Journal of Al-Qadisiyah for Computer Science and Mathematics Vol.18.(1) 2026,pp.Math 40–64

University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

A Numerical Solution of Mixed-Order Fredholm Integro-Differential Equations by Using Trapezoidal, Two-Block, and Modified Two-Block Methods

Kurdistan H. Hama [a,*] ***and Shazad S. Ahmed*** [a]

[a] *Department of Mathematics, College of Science, University of Sulaimani, Sulaymaniyah, Iraq. Email: kurdstanhama.ali@gmail.com, shazad.ahmed@univsul.edu.iq*

ARTICLEINFO

Article history:
Received: 26 /10/2025
Revised form: 16 /12/2025
Accepted : 21/12/2025
Available online: 30 /03/2026

Keywords:

Classical and fractional order integro-differential equations, Fredholm type, Caputo fractional derivatives, Trapezoidal method, Two-point block method, Modify two-point block method and Finite difference approximation.

ABSTRACT

The purpose of this article is to use the trapezoidal rule, two-point block schemes, and a modified two-point block approach to numerically solve classical and fractional-order (Mixed-orders in the Caputo sense) Fredholm integro-differential equations with variable coefficients under limit conditions (IFDEs-CF). Three new algorithms are proposed to find approximate solutions to these equations, firstly transforming them into systems of linear algebraic equations using the trapezoidal method aided with finite difference approximation, then applying two-block techniques with the predictor value obtained, and finally modifying the results by modification technique to obtain correcting values for these equations at each fixed point. This approach is computationally attractive, and illustrative examples with usage explanations are provided. Additionally, we provide specific examples to showcase the accuracy of the method, and we employ the least-squares error methodology to minimize error terms within the given domain. Finally, the most common application suggested for the numerical approaches is developed in the Python program.

MSC..

https://doi.org/10.29304/jqcsm.2026.18.12418

1. Introduction

One of the most significant areas of mathematics that deals with integrals and derivatives of any arbitrary order is fractional calculus (FC). The idea of FC has been successfully applied in recent years to a variety of problems in physics, signal processing, engineering, biology, and other domains. Fluid flow, electrical networks, electromagnetic, control theory, facts, optics, capacity, diffusion, and viscoelasticity are just a few of the domains in which FC finds application (see, for instance, [1], [2], [3], [4], [5], [6], [7], and [8]). Many real-life problems are reduced to fractional differential and integral equations by mathematical modeling [9]. Researchers in a wide range of scientific and technical domains have recently shown a great deal of interest in fractional differential equations and integro-fractional differential equations.

However, because of its numerous applications in the previously listed fields, integro-fractional differential equations have attracted the curiosity of numerous scientists. Fredholm-type mixed classical and fractional order integro-differential equations are among the most prevalent types of these equations. As in many other situations, it is quite difficult to find the correct analytical solutions for fractional differential and integral equations. The importance of approximate ways to avoid this issue has increased. References [10], [11], [12], [13], [14] and [15] demonstrate variety of methods for evaluating the approximate solution of integro-fractional differential equations. In [11], the author used Trapezoid's quadrature to solve a multi-higher fractional order linear integro-differential equation of Fredholm type in the Caputo sense. The author in [15] used the finite difference method in combination with the homotopy perturbation method to numerically solve the linear VI-DEs, while [13] and [14] proposed a method based on Block methods for numerically treating the most general VIFDEs of linear type. Numerical techniques are used to approximate the solution of integro-fractional differential equations because it is challenging to obtain the analytical solution of fractional differential equations.

This study uses a new methodological approach based on quadrature and direct two-point blocks to expand on the basic concepts from previous studies. The trapezoidal technique is used to handle the following problems, which combine an estimated finite difference in matrix form. Second, problems that include the following are tackled using two-point blocks and a modified process approach. Linear Fredholm integro-differential equations of the first classical order and multi-fractional orders $\{\boldsymbol{\sigma_i}, \boldsymbol{\beta_j}\}_{i=0,j=1}^{\bar{n}-1,m}$ are present in the range $(\mathbf{0},\mathbf{1})$ with variable coefficients (FIDEs-CF):

$$\boldsymbol{p(x)y'(x)} + \sum_{i=0}^{\bar{n}-1} \boldsymbol{p_i(x)\,{}_a^C D_x^{\sigma_i} y(x)} + \boldsymbol{p_{\bar{n}}(x)y(x)} = \boldsymbol{f(x)} + \boldsymbol{\lambda_0} \int_a^b \boldsymbol{\mathcal{K}_0(x,s)\, y(s)ds} + \sum_{j=1}^{m} \boldsymbol{\lambda_j} \int_a^b \boldsymbol{\mathcal{K}_j(x,s)\,{}_a^C D_s^{\beta_j} y(s)ds} \qquad (1)$$

Under the boundary condition:

$$\boldsymbol{g_{00}}\mathbf{y}(\boldsymbol{a}) + \boldsymbol{\hbar_{00}}\mathbf{y}(\boldsymbol{b}) = \boldsymbol{\vartheta_0} \qquad (2)$$

where the functions: $\{\boldsymbol{p_i}\}_{i=0}^{\bar{n}}, \boldsymbol{f}$ **and** $\boldsymbol{p} \in \boldsymbol{C}([\boldsymbol{a},\boldsymbol{b}],\mathbb{R})$ and $\{\boldsymbol{\mathcal{K}_j}\}_{j=0}^{m} \in \boldsymbol{C}(\Xi,\mathbb{R})$, that is $\Xi = \{(\boldsymbol{x},\boldsymbol{s}): \boldsymbol{a} \le \boldsymbol{x}, \boldsymbol{s} \le \boldsymbol{b}\}$ are knowns, in addition, $\boldsymbol{y(x)}$ is an unknown continuous function. While, given arbitrary real constants $\boldsymbol{\vartheta_0}, \boldsymbol{g_{00}}, \boldsymbol{\hbar_{00}}$. Where the fractional orders: $\boldsymbol{\sigma_{n-1}} > \cdots > \boldsymbol{\sigma_1} > \boldsymbol{\sigma_0} > \mathbf{0}$ and $\boldsymbol{\beta_m} > \boldsymbol{\beta_{m-1}} > \cdots > \boldsymbol{\beta_1} > \boldsymbol{\beta_0} = \mathbf{0}$, since $\boldsymbol{\sigma_i} < \mathbf{1}$ with $\boldsymbol{\beta_j} \in (\mathbf{0},\mathbf{1})$, for all $\boldsymbol{i} = \mathbf{0},\mathbf{1},\dots,\bar{\boldsymbol{n}}-\mathbf{1}$ and $\boldsymbol{j} = \mathbf{1},\mathbf{2},\dots,\boldsymbol{m}$; where $\boldsymbol{\omega_j^\beta} - \mathbf{1} < \boldsymbol{\beta_j} \le \boldsymbol{\omega_j^\beta}, \boldsymbol{\omega_j^\beta} = \lceil \boldsymbol{\beta_j} \rceil = \mathbf{1}$ for all $\boldsymbol{j} = \mathbf{1},\mathbf{2},\dots,\boldsymbol{m}$, and $\boldsymbol{\omega_i^\sigma} - \mathbf{1} < \boldsymbol{\sigma_i} \le \boldsymbol{\omega_i^\sigma}$, $\boldsymbol{\omega_i^\sigma} = \lceil \boldsymbol{\sigma_i} \rceil = \mathbf{1}$, **for all** $\boldsymbol{i} = \mathbf{0},\mathbf{1},\dots,\bar{\boldsymbol{n}}-\mathbf{1}$ and $\bar{\boldsymbol{n}} \in \mathbb{Z}^+, \boldsymbol{m} \in \mathbb{Z}^+$. $\{\boldsymbol{\lambda_j}\}_{j=0}^{m}$ are known parameters.

The paper is organized as follows: Section 2 includes a basic overview of the introduction to fractional calculus and important lemmas, along with the essential concepts and the formulation of the quadrature method with finite differences approximation for differentiations. In section 3, three numerical algorithms for FIDEs-CF are derived in detail using the trapezoidal system, two-point blocks, and modifications. The algorithm in this section also describes the main stages of the system. The numerical results are shown in section 4, and the conclusions are given in section 5.

2. Preliminaries

2.1 Fractional Calculus:

This section provides the essential concepts and significant characteristics of fractional calculus theory with basic lemmas that are utilized during the study for completeness.

Definition 1 ([16]). The space $C_\zeta(I), \zeta \in \mathbb{R}$, contain all real-valued functions y that is defined on a closed bounded interval $[a\,,b] = I$, if there exists a real number $\upsilon > \zeta$ such that $y(x) = (x - a)^\upsilon y_0(x)$where $y_0 \in C(I)$. Moreover, $y(x)$ is considered to be in the space $C_\zeta^n(I)$, where $n \in \mathbb{Z}^+ \cup \{0\}$, if and only if the n-th derivative of y with respect to x , $y^{(n)}$, belongs to $C_\zeta(I)$.

Definition 2 ([17]). The ρ-Riemann-Liouville fractional integral on $[a,b]$ for a function $y \in C_\zeta(I), \zeta \ge -1$, represented as operator ${}_aJ_x^\rho$, of order $\rho > 0$, is defined as:

$$ {}_aJ_x^{\rho} y(x) = \frac{1}{\Gamma(\rho)} \int_a^x (x-s)^{\rho-1} y(s) ds, \qquad \rho \in \mathbb{R}^+ $$

Here, $\Gamma(.)$ denotes the gamma function and the operator ${}_aJ_x^{\rho}$ became identity, when the zeroth fractional order that is ${}_aJ_x^{0} y(x) = y(x)$. Also, has a commutative property, that is, for any arbitrary fractional orders $(\rho \text{ and } \gamma) \in \mathbb{R}^+$, ${}_aJ_x^{\rho}\, {}_aJ_x^{\gamma} y(x) = {}_aJ_x^{\gamma}\, {}_aJ_x^{\rho} y(x) = {}_aJ_x^{\rho+\gamma} y(x)$. And ${}_aJ_x^{\rho}$ is linear operators.

Lemma 1 ([3]). If we've $x > a$ and for $y(x) = (x-a)^{\omega}$ where $\omega \neq -\lceil \rho \rceil$, then the following statement holds:

$$ {}_aJ_x^{\rho} y(x) = \frac{\Gamma(\omega+1)}{\Gamma(\omega+\rho+1)} (x-a)^{\omega+\rho}, \qquad \rho \in \mathbb{R}^+. $$

Definition 3 ([18]). For a function on interval $I = [a, b]$, $y(x) \in C_{-1}^{T}(I)$, and $\omega = \lceil \rho \rceil, \rho \geq 0$, the Riemann-Liouville (R-L) fractional derivative operator, ${}_a^RD_x^{\rho}$, is often stated as:

$$ {}_a^RD_x^{\rho} y(x) = D_x^{\omega}\, {}_aJ_x^{\omega-\rho} y(x) = \begin{cases} \dfrac{\partial^{\omega}}{\partial x^{\omega}} \left(\displaystyle\int_a^x \frac{(x-s)^{\omega-\rho-1}}{\Gamma(\omega-\rho)} y(s) ds \right), & \rho \in \mathbb{R}^+ \\ \dfrac{d^r}{dx^r} y(x), & r = \lceil \rho \rceil \in \mathbb{Z}^+ \cup \{0\} \end{cases} $$

where $\omega - 1 < \alpha \leq \omega\ (= \lceil \rho \rceil \in \mathbb{Z}^+)$, and ${}_a^RD_x^{\rho}$ is linear operators.

Definition 4 ([9,10, 19]). For a function on interval $[a, b]$, $y(x) \in C_{-1}^{\omega}(I)$, and $\omega = \lceil \rho \rceil, \rho \geq 0$, the Caputo fractional derivative operator, ${}_a^CD_x^{\rho}$, is often stated as:

$$ {}_a^CD_x^{\rho} y(x) = {}_aJ_x^{\omega-\rho} D_x^{\omega} y(x) = \begin{cases} \displaystyle\int_a^x \frac{(x-s)^{\omega-\rho-1}}{\Gamma(\omega-\rho)} \frac{d^{\omega} y(s)}{ds^{\omega}} ds, & \rho \in \mathbb{R}^+ \\ \dfrac{d^r}{dx^r} y(x), & r = \lceil \rho \rceil \in \mathbb{Z}^+ \cup \{0\} \end{cases} $$

where $\omega - 1 < \rho \leq \omega (= \lceil \rho \rceil \in \mathbb{Z}^+)$. The fundamental properties of Caputo fractional derivative are: The Caputo-type fractional derivative is a linear operator for any fractional orders and the α-Caputo fractional derivative of a constant (A) is equal to zero i.e. ${}_a^CD_x^{\rho} A = 0$.

Lemma 2 ([12, 20]). The function $y(x) = (x-a)^{\upsilon}$, for $\upsilon \geq 0$, has a Caputo derivative of order $\rho \geq 0$ with $\varpi = \lceil \rho \rceil$, which is formed as: For $\upsilon \in \{0,1,2, \cdots, \varpi - 1\}$: ${}_a^CD_x^{\rho} y(x) = 0$ and for $\upsilon \in \mathbb{N}$ and $\upsilon \geq \varpi$ or $\upsilon \notin \mathbb{N}$ and $\upsilon > \varpi - 1$:

$$ {}_a^CD_x^{\rho} y(x) = \frac{\Gamma(\upsilon+1)}{\Gamma(\upsilon-\rho+1)} (x-a)^{\upsilon-\rho}. $$

Lemma 3 ([12]). The finite difference approximation of Caputo derivative for $0 < \rho \leq 1$ at given points $x = x_{r+1}$; $r = 0,1, \dots, \mathcal{N}-1$ and $h = \frac{b-a}{\mathcal{N}}$, $(\mathcal{N} \in \mathbb{Z}^+)$ is formed as:

$$ {}_a^CD_x^{\rho} y(x_{r+1}) = \frac{h^{-\rho}}{\Gamma(2-\rho)} \sum_{\kappa=0}^{r} [y(x_{r-\kappa+1}) - y(x_{r-\kappa})]\, \delta_{\kappa}^{\rho}. $$

where $\delta_{\kappa}^{\rho} = (\kappa+1)^{1-\rho} - \kappa^{1-\rho}$.

Lemma 4 ([19, 21]). Let $\rho \in \mathbb{R}^+$ such that $m - 1 < \rho \leq m (= \lceil \rho \rceil \in \mathbb{Z}^+)$. Moreover, assume that $y \in C_{-1}^{m}[a, b]$. then the ρ-Caputo fractional derivative of $y(x)$ is continuous on $[a, b]$ and $\lim_{x \to a} {}_a^CD_x^{\rho} y(x) = 0$.

2.2 Trapezoidal Method:

The quadrature rule is defined as the weighted sum of a finite number of integrand function sample values. Consider $g(x)$, a real-valued function defined on the constrained interval $a \leq x \leq b$. The integral $\int_a^b g(x) dx$ is to be calculated using $\sum_{j=1}^{N} w_j g(x_j) + Rem[g]$. $Rem[g]$ is the remainder, and the quadrature rule $\{w_j, x_j\}_{j=1}^{N}$ can be obtained in tabular form with x_j representing the integration nodes and w_j representing the quadrature weights, or constants [22, 23]. Here, we introduce the Trapezoidal rule, an algorithm for producing the quadrature rule determined by the weight function and node numbers:

The trapezoidal rule relies on linear interpolation of $g(x)$at x_r and x_{r+1}, where $g(x)$ is approximated by the straight line connecting (x_r, g_r) and (x_{r+1}, g_{r+1}) .The interval $[a, b]$ is divided into N-subintervals $[x_r, x_{r+1}]$ of equal length $h = (b - a)/N$, with sample points expressed as $x_r = x_0 + rh$, where $(r = 0,1,2, \ldots N)$, and $b = x_0 + Nh$. The numerical integration of $g(x)$ over $[a, b]$ using the trapezoidal rule can then be expressed as follows:

$$\int_{a}^{b} g(x)dx \cong \frac{h}{2}\left[g(a) + 2\sum_{i=1}^{N-1} g(x_i) + g(b)\right] = h\sum_{\kappa=0}^{N} w_\kappa^T g(x_\kappa) \tag{3}$$

Where w_κ^T is weights for Trapezoidal rule, $w_0^T = w_N^T = \frac{1}{2}; w_\kappa^T = 1; (0 < \kappa < N)$, with global error $-\frac{(b-a)}{12} h^2 g''(\theta), a < \theta < b$.

2.3 Finite Differences Rule:

In mathematics the finite differences, or the corresponding difference quotients, are frequently employed in numerical differentiation and other applications as approximations of derivatives. Conversely, a finite difference is a numerical technique that is used to approximate the derivative of a function without the need to calculate the derivative symbolically. By approximating functions on a grid and substituting differences for derivatives, it is also a reasonably general method for functional (differential or integral) equations.

Let $x_\kappa = x_0 + \kappa h, 0 \le \kappa \le N$, where h is the step size in the space. Denote $\mathcal{Y}_\kappa = \mathcal{y}(x_\kappa)$ while $\mathcal{y}_\kappa^h$ represents the approximation of $\mathcal{Y}_\kappa$ at the collocation point x_κ depending on step size h. According to typical, we present the following notations for the first derivative at selecting point x_κ: The forward, backward, and central difference operators are the three types of difference operators, [23]:

Three-point forward difference with truncation error $O(h^2)$:

$$\mathcal{y}'_\kappa = \frac{-3\mathcal{y}_\kappa^h + 4\mathcal{y}_{\kappa+1}^h - \mathcal{y}_{\kappa+2}^h}{2h} + O(h^2) \tag{4}$$

Two-point central difference with truncation error $O(h^2)$:

$$\mathcal{y}'_\kappa = \frac{\mathcal{y}_{\kappa+1}^h - \mathcal{y}_{\kappa-1}^h}{2h} + O(h^2) \tag{5}$$

Three-point Backward difference with truncation error $O(h^2)$:

$$\mathcal{y}'_\kappa = \frac{\mathcal{y}_{\kappa-2}^h - 4\mathcal{y}_{\kappa-1}^h + 3\mathcal{y}_\kappa^h}{2h} + O(h^2) \tag{6}$$

2.4 Linear Multistep Rules:

Consider the initial value problem for single first-order differential equation: $y'(x) = \mathcal{F}(x, y(x)), y(x_0) = y_0$, we seek a solution in the range $a \le x \le b$, where a and b are finite. Let $x_\kappa = x_0 + \kappa h, 0 \le \kappa \le N$, where h are the step size in the space and let y_κ be an approximation to the theoretical solution at x_κ, that is, to $y(x_\kappa), \mathcal{F}_\kappa = \mathcal{F}(x_\kappa, y_\kappa)$. The approximations built up by using the following linear multistep methods, [14,24]:

$$\left.\begin{array}{ll} y_{\kappa+1} - y_\kappa = \frac{h}{2}[\mathcal{F}_{\kappa+1} + \mathcal{F}_\kappa]; & \text{order } 2 \\ y_{\kappa+2} - y_\kappa = 2h\mathcal{F}_{\kappa+1}; & \text{order } 2 \\ y_{\kappa+1} - y_\kappa = \frac{h}{12}[-\mathcal{F}_{\kappa+2} + 8\mathcal{F}_{\kappa+1} + 5\mathcal{F}_\kappa]; & \text{order } 3 \\ y_{\kappa+2} - y_\kappa = \frac{h}{3}[\mathcal{F}_{\kappa+2} + 4\mathcal{F}_{\kappa+1} + \mathcal{F}_\kappa]; & \text{order } 4 \end{array}\right\} \tag{7}$$

3. Three Algorithms to Solve FIDEs-CF:

This section presents a new three numerical algorithms for approximating solutions to Fredholm integro-differential equations (FIDEs-CF) in the Caputo sense with variable coefficients that are mixed-order, classical, and fractional-order (1) and (2). It uses a quadrature methods finite difference approximations and two-point block approach, with some modifications.

3.1 First Algorithm:

Recall equation (1) for strictly decreasing the fractional orders σ_i and β_j lies in $(0,1)$. Thus, for obtaining an approximation of the solution $y(x)$ in a given set of $(N+1)$-equally spaced grid points $x_r = x_0 + rh, (r = 0,1,\dots,N)$ with $x_0 + Nh = b$, consists in approximating the linear IFDEs-CF (1) in the discretized equations:

$$p(x_r)[y'(x)]_{x=x_r} + \sum_{i=0}^{\bar{n}-1} p_i(x_r)\left[{}_a^C D_x^{\sigma_i} y(x)\right]_{x=x_r} + p_{\bar{n}}(x_r)y(x_r)$$
$$= f(x_r) + \lambda_0 \int_a^b \mathcal{K}_0(x_r,s)\, y(s)ds + \sum_{j=1}^{m} \lambda_j \int_a^b \mathcal{K}_j(x_r,s)\,{}_a^C D_s^{\beta_j} y(s)ds \tag{8}$$

This results in a system of $N+1$ linear algebraic equations that approximate $y(x_r)$ in $N+1$ unknowns: $\hat{y}(x_r) = \hat{y}_r^h$ depending on h-step size. Here, the classical and fractional differential parts are estimated using difference formulae, whereas the Fredholm integral part in (8) is approximated using the quadrature formula: Trapezoidal method. The following classification was created by using the (3) rule to evaluate each integral part in equation (8) for each $r = 0,1,\dots,N$ and considering the formulas (4, 5 and 6) with the lemmas 3 and 4, respectively. This represents the Caputo-fractional order for any continuous function at the starting point $x = x_0 = a$ equal to zero, using lemma 4, it is noted that the second component vanishes. For $r = 0$, we first get:

$$p_0[y'(x)]_{x=x_0} + p_{\bar{n},0}y_0 = f_0 + \lambda_0 \int_a^b \mathcal{K}_0(x_0,s)\, y(s)ds + \sum_{j=1}^{m} \lambda_j \int_a^b \mathcal{K}_j(x_0,s)\,{}_a^C D_s^{\beta_j} y(s)ds$$

Applying the three-point forward difference formula (4) to the classical first derivative and setting $\kappa = 0$ yields equation (9), which is represented by $(N,0)$-Forward formula:

$$\frac{p_0}{2h}[-3\hat{y}_0^h + 4\hat{y}_1^h - \hat{y}_2^h] + p_{\bar{n},0}\hat{y}_0^h$$
$$= f_0 + \frac{\lambda_0 h}{2}\mathcal{K}_0^{0,0}\hat{y}_0^h + \lambda_0 h \sum_{\ell=1}^{N-1} \mathcal{K}_0^{0,\ell}\hat{y}_\ell^h + \frac{\lambda_0 h}{2}\mathcal{K}_0^{0,N}\hat{y}_N^h$$
$$+ h\sum_{j=1}^{m} \lambda_j \mathcal{A}_\beta(j)\left\{\sum_{\ell=1}^{N-1} \mathcal{K}_j^{0,\ell}\left[\sum_{\mathbb{p}=0}^{\ell-1}(\hat{y}_{\ell-\mathbb{p}}^h - \hat{y}_{\ell-\mathbb{p}-1}^h)\delta_{\mathbb{p}}^{\beta_j}\right]\right.$$
$$\left. + \frac{1}{2}\mathcal{K}_j^{0,N}\left[\sum_{\mathbb{p}=0}^{N-1}(\hat{y}_{N-\mathbb{p}}^h - \hat{y}_{N-\mathbb{p}-1}^h)\delta_{\mathbb{p}}^{\beta_j}\right]\right\} \tag{9}$$

Assuming that:

$$\mathcal{A}_\beta(j) = \frac{h^{-\beta_j}}{\Gamma(2-\beta_j)} \tag{10}$$

Clearly that $\mathcal{K}_\ell^{r,p} = \mathcal{K}_\ell(t_r, s_p)$ for all ℓ, r and p, and $\hat{y}_r^h$ is the approximate value of $y(x_r)$ depending on step size h. Also, $p_i = p(x_i)$ and $p_{i,j} = p_i(x_j)$ for all i, j. Furthermore, $\delta_{\mathbb{p}}^{\beta_j}$ defined in lemma 3. Second, for $r = 1,2,\dots,N-1$, also using formula (5) with lemma 3 and 4 to equation (8), we obtain equation (11) which is represented by (N,r)-Central formula:

$$\frac{p_r}{2h}[\hat{y}^h_{r+1}-\hat{y}^h_{r-1}]+\sum_{i=0}^{\bar{n}-1}p_{i,r}\mathcal{A}_\sigma(i)\left[\sum_{\mathbb{q}=0}^{r-1}\left(\hat{y}^h_{r-\mathbb{q}}-\hat{y}^h_{r-\mathbb{q}-1}\right)\delta^{\sigma_i}_{\mathbb{q}}\right]+p_{\bar{n},r}\hat{y}^h_r$$
$$=f_r+\frac{\lambda_0 h}{2}\mathcal{K}_0^{r,0}\hat{y}^h_0+\lambda_0 h\sum_{\ell=1}^{N-1}\mathcal{K}_0^{r,\ell}\hat{y}^h_\ell+\frac{\lambda_0 h}{2}\mathcal{K}_0^{r,N}\hat{y}^h_N$$
$$+h\sum_{j=1}^{m}\lambda_j\mathcal{A}_\beta(j)\left\{\sum_{\ell=1}^{N-1}\mathcal{K}_j^{r,\ell}\left[\sum_{\mathbb{p}=0}^{\ell-1}\left(\hat{y}^h_{\ell-\mathbb{p}}-\hat{y}^h_{\ell-\mathbb{p}-1}\right)\delta^{\beta_j}_{\mathbb{p}}\right]\right.$$
$$\left.+\frac{1}{2}\mathcal{K}_j^{r,N}\left[\sum_{\mathbb{p}=0}^{N-1}\left(\hat{y}^h_{N-\mathbb{p}}-\hat{y}^h_{N-\mathbb{p}-1}\right)\delta^{\beta_j}_{\mathbb{p}}\right]\right\} \tag{11}$$

Finally, for $r=N$, also using formula (6) with lemma 3 and 4 to equation (8), we obtain equation (12) which is represented by (N,N)-Backward formula:

$$\frac{p_N}{2h}[\hat{y}^h_{N-2}-4\hat{y}^h_{N-1}+3\hat{y}^h_N]+\sum_{i=0}^{\bar{n}-1}p_{i,N}\mathcal{A}_\sigma(i)\left[\sum_{\mathbb{q}=0}^{N-1}\left(\hat{y}^h_{N-\mathbb{q}}-\hat{y}^h_{N-\mathbb{q}-1}\right)\delta^{\sigma_i}_{\mathbb{q}}\right]+p_{\bar{n},N}\hat{y}^h_N$$
$$=f_N+\frac{\lambda_0 h}{2}\mathcal{K}_0^{N,0}\hat{y}^h_0+\lambda_0 h\sum_{\ell=1}^{N-1}\mathcal{K}_0^{N,\ell}\hat{y}^h_\ell+\frac{\lambda_0 h}{2}\mathcal{K}_0^{N,N}\hat{y}^h_N$$
$$+h\sum_{j=1}^{m}\lambda_j\mathcal{A}_\beta(j)\left\{\sum_{\ell=1}^{N-1}\mathcal{K}_j^{N,\ell}\left[\sum_{\mathbb{p}=0}^{\ell-1}\left(\hat{y}^h_{\ell-\mathbb{p}}-\hat{y}^h_{\ell-\mathbb{p}-1}\right)\delta^{\beta_j}_{\mathbb{p}}\right]\right.$$
$$\left.+\frac{1}{2}\mathcal{K}_j^{N,N}\left[\sum_{\mathbb{p}=0}^{N-1}\left(\hat{y}^h_{N-\mathbb{p}}-\hat{y}^h_{N-\mathbb{p}-1}\right)\delta^{\beta_j}_{\mathbb{p}}\right]\right\} \tag{12}$$

We developed a linear algebraic system of equations using equations (9), (11) and (12) that may be expressed in matrix form (13) as follows:

$$[L-hI]\hat{Y}^h=F \tag{13}$$

where

$$F=[f_0 \quad f_1 \quad \cdots \quad f_N]^T \text{ and } \hat{Y}^h=[\hat{y}^h_0 \quad \hat{y}^h_1 \quad \cdots \quad \hat{y}^h_N]^T$$

The expression for L, which is a square matrix of dimension $(N+1)$, is as follows:

$$L=L^{\mathcal{P}}+L^{\mathcal{A}}+\frac{1}{2h}L^{\mathcal{G}}$$

where

$$L^{\mathcal{P}}=\left[diag[p_{\bar{n},0} \quad p_{\bar{n},1} \quad \cdots \quad p_{\bar{n},N}]\right]_{N+1} \tag{14}$$

The expression for $L^{\mathcal{A}}=\left[L^{\mathcal{A}}_{k\ell}\right]_{N+1\times N+1}$, for each $k,\ell=0,1,\dots,N$ which is a lower triangular matrix of dimension $(N+1)$, is as follows:

$$\left.\begin{array}{ll}
{L^{\mathcal{A}}}_{k\ell}=0, & \text{for all } k<\ell \\
L^{\mathcal{A}}_{kk}=\begin{cases}\sum_{i=0}^{\bar{n}-1}p_{i,k}\mathcal{A}_\sigma(i) & \text{for all } k=\overline{1:N} \\ 0 & \text{o. w.}\end{cases} & \\
L^{\mathcal{A}}_{k0}=-\sum_{i=0}^{\bar{n}-1}\mathcal{P}_{i,k}A_\sigma(i)\delta^{\sigma_i}_{k-1} & \text{for all } k>0;\, k=\overline{1:N} \\
L^{\mathcal{A}}_{k\ell}=\sum_{i=0}^{\bar{n}-1}\mathcal{P}_{i,k}A_\sigma(i)C^{\sigma_i}_{k-\ell} & \text{for all } k>\ell;\, =\overline{2:N} \text{ and } \ell=\overline{1:k-1}
\end{array}\right\} \tag{15}$$

and the coefficients δ_ℓ^α and $C_\ell^\alpha\left(\ell=\overline{0:N}\right)$ for any real number $\alpha=\sigma$ and β defined as:

$$\begin{array}{c}\delta_\ell^\alpha=(1+\ell)^{1-\alpha}-\ell^{1-\alpha} \quad \text{and} \quad \delta_0^\alpha=1 \\ C_\ell^\alpha=\delta_\ell^\alpha-\delta_{\ell-1}^\alpha;C_0^\alpha=1 \text{ and assume } \delta_{-i}^\alpha=0, \qquad \forall i\in\mathbb{Z}^+\end{array} \tag{16}$$

However, the matrix $L^{\mathcal{G}}=\left[L_{k\ell}^{\mathcal{G}}\right]_{N+1\times N+1}$ defines each element $L_{k\ell}^{\mathcal{G}}$ for every $k,\ell=\overline{0:N}$ as follows:

$$\left.\begin{array}{l} L_{0\ell}^{\mathcal{G}}=\begin{cases}-3p_0 & \text{if } \ell=0\\ 4p_0 & \text{if } \ell=1\\ -p_0 & \text{if } \ell=2\\ 0 & \text{if } \ell>2\end{cases} \\ L_{N\ell}^{\mathcal{G}}=\begin{cases}p_N & \text{if } \ell=N-2\\ -4p_N & \text{if } \ell=N-1\\ 3p_N & \text{if } \ell=N\\ 0 & \text{if } \ell<N-2\end{cases} \\ \text{for each } k=1,2,\dots,N-1: \\ L_{k\ell}^{\mathcal{G}}=\begin{cases}-p_k & \text{if } k=\ell-1\\ p_k & \text{if } k=\ell+1\\ 0 & \begin{matrix}\text{o.w.}\\ \{k=\ell,k>\ell+1,k<\ell-1\}\end{matrix}\end{cases}\end{array}\right\} \tag{17}$$

Moreover, the expression for I, which is a square matrix of dimension $(N+1)$, is as follows:

$$I=\lambda_0 I^{\mathcal{K}}+I^{\mathcal{A}}$$

$I^{\mathcal{K}}=\left[I_{k\ell}^{\mathcal{K}}\right]_{N+1\times N+1}$ has the following expression for each $k,\ell=0,1,\dots,N$ that is a first kernel $(\mathcal{K}_0)$-matrix of dimension $(N+1)$:

$$I_{k\ell}^{\mathcal{K}}=\mathcal{T}_\ell\mathcal{K}_0^{k\ell} \tag{18}$$

Additionally, the square matrix of dimension $N+1$ is $I^{\mathcal{A}}=\left[I_{k\ell}^{\mathcal{A}}\right]_{N+1\times N+1}$, and for any $k,\ell=\overline{0:N}$, define each element $I_{k\ell}^{\mathcal{A}}$ as follows:

$$\left.\begin{array}{l} I_{k,0}^{\mathcal{A}}=-\sum_{j=1}^{m}\lambda_j A_\beta(j)\left[\sum_{d=1}^{N}\mathcal{T}_d^*\mathcal{K}_j^{k,d}\delta_{d-1}^{\beta_j}\right] \\ \text{for all } \ell=1,2,\dots,N-1 \\ I_{k,\ell}^{\mathcal{A}}=\sum_{j=1}^{m}\lambda_j A_\beta(j)\left[\sum_{d=\ell}^{N}\mathcal{T}_d^*\mathcal{K}_j^{k,d}C_{d-\ell}^{\beta_j}\right] \\ I_{k,N}^{\mathcal{A}}=\frac{1}{2}\sum_{j=1}^{m}\lambda_j A_\beta(j)\mathcal{K}_j^{k,N}\end{array}\right\} \tag{19}$$

where,

$$\mathcal{T}_\ell=\begin{cases}1/2 & \text{if} \quad \ell=0 \text{ or } N\\ 1 & \text{if} \quad o.w.\end{cases} \quad \text{and} \quad \mathcal{T}_d^*=\begin{cases}1/2 & \text{if} \quad d=N\\ 1 & \text{if} \quad o.w.\end{cases}$$

Finally, by creating a new row in the system (13) in matrix form, this technique adds a boundary condition (2) of equation (1), which offers:

$$B\hat{Y}^h=V \tag{20}$$

Where $B=\begin{bmatrix}g_{00} & 0 & \cdots & 0 & h_{00}\end{bmatrix}_{N+1}$, $\hat{Y}^h=\begin{bmatrix}\hat{y}_0^h & \hat{y}_1^h & \cdots & \hat{y}_N^h\end{bmatrix}^T$ and $V=[\vartheta_0]$ creating a new matrix by adding (20) to (13), produces

$$D\hat{Y}^h=E \tag{21}$$

where

$$D=\begin{bmatrix}L-hI\\ \dots\dots\\ B\end{bmatrix}_{(N+2)\times(N+1)} \quad \text{and} \quad E=\begin{bmatrix}F\\ \dots\\ V\end{bmatrix}_{(N+2)\times 1}$$

To determine the approximate column vector $\hat{Y}^h$'s in equation (21), store the matrix D and compute $D^T D$ and $D^T E$ to obtain the new square matrices then use any numerical iterative procedure to solve $[D^T D]$ and $[D^T E]$. Then the approximate solution for all $\hat{y}_i$ at each point x_i $(i = 0,1, \dots, \mathrm{N})$ is obtained for fractional order linear FIDEs-CF is called y-approximate $\hat{y}$ to $y(x)$ depending on small step size h.

The Algorithm (ATFIDE-CF). The approximate solution for mixed classical and fractional order integro-differential equations of the Fredholm type with variable coefficients (FIDEs-CF) utilizing the finite difference approximation and the closed Newton-Cotes formula (Trapezoidal rule) is summarized in the following steps:

Step 1.
a. Input $N \in \mathbb{Z}^+$, take $h = (b-a)/N$ and $x_r = a + rh, r = \overline{0{:}\,N}$.
b. Input the coefficients of boundary conditions g_{00}, h_{00} and ϑ_0.
Step 2. For each $k = 0,1, \dots, \ell$ computing $\mathcal{A}_\alpha(k)$ for all fractional orders $\alpha = \sigma$ and β, and $\ell = \bar{n} - 1$ or m, respectively, applied equation (10).
Step 3. For all $\ell = 0,1, \dots, N$ find the constant coefficients (δ_ℓ^α and C_ℓ^α) for fractional orders $\alpha = \sigma$ and β respectively using equation (16).
Step 4. Utilizing equations (14, 15, and 17) and steps (2 and 3), determine the values of each element $L^{\mathcal{P}}, L^{\mathcal{A}}$ and $L^{\mathcal{G}}$, respectively. Finally, construct the matrix $L = [L_{k\ell}]_{N+1 \times N+1}$.
Step 5. The kernel values at every point are calculated using the formula $\mathcal{K}_j^{k\ell} = \mathcal{K}_j(x_k, s_\ell)$, for all $j = 0,1, \dots, m$ and $k, \ell = 0,1, \dots, N$.
Step 6. $I^{\mathcal{K}}$ and $I^{\mathcal{A}}$ are evaluated using formulae (18 and 19), respectively. Create the matrix $I = [I_{k\ell}]_{N+1 \times N+1}$ at the end.
Step 7. Using the formula $f_r = f(x_r)$, find each component of the vector F at locations $x_r = a + rh$ $(r = 0,1, \dots, N)$.
Step 8. Establishing boundary conditions g_{00}, h_{00} and ϑ_0 into matrices B and ϑ to form (20).
Step 9. Construct the matrices D and E which are represented in system (21).
Step 10. To calculate $\hat{Y}^h$, or the column-approximate values of the exact solution (Y), use the numerical iterative approach for the system that is obtained in step 9 after multiplying both sides by D^T.

3.2 Second Algorithm:

The two-point block method provides a useful approach to a number of problems. For differential equations, the two-point block method was explained, and for integral equations, it was provided. This method will thus simultaneously generate approximations $\hat{y}_{n+1}^h$ and $\hat{y}_{n+2}^h$ to $y(x_{n+1})$ and $y(x_{n+2})$ depending on step size h respectively, at each selected κ-th blocks. Let $\hat{\mathbb{Y}}_{\rho,r}, \rho = 1,2$, denoted an approximation to $y(x_{\rho+n})$ which is locally in error by a term of order h^{r+2} i.e. $y(x_{\rho+n}) - \hat{\mathbb{Y}}_{\rho,r} = O(h^{r+2})$, [13], [14] and [24].

Here, we want to solve our Mixed-order (classical and fractional) IDEs of Fredholm type problem formed in equation (8) from the predictors valued $\hat{Y}^h$, represented by ${}^p\hat{Y}^h = \begin{bmatrix} {}^p\hat{y}_0^h & {}^p\hat{y}_1^h & \cdots & {}^p\hat{y}_N^h \end{bmatrix}^T$, the output of algorithm ATFIDE-CF, to the final point. Let $x \in [a = x_0, x_N = b]$ and the interval $[x_0, x_N]$ is discretized for the numerical technique as $\{x_0, x_1, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_N\} \subset \mathbb{R}$ depending on the step size h, $x_\iota = x_0 + \iota h, (\iota = 0,1, \dots, N)$ with $x_0 + Nh = b$. The k-th block in Figure 1 has three points: x_n^k, x_{n+1}^k and x_{n+2}^k. The suggested approach is to utilize the predictor values in vector ${}^p\hat{Y}^h$, depending on the step size h, to compute the approximation values of ${}^c\hat{y}_{n+1}^h$ and ${}^c\hat{y}_{n+2}^h$ simultaneously. The calculating method is carried out in one single iteration on this block. Similarly, there are three locations in the $(k+1)$-th block: $x_{n+2}^k = x_n^{k+1}, x_{n+1}^{k+1}$ and x_{n+2}^{k+1} and so on for all other blocks. Using the new values of ${}^p\hat{Y}^h$, enter ${}^p\hat{y}_{n+1}^h$ and ${}^p\hat{y}_{n+2}^h$ into all correcting values ${}^c\hat{y}_{n+1}^h$ and ${}^c\hat{y}_{n+2}^h$, and continued.

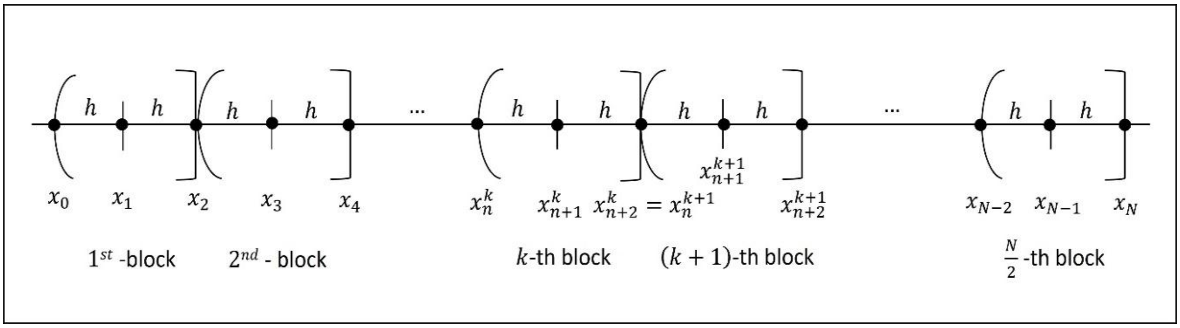

Figure 1. Discretization of a two-point block method

In this study, the numerical solution for mixed classical and fractional order integro-differential equations of the Fredholm type with variable coefficients (FIDEs-CF) was found using two-point block approaches. In the context of the current study, these formulas show how the two-point block technique is applied. Rewrite equation (1) as follows, keeping in mind the initial form:

$$\frac{dy}{dx} = \mathbb{F}\big(x, y(x), \mathbb{D}y(x), \mathbb{I}y(x)\big) \tag{22}$$

where

$$\mathbb{F}\big(x, y(x), \mathbb{D}y(x), \mathbb{I}y(x)\big) = \bar{f}(x) + \bar{p}_{\bar{n}}(x)y(x) + \sum_{i=0}^{\bar{n}-1} \bar{p}_i(x)\,{}_a^C D_x^{\sigma_i} y(x) + \lambda_0 I[Q_0(x,s);s] + \sum_{j=1}^{m} \lambda_j I\big[Q_j(x,s);s\big] \tag{23}$$

and, for each $i = 0,1,\dots,\bar{n}-1;\ j = 0,1,\dots,m$:

$$\bar{f}(x) = \frac{f(x)}{p(x)}, \qquad \bar{p}_i(x) = \frac{-p_i(x)}{p(x)}, \qquad \bar{p}_{\bar{n}}(x) = \frac{-p_{\bar{n}}(x)}{p(x)}, \qquad \bar{\mathcal{K}}_0(x,s) = \frac{\mathcal{K}_0(x,s)}{p(x)}, \quad \bar{\mathcal{K}}_j(x,s) = \frac{\mathcal{K}_j(x,s)}{p(x)}$$

while,

$$\left.\begin{array}{cc} Q_0(x,s) = \bar{\mathcal{K}}_0(x,s)y(s) & Q_j(x,s) = \bar{\mathcal{K}}_j(x,s)\,{}_a^C D_s^{\beta_j} y(s) \\ \text{and} & \text{and} \\ I[Q_0(x,s);s] = \int_a^b Q_0(x,s)ds & I\big[Q_j(x,s);s\big] = \int_a^b Q_j(x,s)ds \end{array}\right\} \tag{24}$$

Therefore, calculating the result of the integration of equation (24) numerically, using the Trapezoidal approach as in equation (3) with aid of lemma 4, produces for input $\bar{N}$ numbers of sub-intervals on the hole interval $[a,b]$ at any select collocation points say $x = x_r = x_0 + rh; r = \overline{0:\bar{N}}$. Therefore, using lemma 3, we can construct the integral part at all collocation points and we get approximately the following:

$$\begin{aligned} I_{\bar{N}}[Q_0(x_r,s);s] &= \int_a^b \bar{\mathcal{K}}_0(x_r,s)y(s)ds = Trap\,[Q_0(x_r,s),a,b,\bar{N}] \\ &= \frac{h}{2}\left[\bar{\mathcal{K}}_0^{r,0}\hat{y}_0^h + 2\sum_{\ell=1}^{\bar{N}-1}\bar{\mathcal{K}}_0^{r,\ell}\hat{y}_\ell^h + \bar{\mathcal{K}}_0^{r,\bar{N}}\hat{y}_{\bar{N}}^h\right] \end{aligned} \tag{25}$$

and, for each $j = 0,1,\dots,m$:

$$
\begin{aligned}
I_{\bar{N}}\left[Q_j(x_r,s);s\right] &= \int_a^b \bar{\mathcal{K}}_j(x_r,s)\,{}_a^C D_s^{\beta_j} y(s)ds = Trap\left[Q_j(x_r,s),a,b,\bar{N}\right] \\
&= \frac{h^{1-\beta_j}}{\Gamma(2-\beta_j)}\left[\sum_{\ell=1}^{\bar{N}-1} \bar{\mathcal{K}}_j^{r,\ell}\left(\sum_{p=0}^{\ell-1}\left[\hat{y}_{\ell-p}^h - \hat{y}_{\ell-p-1}^h\right]\delta_p^{\beta_j}\right)\right. \\
&\left. + \frac{1}{2}\bar{\mathcal{K}}_j^{r,\bar{N}}\left(\sum_{p=0}^{\bar{N}-1}\left[\hat{y}_{\bar{N}-p}^h - \hat{y}_{\bar{N}-p-1}^h\right]\delta_p^{\beta_j}\right)\right] \qquad (26)
\end{aligned}
$$

For every $i = 0,1,\dots,\bar{n}-1$ at every collocation point $\{x_r\}$, we may construct the σ_i-fractional derivative part using lemmas 3 and 4, and we obtain approximately the following:

$$
FDD[y(x_r),\sigma_i] = \left[{}_a^C D_x^{\sigma_i} y(x)\right]_{x=x_r} = \begin{cases} 0 & if\ r = 0 \\ \mathcal{A}_\sigma(i)\sum_{q=0}^{r-1}\left(\hat{y}_{r-q}^h - \hat{y}_{r-q-1}^h\right)\delta_q^{\sigma_i} & if\ r \geq 1 \end{cases} \qquad (27)
$$

Now, the objective is to use a fourth-order, six-stage block technique to obtain approximate solutions for a differential problem within intervals $[x_n, x_{n+2}]$ at fixed block say block κ that is $[x_n^\kappa, x_{n+2}^\kappa]$. The differential problem (22) will be approached using the same approach that was created for problem (1). We shall make our presentation as though we were dealing only with (22) and represented given values as predictors ${}^p\hat{y}_0^h$, ${}^p\hat{y}_1^h$ and ${}^p\hat{y}_2^h$, the output of algorithm ATFIDE-CF. Therefore, let's:

$$
\left.\begin{aligned}
\hat{\mathbb{Y}}_{0,r} &= {}^p\hat{y}_n^h, \\
\hat{\mathbb{Y}}'_{0,r} &= \mathbb{F}\left(x_n^\kappa, {}^p\hat{y}_n^h, [\mathbb{D}y(x)]_{x=x_n^\kappa}, [\mathbb{I}y(x)]_{x=x_n^\kappa}\right),
\end{aligned}\right\}\text{for all } r = 0,1,2,\dots \qquad (28.A)
$$

The procedure uses equations (7) in the following sequence of steps:

$$
\left.\begin{aligned}
\hat{\mathbb{Y}}_{1,0} &= {}^p\hat{y}_n^h + h\hat{\mathbb{Y}}'_{0,0}, \\
\hat{\mathbb{Y}}'_{1,0} &= \mathbb{F}\left(x_{n+1}^\kappa, \hat{\mathbb{Y}}_{1,0}, [\mathbb{D}y(x)]_{x=x_{n+1}^\kappa}, [\mathbb{I}y(x)]_{x=x_{n+1}^\kappa}\right),
\end{aligned}\right\} \qquad (28.B)
$$

$$
\left.\begin{aligned}
\hat{\mathbb{Y}}_{1,1} &= {}^p\hat{y}_n^h + \frac{h}{2}\left[\hat{\mathbb{Y}}'_{1,0} + \hat{\mathbb{Y}}'_{0,0}\right], \\
\hat{\mathbb{Y}}'_{1,1} &= \mathbb{F}\left(x_{n+1}^\kappa, \hat{\mathbb{Y}}_{1,1}, [\mathbb{D}y(x)]_{x=x_{n+1}^\kappa}, [\mathbb{I}y(x)]_{x=x_{n+1}^\kappa}\right),
\end{aligned}\right\} \qquad (28.C)
$$

$$
\left.\begin{aligned}
\hat{\mathbb{Y}}_{2,1} &= {}^p\hat{y}_n^h + 2h\hat{\mathbb{Y}}'_{1,1}, \\
\hat{\mathbb{Y}}'_{2,1} &= \mathbb{F}\left(x_{n+2}^\kappa, \hat{\mathbb{Y}}_{2,1}, [\mathbb{D}y(x)]_{x=x_{n+2}^\kappa}, [\mathbb{I}y(x)]_{x=x_{n+2}^\kappa}\right),
\end{aligned}\right\} \qquad (28.D)
$$

$$
\left.\begin{aligned}
\hat{\mathbb{Y}}_{1,2} &= {}^p\hat{y}_n^h + \frac{h}{12}\left[-\hat{\mathbb{Y}}'_{2,1} + 8\hat{\mathbb{Y}}'_{1,1} + 5\hat{\mathbb{Y}}'_{0,1}\right], \\
\hat{\mathbb{Y}}'_{1,2} &= \mathbb{F}\left(x_{n+1}^\kappa, \hat{\mathbb{Y}}_{1,2}, [\mathbb{D}y(x)]_{x=x_{n+1}^\kappa}, [\mathbb{I}y(x)]_{x=x_{n+1}^\kappa}\right),
\end{aligned}\right\} \qquad (28.E)
$$

$$
\left.\begin{aligned}
\hat{\mathbb{Y}}_{2,2} &= {}^p\hat{y}_n^h + \frac{h}{3}\left[\hat{\mathbb{Y}}'_{2,1} + 4\hat{\mathbb{Y}}'_{1,2} + \hat{\mathbb{Y}}'_{0,2}\right], \\
\hat{\mathbb{Y}}'_{2,2} &= \mathbb{F}\left(x_{n+2}^\kappa, \hat{\mathbb{Y}}_{2,2}, [\mathbb{D}y(x)]_{x=x_{n+2}^\kappa}, [\mathbb{I}y(x)]_{x=x_{n+2}^\kappa}\right), \\
\hat{\mathbb{Y}}_{2,3} &= {}^p\hat{y}_n^h + \frac{h}{3}\left[\hat{\mathbb{Y}}'_{2,2} + 4\hat{\mathbb{Y}}'_{1,2} + \hat{\mathbb{Y}}'_{0,2}\right].
\end{aligned}\right\} \qquad (28.F)
$$

Let us now consider (28 A to F) from a different notational viewpoint. Let:

$$
\aleph_1^\kappa = \mathbb{F}\left(x_n^\kappa, {}^p\hat{y}_n^h, [\mathbb{D}y(x)]_{x=x_n^\kappa}, [\mathbb{I}y(x)]_{x=x_n^\kappa}\right), \quad \aleph_2^\kappa = \hat{\mathbb{Y}}'_{1,0}, \quad \aleph_3^\kappa = \hat{\mathbb{Y}}'_{1,1}
$$
$$
\aleph_4^\kappa = \hat{\mathbb{Y}}'_{2,1} \qquad \aleph_5^\kappa = \hat{\mathbb{Y}}'_{1,2} \quad \aleph_6^\kappa = \hat{\mathbb{Y}}'_{2,2}
$$

Then (28 A to F) may be written in the formulas (29) and (30):

$$\left.\begin{aligned}
\aleph_1^\kappa &= \mathbb{F}\left(x_n^\kappa,\ {}^p\hat{y}_n^h, [\mathbb{D}y(x)]_{x=x_n^\kappa}, [\mathbb{I}y(x)]_{x=x_n^\kappa}\right)\\
\aleph_2^\kappa &= \mathbb{F}\left(x_n^\kappa + h,\ {}^p\hat{y}_n^h + h\aleph_1^\kappa, [\mathbb{D}y(x)]_{x=x_n^\kappa+h}, [\mathbb{I}y(x)]_{x=x_n^\kappa+h}\right)\\
\aleph_3^\kappa &= \mathbb{F}\left(x_n^\kappa + h,\ {}^p\hat{y}_n^h + \frac{h}{2}[\aleph_1^\kappa + \aleph_2^\kappa], [\mathbb{D}y(x)]_{x=x_n^\kappa+h}, [\mathbb{I}y(x)]_{x=x_n^\kappa+h}\right)\\
\aleph_4^\kappa &= \mathbb{F}\left(x_n^\kappa + 2h,\ {}^p\hat{y}_n^h + 2h\aleph_3^\kappa, [\mathbb{D}y(x)]_{x=x_n^\kappa+2h}, [\mathbb{I}y(x)]_{x=x_n^\kappa+2h}\right)\\
\aleph_5^\kappa &= \mathbb{F}\left(x_n^\kappa + h,\ {}^p\hat{y}_n^h + \frac{h}{12}[5\aleph_1^\kappa + 8\aleph_3^\kappa - \aleph_4^\kappa], [\mathbb{D}y(x)]_{x=x_n^\kappa+h}, [\mathbb{I}y(x)]_{x=x_n^\kappa+h}\right)\\
\aleph_6^\kappa &= \mathbb{F}\left(x_n^\kappa + 2h,\ {}^p\hat{y}_n^h + \frac{h}{3}[\aleph_1^\kappa + \aleph_4^\kappa + 4\aleph_5^\kappa], [\mathbb{D}y(x)]_{x=x_n^\kappa+2h}, [\mathbb{I}y(x)]_{x=x_n^\kappa+2h}\right)
\end{aligned}\right\} \tag{29}$$

For each $\kappa(=1,2,\dots,N/2)$-th block, the six-stage two-block technique can therefore be expressed as follows:

$$\left.\begin{aligned}
{}^c\hat{y}_n^h &= {}^p\hat{y}_n^h\\
{}^c\hat{y}_{n+1}^h &= {}^c\hat{y}_n^h + \frac{h}{12}[5\aleph_1^\kappa + 8\aleph_3^\kappa - \aleph_4^\kappa]\\
{}^c\hat{y}_{n+2}^h &= {}^c\hat{y}_n^h + \frac{h}{3}[\aleph_1^\kappa + 4\aleph_5^\kappa + \aleph_6^\kappa]
\end{aligned}\right\} \tag{30}$$

Equations (30) and (29) are applied numerically for a 1-block expansion. With $\kappa = 1$, the first block utilizing the points x_0^1, x_1^1 and x_2^1 referred to as x_0, x_1 and x_2 on interval $[x_0, x_2]$, with values of ${}^p\hat{y}_0^h$, ${}^p\hat{y}_1^h$ and ${}^p\hat{y}_2^h$, respectively, apply the algorithm ATFIDE-CF. After determining the expected value at the first point, ${}^c\hat{y}_n^h$, equation (30) is used to determine the values at the following points, ${}^c\hat{y}_1^h$ and ${}^c\hat{y}_2^h$. For this first block, we determine all values of $\aleph_i^1(i = 1,2,\dots,6)$, for definition of function $\mathbb{F}$ using equation (23) with equations (25 and 26) for numerical integrations (take $\bar{N} = N$) and property of Caputo derivative lemma 4, so:

$$\begin{aligned}
\aleph_1^1 &= \mathbb{F}\left(x_0,\ {}^p\hat{y}_0^h, [\mathbb{D}y(x)]_{x=x_0}, [\mathbb{I}y(x)]_{x=x_0}\right)\\
&= \bar{f}(x_0) + \bar{p}_{\bar{n},0}\,{}^p\hat{y}_0^h + \lambda_0 Trap\,[Q_0(x_0,s),a,b,N] + \sum_{j=1}^{m}\lambda_j Trap\,[Q_j(x_0,s),a,b,N]
\end{aligned}$$

$$\begin{aligned}
\aleph_2^1 &= \mathbb{F}\left(x_0^1 + h,\ {}^p\hat{y}_0^h + h\aleph_1^1, [\mathbb{D}y(x)]_{x=x_0^1+h}, [\mathbb{I}y(x)]_{x=x_0^1+h}\right)\\
&= \bar{f}(x_1) + \bar{p}_{\bar{n},1}\left[{}^p\hat{y}_0^h + h\aleph_1^1\right] + \sum_{i=0}^{\bar{n}-1}\bar{p}_{i,1}FDD[y(x_1),\sigma_i] + \lambda_0 Trap\,[Q_0(x_1,s),a,b,N]\\
&\quad + \sum_{j=1}^{m}\lambda_j Trap\,[Q_j(x_1,s),a,b,N]
\end{aligned}$$

$$\begin{aligned}
\aleph_3^1 &= \mathbb{F}\left(x_0^1 + h,\ {}^p\hat{y}_0^h + \frac{h}{2}[\aleph_1^1 + \aleph_2^1], [\mathbb{D}y(x)]_{x=x_0^1+h}, [\mathbb{I}y(x)]_{x=x_0^1+h}\right)\\
&= \bar{f}(x_1) + \bar{p}_{\bar{n},1}\left[{}^p\hat{y}_0^h + \frac{h}{2}[\aleph_1^1 + \aleph_2^1]\right] + \sum_{i=0}^{\bar{n}-1}\bar{p}_{i,1}FDD[y(x_1),\sigma_i] + \lambda_0 Trap\,[Q_0(x_1,s),a,b,N]\\
&\quad + \sum_{j=1}^{m}\lambda_j Trap\,[Q_j(x_1,s),a,b,N]
\end{aligned}$$

$$\begin{aligned}
\aleph_4^1 &= \mathbb{F}\left(x_0^1 + 2h,\ {}^p\hat{y}_0^h + 2h\aleph_3^1, [\mathbb{D}y(x)]_{x=x_0^1+2h}, [\mathbb{I}y(x)]_{x=x_0^1+2h}\right)\\
&= \bar{f}(x_2) + \bar{p}_{\bar{n},2}\left[{}^p\hat{y}_0^h + 2h\aleph_3^1\right] + \sum_{i=0}^{\bar{n}-1}\bar{p}_{i,2}FDD[y(x_2),\sigma_i] + \lambda_0 Trap\,[Q_0(x_2,s),a,b,N]\\
&\quad + \sum_{j=1}^{m}\lambda_j Trap\,[Q_j(x_2,s),a,b,N]
\end{aligned}$$

$$\aleph_5^1 = \mathbb{F}\left(x_0^1 + h,\ {}^p\hat{y}_0^h + \frac{h}{12}[5\aleph_1^1 + 8\aleph_3^1 - \aleph_4^1], [\mathbb{D}y(x)]_{x=x_0^1+h}, [\mathbb{I}y(x)]_{x=x_0^1+h}\right)$$

$$= \bar{f}(x_1) + \bar{p}_{\bar{n},1}\left[{}^p\hat{y}_0^h + \frac{h}{12}[5\aleph_1^1 + 8\aleph_3^1 - \aleph_4^1]\right] + \sum_{i=0}^{\bar{n}-1} \bar{p}_{i,1} FDD[y(x_1), \sigma_i]$$

$$+ \lambda_0 Trap\,[Q_0(x_1, s), a, b, N] + \sum_{j=1}^{m} \lambda_j Trap\,[Q_j(x_1, s), a, b, N]$$

$$\aleph_6^1 = \mathbb{F}\left(x_0^1 + 2h,\ {}^p\hat{y}_0^h + \frac{h}{3}[\aleph_1^1 + \aleph_4^1 + 4\aleph_5^1], [\mathbb{D}y(x)]_{x=x_0^1+2h}, [\mathbb{I}y(x)]_{x=x_0^1+2h}\right)$$

$$= \bar{f}(x_2) + \bar{p}_{\bar{n},1}\left[{}^p\hat{y}_0^h + \frac{h}{3}[\aleph_1^1 + \aleph_4^1 + 4\aleph_5^1]\right] + \sum_{i=0}^{\bar{n}-1} \bar{p}_{i,2} FDD[y(x_2), \sigma_i]$$

$$+ \lambda_0 Trap\,[Q_0(x_2, s), a, b, N] + \sum_{j=1}^{m} \lambda_j Trap\,[Q_j(x_2, s), a, b, N]$$

Thus, from equation (30), we obtain ${}^c\hat{y}_0^h$, ${}^c\hat{y}_1^h$ and ${}^c\hat{y}_2^h$:

$${}^c\hat{y}_0^h = {}^p\hat{y}_0^h; \quad {}^c\hat{y}_1^h = {}^c\hat{y}_0^h + \frac{h}{12}[5\aleph_1^1 + 8\aleph_3^1 - \aleph_4^1] \quad \text{and} \quad {}^c\hat{y}_2^h = {}^c\hat{y}_0^h + \frac{h}{3}[\aleph_1^1 + 4\aleph_5^1 + \aleph_6^1]$$

We use equation (23) to define the function $\mathbb{F}$ with equations (25 and 26) for numerical integrations (assume $\bar{N} = N$) and lemma 3 to approximate the Caputo derivative by finite difference in order to identify all values of $\aleph_i^2 (i = 1,2, \dots, 6)$ for the second block, i.e., $\kappa = 2$. With values of ${}^c\hat{y}_2^h$, ${}^p\hat{y}_3^h$ and ${}^p\hat{y}_4^h$, using the points x_0^2, x_1^2 and x_2^2 (also known as x_2, x_3 and x_4) on interval $[x_2, x_4]$, The ATFIDE-CF procedure is applied, respectively, with known that $\hat{Y}^h = [\ {}^c\hat{y}_0^h \quad {}^c\hat{y}_1^h \quad {}^c\hat{y}_2^h \quad {}^p\hat{y}_3^h \quad {}^p\hat{y}_4^h \quad \cdots \quad {}^p\hat{y}_{N-1}^h \quad {}^p\hat{y}_N^h]$. Thus:

$$\aleph_1^2 = \bar{f}(x_2) + \bar{p}_{\bar{n},2}\ {}^c\hat{y}_2^h + \lambda_0 Trap\,[Q_0(x_2, s), a, b, N] + \sum_{j=1}^{m} \lambda_j Trap\,[Q_j(x_2, s), a, b, N]$$

$$\aleph_2^2 = \bar{f}(x_3) + \bar{p}_{\bar{n},3}[\ {}^c\hat{y}_2^h + h\aleph_1^2] + \sum_{i=0}^{\bar{n}-1} \bar{p}_{i,3} FDD[y(x_3), \sigma_i] + \lambda_0 Trap\,[Q_0(x_3, s), a, b, N]$$

$$+ \sum_{j=1}^{m} \lambda_j Trap\,[Q_j(x_3, s), a, b, N]$$

$$\aleph_3^2 = \bar{f}(x_3) + \bar{p}_{\bar{n},3}\left[{}^c\hat{y}_2^h + \frac{h}{2}[\aleph_1^2 + \aleph_2^2]\right] + \sum_{i=0}^{\bar{n}-1} \bar{p}_{i,3} FDD[y(x_3), \sigma_i] + \lambda_0 Trap\,[Q_0(x_3, s), a, b, N]$$

$$+ \sum_{j=1}^{m} \lambda_j Trap\,[Q_j(x_3, s), a, b, N]$$

$$\aleph_4^2 = \bar{f}(x_4) + \bar{p}_{\bar{n},4}[\ {}^c\hat{y}_2^h + 2h\aleph_3^2] + \sum_{i=0}^{\bar{n}-1} \bar{p}_{i,4} FDD[y(x_4), \sigma_i] + \lambda_0 Trap\,[Q_0(x_4, s), a, b, N]$$

$$+ \sum_{j=1}^{m} \lambda_j Trap\,[Q_j(x_4, s), a, b, N]$$

$$\aleph_5^2 = \bar{f}(x_3) + \bar{p}_{\bar{n},3}\left[{}^c\hat{y}_2^h + \frac{h}{12}[5\aleph_1^2 + 8\aleph_3^2 - \aleph_4^2]\right] + \sum_{i=0}^{\bar{n}-1} \bar{p}_{i,3} FDD[y(x_3), \sigma_i] + \lambda_0 Trap\,[Q_0(x_3, s), a, b, N]$$

$$+ \sum_{j=1}^{m} \lambda_j Trap\,[Q_j(x_3, s), a, b, N]$$

$$\aleph_6^2 = \bar{f}(x_4) + \bar{p}_{\bar{n},4}\left[{}^c\hat{y}_2^h + \frac{h}{3}[\aleph_1^2 + \aleph_4^2 + 4\aleph_5^2]\right] + \sum_{i=0}^{\bar{n}-1} \bar{p}_{i,4} FDD[y(x_4), \sigma_i] + \lambda_0 Trap\,[Q_0(x_4, s), a, b, N]$$

$$+ \sum_{j=1}^{m} \lambda_j Trap\,[Q_j(x_4, s), a, b, N]$$

Thus, from equation (30), we obtain ${}^{c}\hat{y}_3^h$ and ${}^{c}\hat{y}_4^h$:

$${}^{c}\hat{y}_3^h = {}^{c}\hat{y}_2^h + \frac{h}{12}[5\aleph_1^2 + 8\aleph_3^2 - \aleph_4^2] \quad \text{and} \quad {}^{c}\hat{y}_4^h = {}^{c}\hat{y}_2^h + \frac{h}{3}[\aleph_1^2 + 4\aleph_5^2 + \aleph_6^2]$$

Calculating all corrector values for each selected block for each $\kappa = 3,4,\dots,N/2$ $\hat{Y}^h = [\,{}^{c}\hat{y}_0^h \quad {}^{c}\hat{y}_1^h \quad {}^{c}\hat{y}_2^h \quad {}^{c}\hat{y}_3^h \quad {}^{c}\hat{y}_4^h \quad \cdots \quad {}^{c}\hat{y}_{N-1}^h \quad {}^{c}\hat{y}_N^h]$.

The Algorithm (ABFIDE-CF). The following procedures provide a summary of the corrector approximate solution ${}^{c}\hat{y}_\kappa^h(\kappa = \overline{0{:}N})$ for FIDEs-CF using the finite difference approximation and the two-point block approach using Trapezoidal technique for predictor values ${}^{p}\hat{y}_\kappa^h$:

Step 1.

a. Input $N \in \mathbb{Z}^+$, take $h = (b-a)/N$ and $x_r = a + rh, r = \overline{0{:}N}$ and the number of blocks $(N/2)$.

b. Input the coefficients of boundary conditions g_{00}, h_{00} and ϑ_0.

Step 2. Execute the **ATFIDE-CF** algorithm to obtain the predictor values of ${}^{p}\hat{Y}^h = \left[{}^{p}\hat{y}_0^h \quad {}^{p}\hat{y}_1^h \quad \cdots \quad {}^{p}\hat{y}_N^h\right]^T$.

Step 3. Define the function $\mathbb{F}$, from the form (23).

- **For every $\boldsymbol{\kappa = 1,2,\dots,N/2}$, take the subsequent actions on each block interval $[\boldsymbol{x_n^\kappa, x_{n+2}^\kappa}]$:**

Step 4. Evaluate:

i- $Trap\,[Q_0(x_n^\kappa, s), a, b, N]$ and $Trap\left[Q_j(x_n^\kappa, s), a, b, N\right]$ from equations (25 and 26). Putting in $[\mathbb{I}y(x)]_{x=x_n^\kappa}$.

ii- $FDD[y(x_n^\kappa), \sigma_i]$ from equation (27). Putting in $[\mathbb{D}y(x)]_{x=x_n^\kappa}$.

Step 5. Compute the $\{\aleph_1^\kappa\}$ from equation (29).

Step 6. Return step 4 for point $x = x_n^\kappa + h$. Evaluate: $[\mathbb{I}y(x)]_{x=x_n^\kappa+h}$ and $[\mathbb{D}y(x)]_{x=x_n^\kappa+h}$.

Step 7. Compute the $\{\aleph_2^\kappa$ and $\aleph_3^\kappa\}$, respectively, from equation (29).

Step 8. Return step 4 for point $x = x_n^\kappa + 2h$. Evaluate: $[\mathbb{I}y(x)]_{x=x_n^\kappa+2h}$ and $[\mathbb{D}y(x)]_{x=x_n^\kappa+2h}$.

Step 9. Compute the $\{\aleph_4^\kappa, \aleph_5^\kappa$ and $\aleph_6^\kappa\}$, respectively, from equation (29).

Step 10. By using steps 5, 7 and 9 compute: $\{\,{}^{c}\hat{y}_{n+1}^h$ and ${}^{c}\hat{y}_{n+2}^h\}$ from equation (30).

Step 11. If $\kappa \le N/2$ go to step 4. Otherwise stop.

3.3 Third Algorithm:

In the second strategy ABFIDE-CF, the two-point block method offers a reliable predictor-corrector numerical framework for solving complicated mixed-order Fredholm integro-differential equations (FIDEs-CF). By improving the original two-point block method with superior extrapolation and correction techniques, it substantially improves the answer's accuracy. The method operates in three main stages. Initially, approximate solutions ${}^{p}\hat{y}_0^h, {}^{p}\hat{y}_1^h, \dots, {}^{p}\hat{y}_N^h$ are obtained at all grid points using the algorithm ATFIDE-CF. Using these initial values, the conventional two-point block algorithm ABFIDE-CF is used in the second stage ${}^{c}\hat{y}_0^h, {}^{c}\hat{y}_1^h, \dots, {}^{c}\hat{y}_N^h$. The process then moves into an iterative refinement phase, where the modified two block algorithm is used to gradually improve the results at each grid point. This is done by employing improved techniques for increased precision.

We turn next to methods for decreasing the step size (h) in the present block. Suppose that after $\widehat{\mathbb{Y}}_{1,1}$ in equation (28.C) we wish to halve h. We first perform $\widehat{\mathbb{Y}}'_{1,1}$ in (28.C). set $\mathbb{h} = h/2$, then the $\widehat{\mathbb{Y}}_{1,1}$ and $\widehat{\mathbb{Y}}'_{1,1}$ that we just computed are values at x_2, and so we relabel them $\widehat{\mathbb{Y}}_{2,1}$ and $\widehat{\mathbb{Y}}'_{2,1}$. i.e. set $\widehat{\mathbb{Y}}_{2,1} = \widehat{\mathbb{Y}}_{1,1}$ and $\widehat{\mathbb{Y}}'_{2,1} = \widehat{\mathbb{Y}}'_{1,1}$. After using the Hermite interpolation formula [24], to find new $\widehat{\widehat{\mathbb{Y}}}$:

$$\left.\begin{aligned} \widehat{\widehat{\mathbb{Y}}}_{1,1} &= \frac{1}{2}\left[\widehat{\mathbb{Y}}_{0,1} + \widehat{\mathbb{Y}}_{2,1}\right] + \frac{\mathbb{h}}{4}\left[\widehat{\mathbb{Y}}'_{0,1} - \widehat{\mathbb{Y}}'_{2,1}\right], \\ \widehat{\widehat{\mathbb{Y}}}'_{1,1} &= \mathbb{F}\left(\bar{x}_{n+1}^\kappa, \widehat{\widehat{\mathbb{Y}}}_{1,1}, [\mathbb{D}y(x)]_{x=\bar{x}_{n+1}^\kappa}, [\mathbb{I}y(x)]_{x=\bar{x}_{n+1}^\kappa}\right), \end{aligned}\right\} \text{where } \bar{x}_{n+q}^\kappa = x_n + q\mathbb{h} \quad (31.C)$$

$$\left.\begin{aligned} \widehat{\mathbb{Y}}_{1,2} &= {}^{c}\hat{y}_n^h + \frac{\mathbb{h}}{12}\left[-\widehat{\mathbb{Y}}'_{2,1} + 8\widehat{\widehat{\mathbb{Y}}}_{1,1} + 5\widehat{\mathbb{Y}}'_{0,1}\right], \\ \widehat{\mathbb{Y}}'_{1,2} &= \mathbb{F}\left(\bar{x}_{n+1}^\kappa, \widehat{\mathbb{Y}}_{1,2}, [\mathbb{D}y(x)]_{x=\bar{x}_{n+1}^\kappa}, [\mathbb{I}y(x)]_{x=\bar{x}_{n+1}^\kappa}\right), \end{aligned}\right\} \quad (31.E)$$

$$\left.\begin{aligned} \widehat{\mathbb{Y}}_{2,2} &= {}^{c}\hat{y}_n^h + \frac{\mathbb{h}}{3}\left[\widehat{\mathbb{Y}}'_{2,1} + 4\widehat{\mathbb{Y}}'_{1,2} + \widehat{\mathbb{Y}}'_{0,2}\right], \\ \widehat{\mathbb{Y}}'_{2,2} &= \mathbb{F}\left(\bar{x}_{n+2}^{\kappa}, \widehat{\mathbb{Y}}_{2,2}, [\mathbb{D}y(x)]_{x=\bar{x}_{n+2}^{\kappa}}, [\mathbb{I}y(x)]_{x=\bar{x}_{n+2}^{\kappa}}\right), \\ \widehat{\mathbb{Y}}_{2,3} &= {}^{c}\hat{y}_n^h + \frac{\mathbb{h}}{3}\left[\widehat{\mathbb{Y}}'_{2,2} + 4\widehat{\mathbb{Y}}'_{1,2} + \widehat{\mathbb{Y}}'_{0,2}\right]. \end{aligned}\right\} \quad (31.F)$$

Let us now consider (28 A and B) with (31 C to F) from a different notational viewpoint. Let:

$$\Omega_1^{\kappa} = \mathbb{F}\left(x_n^{\kappa}, {}^{c}\hat{y}_n^h, [\mathbb{D}y(x)]_{x=x_n^{\kappa}}, [\mathbb{I}y(x)]_{x=x_n^{\kappa}}\right), \quad \Omega_2^{\kappa} = \widehat{\mathbb{Y}}'_{1,0}, \quad \Omega_3^{\kappa} = \widehat{\mathbb{Y}}'_{1,1}$$
$$\bar{\Omega}_3^{\kappa} = \widehat{\widehat{\mathbb{Y}}}'_{1,1} \qquad \Omega_4^{\kappa} = \widehat{\mathbb{Y}}'_{1,2} \quad \Omega_5^{\kappa} = \widehat{\mathbb{Y}}'_{2,2}$$

we can rewrite the formula in the new corrector strategy as follows at block κ, the interval $[x_n^{\kappa}, x_{n+2}^{\kappa}]$. That equations (28 A and B) with (31 C to F) may be written in the formulas (32) and (33):

$$\left.\begin{aligned} \Omega_1^{\kappa} &= \mathbb{F}\left(x_n^{\kappa}, {}^{c}\hat{y}_n^h, [\mathbb{D}y(x)]_{x=x_n^{\kappa}}, [\mathbb{I}y(x)]_{x=x_n^{\kappa}}\right) \\ \Omega_2^{\kappa} &= \mathbb{F}\left(x_{n+1}^{\kappa}, {}^{c}\hat{y}_n^h + h\Omega_1^{\kappa}, [\mathbb{D}y(x)]_{x=x_{n+1}^{\kappa}}, [\mathbb{I}y(x)]_{x=x_{n+1}^{\kappa}}\right) \\ \Omega_3^{\kappa} &= \mathbb{F}\left(x_{n+1}^{\kappa}, {}^{c}\hat{y}_n^h + \frac{h}{2}[\Omega_1^{\kappa} + \Omega_2^{\kappa}], [\mathbb{D}y(x)]_{x=x_{n+1}^{\kappa}}, [\mathbb{I}y(x)]_{x=x_{n+1}^{\kappa}}\right) \\ \bar{\Omega}_3^{\kappa} &= \mathbb{F}\left(\bar{x}_{n+1}^{\kappa}, {}^{c}\hat{y}_n^h + \frac{h}{4}[\Omega_1^{\kappa} + \Omega_2^{\kappa}] + \frac{\mathbb{h}}{4}[\Omega_1^{\kappa} - \Omega_3^{\kappa}], [\mathbb{D}y(x)]_{x=\bar{x}_{n+1}^{\kappa}}, [\mathbb{I}y(x)]_{x=\bar{x}_{n+1}^{\kappa}}\right) \\ \Omega_4^{\kappa} &= \mathbb{F}\left(\bar{x}_{n+1}^{\kappa}, {}^{c}\hat{y}_n^h + \frac{\mathbb{h}}{12}[5\Omega_1^{\kappa} + 8\bar{\Omega}_3^{\kappa} - \Omega_3^{\kappa}], [\mathbb{D}y(x)]_{x=\bar{x}_{n+1}^{\kappa}}, [\mathbb{I}y(x)]_{x=\bar{x}_{n+1}^{\kappa}}\right) \\ \Omega_5^{\kappa} &= \mathbb{F}\left(\bar{x}_{n+2}^{\kappa}, {}^{c}\hat{y}_n^h + \frac{\mathbb{h}}{3}[\Omega_1^{\kappa} + \Omega_3^{\kappa} + 4\Omega_4^{\kappa}], [\mathbb{D}y(x)]_{x=\bar{x}_{n+2}^{\kappa}}, [\mathbb{I}y(x)]_{x=\bar{x}_{n+2}^{\kappa}}\right) \end{aligned}\right\} \quad (32)$$

Where $\bar{x}_{n+q}^{\kappa} = x_n + q\mathbb{h}$ such that $\mathbb{h} = h/2$ and for all $q = 0,1,2, \dots$. For each κ $(= 1,2, \dots, N/2)$-th block, the modify six-stage two-block technique can therefore be expressed as follows:

$$\left.\begin{aligned} {}^{c*}\hat{y}_n^h &= {}^{c}\hat{y}_n^h \\ {}^{c*}\hat{y}_{n+1}^h &= {}^{c*}\hat{y}_n^h + \frac{\mathbb{h}}{12}[5\Omega_1^{\kappa} + 8\bar{\Omega}_3^{\kappa} - \Omega_3^{\kappa}] \\ {}^{c*}\hat{y}_{n+2}^h &= {}^{c*}\hat{y}_n^h + \frac{\mathbb{h}}{3}[\Omega_1^{\kappa} + 4\Omega_4^{\kappa} + \Omega_5^{\kappa}] \end{aligned}\right\} \quad (33)$$

The Algorithm (AMBFIDE-CF). The following procedures provide a summary of the corrector approximate solution ${}^{c*}\hat{y}_{\kappa}^h(\kappa = \overline{0:N})$ for FIDEs-CF using the finite difference approximation and the two-point block approach ${}^{c}\hat{y}_{\kappa}^h$ using Trapezoidal technique for predictor values ${}^{p}\hat{y}_{\kappa}^h$:

Step 1.
a. Input $N \in \mathbb{Z}^+$, take $h = (b-a)/N$ and $x_r = a + rh, r = \overline{0:N}$ and the number of blocks $(N/2)$.
b. Input the coefficients of boundary conditions g_{00}, h_{00} and ϑ_0.
Step 2. Execute the **ABFIDE-CF** algorithm to obtain the predictor values of ${}^{c}\hat{Y}^h = [\,{}^{c}\hat{y}_0^h \quad {}^{c}\hat{y}_1^h \quad \cdots \quad {}^{c}\hat{y}_N^h]^T$.
Step 3. Define the function $\mathbb{F}$, from the form (23).

- **For every $\boldsymbol{\kappa = 1,2,\dots,N/2}$, take the subsequent actions on each block interval $[x_n^{\kappa}, x_{n+2}^{\kappa}]$:**

Step 4. Evaluate:
iii- $Trap\,[Q_0(x_n^{\kappa}, s), a, b, N]$ and $Trap\,[Q_j(x_n^{\kappa}, s), a, b, N]$ from equations (25 and 26). Putting in $[\mathbb{I}y(x)]_{x=x_n^{\kappa}}$.
iv- $FDD[y(x_n^{\kappa}), \sigma_i]$ from equation (27). Putting in $[\mathbb{D}y(x)]_{x=x_n^{\kappa}}$.
Step 5. Compute the $\{\Omega_1^{\kappa}\}$ from equation (32).
Step 6. Return step 4 for point $x = x_{n+1}^{\kappa}$. Evaluate: $[\mathbb{I}y(x)]_{x=x_{n+1}^{\kappa}}$ and $[\mathbb{D}y(x)]_{x=x_{n+1}^{\kappa}}$.
Step 7. Compute the $\{\Omega_2^{\kappa}$ and $\Omega_3^{\kappa}\}$, respectively, from equation (32).
Step 8. Return step 4 for point $x = \bar{x}_{n+1}^{\kappa}$. Evaluate: $[\mathbb{I}y(x)]_{x=\bar{x}_{n+1}^{\kappa}}$ and $[\mathbb{D}y(x)]_{x=\bar{x}_{n+1}^{\kappa}}$.
Step 9. Compute the $\{\bar{\Omega}_3^{\kappa}, \Omega_4^{\kappa}$ and $\Omega_5^{\kappa}\}$, respectively, from equation (32).
Step 10. By using steps 5, 7 and 9 compute: $\{{}^{c*}\hat{y}_{n+1}^h$ and ${}^{c*}\hat{y}_{n+2}^h\}$ from equation (33).

Step 11. If $\kappa \leq N/2$ go to step 4. Otherwise stop

4. Numerical Experimental Tests:

In this section, the numerical results of some representative problems are used to demonstrate the feasibility of the current three algorithms (**ATFIDE-CF, ABFIDE-CF and AMBFIDE-CF**). These findings are also explained in tables and figures. They were all finished using a computer program that was running Python 3.8.8 (2021). $\sum_{\ell=0}^{M}[y(x_\ell)-\hat{y}_N(x_\ell)]^2$ The least square errors (LSE) in the tables are represented by $M \in \mathbb{N}$ at M-selected collocation points x_ℓ locations. The running time is also given in tabular form. Additionally, apply the absolute error function $E(x_\ell)=|y(x_\ell)-\hat{y}_N(x_\ell)|$, for all $\ell=0,1,\ldots,M$, to each test problem at the chosen interval points. Additionally, use graphs to show it.

Test Problem 1. Consider the mixed classical and fractional order integro-differential equations of the Fredholm type with variable coefficients on the closed bounded interval $[a,b]=[0,1]$:

$$y'(x)-\frac{2}{3}x\,{}_{0}^{C}D_{x}^{0.2}y(x)+2xy(x)=f(x)+\int_{0}^{1}s\sin(x)\,y(s)ds-\int_{0}^{1}\mathrm{e}^{x}\,{}_{0}^{C}D_{s}^{0.4}y(s)ds$$

where

$$f(x)=\frac{-2}{\Gamma(2.8)}x^{2.8}+\frac{3\mathrm{e}^{x}}{2.6\Gamma(2.3)}+3x^{3}+2.6\,x\,-0.275\,\sin(x)$$

whereas the boundary condition: $y(0)-y(1)=\frac{-2}{3}$. While the exact solution is $y(x)=\frac{3}{2}x^2-\frac{1}{5}$.

This example obtained the numerical computation at every point applying the suggested strategy which is a linear IFDEs-CF of Fredholm type with variable coefficients for all derivative orders (C-F) lies in (0,1]: Take $N=10$ and $x_r=x_0+rh$, $(r=\overline{0:N})$. Since $\bar{n}=m=1$ and the fractional orders are $\sigma_0=0.2$ and $\beta_1=0.4$, $\beta_0=0$ with boundary coefficients $g_{00}=1; h_{00}=-1$ and $\vartheta_0=-2/3$. While, the kernels are $\mathcal{K}_0(x,s)=s\sin(x)$ with $\mathcal{K}_1(x,s)=\mathrm{e}^{x}$ and the eigenvalues parameters $\lambda_0=1$; $\lambda_1=-1$ and the functions: $p(x)=1$, $p_0(x)=\frac{-2}{3}x$ and $p_1(x)=2x$. We obtained the numerical results by using the suggested approach to the mentioned situation. Set the Python application for all three algorithms, ATFIDE-CF, ABFIDE-CF, and AMBFIDE-CF, and run all of them to obtain the following data solution for our test problem. Use equation (10) to get the following results:

$$\mathcal{A}_\sigma(0)=1.7016542932 \quad \text{and} \quad \mathcal{A}_\beta(0)=1 \quad \mathcal{A}_\beta(1)=2.8112403816$$

Also, the various values of δ_ℓ^α and C_ℓ^α for fractions $\alpha=\sigma$ and β for all $\ell=0,1,\ldots,10$ are presented in table 1 and 2 respectively, by applying equation (16).

Table 1. Contain all values of $\boldsymbol{\delta_\ell^\alpha}$ for fractions $\boldsymbol{\alpha=\sigma}$ and $\boldsymbol{\beta}$ for all $\boldsymbol{\ell=0,1,\ldots,10}$

Orders:	σ-fractional	β-fractional	
ℓ	$\delta_\ell^{\sigma_0}$	$\delta_\ell^{\beta_0}$	$\delta_\ell^{\beta_1}$
0	1	1	1
1	0.7411011266	1	0.5157165665
2	0.6671235587	1	0.4174654784
3	0.6232084477	1	0.3642146651
4	0.5924651854	1	0.3291310944
5	0.5690643942	1	0.3036282472
6	0.5503136812	1	0.2839397981
7	0.5347552493	1	0.2681064035
8	0.5215144917	1	0.2549905657
9	0.5100273100	1	0.2438788867
10	0.4999096827	1	0.2342974276

Table 2. Contain all values of C_ℓ^σ for fractions $\alpha = \sigma$ and β for all $\ell = 0, 1, \ldots, 10$

Orders:	σ-fractional	β-fractional	
ℓ	$C_\ell^{\sigma_0}$	$C_\ell^{\beta_0}$	$C_\ell^{\beta_1}$
0	1	1	1
1	−0.2588988734	0	−0.4842834335
2	−0.0739775679	0	−0.0982510881
3	−0.0439151109	0	−0.0532508134
4	−0.0307432624	0	−0.0350835707
5	−0.0234007911	0	−0.0255028472
6	−0.0187507131	0	−0.0196884490
7	−0.0155584319	0	−0.0158333947
8	−0.0132407576	0	−0.0131158378
9	−0.0114871817	0	−0.0111116790
10	−0.0101176273	0	−0.0095814591

For the finite differences quadratic Trapezoidal techniques, the matrices $L\left(= L^{\mathcal{P}} + L^{\mathcal{A}} + \frac{1}{2h} L^{\mathcal{G}}\right)$ and $I(= \lambda_0 I^{\mathcal{K}} + I^{\mathcal{A}})$ are created as equations (14 for $L^{\mathcal{P}}$-matrix, 15 for $L^{\mathcal{A}}$-matrix, 17 for $L^{\mathcal{G}}$-matrix) and (18 for $I^{\mathcal{K}}$-matrix, 19 for $I^{\mathcal{A}}$-matrix). The Python program which written for this purpose runs to obtain:

$$\boldsymbol{L} = \begin{bmatrix} -15 & 20 & -5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -4.88656 & 0.08656 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.16815 & -4.94126 & 0.17311 & 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.22704 & 0.02518 & -4.91189 & 0.25967 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2828 & 0.01993 & 0.03357 & -4.88252 & 0.34623 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0.33606 & 0.01744 & 0.02491 & 0.04196 & -4.85315 & 0.43278 & 5 & 0 & 0 & 0 & 0 \\ 0.38734 & 0.01593 & 0.02093 & 0.02989 & 0.05035 & -4.82378 & 0.51934 & 5 & 0 & 0 & 0 \\ 0.43701 & 0.01489 & 0.01858 & 0.02441 & 0.03487 & 0.05875 & -4.79441 & 0.60589 & 5 & 0 & 0 \\ 0.48532 & 0.01412 & 0.01702 & 0.02124 & 0.0279 & 0.03986 & 0.06714 & -4.76504 & 0.69245 & 5 & 0 \\ 0.53246 & 0.01352 & 0.01589 & 0.01914 & 0.02389 & 0.03139 & 0.04484 & 0.07553 & -4.73567 & 0.77901 & 5 \\ 0.57859 & 0.01303 & 0.01502 & 0.01765 & 0.02127 & 0.02655 & 0.03488 & 0.04982 & 5.08392 & -19.7063 & 15.86556 \end{bmatrix}_{11\times 11}$$

and

$$\boldsymbol{I} = \begin{bmatrix} 10.84895 & -0.70122 & -0.73528 & -0.77597 & -0.8259 & -0.88942 & -0.97458 & -1.09875 & -1.3117 & -2.13052 & -1.40562 \\ 11.98994 & -0.76499 & -0.79264 & -0.82763 & -0.87282 & -0.93304 & -1.01718 & -1.14442 & -1.36979 & -2.26474 & -1.50353 \\ 13.25094 & -0.83661 & -0.85833 & -0.88817 & -0.92929 & -0.987 & -1.07115 & -1.20294 & -1.44318 & -2.42342 & -1.61749 \\ 14.64455 & -0.917 & -0.93341 & -0.95879 & -0.99664 & -1.05283 & -1.13823 & -1.27629 & -1.53419 & -2.60994 & -1.74963 \\ 16.18473 & -1.00716 & -1.01902 & -1.04078 & -1.07633 & -1.13215 & -1.22025 & -1.36654 & -1.64529 & -2.82789 & -1.90223 \\ 17.88689 & -1.10818 & -1.11638 & -1.13553 & -1.1699 & -1.22669 & -1.31916 & -1.47593 & -1.77909 & -3.08115 & -2.07776 \\ 19.76807 & -1.22124 & -1.22683 & -1.24451 & -1.27903 & -1.33831 & -1.43702 & -1.60679 & -1.93836 & -3.37389 & -2.27889 \\ 21.8471 & -1.34766 & -1.35182 & -1.36934 & -1.40547 & -1.46896 & -1.57603 & -1.76165 & -2.12606 & -3.71055 & -2.50846 \\ 24.14478 & -1.48886 & -1.49291 & -1.51174 & -1.55113 & -1.62076 & -1.73856 & -1.94315 & -2.34536 & -4.09594 & -2.76959 \\ 26.68411 & -1.64639 & -1.65182 & -1.67357 & -1.71805 & -1.79596 & -1.92709 & -2.15415 & -2.5996 & -4.53524 & -3.0656 \\ 29.4905 & -1.82197 & -1.83039 & -1.85686 & -1.90843 & -1.99696 & -2.1443 & -2.39767 & -2.89239 & -5.03403 & -3.40014 \end{bmatrix}_{11\times 11}$$

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

$$[B; C] = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 - 1; -2/3]$$

substituting the above matrices for fundamental equation (21), the augmented matrix is obtained based on a condition which is:

$[\boldsymbol{D};\boldsymbol{E}]$

$$
= \left[\begin{array}{ccccccccccccc}
-16.08489485 & 20.0701221 & -4.92647243 & 0.07759673 & 0.08258975 & 0.08894193 & 0.09745808 & 0.10987454 & 0.13116995 & 0.21305218 & 0.14056202 & : & 0.80709732 \\
-6.08555061 & 0.16305495 & 5.07926386 & 0.08276265 & 0.08728245 & 0.09330436 & 0.10171783 & 0.11444181 & 0.13697854 & 0.22647407 & 0.15035338 & : & 1.12563557 \\
-1.15694717 & -4.85759852 & 0.25894615 & 5.08881678 & 0.09292858 & 0.09870045 & 0.10711541 & 0.12029421 & 0.1443178 & 0.24234228 & 0.16174937 & : & 1.46198904 \\
-1.23741213 & 0.11687658 & -4.8185473 & 0.35554816 & 5.09966369 & 0.10528304 & 0.11382344 & 0.1276287 & 0.1534193 & 0.26099354 & 0.17496287 & : & 1.82821971 \\
-1.33567684 & 0.12064325 & 0.1354712 & -4.77844013 & 0.45385822 & 5.11321485 & 0.12202527 & 0.13665427 & 0.16452911 & 0.28278885 & 0.19022297 & : & 2.23725188 \\
-1.45263223 & 0.12825567 & 0.1365474 & 0.15551403 & -4.73615742 & 0.55545108 & 5.13191568 & 0.1475927 & 0.17790865 & 0.30811536 & 0.20777631 & : & 2.70254144 \\
-1.58946695 & 0.13805239 & 0.14360887 & 0.15434252 & 0.17825634 & -4.68994681 & 0.66303993 & 5.16067949 & 0.19383584 & 0.33738856 & 0.22788857 & : & 3.23795161 \\
-1.7477029 & 0.14965643 & 0.15376467 & 0.16134748 & 0.17541985 & 0.20564215 & -4.63680361 & 0.78205958 & 5.21260643 & 0.37105481 & 0.25084626 & : & 3.85769076 \\
-1.92916131 & 0.16300608 & 0.16630868 & 0.17241138 & 0.18301364 & 0.20193122 & 0.24099384 & -4.57072124 & 0.92698665 & 5.40959432 & 0.27695872 & : & 4.57627892 \\
-2.13594831 & 0.17815798 & 0.18106714 & 0.18650169 & 0.19569697 & 0.21098415 & 0.23754558 & 0.29094542 & -4.4757063 & 1.23253181 & 5.30656043 & : & 5.40853015 \\
-2.3704565 & 0.19522839 & 0.19806002 & 0.2033357 & 0.21211487 & 0.22624239 & 0.24930654 & 0.28958589 & 5.37316205 & -19.20289226 & 16.20557744 & : & 6.36954538 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & : & \cdots \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & : & -\frac{2}{3}
\end{array}\right]
$$

solving the system above, by procedure that $[D^T D; D^T E]$, the approximate solutions $\hat{Y}^h$ are obtained. Table 3 shows a comparison between the exact solution and numerical solutions $y(x) = \hat{y}_N^h(x)$ for $N = 10$ of the three strategies: Trapezoidal method, Two-point block method and modify two-point block method depending on the least square error and running time. Also, the results, least square error, and the required time for running the programs for different values, i.e., different step size h, are shown in Table 4. Figures 1 displays a comparison between the approximate solutions by three algorithms and the exact solution with a step size of $h = 0.1$ and $N = 10$ for test example 1. The absolute errors are also depicted in Figure 2.

Table 3. Numerical results in that shows comparing the exact $y(x)$ and numerical solutions $\hat{y}_N^h(x)$ for test example 1 using algorithms (ATFIDE-CF, ABFIDE-CF and AMBFIDE-CF) depending on least square error and running time for $N = 10$.

x_i	Exact Solution	*Approximate Solutions*		
		Trapezoidal	*Two-point block*	*Modify two-point block*
0.0	−0.200	−0.18273375	−0.18273375	−0.18273375
0.1	−0.185	−0.16703029	−0.16703133	−0.16739067
0.2	−0.140	−0.12148536	−0.12148943	−0.12228406
0.3	−0.065	−0.04612347	−0.04613745	−0.04740482
0.4	0.040	0.05907293	0.05907504	0.05726772
0.5	0.175	0.19408961	0.19404502	0.19176367
0.6	0.340	0.35895958	0.35896959	0.35612065
0.7	0.535	0.55367793	0.55358522	0.55038113
0.8	0.760	0.77829193	0.77831431	0.77459073
0.9	1.015	1.03280208	1.03264302	1.02879698
1.0	1.300	1.31726626	1.31730722	1.31306533
$L.S.E.$		3.706303×10^{-3}	3.697489×10^{-3}	2.881255×10^{-3}
$R.Time/Sec$		0.106966495519	0.29590439796	2.1884922981

Table 4. The mean least squared errors and running times $N = 10, 50$ and 100 for the Trapezoidal, Two-point block, and modify two-point block methods for various step sizes.

Methods	*Least square Error (LSE.) and R.Time/Sec for different input N*					
	$N = 10$		$N = 50$		$N = 100$	
	$LSE.$	$R.Time/Sec$	$LSE.$	$R.Time/Sec$	$LSE.$	$R.Time/Sec$
Trapezoidal	3.706303×10^{-3}	0.106966	1.603494×10^{-4}	10.017458	3.859712×10^{-5}	46.913775
Two-point block	3.697489×10^{-3}	0.295904	1.603414×10^{-4}	14.526843	3.859676×10^{-5}	69.572829
Modify two-point block	2.881255×10	2.188492	1.231219×10^{-4}	70.886375	2.959983×10^{-5}	442.463386

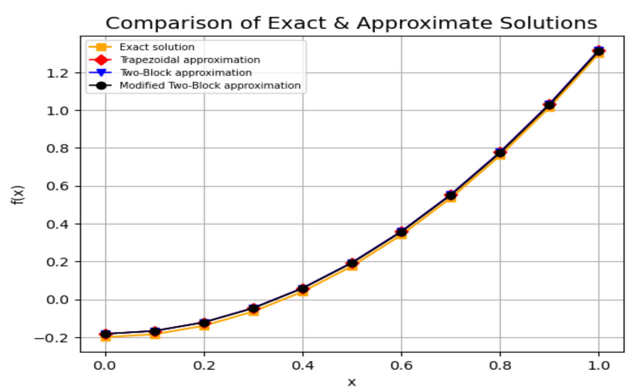

Figure 2. For test example, 1 with a step size of $h = 0.1$, the approximate solutions, while the precise solution is shown by a solid line.

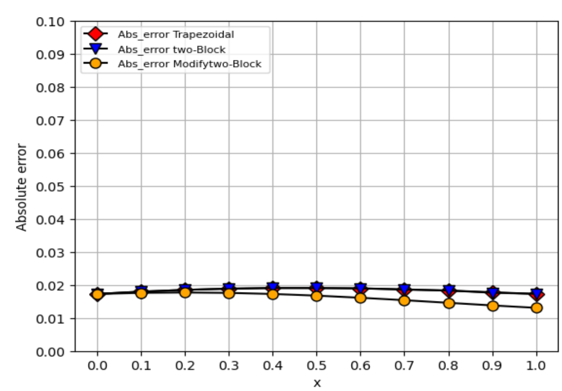

Figure 3. Absolute error plot function $\left|y(x) - \hat{y}_N^h(x)\right|$ for $N = 10, h = 0.1$ for test example 1.

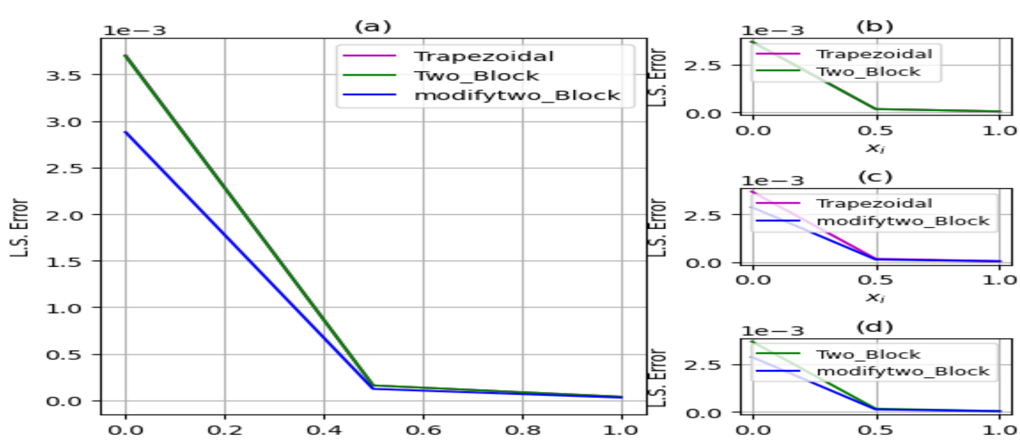

Figure 4. (a) LSE. comparison of Trapezoidal, Two-point block and modify two-point block methods for Example 1 **(b)** LSE. comparison of Trapezoidal and Two-point block methods. **(c)** LSE. comparison of Trapezoidal and Modify Two-Block methods. **(d)** LSE. comparison of Two-point block and modify two-point block methods.

Test Problem 2. Consider the mixed classical and fractional order integro-differential equations of the Fredholm type with variable coefficients on the closed bounded interval $[a,b]=[0,1]$:

$$y'(x)+{}_{0}^{C}D_x^{0.3}y(x)+{}_{0}^{C}D_x^{0.5}y(x)+xy(x)$$
$$=f(x)+\left(\frac{1}{10}\right)\int_0^1(2s-x)y(s)ds-\left(\frac{1}{10}\right)\int_0^1(\cosh(x)+5x){}_{0}^{C}D_s^{0.6}y(s)ds$$

where

$$f(x)=\frac{1}{\Gamma(1.7)}x^{0.7}-\frac{2}{\Gamma(3.7)}x^{2.7}+\frac{1}{\Gamma(1.5)}x^{0.5}-\frac{2}{\Gamma(3.5)}x^{2.5}+\frac{1}{10}(\cosh(x)+5x)\left[\frac{1}{\Gamma(2.4)}-\frac{2}{\Gamma(4.4)}\right]-\frac{1}{3}x^4$$
$$+\frac{1}{24}x+0.9466667$$

The suitable condition is: $y(1)=\frac{2}{3}$. While the exact solution is $y(x)=x-\frac{1}{3}x^3$. This example obtained the numerical computation at every point applying the suggested three strategy which is a linear IFDEs-CF of Fredholm type with variable coefficients for all derivative orders (C-F) lies in (0,1]: Take $N=10$ and $x_r=x_0+rh,\ (r=\overline{0:N})$. Since $\bar{n}=2; m=1$ and the fractional orders are $\sigma_0=0.3, \sigma_1=0.5$ and $\beta_1=0.6,\ \beta_0=0$ with boundary coefficients $g_{00}=0; h_{00}=1$ and $\vartheta_0=2/3$. While, the kernels are $\mathcal{K}_0(x,s)=2s-x$ with $\mathcal{K}_1(x,s)=\cosh(x)+5x$ and the eigenvalues parameters $\lambda_0=1/10;\ \lambda_1=-1/10$ and the functions: $p(x)=1, p_0\equiv p_1\equiv 1$ and $p_2(x)=x$. We obtained the numerical results by using the suggested approach to the mentioned situation. Set the Python application for all three algorithms, ATFIDE-CF, ABFIDE-CF, and AMBFIDE-CF, and run all of them to obtain the following data solution for our test problem. Use equation (10) to get the following results:

$\mathcal{A}_\sigma(0)=2.1958807641 \quad \mathcal{A}_\sigma(1)=3.5682482323$ and $\mathcal{A}_\beta(0)=1 \quad \mathcal{A}_\beta(1)=4.4869086589$

Also, the various values of δ_ℓ^α and C_ℓ^α for fractions $\alpha=\sigma$ and β for all $\ell=0,1,\dots,10$ are presented in table 5 and 6 respectively, by applying equation (16).

Table 5. Contain all values of δ_ℓ^α for fractions $\alpha=\sigma$ and β for all $\ell=0,1,\dots,10$

Orders:	σ-fractional		β-fractional	
ℓ	$\delta_\ell^{\sigma_0}$	$\delta_\ell^{\sigma_1}$	$\delta_\ell^{\beta_0}$	$\delta_\ell^{\beta_1}$
0	1	1	1	1
1	0.6245047927	0.4142135624	1	0.3195079108
2	0.5331644873	0.3178372452	1	0.2323376631
3	0.4813465416	0.2679491924	1	0.1892555527
4	0.4461534921	0.2360679775	1	0.1625528121
5	0.4199747728	0.2134217653	1	0.1440185724
6	0.3993846907	0.1962615683	1	0.1302339134
7	0.3825650730	0.1826758137	1	0.1194902855
8	0.3684428716	0.1715728753	1	0.1108279753
9	0.3563356145	0.1622776602	1	0.1036617462
10	0.3457843332	0.1543471302	1	0.0976122038

Table 6. Contain all values of C_ℓ^α for fractions $\alpha = \sigma$ and β for all $\ell = 0,1,\dots,10$

Orders:	σ-fractional		β-fractional	
ℓ	$C_\ell^{\sigma_0}$	$C_\ell^{\sigma_1}$	$C_\ell^{\beta_0}$	$C_\ell^{\beta_1}$
0	1	1	1	1
1	−0.3754952073	−0.5857864376	0	−0.6804920892
2	−0.0913403055	−0.0963763172	0	−0.0871702476
3	−0.0518179457	−0.0498880528	0	−0.0430821105
4	−0.0351930495	−0.0318812149	0	−0.0267027406
5	−0.0261787192	−0.0226462122	0	−0.0185342398
6	−0.0205900821	−0.0171601970	0	−0.0137846590
7	−0.0168196177	−0.0135857546	0	−0.0107436279
8	−0.0141222014	−0.0111029384	0	−0.0086623102
9	−0.0121072571	−0.0092952151	0	−0.0071662291
10	−0.0105512813	−0.0079305300	0	−0.0060495425

Table 7 shows a comparison between the exact solution and numerical solutions $y(x) = \hat{y}_N^h(x)$ for $N = 10$ of the three strategies: Trapezoidal method, Two-point block method and modify two-point block method depending on the least square error and running time. Also, the results, least square error, and the required time for running the programs for different values, i.e., different step size h, are shown in Table 8. Figures 4 displays a comparison between the approximate solutions by three algorithms and the exact solution with a step size of $h = 0.1$ and $N = 10$ for test example 2. The absolute errors are also depicted in Figure 5.

Table 7. Numerical results in that shows comparing the exact $y(x)$ and numerical solutions $\hat{y}_N^h(x)$ for test example 2 using algorithms (ATFIDE-CF, ABFIDE-CF and AMBFIDE-CF) depending on least square error and running time for $N = 10$.

x_i	Exact Solution	*Approximate Solutions*		
		Trapezoidal	*Two-point block*	*Modify two-point block*
0.0	0.00000000	0.00262389	0.00262389	0.00262389
0.1	0.09966667	0.10213647	0.10239117	0.10231600
0.2	0.19733333	0.20023795	0.19991350	0.19985559
0.3	0.29100000	0.29370573	0.29360413	0.29332099
0.4	0.37866667	0.38130687	0.38077218	0.38070664
0.5	0.45833333	0.46071284	0.46045080	0.46003975
0.6	0.52800000	0.53007016	0.52939046	0.52933072
0.7	0.58566667	0.58741241	0.58707058	0.58658722
0.8	0.62933333	0.63056866	0.62987389	0.62983338
0.9	0.65700000	0.65799280	0.65744933	0.65704657
1.0	0.66666667	0.66674359	0.66620614	0.66629882
LSE.		5.122598×10^{-5}	4.127387×10^{-5}	3.573159×10^{-5}
R. Time/Sec		0.22990847	0.35886931	3.05152345

Table 8. The mean least squared errors and running times $N = 10, 50$ and 100 for the Trapezoidal, Two-point block, and modify two-point block methods for various step sizes.

Methods	Least square Error (LSE.) and $R.Time/Sec$ for different input N					
	$N = 10$		$N = 50$		$N = 100$	
	$LSE.$	$R.Time$ $/Sec$	$LSE.$	$R.Time$ $/Sec$	$LSE.$	$R.Time$ $/Sec$
Trapezoida l	5.122598×10^{-5}	0.229908	2.360518×10^{-6}	13.327582	6.234882×10^{-7}	46.913775
Two-point block	4.127387×10^{-5}	0.358869	2.282876×10^{-6}	20.416350	6.161447×10^{-7}	105.80631
Modify two-point block	3.573159×10^{-5}	3.051523	2.065440×10^{-6}	118.41427	5.933845×10^{-7}	993.2877

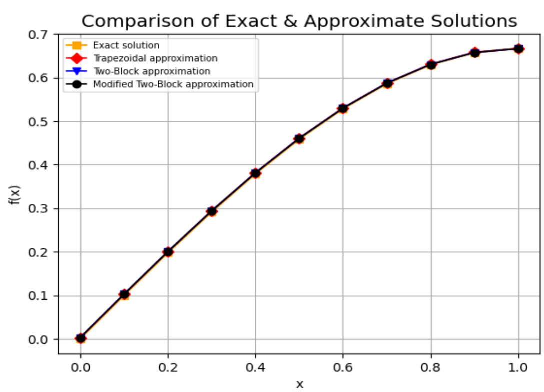

Figure 5. For test example, 2 with a step size of $h = 0.1$, the approximate solutions, while the precise solution is shown by a solid line.

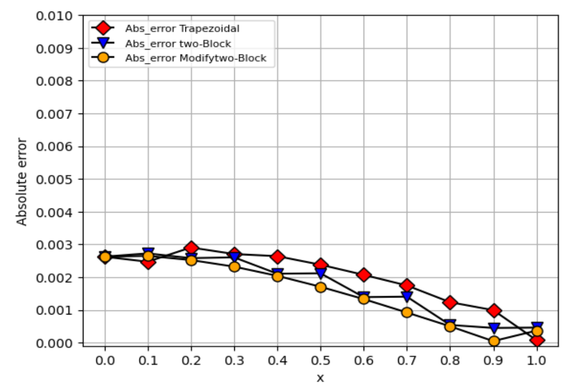

Figure 6. Absolute error plot function $\left|y(x) - \hat{y}_N^h(x)\right|$ for $N = 10, h = 0.1$ for test example 2.

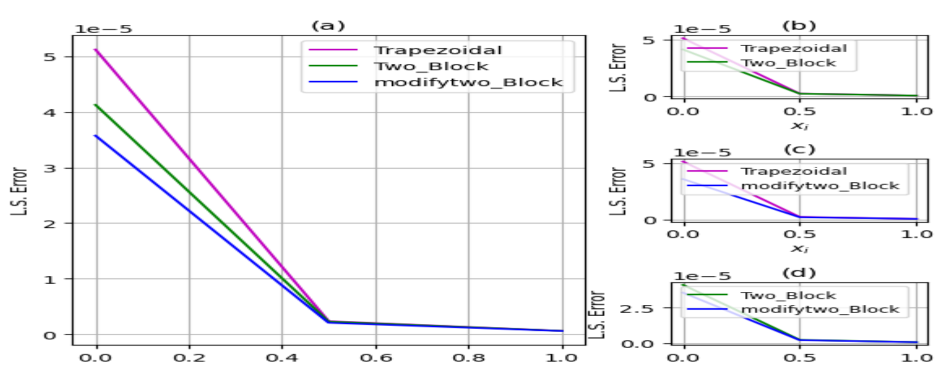

Figure 7. (a) LSE. comparison of Trapezoidal, Two-point block and modify two-point block methods for Example 1 **(b)** LSE. comparison of Trapezoidal and Two-point block methods. **(c)** LSE. comparison of Trapezoidal and Modify Two-Block methods. **(d)** LSE. comparison of Two-point block and modify two-point block methods.

Test Problem 3. Consider the mixed classical and fractional order integro-differential equations of the Fredholm type with variable coefficients on the closed bounded interval $[a,b]=[0,1]$:

$$y'(x)+{}_0^CD_x^{0.1}y(x)+(x+1)\,{}_0^CD_x^{0.3}y(x)+\left(\frac{1}{3}x\right){}_0^CD_x^{0.4}y(x)+(x^2+2)y(x)$$
$$=f(x)+\frac{1}{5}\int_0^1(2s)y(s)ds-\frac{1}{5}\int_0^1\left(\cos\left(\frac{x}{2}\right)-x\right){}_0^CD_s^{0.7}y(s)ds$$

where

$$f(x)=\frac{-2}{\Gamma(2.9)}x^{1.9}-\frac{2}{\Gamma(2.7)}(x^{2.7}+x^{1.7})-\frac{\frac{2}{3}}{\Gamma(2.6)}x^{2.6}-\frac{\frac{2}{5}}{\Gamma(3.3)}\left(\cos\left(\frac{x}{2}\right)-x\right)-x^4-2x^2-2x+\frac{1}{10}$$

The boundary condition: $3y(0)+y(1)=-1$. While the exact solution is $y(x)=-x^2$. This example obtained the numerical computation at every point applying the suggested three strategy which is a linear IFDEs-CF of Fredholm type with variable coefficients for all derivative orders (C-F) lies in (0,1]: Take $N=10$ and $x_r=x_0+rh,\ (r=\overline{0:N})$. Since $\bar{n}=3; m=1$ and the fractional orders are $\sigma_0=0.1,\sigma_1=0.3,\sigma_2=0.4$ and $\beta_1=0.7,\ \beta_0=0$ with boundary coefficients $g_{00}=3; h_{00}=1$ and $\vartheta_0=-1$. While, the kernels are $\mathcal{K}_0(x,s)=2s$ with $\mathcal{K}_1(x,s)=\cos(x/2)-x$ and the eigenvalues parameters $\lambda_0=1/5;\ \lambda_1=-1/5$ and the functions: $p(x)=1, p_0(x)=1, p_1(x)=x+1$, $p_2(x)=x/3$ and $p_3(x)=x^2+2$ We obtained the numerical results by using the suggested approach to the mentioned situation. Set the Python application for all three algorithms, ATFIDE-CF, ABFIDE-CF, and AMBFIDE-CF, and run all of them to obtain the following data solution for our test problem. Use equation (10) to get the following results:

$\mathcal{A}_\sigma(0)=1.3089729017$ $\mathcal{A}_\sigma(1)=2.19588076$ $\mathcal{A}_\sigma(2)=2.811240381$ and $\mathcal{A}_\beta(0)=1$ $\mathcal{A}_\beta(1)=5.58444120$

The exact solution and numerical solutions for $y(x)=\hat{y}_N^h(x)$ for $N=10$ of the three strategies-the trapezoidal technique, the two-point block method, and the modified two-point block method-based on the running time and least square error are compared in Table 9. Additionally, Table 10 displays the outcomes, least square error, and time needed to execute the algorithms for various values, or step sizes h. For test example 3, Figure 7 compares the exact answer with a step size of $h=0.1\ and\ N=10$ with the approximate solutions obtained by three algorithms. Figure 8 shows the absolute errors as well.

Table 9. Numerical results in that shows comparing the exact $y(x)$ and numerical solutions $\hat{y}_N^h(x)$ for test example 3 using algorithms (ATFIDE-CF, ABFIDE-CF and AMBFIDE-CF) depending on least square error and running time for $N=10$.

x_i	Exact Solution	*Approximate Solutions $\hat{y}_N^h(x)$*		
		Trapezoidal	*Two-point block*	*Modify two-point block*
0.0	−0.00	0.00072054	0.00072054	0.00072054
0.1	−0.01	−0.01006724	−0.00996222	−0.00983996
0.2	−0.04	−0.04078291	−0.04081348	−0.04027094
0.3	−0.09	−0.09118969	−0.09106657	−0.09057865
0.4	−0.16	−0.16152601	−0.16157594	−0.16079390
0.5	−0.25	−0.25174146	−0.25155096	−0.25094407
0.6	−0.36	−0.36190286	−0.36193282	−0.36104488
0.7	−0.49	−0.49200872	−0.49176476	−0.49110902
0.8	−0.64	−0.64207961	−0.64211555	−0.64114496
0.9	−0.81	−0.81212011	−0.81179837	−0.81115950
1.0	−1.00	−1.00213953	−1.00218459	−1.00115753
LSE.		2.896651×10^{-5}	2.654127×10^{-5}	8.791524×10^{-6}
R. Time/Sec		0.17188239	0.34373093	2.48437190

Table 10. The mean least squared errors and running times $N = 10, 50$ and 100 for the Trapezoidal, Two-point block, and modify two-point block methods for various step sizes.

Methods	Least square Error (LSE.) and $R.Time/Sec$ for different input N					
	$N = 10$		$N = 50$		$N = 10$	
	$LSE.$	$R.Time/Sec$	$LSE.$	$R.Time/Sec$	$LSE.$	$R.Time/Sec$
Trapezoidal	2.896651×10^{-5}	0.171882	1.073431×10^{-6}	14.785652	2.740044×10^{-7}	90.684154
Two-point block	2.654127×10^{-5}	0.343731	1.071960×10^{-6}	16.072447	2.739603×10^{-7}	109.72937
Modify two-point block	8.791524×10^{-6}	2.484371	2.545316×10^{-7}	119.14772	5.744475×10^{-8}	807.42086

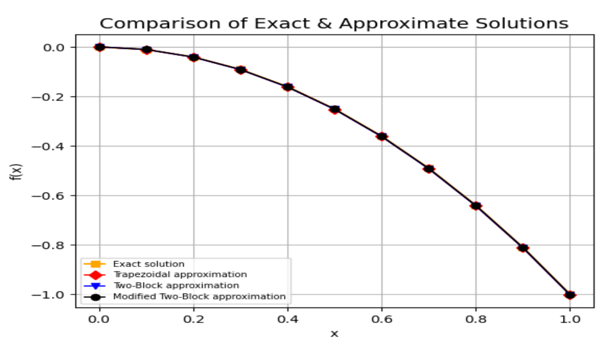

Figure 8. For test example, 3 with a step size of $h = 0.1$, the approximate solutions, while the precise solution is shown by a solid line.

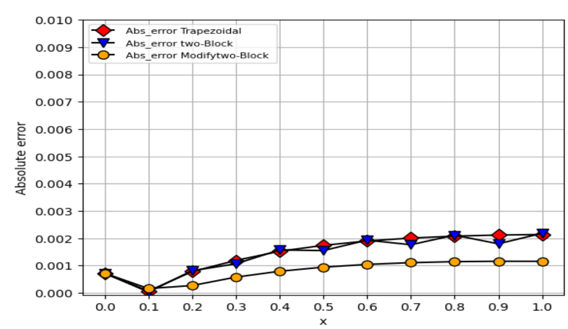

Figure 9. Absolute error plot function $|y(x) - \hat{y}_N^h(x)|$ for $N = 10, h = 0.1$ for test example 3.

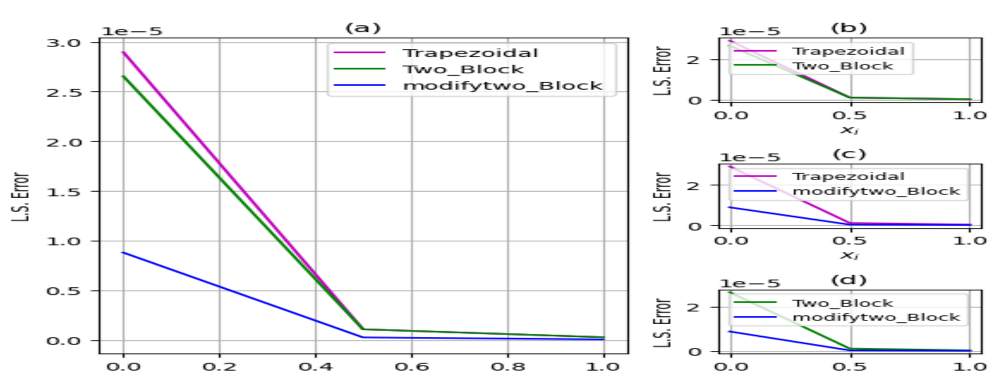

Figure 10. (a) LSE. comparison of Trapezoidal, Two-point block and modify two-point block methods for Example 3 **(b)** LSE. comparison of Trapezoidal and Two-point block methods. **(c)** LSE. comparison of Trapezoidal and Modify Two-Block methods. **(d)** LSE. comparison of Two-point block and modify two-point block methods.

5. Conclusion:

In this study, we have presented and implemented three distinct numerical techniques-namely, the Trapezoidal Method, the Two-Point Block Method, and the Modified Two-Point Block Method-to solve mixed classical and fractional-order integro-differential equations of Fredholm type with variable coefficients under boundary conditions (IFDEs-CF) in the Caputo sense. The primary aim was to assess the performance, accuracy, and practical applicability of each method in approximating the solutions to these complex equations. All algorithms were thoroughly formulated and implemented using the Python programming language, and their effectiveness was evaluated through a series of carefully selected numerical examples.

Based on the numerical outcomes and graphical analyses, we draw the following conclusions:

1. **Accuracy of the Modified Two-Point Block Method:**

Among the three numerical approaches examined, the Modified Two-Point Block Method consistently yielded the highest level of accuracy. This method demonstrated superior approximation quality, particularly in terms of minimizing the least squares error, and showed strong agreement with the exact solutions of the test problems.

2. **Performance of the Two-Point Block Method:**

The Two-Point Block Method also performed well across all test cases. It produced accurate results and outperformed the classical Trapezoidal Method in most scenarios. This indicates its reliability as a robust and efficient method for solving IFDEs-CF, especially when the Modified version is not required.

3. **Effectiveness of the Trapezoidal Method:**

While the Trapezoidal Method is a more conventional approach, it still provided accurate approximations, particularly when applied with sufficiently small step sizes h. However, its performance was relatively less effective compared to the other two methods, especially for problems involving strong fractional behavior.

4. **Impact of Step Size and Number of Intervals:**

For all three methods, the accuracy of the numerical solution improves as the number of subintervals N increases, which corresponds to a reduction in the step size h. A finer discretization led to a noticeable decrease in the least squares error, confirming the convergence behavior of each method.

5. **General Observations on Convergence and Reliability:**

Although all three methods yielded convergent and practically useful solutions, the Modified Two-Point Block Method emerged as the most reliable and accurate technique for this particular class of equations. It not only ensured better convergence rates but also maintained numerical stability across different problem settings.

In summary, the study confirms that while each method has its merits, the Modified Two-Point Block Method is the most suitable choice for solving IFDEs-CF in the Caputo sense, due to its superior accuracy, consistency, and computational effectiveness. These findings contribute to the broader understanding of numerical strategies for fractional integro-differential equations and provide a solid foundation for further exploration in this area.

References

[1] V. S. Panwar, P. S. Uduman, and J. F. Gómez-Aguilar, "Mathematical modeling of coronavirus disease COVID-19 dynamics using CF and ABC non-singular fractional derivatives," *Chaos, Solitons & Fractals*, vol. 145, Art. no. 110757, 2021.

[2] C. Volos, "Introductory chapter: Fractional calculus – From theory to applications," in *Fractional Calculus – From Theory to Applications*. London, U.K.: IntechOpen, 2025, ch. 1.

[3] I. Podlubny, "Fractional-order systems and PI $^{\lambda}$D $^{\mu}$ controllers," *IEEE Trans. Autom. Control*, vol. 44, no. 1, pp. 208–214, Jan. 1999.

[4] H. Sun, Y. Zhang, D. Baleanu, W. Chen, and Y. Chen, "A new collection of real-world applications of fractional calculus in science and engineering," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 64, pp. 213–231, 2018.

[5] H. M. Srivastava, "Editorial for the Special Issue 'Operators of fractional calculus and their multidisciplinary applications,'" *Fractal Fract.*, vol. 7, no. 5, Art. no. 415, 2023.

[6] T. V. Moreira Sampaio *et al.*, "Procedimentos para estimativa da velocidade de disseminação de doenças: Um estudo de caso com dados da COVID-19 no estado do Paraná," *Hygeia*, vol. 20, pp. 1–15, 2024.

[7] D. Valério, J. T. Machado, and V. Kiryakova, "Some pioneers of the applications of fractional calculus," *Fractional Calculus and Applied Analysis*, vol. 17, no. 2, pp. 552–578, 2014.

[8] D. Valério, J. T. Machado, and V. Kiryakova, "Some pioneers of the applications of fractional calculus," *Fractional Calculus and Applied Analysis*, vol. 17, no. 2, pp. 552–578, 2014.

[9] A. Roohollahi, B. Ghazanfari, and S. Akhavan, "Numerical solution of the mixed Volterra–Fredholm integro-differential multi-term equations of fractional order," *J. Comput. Appl. Math.*, vol. 376, Art. no. 112828, 2020.

[10] S. S. Ahmed and S. J. Mohammed Faeq, "Bessel collocation method for solving Fredholm–Volterra integro-fractional differential equations of multi-high order in the Caputo sense," *Symmetry*, vol. 13, no. 12, Art. no. 2354, 2021.

[11] V. Samadpour Khalifeh Mahaleh and R. Ezzati, "Numerical solution of linear fuzzy Fredholm integral equations of second kind using iterative method and midpoint quadrature formula," *J. Intell. Fuzzy Syst.*, vol. 33, no. 2, pp. 1293–1302, 2017.

[12] S. S. Ahmed, "On system of linear Volterra integro-fractional differential equations," presented at an academic seminar, Jul. 2009.

[13] M. A. Rufai, B. Carpentieri, and H. Ramos, "A new hybrid block method for solving first-order differential system models in applied sciences and engineering," *Fractal Fract.*, vol. 7, no. 10, Art. no. 703, 2023.

[14] R. K. Salih, "Block and Weddle methods for solving nth-order linear retarded Volterra integro-differential equations," *Emirates J. Eng. Res.*, vol. 25, no. 2, pp. 3–10, 2020.

[15] B. Raftari, "Numerical solutions of the linear Volterra integro-differential equations: Homotopy perturbation method and finite difference method," *World Appl. Sci. J.*, vol. 9, no. 1, pp. 7–12, 2010.

[16] O. Tunç, Ö. Atan, C. Tunç, and J. C. Yao, "Qualitative analyses of integro-fractional differential equations with Caputo derivatives and retardations via the Lyapunov–Razumikhin method," *Axioms*, vol. 10, no. 2, Art. no. 58, 2021.

[17] K. Saad and M. Khirallah Yemen, "A spectral collocation method for solving Caputo–Liouville fractional-order Fredholm integro-differential equations," *Eur. J. Pure Appl. Math.*, vol. 17, no. 1, pp. 1–20, 2022.

[18] A. Shukla, N. Sukavanam, and D. N. Pandey, "Approximate controllability of semilinear fractional control systems of order $\alpha \in (1,2]$," in *Proc. Conf. Control and Its Applications*, SIAM, Jul. 2015, pp. 175–180.

[19] S. S. Ahmed and H. A. Rasol, "Numerical computation of mixed Volterra–Fredholm integro-fractional differential equations by using Newton–Cotes methods," *Symmetry*, vol. 14, no. 8, Art. no. 1693, 2022.

[20] K. Diethelm and N. J. Ford, "Analysis of fractional differential equations," *J. Math. Anal. Appl.*, vol. 265, no. 2, pp. 229–248, 2002.

[21] K. B. Oldham, "Fractional differential equations in electrochemistry," *Adv. Eng. Softw.*, vol. 41, no. 1, pp. 9–12, 2010.

[22] R. I. Esa, "Approximate solution of Fredholm integro-differential equation using quadrature formulas methods," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 9, no. 6, pp. 284–291, 2022.

[23] R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th ed. Boston, MA, USA: Cengage Learning, 2011.

[24] S. Scholz, "Review of J. D. Lambert, *Computational Methods in Ordinary Differential Equations*," *Z. Angew. Math. Mech.*, vol. 54, no. 7, p. 523, 1974.