_{\ell=0}^{r}\left[u(t_{r-\ell+1}) - u(t_{r-\ell})\right] b_\ell^\rho \tag{3} $$

where $b_\ell^\rho = (\ell+1)^{1-\rho} - \ell^{1-\rho}$.

Lemma 4. ([15,16]) Let say $\rho \geq 0$, $\rho \notin \mathbb{N}$. and for $u \in C_{-1}^{\lceil\rho\rceil}[a,b]$. Then $[{}_a^C D_t^\rho u(t)]_{t=a} = 0$, that is $\lim_{t\to a}[{}_a^C D_t^\rho u(t)] = 0$, and the Caputo fractional derivative ${}_a^C D_t^\rho u(t)$ is continuous on $[a,b]$.

3. Quadrature-Rule ([16,17,18,19]):

The weighted sum of a finite number of integrand function sample values is known as the quadrature rule. Consider the real-valued function $g(t)$, which is defined on $[a,b]$. By using $\sum_{j=1}^{N} w_j g(t_j) + R[g]$, we want to calculate the value of the integral $\int_a^b g(t)dt$. With $R[g]$ as the remainder and the quadrature rule $\{w_j, t_j\}_{j=1}^{N}$ may be found in tabular form, where the integration nodes are represented by the real numbers t_j and the quadrature weights, or constants, are represented by w_j, [17,18]. We present here two algorithms for generating the quadrature rule defined by the weight function and number of nodes:

For $[a,b]$, we subdivide the interval from a to b into N-subintervals of size h, $h = \frac{b-a}{N}$; $N \geq 1$ with grid points $x_i = a + ih$ $(i = 0,1,2,\ldots N)$. Then we can write the integration by the Trapezoidal rule as:

$$ \int_a^b f(x)dx = \frac{h}{2}\left[f(a) + 2\sum_{i=1}^{N-1} f(x_i) + f(b)\right] = h\sum_{k=0}^{N} w_k^t f(x_k) \tag{4} $$

Hence w_k^t are weights for the trapezoidal rule, where $w_0^t = w_N^t = \frac{1}{2}$; $w_i^t = 1$; $(0 < i < n)$.

Also, we can write the integration by Simpson's rule as in the following generalization formula:

(N- is even):

$$\int_{a}^{b} f(x)dx = h\sum_{i=1}^{N/2}\sum_{k=0}^{2} w_k^s\, f(x_{2i-k}) \tag{5}$$

(N- is odd):

$$\int_{a}^{b} f(x)dx = h\sum_{i=1}^{(N-1)/2}\sum_{k=0}^{2} w_k^s f(x_{2i-k}) + h\sum_{k=0}^{1} w_k^t f(x_{N-k}) \tag{6}$$

while $w_k^s and\ w_k^t$ are the weights for Simpson's and trapezoidal rules, respectively, where $w_0^s = w_2^s = \frac{1}{3}$, $w_1^s = \frac{4}{3}$ and $w_0^t = w_1^t = \frac{1}{2}$.

4. A Numerical Technique Utilizing the Quadrature-Rule:

This section presents a suitable approach that uses quadrature methods with the aid of the finite difference approximation to treat multi-fractional orders of FIDEs with variable coefficients. Recall equation (1) for $0 < \max_{i,j}\{\alpha_i, \beta_j\} \leq 1$ with strictly decreasing for α_i and β_j for all $i = 0,1,\dots,n; j = 0,1,\dots,m$. Thus, for obtaining an approximation of the solution $u(t)$ in a given set of $(N+1)$-equally spaced grid points $t_r = t_0 + rh, (r = 0,1,\dots,N)$ with $t_0 + Nh = b$, consists of approximating the linear Fredholm IFDEs (1) in the discretized equations:

$$\left[\sum_{i=0}^{n-1} \mathcal{P}_i(t)\, {}_a^C D_t^{\alpha_{n-i}} u(t) + \mathcal{P}_n(t)u(t)\ \right]_{t=t_r}$$
$$= f(t_r) + \lambda \sum_{j=0}^{m-1} \int_a^b \mathcal{K}_j(t_r, s)\, {}_a^C D_s^{\beta_{m-j}} u(s)ds$$
$$+ \lambda \int_a^b \mathcal{K}_m(t_r, s)u(s)ds \tag{7}$$

This leads to a system of $N+1$ linear algebraic equations in $N+1$ unknowns. $\tilde{u}(t_r) = \tilde{u}_r$, which approximates $u(t_r)$. Here, the Fredholm integral part in (7) is approximated by the closed Newton-Cotes formula (Trapezoidal rule and Simpson's rule), and the fractional differential parts are approximated by using forward difference as stated in the following proposition:

4.1 Trapezoidal Method:

By applying (4) rule to evaluate each integral part in equation (7) for each $r = 0,1,\dots,N$ and taking into account the formula (3) with lemma (4), that is, the Caputo-fractional order for any continuous function at the starting point $t = t_0 = a$ equal to zero, then its results are formed in the following classification: First, for $r = 0$, we obtain:

$$P_{n,0}\tilde{u}_0 = f_0 + \lambda h \sum_{j=0}^{m-1} \left\{ \sum_{p=1}^{N-1} \mathcal{K}_{0,p}^j A_m^\beta(j) \left[\sum_{q=0}^{p-1} [\tilde{u}_{p-q} - \tilde{u}_{p-q-1}] b_q^{\beta_{m-j}} \right] \right.$$
$$\left. + \frac{1}{2}\mathcal{K}_{0,N}^j A_m^\beta(j) \left[\sum_{\ell=0}^{N-1} [\tilde{u}_{N-\ell} - \tilde{u}_{N-\ell-1}] b_\ell^{\beta_{m-j}} \right] \right\} + \frac{\lambda h}{2}\mathcal{K}_{0,0}^m \tilde{u}_0 + \lambda h \sum_{p=1}^{N-1} \mathcal{K}_{0,p}^m \tilde{u}_p$$
$$+ \frac{\lambda h}{2}\mathcal{K}_{0,N}^m \tilde{u}_N \tag{8}$$

Assuming that:

$$A_\ell^\sigma(j) = \frac{h^{-\sigma_{\ell-j}}}{\Gamma(2 - \sigma_{\ell-j})} \tag{9}$$

Where σ is the fractional order α or β with $\ell = n$ or m, respectively, for all $j = 0,1,\dots,\ell$. That $A_\ell^\sigma(\ell) = 1$ and $\mathcal{K}_{rp}^\ell = \mathcal{K}_\ell(t_r, s_p)$ for all $\ell, r,$ and $p, \tilde{u}_r$ is the approximate value of $u(t_r)$.

Second, for $r = 1,2,\dots,N$, replace it by $\bar{r} = r-1$ so $\bar{r} = 0,1,\dots,N-1$ also using equation (3) with lemma (4) to equation (7), yields:

$$\sum_{i=0}^{n-1} P_{i,\bar{r}+1} A_n^{\alpha}(i) \left\{ \sum_{\ell=0}^{\bar{r}} [\tilde{u}_{\bar{r}-\ell-1} - \tilde{u}_{\bar{r}-\ell}] b_{\ell}^{\alpha_{n-i}} \right\} + P_{n,\bar{r}+1}\tilde{u}_{\bar{r}+1}$$
$$= f_{\bar{r}+1} + \lambda \sum_{j=0}^{m-1} \left\{ h \sum_{p=1}^{N-1} \mathcal{K}_{\bar{r}+1,p}^{j} \left[A_m^{\beta}(j) \sum_{q=0}^{p-1} [\tilde{u}_{p-q} - \tilde{u}_{p-q-1}] b_q^{\beta_{m-j}} \right] \right.$$
$$\left. + \frac{h}{2} \mathcal{K}_{\bar{r}+1,N}^{j} \left[A_m^{\beta}(j) \sum_{k=0}^{N-1} [\tilde{u}_{N-k} - \tilde{u}_{N-k-1}] b_k^{\beta_{m-j}} \right] \right\} + \frac{\lambda h}{2} \mathcal{K}_{\bar{r}+1,0}^{m} \tilde{u}_0$$
$$+ \lambda h \sum_{p=1}^{N-1} \mathcal{K}_{\bar{r}+1,p}^{m} \tilde{u}_p$$
$$+ \frac{\lambda h}{2} \mathcal{K}_{\bar{r}+1,N}^{m} \tilde{u}_N \tag{10}$$

From the linear algebraic equations (8) and (10), we construct a linear system of equations; this can be written in a matrix form:

$$[L - I]\tilde{U} = F \tag{11}$$

where $L = [L_{k\ell}]_{N+1\times N+1}$ is a lower triangular matrix and defines each element $L_{k\ell}$ for all $k,\ell = \overline{0:N}$ as:

$$\left.\begin{array}{ll} L_{k,\ell} = 0 & \text{for all } k < l \\ L_{k,k} = \mathcal{H}_n^{\alpha}(k) & \text{for each } k = \overline{0:N} \\ L_{k,0} = -\sum_{i=0}^{n-1} \mathcal{P}_{i,k} A_n^{\alpha}(i) b_{k-1}^{\alpha_{n-i}} & \text{for all } k = \overline{1:N} \\ L_{k,\ell} = \sum_{i=0}^{n-1} \mathcal{P}_{i,k} A_n^{\alpha}(i) C_{k-\ell}^{\alpha_{n-i}} & \text{such that all } k > l \\ \text{for each } k = 2,3,\dots,N \text{ and with } \ell = 1,2,\dots,k-1 & \end{array}\right\} \tag{12}$$

while

$$\mathcal{H}_n^{\alpha}(r) = \begin{cases} P_{n,0} & \text{if } r = 0 \\ P_{n,r} + \sum_{i=0}^{n-1} P_{i,r} A_n^{\alpha}(i) & \text{o.w.} \end{cases} \tag{13}$$

and the coefficients b_{ℓ}^{σ} and $C_{\ell}^{\sigma}\,(\ell = \overline{0:N})$ for any real number $\sigma \in (0,1]$, $(\sigma = \alpha$ or $\beta)$ defined as:

$$\left.\begin{array}{c} b_{\ell}^{\sigma} = (1+\ell)^{1-\sigma} - \ell^{1-\sigma} \quad ; \quad b_0^{\sigma} = 1 \\ C_{\ell}^{\sigma} = b_{\ell}^{\sigma} - b_{\ell-1}^{\sigma} \quad ; \quad C_0^{\sigma} = 1 \quad \text{and assume } b_{-i}^{\sigma} = 0, \forall i \end{array}\right\} \tag{14}$$

Moreover, the $I = [I_{s\ell}]_{N+1\times N+1}$ is a square matrix of dimension $N+1$and define each element $I_{s\ell}$ for all $s,\ell = \overline{0:N}$ as:

$$\left.\begin{array}{l} I_{s,0} = \frac{\lambda h}{2}\mathcal{K}_{s,0}^{m} - \lambda h \sum_{j=0}^{m-1} A_m^{\beta}(j) \left[\sum_{d=1}^{N}{}^{*} \mathcal{K}_{s,d}^{j} b_{d-1}^{\beta_{m-j}} \right] \\ I_{s,\ell} = \lambda h \mathcal{K}_{s,\ell}^{m} + \lambda h \sum_{j=0}^{m-1} A_m^{\beta}(j) \left[\sum_{d=\ell}^{N}{}^{*} \mathcal{K}_{s,d}^{j} C_{d-\ell}^{\beta_{m-j}} \right] \quad \ell = \overline{1:N-1} \\ I_{s,N} = \frac{\lambda h}{2} \left[\mathcal{K}_{s,N}^{m} + \sum_{j=0}^{m-1} A_m^{\beta}(j) \mathcal{K}_{s,N}^{j} \right] \end{array}\right\} \tag{15}$$

where the sign $(*)$ denotes that the last term of the summation is multiplied by ½ (half). Furthermore,

$$F = [f_0 \quad f_1 \quad \cdots \quad f_N]^T \quad \text{and} \quad \tilde{U} = [\tilde{u}_0 \quad \tilde{u}_1 \quad \cdots \quad \tilde{u}_N]^T$$

since $f_i = f(t_i)$ and $\tilde{u}_i(i = \overline{0:N})$ is the approximate value of $u_i = u(t_i)$.
Finally, in this technique, a boundary condition of equation (2) is added as a new row in the system (11) can be formed in matrix form, this gives:

$$B\widetilde{U} = C \tag{16}$$

where

$$B = [g_{11} \quad 0 \quad \cdots \quad 0 \quad h_{11}]_{N+1}, \quad \widetilde{U} = [\tilde{u}_0 \quad \tilde{u}_1 \quad \cdots \quad \tilde{u}_N]^T \quad \text{and} \quad C = [C_1]$$

obtaining a new matrix by adding (16) to (11), yields

$$D\widetilde{U} = E \tag{17}$$

where

$$D = \begin{bmatrix} L - I \\ \cdots\cdots \\ B \end{bmatrix}_{(N+\mu+1)\times(N+1)} \quad \text{and} \quad E = \begin{bmatrix} F \\ \cdots \\ C \end{bmatrix}_{(N+\mu+1)\times 1}$$

To determine the approximate column vector $\widetilde{U}$'s in equation (17), store the matrix D and compute $D^T D$ and $D^T E$ then use the LU-factorization procedure to solve $[D^T D]\widetilde{U} = [D^T E]$. Then the approximate solution for all $\tilde{u}_i$ at each point $t_i(i = \overline{0:N})$ is obtained for fractional order linear FIDEs (1).

The Algorithm (AFIFT)

The approximate solution for linear IFDEs of Fredholm type with variable coefficients by using the closed Newton-Cotes formula (Trapezoidal rule) with the aid of finite difference approximation can be summarized by the following stages:

Step 1:

a. Input $N \in \mathbb{Z}^+$, take $h = (b - a)/N$ and $t_r = a + rh$.
b. Input the coefficients of the boundary conditions. g_{11}, h_{11} and C_1.

Step 2: To compute $A_\ell^\sigma(k)$ for each $k = 0,1,\dots,\ell, (\in \mathbb{Z}^+)$ and for all $\sigma = \alpha$ or β and $\ell = n$ or m, respectively, applied equation (9).

Step 3: Using equation (13) and step 2 for all fractional orders $\alpha_n > \alpha_{n-1} > \cdots > \alpha_1 > \alpha_0 = 0$ to evaluate $\mathcal{H}_n^\alpha(r), r \in \mathbb{Z}^+$.

Step 4: For all $\ell = 0,1,\dots,N$ find the constant coefficients (b_ℓ^σ and C_ℓ^σ). For fractional orders $\sigma = \alpha$ and β, respectively, using equation (14).

Step 5: For all $k, \ell = 0,1,\dots,N$ evaluate each element $L_{k,\ell}$ using formulas in equation (12) with steps (2,3, and 4). Finally, construct the lower triangular matrix $L = [L_{k\ell}]_{N+1\times N+1}$.

Step 6: Evaluate the values of kernels at each given point, $\mathcal{K}_{s\ell}^j = \mathcal{K}_j(t_s, t_\ell)$ for all $j = 0,1,\dots,m$ and $s, \ell = 0,1,\dots,N$.

Step 7: For all $s, \ell = 0,1,\dots,N$ calculate each element $I_{s\ell}$ using formulas in equation (15) with steps (4 and 6). Finally, construct the matrix $I = [I_{s\ell}]_{N+1\times N+1}$.

Step 8: Compute all elements of the column vector F at points t_r by $f_r = f(t_r), t_r = a + rh \ (r = 0,1,\dots,N)$.

Step 9: Putting boundary conditions g_{11}, h_{11} and C_1 into matrices B and C to form (16).

Step 10: Construct the matrices D and E, which are represented in the system (17).

Step 11: Apply the LU-factorization method to the system which is obtained in step 10, after multiplying both sides by D^T, to compute the column-approximate values $\widetilde{U}$ of the exact solution U.

4.2 Simpson's Method:

The Simpson's rule is a second case of (unweighted) closed-Newton-Cotes formula, which is the most important rule for evaluating bounded integrals numerically. Here, we use parabolas to approximate each part of the curve. The given integral of integration can be divided into N-subintervals of equal length $h=(b-a)/N$, $N\geq 2$, and points $t_r=t_0+rh\ (r=0,1,\dots,N)$ and $b=t_0+Nh$. If ***N*-is even**, then the numerical integration of $g(t)$ over $[a,b]$ by Simpson's rule can be written as [6,7,9,10]:

$$\int_a^b g(t)dt=\frac{h}{3}\sum_{d=1}^{N/2}[g(t_{2d})+4g(t_{2d-1})+g(t_{2d-2})]$$
$$=h\sum_{d=1}^{N/2}\sum_{\ell=0}^{2}w_\ell^s g(t_{2d-\ell}) \tag{18}$$

If ***N*-is odd**, we formulated Simpson's rule as:

$$\int_a^b g(t)dt=h\sum_{d=1}^{(N-1)/2}\sum_{\ell=0}^{2}w_\ell^s g(t_{2d-\ell})+h\sum_{\ell=0}^{1}w_\ell^t g(t_{N-\ell}) \tag{19}$$

whilew_ℓ^s and w_ℓ^t are the weights for Simpson's rule, where $w_0^s=w_2^s=\frac{1}{3}, w_1^s=\frac{4}{3}$; and $w_0^t=w_1^t=\frac{1}{2}$; also, the set of points $t_r=a+rh\ (r=\overline{0:N})$.

By applying the equations (18) or (19) for the number of sub-intervals even or odd, respectively, to evaluate each integral part in equation (7) with taken formula (3) and the lemma (4), then it results in the following classification:

- For N-is even:
  First for $r=0$, i.e. take $t=t_0=a$ in to equation (7) and using formula (18) with proposition (1), we get:

$$P_{n,0}\tilde{u}_0=f_0+\frac{\lambda h}{3}\sum_{j=0}^{m-1}\sum_{d=1}^{N/2}\Bigg\{\mathcal{K}_{0,2d}^{j}A_m^{\beta}(j)\sum_{\ell=0}^{2d-1}[\tilde{u}_{2d-\ell}-\tilde{u}_{2d-\ell-1}]\,b_\ell^{\beta_{m-j}}$$
$$+4\mathcal{K}_{0,2d-1}^{j}A_m^{\beta}(j)\sum_{\ell=0}^{2d-2}[\tilde{u}_{2d-\ell-1}-\tilde{u}_{2d-\ell-2}]\,b_\ell^{\beta_{m-j}}\Bigg\}$$
$$+\frac{\lambda h}{3}\sum_{j=0}^{m-1}\sum_{d=2}^{N/2}\mathcal{K}_{0,2d-2}^{j}A_m^{\beta}(j)\sum_{\ell=0}^{2d-3}[\tilde{u}_{2d-\ell-2}-\tilde{u}_{2d-\ell-3}]\,b_\ell^{\beta_{m-j}}$$
$$+\frac{\lambda h}{3}\sum_{d=1}^{N/2}\left\{\mathcal{K}_{0,2d}^{m}\tilde{u}_{2d}+4\mathcal{K}_{0,2d-1}^{m}\tilde{u}_{2d-1}+\mathcal{K}_{0,2d-2}^{m}\tilde{u}_{2d-2}\right\} \tag{20}$$

In the next step for $r=1,2,\dots,N$ replace it by$\bar{r}=r-1$ so $\bar{r}=0,1,\dots,N-1$ also take $t=t_{\bar{r}+1}$ into equation (7) and also using formula (18) for integral parts and proposition (1), we obtain:

$$\sum_{i=0}^{n-1}P_{i,\bar{r}+1}\,A_n^{\alpha}(i)\sum_{\ell=0}^{\bar{r}}[\tilde{u}_{\bar{r}-\ell+1}-\tilde{u}_{\bar{r}-\ell}]b_\ell^{\alpha_{n-i}}+P_{n,\bar{r}+1}\tilde{u}_{\bar{r}+1}=f_{\bar{r}+1}$$
$$+\frac{\lambda h}{3}\sum_{j=0}^{m-1}\sum_{d=1}^{N/2}\Bigg\{\mathcal{K}_{\bar{r}+1,2d}^{j}A_m^{\beta}(j)\sum_{\ell=0}^{2d-1}[\tilde{u}_{2d-\ell}-\tilde{u}_{2d-\ell-1}]\,b_\ell^{\beta_{m-j}}$$
$$+4\mathcal{K}_{\bar{r}+1,2d-1}^{j}A_m^{\beta}(j)\sum_{\ell=0}^{2d-2}[\tilde{u}_{2d-\ell-1}-\tilde{u}_{2d-\ell-2}]\,b_\ell^{\beta_{m-j}}\Bigg\}$$
$$+\frac{\lambda h}{3}\sum_{j=0}^{m-1}\sum_{d=2}^{N/2}\mathcal{K}_{\bar{r}+1,2d-2}^{j}A_m^{\beta}(j)\sum_{\ell=0}^{2d-3}[\tilde{u}_{2d-\ell-2}-\tilde{u}_{2d-\ell-3}]\,b_\ell^{\beta_{m-j}}$$
$$+\frac{\lambda h}{3}\sum_{d=1}^{N/2}\Big\{\mathcal{K}_{\bar{r}+1,2d}^{m}\tilde{u}_{2d}+4\mathcal{K}_{\bar{r}+1,2d-1}^{m}\tilde{u}_{2d-1}$$
$$+\mathcal{K}_{\bar{r}+1,2d-2}^{m}\tilde{u}_{2d-2}\Big\} \tag{21}$$

were $A_\ell^\sigma(k)$ for fractional orders $\sigma = \alpha$ or β and $\ell = n$ or m respectively for all $k = \overline{0:\ell}(\ell \in \mathbb{Z}^+)$ are defined in equation (9) and $\mathcal{K}_{rp}^j = \mathcal{K}_j(t_r, s_p)$ all kernel values for all $r, p = \overline{0:N}$ and $j = \overline{0:m}$.

After some simple manipulation of linear algebraic equations (19) and (21), we construct a linear system of equations that can be written in matrix form:

$$\left[L - \frac{\lambda h}{3} I\right] \tilde{U} = F \tag{22}$$

where $L = [L_{k\ell}]_{N+1\times N+1}$ is a lower triangular matrix and defines each element $L_{k\ell}$ for all $k, \ell = 0,1,\dots,N$ in the equations (12, 13, and 14). Moreover, the $I = [I_{s\ell}]_{N+1\times N+1}$ is a square matrix of dimension $N + 1$and define each element $I_{s\ell}$ for all $s, \ell = 0,1,\dots,N$ as:

$$\left.\begin{aligned} I_{s,0} &= \mathcal{K}_{s,0}^m - \sum_{j=0}^{m-1} A_m^\beta(j) \left[\sum_{d=1}^{N} w_d \mathcal{K}_{s,d}^j b_{d-1}^{\beta_{m-j}}\right] \\ I_{s,\ell} &= w_\ell \mathcal{K}_{s,\ell}^m + \sum_{j=0}^{m-1} A_m^\beta(j) \left[\sum_{d=\ell}^{N} w_d \mathcal{K}_{s,d}^j C_{d-\ell}^{\beta_{m-j}}\right] \quad \ell = \overline{1:N-1} \\ I_{s,N} &= \mathcal{K}_{s,N}^m + \sum_{j=0}^{m-1} A_m^\beta(j) \mathcal{K}_{s,N}^j \end{aligned}\right\} \tag{23}$$

with

$$w_d = \begin{cases} 1 & \text{if } d = N \\ 2 & \text{if } d \neq N \text{ and } d \text{ is even} \\ 4 & \text{if } d \neq N \text{ and } d \text{ is odd} \end{cases}$$

Furthermore,

$$F = [f_0 \quad f_1 \quad \cdots \quad f_N]^T \quad \text{and} \quad \tilde{U} = [\tilde{u}_0 \quad \tilde{u}_1 \quad \cdots \quad \tilde{u}_N]^T$$

Since $f_r = f(t_r)$ and $\tilde{u}_r(r = \overline{0:N})$ is the approximate value of $u_r = u(t_r)$.

- For N-is odd: First, for $r = 0$, using the formula (19) and applying the proposition (1) into equation (7) after putting $t = t_0 = a$, we obtain:

$$\begin{aligned} P_{n,0}\tilde{u}_0 = f_0 &+ \frac{\lambda h}{3} \sum_{j=0}^{m-1} A_m^\beta(j) \sum_{d=1}^{(N-1)/2} \left\{ \mathcal{K}_{0,2d}^j \sum_{\ell=0}^{2d-1} [\tilde{u}_{2d-\ell} - \tilde{u}_{2d-\ell-1}]\, b_\ell^{\beta_{m-j}} \right. \\ &\left. + 4\mathcal{K}_{0,2d-1}^j \sum_{\ell=0}^{2d-2} [\tilde{u}_{2d-\ell-1} - \tilde{u}_{2d-\ell-2}]\, b_\ell^{\beta_{m-j}} \right\} \\ &+ \frac{\lambda h}{3} \sum_{j=0}^{m-1} A_m^\beta(j) \sum_{d=2}^{(N-1)/2} \left\{ \mathcal{K}_{0,2d-2}^j \sum_{\ell=0}^{2d-3} [\tilde{u}_{2d-\ell-2} - \tilde{u}_{2d-\ell-3}]\, b_\ell^{\beta_{m-j}} \right\} \\ &+ \frac{\lambda h}{2} \sum_{j=0}^{m-1} A_m^\beta(j) \left\{ \mathcal{K}_{0,N}^j \sum_{\ell=0}^{N-1} [\tilde{u}_{N-\ell} - \tilde{u}_{N-\ell-1}] b_\ell^{\beta_{m-j}} \right. \\ &\left. + \mathcal{K}_{0,N-1}^j \sum_{\ell=0}^{N-2} [\tilde{u}_{N-\ell-1} - \tilde{u}_{N-\ell-2}] b_\ell^{\beta_{m-j}} \right\} \\ &+ \frac{\lambda h}{3} \sum_{d=1}^{(N-1)/2} \left\{ \mathcal{K}_{0,2d}^m \tilde{u}_{2d} + 4\mathcal{K}_{0,2d-1}^m \tilde{u}_{2d-1} + \mathcal{K}_{0,2d-2}^m \tilde{u}_{2d-2} \right\} \\ &+ \frac{\lambda h}{2} \left[\mathcal{K}_{0,N}^m \tilde{u}_N \right. \\ &\left. + \mathcal{K}_{0,N-1}^m \tilde{u}_{N-1} \right] \end{aligned} \tag{24}$$

for $r = 1,2,\dots,N$, putting $\bar{r} = r - 1$. So, $\bar{r} = 0,1,\dots,N-1$ and put $t = t_{\bar{r}+1}$ into equation (2) and using equation (19) for integral terms with using proposition (1), to obtain:

$$\sum_{i=0}^{n-1} P_{i,\bar{r}+1}\, A_n^{\alpha}(i) \sum_{\ell=0}^{\bar{r}} [\tilde{u}_{\bar{r}-\ell+1} - \tilde{u}_{\bar{r}-\ell}] b_\ell^{\alpha_{n-i}} + P_{n,\bar{r}+1}\tilde{u}_{\bar{r}+1}$$
$$= f_{\bar{r}+1}$$
$$+ \frac{\lambda h}{3} \sum_{j=0}^{m-1} A_m^{\beta}(j) \sum_{d=1}^{(N-1)/2} \left\{ \mathcal{K}_{\bar{r}+1,2d}^{j} \sum_{\ell=0}^{2d-1} [\tilde{u}_{2d-\ell} - \tilde{u}_{2d-\ell-1}] b_\ell^{\beta_{m-j}} \right.$$
$$\left. + 4\mathcal{K}_{\bar{r}+1,2d-1}^{j} \sum_{\ell=0}^{2d-2} [\tilde{u}_{2d-\ell-1} - \tilde{u}_{2d-\ell-2}] b_\ell^{\beta_{m-j}} \right\}$$
$$+ \frac{\lambda h}{3} \sum_{j=0}^{m-1} A_m^{\beta}(j) \sum_{d=2}^{(N-1)/2} \mathcal{K}_{\bar{r}+1,2d-2}^{j} \sum_{\ell=0}^{2d-3} [\tilde{u}_{2d-\ell-2} - \tilde{u}_{2d-\ell-3}] b_\ell^{\beta_{m-j}}$$
$$+ \frac{\lambda h}{2} \sum_{j=0}^{m-1} A_m^{\beta}(j) \left\{ \mathcal{K}_{\bar{r}+1,N}^{j} \sum_{\ell=0}^{N-1} [\tilde{u}_{N-\ell} - \tilde{u}_{N-\ell}] b_\ell^{\beta_{m-j}} \right.$$
$$\left. + \mathcal{K}_{\bar{r}+1,N-1}^{j} \sum_{\ell=0}^{N-2} [\tilde{u}_{N-\ell-1} - \tilde{u}_{N-\ell-2}] b_\ell^{\beta_{m-j}} \right\}$$
$$+ \frac{\lambda h}{3} \sum_{d=1}^{(N-1)/2} \left\{ \mathcal{K}_{\bar{r}+1,2d}^{m} \tilde{u}_{2d} + 4\mathcal{K}_{\bar{r}+1,2d-1}^{m} \tilde{u}_{2d-1} + \mathcal{K}_{r+1,2d-2}^{m} \tilde{u}_{2d-2} \right\}$$
$$+ \frac{\lambda h}{2} \left[\mathcal{K}_{\bar{r}+1,N}^{m} \tilde{u}_N + \mathcal{K}_{\bar{r}+1,N-1}^{m} \tilde{u}_{N-1} \right] \quad (25)$$

where $A_\ell^{\sigma}(k)$ for fractional order $\sigma = \alpha$ or β and $\ell = n$ or m, respectively, for all $k = \overline{0{:}\,\ell}(\ell \in \mathbb{Z}^{+})$ are defined in equation (9) and $\mathcal{K}_{rp}^{j} = \mathcal{K}_j(t_r, s_p)$ all kernel values for each $r, p = \overline{0{:}\,N}$ and $j = \overline{0{:}\,m}$.

From Linear algebraic equations (24) and (25), construct a linear system of equations which can be written in matrix form:

$$\left[L - \frac{\lambda h}{3} I \right] \tilde{U} = F \quad (26)$$

where $L = [L_{k\ell}]_{N+1\times N+1}$ is a lower triangular matrix and defines each element $L_{k\ell}$ for all $k, \ell = 0,1,\dots,N$ in the equations (12-14). Moreover, the matrix $I = [I_{s\ell}]_{N+1\times N+1}$ is a square dimension and defines each element $I_{s\ell}$ for all $s, \ell = \overline{0{:}\,N}$ as:

$$\left. \begin{aligned} I_{s,0} &= \mathcal{K}_{s,0}^{m} - \sum_{j=0}^{m-1} A_m^{\beta}(j) \left[\sum_{d=1}^{N} w_d \mathcal{K}_{s,d}^{j} b_{d-1}^{\beta_{m-j}} \right] \\ I_{s,\ell} &= w_\ell \mathcal{K}_{s,\ell}^{m} + \sum_{j=0}^{m-1} A_m^{\beta}(j) \left[\sum_{d=\ell}^{N} w_d \mathcal{K}_{s,d}^{j} C_{d-\ell}^{\beta_{m-j}} \right] \quad \ell = \overline{1{:}\,N-2} \\ I_{s,N-1} &= w_{N-1} \mathcal{K}_{s,N-1}^{m} + \sum_{j=0}^{m-1} A_m^{\beta}(j) \left[\sum_{d=N-1}^{N} w_d\, \mathcal{K}_{s,d}^{j} C_{d-(N-1)}^{\beta_{m-j}} \right] \\ I_{s,N} &= w_N \mathcal{K}_{s,N}^{m} + \sum_{j=0}^{m-1} A_m^{\beta}(j) \left[w_N \mathcal{K}_{s,N}^{j} \right] \end{aligned} \right\} \quad (27)$$

with

$$w_d = \begin{cases} 3/2 & d = N \\ 5/2 & d = N-1 \\ 4 & d \neq N, N-1, \text{and } d \text{ is odd} \\ 2 & d \neq N, N-1, \text{and } d \text{ is even} \end{cases}$$

Furthermore: $F = [f_0 \quad f_1 \quad \cdots \quad f_N]^T$ and $\widetilde{U} = [\tilde{u}_0 \quad \tilde{u}_1 \quad \cdots \quad \tilde{u}_N]^T$ .More, $f_i = f(t_i)$ and $\tilde{u}_i (i = \overline{0:N})$are the approximate values of $u_i = u(t_i)$.
Finally, from using the boundary equation in matrix form(16) and obtaining a new matrix by adding (16) to (22) or (26) for different values of N yields:

$$D\widetilde{U} = E \tag{28}$$

where $D = \begin{bmatrix} L - \dfrac{\lambda h}{3} I \\ \cdots\cdots\cdots \\ B \end{bmatrix}_{(N+\mu+1)\times(N+1)}$ and $E = \begin{bmatrix} F \\ \cdots \\ C \end{bmatrix}_{(N+\mu+1)\times 1}$

To determine the approximate column vector $\widetilde{U}$'s, store the matrix D and compute $D^T D$ and $D^T E$ then use the LU-factorization procedure to solve $[D^T D]\widetilde{U} = [D^T E]$. The approximate solution for all $\tilde{u}_i$ at each point $t_i (i = \overline{0:N})$ is obtained for fractional order linear FIDEs (1).

The Algorithm (AFIFS)

The approximate solution for linear IFDEs of Fredholm type with variable coefficients by using the closed Newton-Cotes formula (Simpson's 1/3 h rule) with the aid of finite difference approximation can be summarized by the following steps:

Step 1:

a. Input $N \in \mathbb{Z}^+$, take $h = (b - a)/N$ and $t_r = a + rh$.
b. Input the coefficients of the boundary conditions. g_{11}, h_{11} and C_1.

Step 2: To compute $A_\ell^\sigma(k)$ for each $k = 0,1, \ldots, \ell, (\in \mathbb{Z}^+)$ and for all $\sigma = \alpha$ or β and $\ell = n$ or m, respectively, applied equation (9).

Step 3: Using the equation (13) and step 2 for all fractional orders $\alpha_n > \alpha_{n-1} > \cdots > \alpha_1 > \alpha_0 = 0$ to evaluate $\mathcal{H}_n^\alpha(r), r \in \mathbb{Z}^+$.

Step 4: For all $\ell = 0,1, \ldots, N$ find the constant coefficients (b_ℓ^σ and C_ℓ^σ) for fractional orders $\sigma = \alpha$ and β, respectively, using equation (14).

Step 5: For all $k, \ell = 0,1, \ldots, N$ evaluate each element $L_{k,\ell}$ using formulas in equations (12) with steps (2,3, and 4). Finally, construct the lower triangular matrix $L = [L_{k\ell}]_{N+1\times N+1}$.

Step 6: Evaluate the values of kernels at each given point, $\mathcal{K}_{s\ell}^j = \mathcal{K}_j(t_s, t_\ell)$ for all $j = 0,1, \ldots, m$ and $s, \ell = 0,1, \ldots, N$.

Step 7: For all $s, \ell = 0,1, \ldots, N$evaluate each element $I_{s\ell}$ using formulas in equations (23) for N-is even and formulas in the equation (27) for N-is odd with steps (4 and 6). Finally, construct the matrix $I = [I_{s\ell}]_{N+1\times N+1}$.

Step 8: Compute all elements of the column vector F at points t_r by $f_r = f(t_r), t_r = a + rh \ (r = 0,1, \ldots, N)$.

Step 9: Putting boundary conditions g_{11}, h_{11} and C_1 into matrices B and C to form (16).

Step 10: Construct the matrices D and E represented in the system (28).

Step 11: Apply the LU-factorization method for the system, which is obtained in step 10, after multiplying both sides by D^T, to compute the column-approximate values $\widetilde{U}$ of the exact solution U.

5. Numerical Performance:

The numerical section uses the L_2 error norm to verify the correctness and efficacy of the proposed schemes. The suggested algorithm AFIFM produces numerical results that are compared, and MATLAB is used to generate both the numerical and graphical results.

Test example 1. Consider a higher-order linear IFDE of Fredholm type with variable coefficients for a fractional order that lies in $(0,1]$:

$$
\begin{aligned}
&{}_{0}^{C}D_{t}^{0.7}u(t) + \sinh(t)u(t) \\
&= \frac{6}{\Gamma(2.3)}t^{1.3} + \sinh(t)(3t^2+2) - \frac{6e^t}{4.2\Gamma(2.2)} - \frac{6}{3.5\Gamma(2.5)}t^2 + \frac{6}{\Gamma(3.5)} - 5e^{t+1} \\
&+ 8e^t + \int_0^1 \left[(s^2e^t)\,{}_{0}^{C}D_{s}^{0.8}u(s) + (st^2-1)\,{}_{0}^{C}D_{s}^{0.5}u(s) + (e^{s+t})u(s)\right]ds
\end{aligned}
$$

subjected to the boundary conditions: $u(0)+u(1)=7$, while the exact solution is: $u(t)=3t^2+2$.

Take $N=10$ and $t_r=t_0+rh$, $(r=\overline{0:N})$. Since $(n,m)=(1,2)$ and the fractional orders are $\alpha_1=0.7$, $\alpha_0=0$ and $\beta_2=0.8$, $\beta_1=0.5$, $\beta_0=0$ with boundary coefficients $g_{11}=h_{11}=1$ and $C_1=7$ by running the programs Main N-CTrap and Main N-CSimp, the following was obtained:

$$A_1^{\alpha}(0)=5.5844412044 \quad A_1^{\alpha}(1)=1$$

$$A_2^{\beta}(0)=6.8719105251 \quad A_2^{\beta}(1)=3.5682482323 \quad A_2^{\beta}(2)=1$$

Table 1. Contain the value of $\mathcal{H}_2^{\alpha}(r)$ for each $t_r(r=0,1,\dots,10)$ with $\mathcal{H}_2^{\alpha}(0)=0$

t_r	0.1	0.2	0.3	0.4	0.5
$\mathcal{H}_2^{\alpha}(r)$	5.6846079545	5.7857772070	5.888961497	5.9951935302	6.1055365099
t_r	0.6	0.7	0.8	0.9	1.0
$\mathcal{H}_2^{\alpha}(r)$	6.2210947866	6.3430249063	6.472547186	6.6109579301	6.7596423981

Table 2. Contain all values of b_{ℓ}^{σ} for fractions $\sigma=\alpha$ and β for all $\ell=0,1,\dots,10$

orders	α**-fractional**		β**-fractional**		
ℓ	$b_{\ell}^{\alpha_0}$	$b_{\ell}^{\alpha_1}$	$b_{\ell}^{\beta_0}$	$b_{\ell}^{\beta_1}$	$b_{\ell}^{\beta_2}$
1	1	1	1	1	1
2	1	0.2311444133	1	0.4142135623	0.1486983549
3	1	0.1592447569	1	0.3178372451	0.0970325846
4	1	0.1253273961	1	0.2679491924	0.0737769711
5	1	0.1049400301	1	0.2360679774	0.0602217506
6	1	0.0911132627	1	0.2134217652	0.0512394196
7	1	0.0810201031	1	0.1962615682	0.0448040804
8	1	0.0732760205	1	0.1826758136	0.0399434049
9	1	0.0671160618	1	0.1715728752	0.0361290074
10	1	0.0620802700	1	0.1622776601	0.0330476185

Table 3. Contain all values of C_{ℓ}^{σ} for fractions $\sigma=\alpha$ and β for all $\ell=0,1,\dots,10$

orders	α**-fractional**		β**-fractional**		
ℓ	$C_{\ell}^{\alpha_0}$	$C_{\ell}^{\alpha_1}$	$C_{\ell}^{\beta_0}$	$C_{\ell}^{\beta_1}$	$C_{\ell}^{\beta_2}$
1	1	1	1	1	1
2	0	−0.7688555866	0	−0.5857864376	−0.8513016450
3	0	−0.0718996563	0	−0.0963763171	−0.0516657703
4	0	−0.0339173607	0	−0.0498880527	−0.0232556134
5	0	−0.0203873660	0	−0.0318812149	−0.0135552204
6	0	−0.0138267674	0	−0.0226462122	−0.0089823310
7	0	−0.0100931596	0	−0.0171601970	−0.0064353391
8	0	−0.0077440825	0	−0.0135857545	−0.0048606755
9	0	−0.0061599586	0	−0.0111029384	−0.0038143975
10	0	−0.0050357918	0	−0.0092952150	−0.0030813888

The matrices L and I in the methods (Trapezoidal and Simpson), which are formed as in equations (3.9 for L-matrix) and (3.12, 3.20, 3.24, and 3.34 for I-matrix), running programs to obtain:

$$\boldsymbol{L}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -5.5844 & 5.6846 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.2908 & -4.2936 & 5.7857 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.8892 & -0.4015 & -4.2936 & 5.8889 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.6998 & -0.1894 & -0.4015 & -4.2936 & 5.9951 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.5860 & -0.1138 & -0.1894 & -0.4015 & -4.2936 & 6.1055 & 0 & 0 & 0 & 0 & 0 \\ -0.5088 & -0.0772 & -0.1138 & -0.1894 & -0.4015 & -4.2936 & 6.2210 & 0 & 0 & 0 & 0 \\ -0.4524 & -0.0563 & -0.0772 & -0.1138 & -0.1894 & -0.4015 & -4.2936 & 6.3430 & 0 & 0 & 0 \\ -0.4092 & -0.0432 & -0.0563 & -0.0772 & -0.1138 & -0.1894 & -0.4015 & -4.2936 & 6.4725 & 0 & 0 \\ -0.3748 & -0.0343 & -0.0432 & -0.0563 & -0.0772 & -0.1138 & -0.1894 & -0.4015 & -4.2936 & 6.6109 & 0 \\ -0.3466 & -0.0281 & -0.0343 & -0.0432 & -0.0563 & -0.0772 & -0.1138 & -0.1894 & -0.4015 & -4.2936 & 6.7596 \end{bmatrix}_{11\times 11}$$

$$\boldsymbol{I_{Trap}}=\begin{bmatrix} 1.0372 & 0.0166 & 0.0108 & 0.0072 & 0.006 & 0.0076 & 0.0126 & 0.0224 & 0.0401 & 0.2577 & 0.3010 \\ 1.0265 & 0.0241 & 0.0182 & 0.0148 & 0.0141 & 0.0168 & 0.0236 & 0.0362 & 0.0589 & 0.3135 & 0.3533 \\ 1.0070 & 0.0316 & 0.0257 & 0.0227 & 0.0229 & 0.0270 & 0.0361 & 0.0522 & 0.0811 & 0.3793 & 0.4143 \\ 0.9784 & 0.0390 & 0.0333 & 0.0309 & 0.0323 & 0.0383 & 0.0502 & 0.0706 & 0.1070 & 0.4557 & 0.4849 \\ 0.9409 & 0.0465 & 0.0412 & 0.0397 & 0.0426 & 0.0508 & 0.0661 & 0.0917 & 0.1369 & 0.5433 & 0.5654 \\ 0.8942 & 0.0541 & 0.0494 & 0.0490 & 0.0537 & 0.0646 & 0.0839 & 0.1154 & 0.1709 & 0.6428 & 0.6567 \\ 0.8382 & 0.0621 & 0.0581 & 0.0591 & 0.0658 & 0.0799 & 0.1037 & 0.1421 & 0.2093 & 0.7551 & 0.7595 \\ 0.7729 & 0.0706 & 0.0674 & 0.0699 & 0.0791 & 0.0968 & 0.1258 & 0.1720 & 0.2525 & 0.8810 & 0.8746 \\ 0.6982 & 0.0796 & 0.0775 & 0.0817 & 0.0937 & 0.1154 & 0.1503 & 0.2053 & 0.3006 & 1.0215 & 1.0029 \\ 0.6138 & 0.0893 & 0.0884 & 0.0947 & 0.1098 & 0.1361 & 0.1775 & 0.2423 & 0.3542 & 1.1777 & 1.1455 \\ 0.5197 & 0.1 & 0.1004 & 0.1090 & 0.1276 & 0.1589 & 0.2075 & 0.2832 & 0.4136 & 1.3508 & 1.3034 \end{bmatrix}_{11\times 11}$$

$$\boldsymbol{I_{Simp}}=\begin{bmatrix} 33.2904 & -3.5352 & 3.7650 & -2.3579 & 1.6365 & 0.1656 & -1.2014 & 4.1867 & -4.4938 & 14.0705 & 6.0219 \\ 32.9114 & -3.1558 & 3.7657 & -1.8123 & 1.4405 & 1.0349 & -1.6445 & 5.5774 & -5.1529 & 16.9392 & 7.0662 \\ 32.2550 & -2.7500 & 3.7213 & -1.1906 & 1.1715 & 2.0504 & -2.1887 & 7.2135 & -5.9223 & 20.3133 & 8.2879 \\ 31.3186 & -2.3135 & 3.6320 & -0.4868 & 0.8274 & 3.2214 & -2.8385 & 9.1097 & -6.8090 & 24.2235 & 9.6982 \\ 30.0991 & -1.8416 & 3.4982 & 0.3055 & 0.4063 & 4.5581 & -3.5991 & 11.2825 & -7.8208 & 28.7034 & 11.3095 \\ 28.5929 & -1.3292 & 3.3200 & 1.1938 & -0.0940 & 6.0720 & -4.4760 & 13.7498 & -8.9663 & 33.7905 & 13.1353 \\ 26.7963 & -0.7703 & 3.0980 & 2.1860 & -0.6762 & 7.7755 & -5.4754 & 16.5318 & -10.2550 & 39.5258 & 15.1907 \\ 24.7051 & -0.1587 & 2.8326 & 3.2909 & -1.3430 & 9.6825 & -6.6042 & 19.6505 & -11.6974 & 45.9551 & 17.4924 \\ 22.3146 & 0.5127 & 2.5241 & 4.5182 & -2.0974 & 11.8084 & -7.8698 & 23.1303 & -13.3053 & 53.1286 & 20.0587 \\ 19.6199 & 1.2517 & 2.1731 & 5.8788 & -2.9430 & 14.1700 & -9.2808 & 26.9981 & -15.0913 & 61.1019 & 22.9100 \\ 16.6152 & 2.0670 & 1.7801 & 7.3847 & -3.8833 & 16.7862 & -10.8462 & 31.2838 & -17.0698 & 69.9366 & 26.0688 \end{bmatrix}_{11\times 11}$$

From the boundary condition equation, the matrix form is computed as:

$$[B;C]=[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1;7]$$

Substituting the above matrices for the fundamental equation, the augmented matrix is obtained based on the condition, which is:

$$[\boldsymbol{D};\boldsymbol{E}]_{\boldsymbol{Trap}}=\begin{bmatrix} -1.0372 & -0.0166 & -0.0108 & -0.0072 & -0.0060 & -0.0076 & -0.0126 & -0.0224 & -0.0401 & -0.2577 & -0.3010 & ; -5.0825 \\ -6.6110 & 5.6604 & -0.0182 & -0.0148 & -0.0141 & -0.0168 & -0.0236 & -0.0362 & -0.0589 & -0.3135 & -0.3533 & ; -5.3588 \\ -2.2978 & -4.3252 & 5.7600 & -0.0227 & -0.0229 & -0.0270 & -0.0361 & -0.0522 & -0.0811 & -0.3793 & -0.4143 & ; -5.5977 \\ -1.8677 & -0.4405 & -4.3270 & 5.8579 & -0.0323 & -0.0383 & -0.0502 & -0.0706 & -0.1070 & -0.4557 & -0.4849 & ; -5.8421 \\ -1.6408 & -0.2359 & -0.4427 & -4.3333 & 5.9525 & -0.0508 & -0.0661 & -0.0917 & -0.1369 & -0.5433 & -0.5654 & ; -6.0952 \\ -1.4802 & -0.1680 & -0.2389 & -0.4505 & -4.3473 & 6.0408 & -0.0839 & -0.1154 & -0.1709 & -0.6428 & -0.6567 & ; -6.3517 \\ -1.3470 & -0.1394 & -0.1720 & -0.2485 & -0.4673 & -4.3735 & 6.1173 & -0.1421 & -0.2093 & -0.7551 & -0.7595 & ; -6.6014 \\ -1.2254 & -0.1269 & -0.1447 & -0.1838 & -0.2685 & -0.4983 & -4.4194 & 6.1709 & -0.2525 & -0.8810 & -0.8746 & ; -6.8303 \\ -1.1074 & -0.1228 & -0.1338 & -0.1590 & -0.2076 & -0.3049 & -0.5518 & -4.4989 & 6.1718 & -1.0215 & -1.0029 & ; -7.0203 \\ -0.9886 & -0.1237 & -0.1316 & -0.1511 & -0.1870 & -0.2499 & -0.3669 & -0.6438 & -4.6479 & 5.4332 & -1.1455 & ; -7.1490 \\ -0.8664 & -0.1282 & -0.1348 & -0.1522 & -0.1839 & -0.2361 & -0.3214 & -0.4726 & -0.8151 & -5.6444 & 5.4562 & ; -7.1889 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & ; 7 \end{bmatrix}$$

$$[\boldsymbol{D};\boldsymbol{E}]_{\boldsymbol{Simp}}=\begin{bmatrix} -1.1096 & 0.1178 & -0.1255 & 0.0785 & -0.0545 & -0.0055 & 0.0400 & -0.1395 & 0.1497 & -0.4690 & -0.2007 & ; -5.0825 \\ -6.6814 & 5.7898 & -0.1255 & 0.0604 & -0.0480 & -0.0344 & 0.0548 & -0.1859 & 0.1717 & -0.5646 & -0.2355 & ; -5.3588 \\ -2.3659 & -4.2019 & 5.6617 & 0.0396 & -0.0390 & -0.0683 & 0.0729 & -0.2404 & 0.1974 & -0.6771 & -0.2762 & ; -5.5977 \\ -1.9332 & -0.3244 & -4.4146 & 5.9051 & -0.0275 & -0.1073 & 0.0946 & -0.3036 & 0.2269 & -0.8074 & -0.3232 & ; -5.8421 \\ -1.7031 & -0.1280 & -0.5181 & -4.3038 & 5.9816 & -0.1519 & 0.1199 & -0.3760 & 0.2606 & -0.9567 & -0.3769 & ; -6.0952 \\ -1.5391 & -0.0695 & -0.3000 & -0.4413 & -4.2904 & 5.9031 & 0.1492 & -0.4583 & 0.2988 & -1.1263 & -0.4378 & ; -6.3517 \\ -1.4020 & -0.0515 & -0.2171 & -0.2622 & -0.3789 & -4.5528 & 6.4036 & -0.5510 & 0.3418 & -1.3175 & -0.5063 & ; -6.6014 \\ -1.2759 & -0.0510 & -0.1716 & -0.2235 & -0.1446 & -0.7242 & -4.0734 & 5.6880 & 0.3899 & -1.5318 & -0.5830 & ; -6.8303 \\ -1.1530 & -0.0603 & -0.1405 & -0.2278 & -0.0439 & -0.5830 & -0.1391 & -5.0646 & 6.9160 & -1.7709 & -0.6686 & ; -7.0203 \\ -1.0288 & -0.0761 & -0.1156 & -0.2523 & 0.0208 & -0.5861 & 0.1199 & -1.3014 & -3.7905 & 4.5742 & -0.7636 & ; -7.1490 \\ -0.9005 & -0.0970 & -0.0937 & -0.2894 & 0.0730 & -0.6367 & 0.2476 & -1.2322 & 0.1674 & -6.6248 & 5.8906 & ; -7.1889 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & ; 7 \end{bmatrix}$$

solving the three systems above, by a procedure that $[D^T D; D^T E]$, the approximate solutions $\tilde{u}(t)$ are obtained. Table 4 shows a comparison between the exact solution $u(t)$ and approximate solutions $\tilde{u}(t)$ for both methods, depending on the least square error and running time.

Table 4. Numerical results for different values of t and comparison between methods

t_r	Exact Solution	Approximate Solution and Absolute Errors			
		Trapezoidal Method	**Trap. Absolute Error**	**Simpson Method**	**Simp. Absolute Error**
0	2	1.9247264228	0.075273577	1.9695561787	0.030443821
0.1	2.03	1.9709652084	0.059034792	2.0125419097	0.01745809
0.2	2.12	2.0763138666	0.043686133	2.1131947559	0.006805244
0.3	2.27	2.2416391970	0.028360803	2.2722603624	0.0022603624
0.4	2.48	2.4671462581	0.012853742	2.4902016222	0.010201622
0.5	2.75	2.7528411826	0.0028411827	2.7670192724	0.017019272
0.6	3.08	3.0986472401	0.01864724	3.1029149354	0.022914935
0.7	3.47	3.5044460741	0.034446074	3.4978462512	0.027846251
0.8	3.92	3.9700966058	0.050096606	3.9519418411	0.031941841
0.9	4.43	4.4954462045	0.065446205	4.4652026508	0.035202651
1	5	5.0803238017	0.080323802	5.0376880152	0.037688015
$L.S.E.$		2.681636×10^{-02}		6.657162×10^{-03}	
$R.Time/Sec$		0.773492		6.73552	

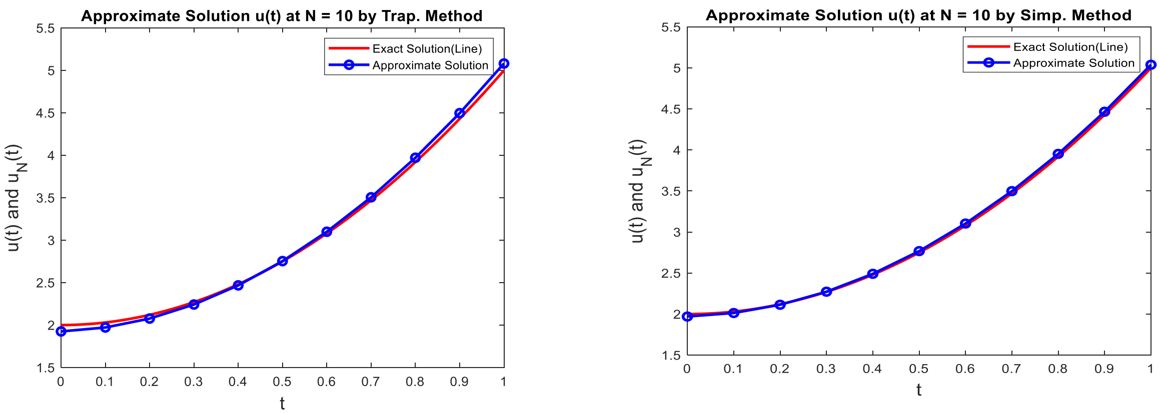

Fig. 1. For example, 1 with a step size of $h = 0.1$, the approximate solutions are shown by bullets, while the precise solution is shown by a solid line.

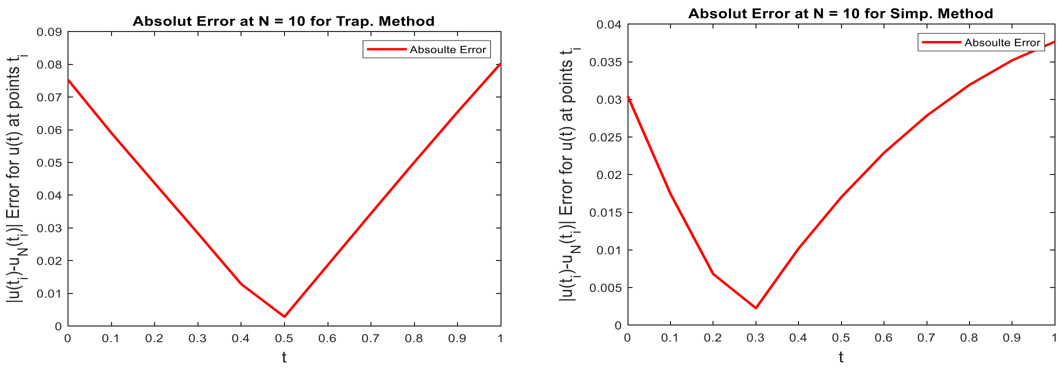

Fig. 2. Absolute error plot function $|u(t) - u_N(t)|$ For $N = 10, h = 0.1$, for example 1.

Table 5. Shows the running times and least square errors for the suggested quadrature techniques, with varying step size h values.

h	0.1	0.02	0.01

Methods	$L.S.E.$	$R.Time$ $/Sec$	$L.S.E.$	$R.Time$ $/Sec$	$L.S.E.$	$R.Time$ $/Sec$
Trap. Method	2.681636×10^{-02}	0.773492	5.058864×10^{-04}	5.540379	1.246178×10^{-04}	19.32022
Simp. Method	6.657162×10^{-03}	6.73552	3.614007×10^{-04}	544.96858	1.111846×10^{-04}	4290.0595

Test example 2. Consider a higher-order linear FIFDE with variable coefficients:

$$ {}_0^C D_t^{2\alpha} u(t) - t^2 {}_0^C D_t^{\alpha} u(t) + \sin(t) u(t) = f(t) $$
$$ = \frac{12}{\Gamma(4-2\alpha)} t^{3-2\alpha} - \frac{6}{\Gamma(3-2\alpha)} t^{2-2\alpha} - \frac{12}{\Gamma(4-\alpha)} t^{5-\alpha} + \frac{6}{\Gamma(3-\alpha)} t^{4-\alpha} $$
$$ + (2t^3 - 3t^2 + 1)\sin(t) - \left[\frac{24}{(5-\beta)\Gamma(4-\beta)} - \frac{12}{(4-\beta)\Gamma(3-\beta)} + \frac{3}{20}\right] t^2 - \frac{1}{2} $$
$$ + \int_0^1 \left[2st^2 {}_0^C D_s^{\beta} u(s) + (1 + st^2) u(s)\right] ds $$

with the boundary conditions: if $0 < \alpha \le 0.5$ and $0 < \beta \le 1$ then: $u(0) + u(1) = 1$, while the exact solution is $u(t) = 2t^3 - 3t^2 + 1$.
for $\alpha = 0.2$ and $\beta = 0.5$ Take $N = 10$ and $t_r = 0{:}0.1{:}1$ for $r = 0,1,2 \ldots N$. Here $n = 2, m = 1$, and by running the programs, we obtain:

$$ A_2^{\alpha}(0) = 2.8112403816,\ A_2^{\alpha}(1) = 1.7016542931,\ A_2^{\alpha}(2) = 1 $$
$$ A_1^{\beta}(0) = 3.5682482323,\ A_1^{\beta}(1) = 1 $$

Table 6 contains all values of $\mathcal{H}_2^{\alpha}(r)$ for each $t_r = 0(0.1)1$ for $r = \overline{1{:}10}$ with $\mathcal{H}_2^{\alpha}(0) = 0$. Moreover, Table 7 shows a comparison between the exact solution and numerical solutions of Trapezoidal and Simpson Methods for $u(t)$ depending on the least square error and running time for running the MATLAB programs for different values of N, i.e., different step sizes h as shown below.

Table 6. The values of $\mathcal{H}_2^{\alpha}(r)$ for each $t_r(r = \overline{1{:}10})$ with $\mathcal{H}_2^{\alpha}(0) = 0$

t_r	0.1	0.2	0.3	0.4	0.5
$\mathcal{H}_2^{\alpha}(r)$	2.8940572553	2.9418435406	2.9536117018	2.9283940370	2.8652523469
t_r	0.6	0.7	0.8	0.9	1.0
$\mathcal{H}_2^{\alpha}(r)$	2.7632873094	2.6216474652	2.4395377248	2.2162273137	1.9510570732

Table 7. Numerical results for different values of t and comparison between methods

t_r	Exact Solution	Approximate Solution and Absolute Errors			
		Trapezoidal Method	Trap. Absolute Error	Simpson Method	Simp. Absolute Error
0	1	1.0128515271	0.012851527	0.01354288	1.0135428804
0.1	0.972	0.9776809241	0.0056809241	0.0064876144	0.9784876143
0.2	0.896	0.8986620799	0.00266208	0.0034831238	0.8994831238
0.3	0.784	0.7851674849	0.0011674849	0.0019527299	0.7859527299
0.4	0.648	0.6483087225	0.00030872255	0.0010279319	0.6490279319
0.5	0.5	0.4996302203	0.00036977962	0.00024447994	0.5002444799
0.6	0.352	0.3508188633	0.0011811366	0.00069929574	0.3513007042
0.7	0.216	0.2135916114	0.0024083885	0.0021117216	0.2138882783
0.8	0.104	0.0996167036	0.0043832963	0.0043146937	0.0996853063
0.9	0.028	0.0204121447	0.0075878553	0.007844095	0.0201559050

1	0	−0.0128012553	0.012801255	0.013521447	−0.0135214467
$L.S.E.$		4.539727×10^{-04}		5.104847×10^{-04}	
$R.Time/Sec$		1.189265		4.275868	

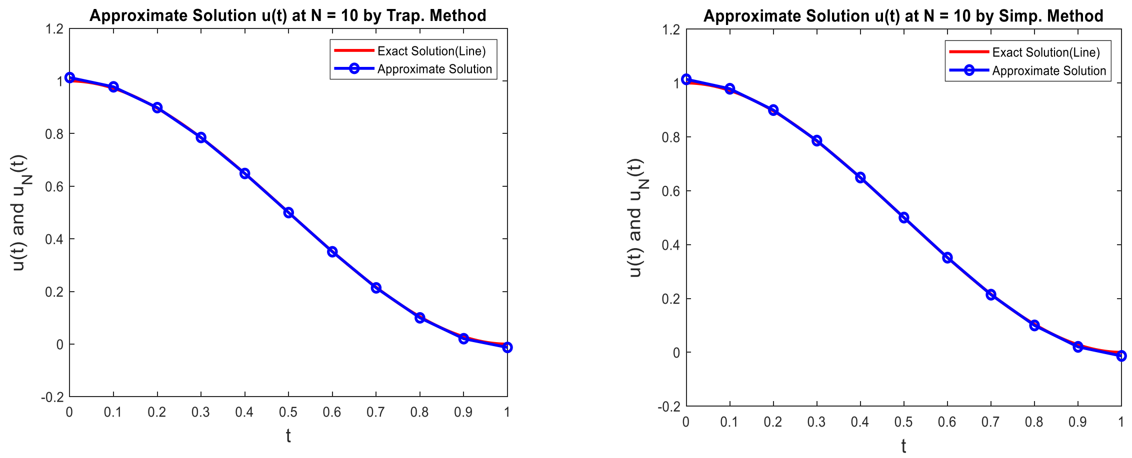

Figure 3. For example, 2 with a step size of $h = 0.1$, the approximate solutions are shown by bullets, while the precise solution is shown by a solid line.

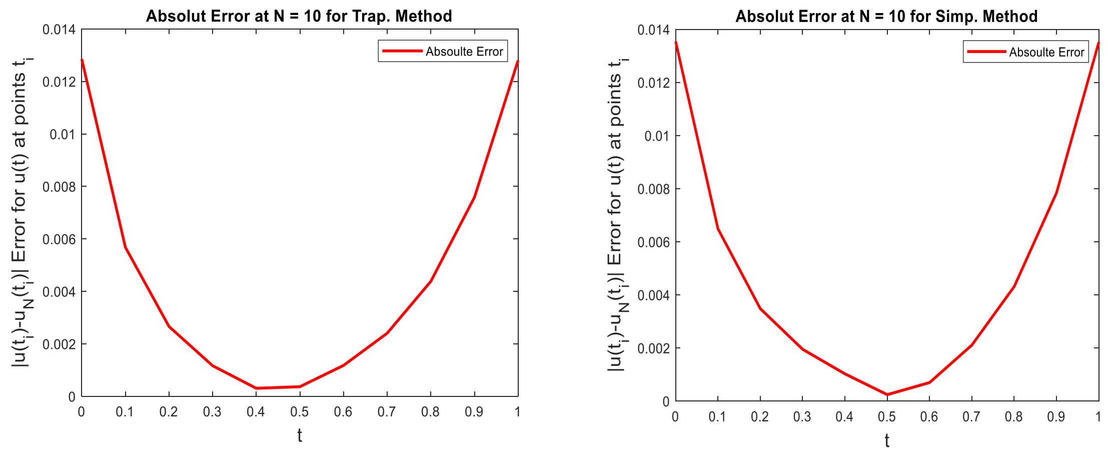

Figure 4. Absolute error plot function $|u(t) - u_N(t)|$ For $N = 10, h = 0.1$, for example 2.

Table 8. Shows the running times and least square errors for the suggested quadrature technique, with varying step size h values.

h	$\mathbf{0.1}$		$\mathbf{0.02}$		$\mathbf{0.01}$	
Methods	$L.S.E.$	$R.Time/Sec$	$L.S.E.$	$R.Time/Sec$	$L.S.E.$	$R.Time/Sec$
Trap. M.	4.539727×10^{-04}	1.176075	1.202408×10^{-05}	14.596982	3.006153×10^{-06}	54.893073
Simp. M.	5.104847×10^{-04}	4.238554	1.375432×10^{-05}	301.97256	3.344898×10^{-06}	2261.5876

Test example 3. Consider a multi-fractional order linear IFDE with variable coefficients on the closed, bounded interval $[a, b]; a, b \in \mathbb{R}$:

$$
\begin{aligned}
&{}_{a}^{C}D_{t}^{\alpha_1}u(t) + t^2u(t) \\
&= t^2 - t^2e^{t-a} \\
&- \lim_{M\to\infty}\sum_{k=0}^{M}\left[\frac{(t-a)^{k-\alpha_1+1}}{\Gamma(k-\alpha_1+2)} - \frac{\lambda\sin(t)\,(b-a)^{k-\beta_2+2}}{\Gamma(k-\beta_2+3)} - \frac{\lambda t(b-a)^{k-\beta_1+2}}{\Gamma(k-\beta_1+3)}\right. \\
&\left. + \frac{\lambda(b-a)^{k-\beta_1+4}}{(k-\beta_1+4)\Gamma(k-\beta_1+2)}\right] \\
&+ \lambda\int_a^b \left[\sin(t)\,{}_{a}^{C}D_{s}^{\beta_2}u(s) + (t-(s-a)^2)\,{}_{a}^{C}D_{s}^{\beta_1}u(s)\right]ds
\end{aligned}
$$

For all $\alpha_1, \beta_2, \beta_1$ are real fractional order lies in $(0,1]$ with boundary condition $u(a) - u(b) = e^{b-a} - 1$ and $\lambda \in \mathbb{R}$. While the exact solution is $u(t) = 1 - e^{t-a}$.
Here, the proposed finite difference quadrature-midpoint method is used to obtain its numerical computation on the bounded interval $[a, b] = [1,2]$. Values of the approximate solution by Trapezoidal and Simpson methods for the mentioned equation with different fractional orders α_1, β_2 and β_1 found by taking $N = 10$ and number of Mittag-Leffler terms $M = 2$ and 4. Tables 9 and 10 illustrate a comparison between the approximate and exact solutions for $N = 10, \lambda = \frac{1}{2}$, and fractional orders $(\alpha_1, \beta_2, \beta_1) = (0.6, 0.4, 0.3)$, while the error profile is depicted in figures 5,6, which present a comparison of absolute errors and approximate solutions for the two methods. Also, the results, least square error, and the required time for running the programs for different values, i.e., different step sizes h, are presented in Tables 11 and 12.

Table 9. Numerical results for different values of $M = 2$ and 4 on the interval [1,2].

t_r	Exact solution	**Trapezoidal method, for example 3 with $N = 10, \lambda = \frac{1}{2}$, and $(\alpha_1, \beta_2, \beta_1) = (0.6, 0.4, 0.3)$**			
		For $M = 2$		**For $M = 4$**	
		Approximate Solution	**Absolute Error**	**Approximate Solution**	**Absolute Error**
1.0	0.0	0.0022302783	0.0022302783	0.0014128639	0.0014128639
1.1	−0.10517092	−0.10614536	0.00097443754	−0.10502418	0.00014674089
1.2	−0.22140276	−0.22434832	0.0029455651	−0.22216879	0.00076603108
1.3	−0.34985881	−0.35402527	0.004166462	−0.3513508	0.0014919909
1.4	−0.4918247	−0.49650258	0.0046778831	−0.49391671	0.0020920137
1.5	−0.64872127	−0.65312464	0.0044033737	−0.65131424	0.0025929698
1.6	−0.8221188	−0.82534904	0.0032302406	−0.82512463	0.0030058298
1.7	−1.0137527	−1.0147931	0.0010403721	−1.0170834	0.0033306683
1.8	−1.2255409	−1.2232654	0.0022755591	−1.2290984	0.0035574413
1.9	−1.4596031	−1.452795	0.0068081387	−1.4632668	0.0036636457
2.0	−1.7182818	−1.7056877	0.012594132	−1.7218792	0.0035973981
	$L.S.E.$	$2.94889633383 \times 10^{-04}$		$7.50778975764 \times 10^{-05}$	
	$R.Time/Sec$	0.8954315		0.9111462	

Table 10. Numerical results for different values of $M = 2$ and 4 on the interval [1,2].

t_r	Exact solution	**Simpson method, for example 3 with $N = 10, \lambda = \frac{1}{2}$, and $(\alpha_1, \beta_2, \beta_1) = (0.6, 0.4, 0.3)$**			
		For $M = 2$		**For $M = 4$**	
		Approximate solution	**Absolute error**	**Approximate solution**	**Absolute error**
1.0	0.0	−0.00000869868	0.00000869868	−0.0007535077	0.00075350779
1.1	−0.10517092	−0.10841569	0.0032447702	−0.1072408	0.0020698865
1.2	−0.22140276	−0.22656374	0.0051609772	−0.22431554	0.0029127854

1.3	-0.34985881	-0.35610609	0.0062472821	-0.35338179	0.003522985
1.4	-0.4918247	-0.49844337	0.0066186746	-0.4957964	0.0039717069
1.5	-0.64872127	-0.65489312	0.0061718519	-0.65303787	0.0043165991
1.6	-0.8221188	-0.82696248	0.0048436798	-0.82668526	0.004566462
1.7	-1.0137527	-1.0162421	0.0024893462	-1.0184926	0.004739855
1.8	-1.2255409	-1.2245742	0.00096675685	-1.2303625	0.0048215412
1.9	-1.4596031	-1.4539671	0.0056360568	-1.4644037	0.0048005402
2.0	-1.7182818	-1.7067516	0.011530188	-1.7229073	0.0046254751
$L.S.E.$		$3.53394505666 \times 10^{-04}$		$1.71161696191 \times 10^{-04}$	
$R.Time/Sec$		2.4475037		2.4938223	

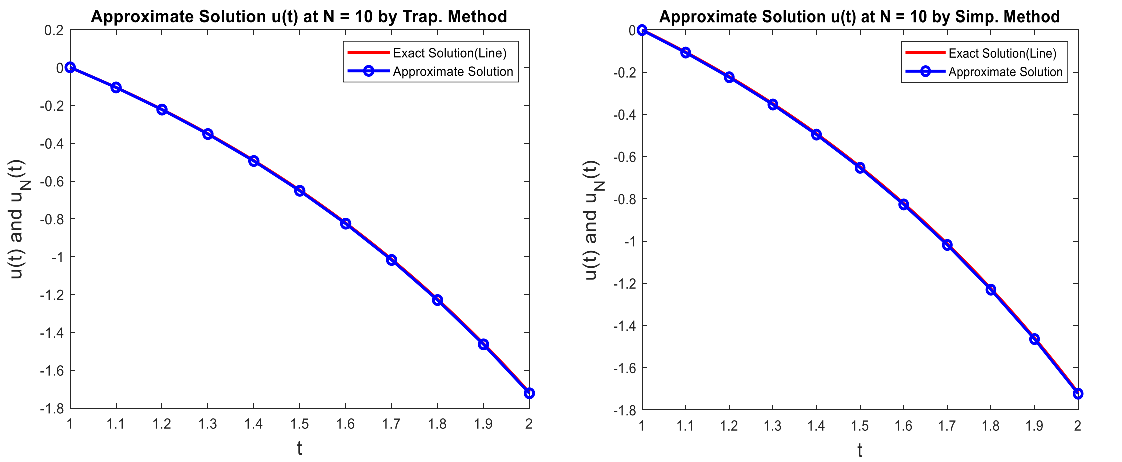

Figure 5. The exact solution (shown by a solid line) and the approximate solutions (shown b bullets) for example 3 with a step size of $h = 0.1$ and $M = 4$.

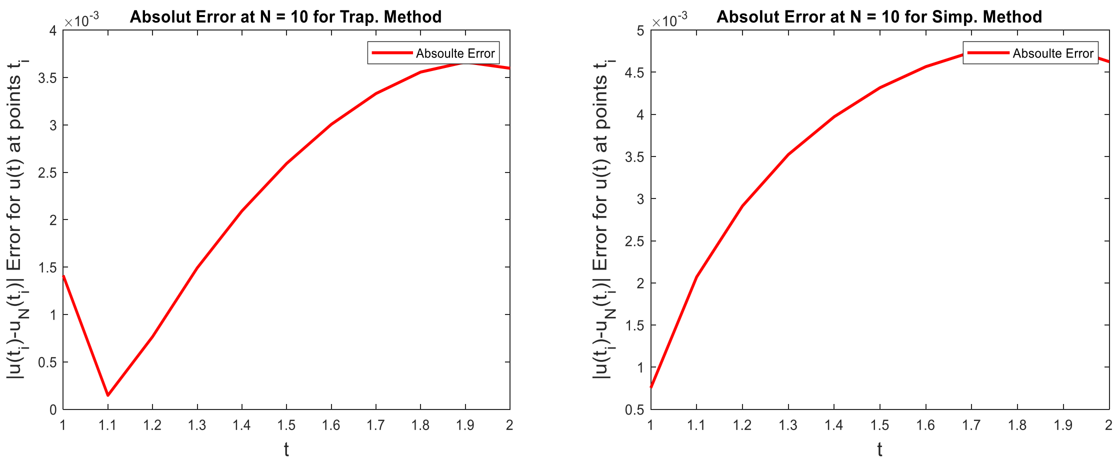

Figure 6. Absolute error plot function $|u(t) - u_N(t)|$ for $N = 10, h = 0.1$ and $M = 4$ fo example 3.

Table 11. LSEs of approximate solution by Trapezoidal method for various values of fractional orders and eigenvalues λ when $M = 4$ in example 3 on the interval $[a, b] = [1,2]$.

Fractional orders: $(\boldsymbol{\alpha_1}, \boldsymbol{\beta_2}, \boldsymbol{\beta_1})$	$(\mathbf{0.8}, \mathbf{0.5}, \mathbf{0.3})$		$(\mathbf{0.5}, \mathbf{0.3}, \mathbf{0}$		$(\mathbf{0.2}, \mathbf{0.3}, \mathbf{0.1})$	
Eigenvalues: λ	$6/5$	$1/5$	$6/5$	$1/5$	$6/5$	$1/5$

$N = 10$	$L.S.E.$	7.794545×10^{-04}	$1.2683819 \times 10^{-03}$	4.27680×10^{-05}	$8.1401588 \times 10^{-05}$	$9.3227264 \times 10^{-05}$	$2.0319866 \times 10^{-06}$
	$R.Time/Sec$	0.983350	0.9028357	0.95748	0.9434983	0.9339204	0.9415094
$N = 50$	$L.S.E.$	4.892593×10^{-05}	$2.8815882 \times 10^{-05}$	1.83678×10^{-06}	$5.2718987 \times 10^{-07}$	$1.4629413 \times 10^{-07}$	$2.6679147 \times 10^{-07}$
	$R.Time/Sec$	9.175369	9.144518	9.12176	10.353755	10.177932	10.283534
$N = 100$	$L.S.E.$	1.329773×10^{-05}	$4.1747311 \times 10^{-06}$	9.28155×10^{-07}	$3.5738073 \times 10^{-07}$	$3.2859089 \times 10^{-07}$	$2.6169685 \times 10^{-07}$
	$R.Time/Sec$	35.97653	33.006919	33.3140	33.202534	35.474032	33.022882

Table 12. LSEs of approximate solution by Simpson method for various values of fractional orders and eigenvalues λ when $M = 4$ in example 3 on the interval $[a, b] = [1,2]$.

Fractional orders: $(\boldsymbol{\alpha_1}, \boldsymbol{\beta_2}, \boldsymbol{\beta_1})$		$(\mathbf{0.8}, \mathbf{0.5}, \mathbf{0.3})$		$(\mathbf{0.5}, \mathbf{0.3}, \mathbf{0.1})$		$(\mathbf{0.2}, \mathbf{0.3}, \mathbf{0.1})$	
Eigenvalues: λ		$6/5$	$1/5$	$6/5$	$1/5$	$6/5$	$1/5$
$N = 10$	$L.S.E.$	$1.4760476 \times 10^{-03}$	$1.4705611 \times 10^{-03}$	$1.0912289 \times 10^{-04}$	$1.0691410 \times 10^{-04}$	$1.9225015 \times 10^{-05}$	$4.1497246 \times 10^{-07}$
	$R.Time/Sec$	2.5410192	2.5712627	2.5816489	2.5661209	2.5579086	2.6615136
$N = 50$	$L.S.E.$	$6.3393839 \times 10^{-05}$	$3.1132111 \times 10^{-05}$	$2.8367448 \times 10^{-06}$	$6.1593748 \times 10^{-07}$	$2.1160329 \times 10^{-07}$	$2.2710878 \times 10^{-07}$
	$R.Time/Sec$	161.82672	163.09490	160.04967	164.70403	156.09650	156.60117
$N = 100$	$L.S.E.$	$1.5846702 \times 10^{-05}$	$4.4595419 \times 10^{-06}$	$1.1116152 \times 10^{-06}$	$3.5224767 \times 10^{-07}$	$4.1910725 \times 10^{-07}$	$2.5060134 \times 10^{-07}$
	$R.Time/Sec$	1180.1969	1184.0819	1172.5111	1190.9019	1187.7246	1176.2318

6. Conclusion:

This research uses a forward finite difference method for the Caputo derivative and introduces two numerical algorithms - the Trapezoidal and Simpson methods - to solve multi-fractional order IDEs of the Fredholm type with changing coefficients for the first time. First, we demonstrate that IFDEs (1)–(2) are equivalent to linear algebraic systems in matrix forms (22) for the Simpson technique with a vector of boundary conditions (16) or in matrix forms (11) for the Trapezoidal method. After that, we solve it using any numerical rule, which gives us a solution to our problem (1-2). This is why the computer programs and unique algorithms were created. Furthermore, we solved several examples related to the proposed equations using numerical techniques. The numerical findings created an exceptional absolute inaccuracy among the numerical approaches used in the literature. The tabular representations of the running time and least square error for the accuracy and speed comparison lead to the following conclusions:

1. With equal step sizes, the AFIFS and AFIFT algorithms give better accuracy in running test examples.
2. The accuracy of the results is influenced by both the process and the step length h; that is, when h is reduced, the accuracy rises, increasing the number of partitions N.

3. To minimize the error terms on the specified domain, we need to increase the value of N, and only a few numbers of M (in the stale example 3) can be used for numerical purposes with a high degree of accuracy, see all the tables.

Data Availability Statement: Authors can confirm that all relevant data are included in the article.
Conflicts of Interest: The article declares no conflicts of interest.
Funding: This research was self-funded by the authors. No external funding was received for this study.
Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

References

[1]. A. Hayotov, S. Babaev, and B. Boytillayev, "The numerical solution of an Abel integral equation by the optimal quadrature formula," *Results Appl. Math.*, vol. 25, 100542, Feb. 2025, DOI: 10.2139/ssrn.. 5005952.

[2]. R. H. Al-Nasir, *Numerical Solution of Volterra Integral Equations of the Second Kind*, M.Sc. dissertation, Univ. of Technology, Baghdad, 1999.

[3]. J. Saberi-Nadjafi and M. Heidari, "Solving linear integral equations of second kind with repeated modified trapezoid quadrature method," *Appl. Math. Comput.*, vol. 189, pp. 980–985, Jun. 2007. DOI: 10.1016/j.amc.2006.11.165.

[4]. S. Rahbar and E. Hashemizadeh, "A computational approach to the Fredholm integral equation of the second kind," in *Proc. World Congr. Eng.*, London, U.K., vol. II, pp. 978–988, Jul. 2008.

[5]. M. J. Emamzadeh and M. T. Kajani, "Nonlinear Fredholm integral equation of the second kind with quadrature methods," *J. Math. Ext.*, vol. 4, no. 2, pp. 51–58, 2010. DOI: 10.1155/2013/426916.

[6]. S. A. Isaacson and R. M. Kirby, "Numerical solution of linear Volterra integral equations of the second kind with sharp gradients," *J. Comput. Appl. Math.*, vol. 235, pp. 4283–4301, Feb. 2011. DOI: 10.1016/j.cam.2011.03.029.

[7]. R. Saadati, B. Raftari, H. Adibi, S. M. Vaezpour, and S. Shakeri, "A comparison between the variational iteration method and trapezoidal rule for solving linear integro-differential equations," *World Appl. Sci. J.*, vol. 4, pp. 321–325, 2008.

[8]. M. A. W. Al-Jawary, *Numerical Methods for System of Integral Equations*, M.Sc. dissertation, Univ. of Baghdad, Baghdad, 2005.

[9]. S. S. Ahmed and S. A. Hamasalih, "Numerical treatment of the most general linear Volterra integro-fractional differential equations with Caputo derivatives by quadrature methods," *J. Math. Comput. Sci.*, vol. 2, pp. 1293–1311, 2012.

[10]. S. S. Ahmed and S. J. Mohammedfaeq, "Operational matrix of generalized block pulse function for solving fractional Volterra–Fredholm integro-differential equations," *J. Southwest Jiaotong Univ.*, vol. 57, pp. 39-59, 2022. DOI: 10.35741/issn 0258-2724.57.3.4.

[11]. A. A. Kilbas, H. M. Srivastava, and J. J. Trujillo, *Theory and Applications of Fractional Differential Equations*. Netherlands: Elsevier, 2006.

[12]. M. R. Ahmed, S. S. Ahmed, and P. O. Sabir, "Approximate Solution to The System of Nonlinear Volterra Integro-Fractional Differential Equations with Variable Coefficients

Using Linear B-Spline Functions," J. SouthWest Jiaotong Univ., vol. 59, no. 4, 2024, DOI: 10.35741/issn 0258-2724.59.4.26.

[13]. S. S. Ahmed and S. J. Mohammedfaeq, "Bessel collocation method for solving Fredholm–Volterra integro-fractional differential equations of multi-high order in the Caputo sense," *Symmetry*, vol. 13, p. 2354, Dec. 2021. DOI: 10.3390/sym13122354.

[14]. S. S. Ahmed, On System of Linear Volterra Integro-Fractional Differential Equations, Ph.D. dissertation, Univ. of Sulaimani, Sulaymaniyah, 2009.

[15]. F. Mainardi, *Fractional calculus and waves in linear viscoelasticity: an introduction to mathematical models*, World Scientific, 2022.

[16]. S.S. Ahmed, H.A. Rasol, Numerical Computation of Mixed Volterra–Fredholm Integro-Fractional Differential Equations by Using Newton-Cotes Methods, *Symmetry* 2022, *14*(8), 1693; DOI: 10.3390/sym14081693.

[17]. K. E. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed., John Wiley & Sons, 1989.

[18]. R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th ed., Int. Thomson Publishing, 2011.

[19]. H. A. M. Saed Ben Hamdin and F. S. M. Musbah, "Hybrid triple quadrature rule blending some Gauss-type rules with the classical or the derivative-based Newton-Cotes-type rules," *Al-Mukhtar J. Basic Sci.*, vol. 21, no. 2, pp. 63-72, 2023.

Journal of Basrah Researches (Sciences) **51**(2), 1 (2025)
DOI: https://doi.org/10.56714/bjrs.51.2.1

تقريب الفروق المحدودة مع طريقة التربيع لحل معادلات فردهولم التكاملية-التفاضلية ذا الرتب الكسرية

دشنى جابوك ظاهر [1] ،شازاد شوقى احمد [2]

[1]قسم الرياضيات،كلية العلوم والصحة، جامعة كويه
[2]قسم الرياضيات،كلية العلوم ، جامعة سليمانيه

معلومات البحث

الاستلام	12 أيار 2025
المراجعة	19 حزيران 2025
القبول	28 حزيران 2025
النشر	31 كانون أول 2025

الكلمات المفتاحية

حساب الكسور، المشتقة الكسرية كابوتو، المعادلة التكاملية التفاضلية، تقنية تربيع نيوتن-كوت، طريقة شبه المنحرف، طريقة سيمبسون، تقريب الفرق الأمامي.

Citation: D. C. Zahir, S. S. Ahmed , J. Basrah Res. (Sci.) **50**(2), 1 (2025). DOI:https://doi.org/10.56714/bjrs.51.2.1

الملخص

في هذه المقالة، تُعرض تقنيات فعّالة لحل معادلات فردهولم التكاملية-التفاضلية ذات الرتب الكسرية المتعددة عددياً، والتي تقع رتبها ضمن المجال المفتوح (0,1]وفقاً لتعريف كابوتو .وتعتمد المنهجية المقترحة على تقريب الفروق المحدودة لمشتقة كابوتو باستخدام نقاط التجميع مستندةً إلى قواعد التربيع، وتحديداً طريقتي شبه المنحرف وسيمبسون.

تُسهم هذه الطريقة في تبسيط عمليات المعالجة من خلال تحويل معادلات فردهولم التكاملية-الكسيرية إلى معادلات جبرية باستخدام مصفوفات تشغيلية .وبعد حساب مشتقة كابوتو في نقطة معينة باستخدام طريقة الفروق المحدودة، يتم تطبيق طريقة التربيع التي تتضمن قواعد شبه المنحرف وسيمبسون، لبناء صيغة عددية تعتمد على الفروق المحدودة للمعادلة الكسرية.

كما تتضمن الدراسة عرضاً لأمثلة عددية تُبيّن صحة وكفاءة المنهج المتبع، إلى جانب مقارنات مع نتائج سابقة منشورة .وقد استُخدم هذا الأسلوب في تصميم خوارزميات لحل معادلات فردهولم التكاملية-الكسيرية، وتم تنفيذ الحلول باستخدام برنامج ماتلاب .إضافةً إلى ذلك، تم إجراء اختبارات عددية لتوضيح دقة الطريقة المقترحة وفعاليتها.

.

*Corresponding author email: dashne.zahir@koyauniversity.org

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq