Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. *, No. *, *, pp. 1-29
DOI:10.22034/cmde.2026.71068.3570

An efficient computational framework for predicting the behavior of multi-term Caputo fractional differential equations with a delay

Payam Moradpour, Khosro Sayevand*, **and Iman Masti**

Faculty of Mathematical Sciences, Malayer University, P. O. Box 16846-13114, Malayer, Iran.

Abstract

Developing robust numerical techniques for approximating solutions to fractional-order differential equations holds significant value, as exact analytical solutions are often unattainable for this class of problems. In this paper, a novel and efficient numerical method based on modified linear B-spline basis functions has been developed for solving multi-term Caputo fractional delay differential equations (FDDEs).The proposed technique employs operational matrices for fractional integration and delay transformation, incorporating the Caputo fractional derivative ${}^{C}D^{\alpha}$ along with both the initial conditions at $t = 0$ and the history function $g(t)$ for $t < 0$ (which provides the necessary prehistory for the delay terms), converting the original FDDE into an equivalent algebraic system (linear or nonlinear) that can be solved by standard linear solvers or iterative schemes. A rigorous convergence analysis is provided, establishing an optimal algebraic convergence rate (order $O(2^{-2J})$, equivalently $O(N^{-2})$ for $N = 2^J + 1$ basis functions) for sufficiently smooth solutions, while the method retains the simplicity and sparsity of finite-element approaches. Several numerical examples including linear, nonlinear, system and multi-term FDDEs have been solved to illustrate the efficiency and reliability of the approach.The results demonstrate excellent agreement with exact solutions and, for the considered smooth test problems, attain errors close to machine precision with a relatively small number of basis functions, consistent with the theoretical algebraic rate. Comparisons with existing methods confirm that the present scheme is both more accurate and computationally economical.

Keywords. Multi-term Caputo fractional delay differential equations, Linear B-spline, Operational matrix, Convergence analysis.
2010 Mathematics Subject Classification. 26A33, 34A08, 41A15.

1. Introduction

Fractional-order differential equations have attracted much attention in science and engineering in recent decades due to their ability to accurately model complex phenomena with memory and nonlocal behavior. These equations, which are generalizations of ordinary differential equations with integer-order derivatives, have found wide application in fields such as physics, biology, control engineering, and economics. Phenomena such as non-Fourier heat transfer, viscoelastic dynamics, systems control, biological processes and epidemiological disease dynamics, artificial neural networks with synaptic delays , and economic models are among the successful applications of these equations [1, 8, 15, 18, 19, 24, 31].

However, in many real systems, the current state of a phenomenon depends not only on its instantaneous derivatives, but also on the values of the variables or their derivatives at past moments (time delays). Delayed differential equations provide a convenient framework for modeling such systems; combining these two concepts, namely the consideration of fractional-order derivatives with time delays, leads to a new class of equations called fractional-order delayed differential equations. These equations, especially when combined with stochastic or nonlinear behaviors, are able to simultaneously include the effects of long-term memory (via fractional order) and limited history dependence (via delay) in the model and are therefore very effective in describing more accurately complex real world phenomena such

Received: 04 January 2026; Accepted: 07 September 2026.
∗ Corresponding author. Emails: ksayehvand@malayeru.ac.ir.

as the spread of infectious diseases with latency, the behavior of viscoelastic smart materials with delay, or the control of systems with delayed feedback [12, 14, 25–27].

Delays, which often arise due to the presence of response times or past effects in systems, add significant computational and analytical complexity to these equations. Despite the high modeling power, analytical solutions of these equations are in most cases impossible or very difficult. For this reason, the development of stable, accurate, and convergent numerical methods for approximating the solutions of these equations is of particular importance [4, 26, 27, 32].

Over the past decade, various numerical approaches have been developed for fractional differential equations and their delay counterparts. Among these, operational matrix methods have gained significant attention due to their ability to convert differential equations into algebraic systems through basis expansions. For instance, [28] introduced an operational matrix for fractional-order differential equations using orthogonal polynomials, while [16] constructed an operational matrix of fractional derivatives based on B-spline functions, demonstrating the flexibility of spline bases in fractional calculus. Extensions to delay problems include the work of [29] on nonlinear systems of Volterra integro-differential equations with delay arguments, and the spectral element method proposed in [6] for neutral delay distributed-order fractional damped diffusion-wave equations. More recently, Taylor series expansions have been employed to derive orthogonal operational matrices [9], and homotopy analysis methods have been applied to nonlinear fractional partial differential equations [7].

Methods such as the extension of Adams-Bashforth-Moulton methods to delay equations [3], the operational matrix method using shifted Chebyshev polynomials of the second kind [12], the collocation Laguerre operational matrix (CLOM) method, simplified reproducing kernel methods (SRKM) which solve the problem in a reproducing kernel Hilbert space [32], a Tau-like numerical method [30], a novel predictor-corrector scheme with Mittag-Leffler kernel [5] have been proposed in recent years to approximate the solutions of these equations, each with its own advantages and challenges. Some (such as wavelet methods) are only optimal for linear problems or under specific conditions. Others, despite high accuracy, suffer from high computational costs due to the need to compute dense operational matrices and solve complex algebraic systems. Furthermore, the convergence and error analysis of these methods, especially for nonlinear equations, often requires complex investigations.

The multi-term Caputo fractional delay differential equations is given as:

$$ {}^{C}D^{\alpha}w(t) = f\big(t, w(t), {}^{C}D^{\beta}w(t), w(ct-\tau), {}^{C}D^{\gamma}w(ct-\tau)\big), \quad t \in [0, R], \tag{1.1}$$

with initial conditions

$$ w^{(r)}(0) = \lambda_r, \quad r = 0, 1, \dots, \lceil \alpha \rceil - 1, \tag{1.2}$$

and

$$ w(t) = g(t), \quad t < 0. \tag{1.3}$$

where ${}^{C}D^{\alpha}, {}^{C}D^{\beta}, {}^{C}D^{\gamma}$ denotes the Caputo fractional derivative of order α, β, γ. The known functions f and g are assumed to be sufficiently smooth; specifically, f is Lipschitz continuous with respect to all its arguments. Also, $ct-\tau$ represents the delay argument. The delay argument is of the constant proportional form $ct-\tau$, where $0 < c \leq 1$ and $\tau \geq 0$, which includes pure delay ($c = 1$) and proportional delay ($c < 1$) as special cases. In addition to the standard initial conditions $w^{(r)}(0) = \lambda_r$ for $r = 0, 1, \dots, \lceil \alpha \rceil - 1$, the problem requires a history function $w(t) = g(t)$ for $t < 0$, which provides the necessary values of the solution and its derivatives for the delayed arguments when $ct - \tau < 0$.

In this paper, inspired by these developments, particularly the use of B-spline bases for fractional operators [16] and the spectral element framework for delay problems [6], proposes a novel collocation method based on linear cardinal B-spline functions for a broad class of linear and nonlinear multi-term Caputo fractional delay differential equations (1.1) and (1.2). Unlike existing approaches that often rely on dense operational matrices or complex analytical derivations, our method exploits the sparsity and Kronecker delta property of cardinal B-splines to achieve high accuracy with minimal computational cost, while maintaining a straightforward implementation. The proposed method uses a numerical collocation approach based on the fractional integral operational matrix using linear cardinal

B-spline functions to solve fractional delayed differential equations. The analytical construction of the Riemann-Liouville fractional integral matrix significantly simplifies the calculations, increases numerical stability, and leads to the formation of a linear algebraic system (for linear problems) or a nonlinear system (for nonlinear problems).

The structure of this paper is as follows: In Section 2, basic definitions of fractional calculus and the introduction of linear cardinal B-spline functions are presented. Section 3 provides a complete description of the numerical method, including the construction of the fractional integration matrix and the formulation of the resulting algebraic system. Convergence analysis and error estimation are presented in Section 4. In Section 5, the efficiency of the method is demonstrated by solving several benchmark examples (linear and nonlinear), with comparison to exact solutions or other methods provided in tables and figures. Finally, conclusions and discussion of future work are presented in Section 6.

Before presenting the detailed definitions, we provide a brief roadmap of the construction to guide the reader through the notation hierarchy. The development follows four logical steps:

(1) **Basis definition:** We begin with the cardinal B-spline $B_n(x)$ of order n, defined recursively via convolution. The linear case $n = 2$ gives the piecewise linear hat function $B_2(x)$, which is the building block of our approximation space. Through the two-scale relation $B_{J,q}(x) = B_2(2^J x - q)$, we generate scaled and translated copies of B_2 at refinement level J. These are then restricted to the interval $[0, 1]$ and denoted by $\varphi_{J,q}(x)$ to form a finite-dimensional basis with compact support and the Kronecker delta property at the collocation points $\mu_m = (m + 1)/2^J$.

(2) **Function expansion:** Any sufficiently smooth function $\omega(t)$ is approximated by

$$\omega_J(t) = \sum_{q=-1}^{2^J-1} \ell_q \varphi_{J,q}(t) = \boldsymbol{\mathcal{L}}^\top \phi_J(t),$$

where $\phi_J(t)$ is the vector of basis functions and $\boldsymbol{\mathcal{L}}$ is the vector of expansion coefficients. Thanks to the Kronecker delta property, the coefficients are simply the nodal values $\ell_q = \omega(\mu_q)$, which greatly simplifies the implementation.

(3) **Operational matrices:** The key idea is to replace integral and derivative operators by matrix operations acting on the coefficient vector $\boldsymbol{\mathcal{L}}$. We construct:
- The ordinary integration matrix $\mathbf{M}_{\text{int}}$ (Theorem 3.1);
- The fractional Riemann–Liouville integration matrix $\mathbf{M}_\nu$ (Theorem 3.3);
- The delay matrix $\mathbf{D}_{\text{delay}}$ (Theorem 3.4);
- The derivative matrix $\mathbf{D}_{\text{der}}$ (Theorem 3.9);
- The product matrix for nonlinear terms (Theorem 3.7).

These matrices are sparse and can be precomputed once for a given refinement level J.

(4) **Algebraic system:** Using the operational matrices, the fractional delay differential equation (1.1) is transformed into a system of algebraic equations by collocation at the points μ_m. The initial conditions are imposed directly through the Kronecker property. The resulting system (linear or nonlinear) is then solved by standard numerical solvers.

This hierarchical structure ensures that each definition has a clear purpose in the overall algorithm, and the reader can refer back to this roadmap when encountering the detailed constructions below.

2. Basic definitions

2.1. **Notation.** To avoid ambiguity and resolve the notational conflict noted in standard conventions, we adopt the following notation throughout this paper:

2.2. **Integrals and derivatives of fractional order.**

Table 1. Summary of the main notation used in this paper.

Symbol	Description
$\mathcal{J}_a^\alpha$	Riemann–Liouville fractional integral operator (Definition 2.1)
$\mathcal{D}_a^\alpha$	Riemann–Liouville fractional derivative operator (Definition 2.2)
${}^C D^\alpha$	Caputo fractional derivative operator (Definition 2.3)
J	Refinement level (resolution index)
$N = 2^J + 1$	Number of basis functions
$\phi_J(x)$	Vector of linear cardinal B-spline scaling functions at level J
$\varphi_{J,j}(x)$	j-th scaling function at level J
$\mu_m = (m+1)/2^J$	Collocation points ($m = -1, 0, \dots, 2^J - 1$)
$\boldsymbol{\mathcal{L}}$	Vector of unknown expansion coefficients
$\mathbf{M}_{\text{int}}$	Operational matrix of ordinary integration (Theorem 3.1)
$\mathbf{M}_\nu$	Operational matrix of fractional integration of order ν (Theorem 3.3)
$\mathbf{D}_{\text{delay}}$	Operational matrix of delay (Theorem 3.4)
$\mathbf{D}_{\text{der}}$	Operational matrix of first derivative (Theorem 3.9)
$\mathbf{Id}$	Identity matrix of appropriate dimension

Definition 2.1. *For a real number $\alpha > 0$ and a locally integrable function $\omega(\xi)$ on $[a, x]$, the fractional integral of order α starting at a is defined by [23, 33]:*

$$\mathcal{J}_a^\alpha \omega(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} \omega(\tau)\, d\tau, \qquad t > a. \tag{2.1}$$

Definition 2.2. *Let $m - 1 \le \alpha < m$ where $m \in \mathbb{N}$. The Riemann-Liouville fractional derivative of order α is given as [23, 33]:*

$$\mathcal{D}_a^\alpha \omega(t) = \frac{d^m}{dx^m}\Big(\mathcal{J}_a^{m-\alpha}\omega(t)\Big) = \frac{1}{\Gamma(m-\alpha)} \frac{d^m}{dt^m} \int_a^t \frac{\omega(\tau)}{(t-\tau)^{\alpha-m+1}}\, d\tau. \tag{2.2}$$

It is worth noting that, under this definition, the fractional derivative of a constant is not zero, i.e. $\mathcal{D}_a^\alpha(\text{constant}) \neq 0$. This feature sometimes limits its applicability in physical models where a constant state should remain unchanged under differentiation.

Definition 2.3. *For $\alpha > 0$ and $m = \lceil \alpha \rceil$, the Caputo fractional derivative of order α is defined by [23, 33]:*

$${}^C D^\alpha \omega(t) = \begin{cases} \dfrac{1}{\Gamma(m-\alpha)} \displaystyle\int_a^t \frac{\omega^{(m)}(\tau)}{(t-\tau)^{\alpha-m+1}}\, d\tau, & m-1 < \alpha < m, \\[2ex] \dfrac{d^m}{dt^m}\omega(t), & \alpha = m \in \mathbb{N}. \end{cases} \tag{2.3}$$

The Caputo formulation remedies the aforementioned drawback: the derivative of a constant is zero,

$${}^C D^\alpha(\text{constant}) = 0.$$

Moreover, the operator is linear:

$${}^C D^\alpha \big[c_1\omega_1(t) + c_2\omega_2(t)\big] = c_1\, {}^C D^\alpha \omega_1(t) + c_2\, {}^C D^\alpha \omega_2(t), \tag{2.4}$$

for any constants c_1, c_2. A particularly useful formula for power functions is

$${}^C D^\alpha\, t^m = \begin{cases} 0, & m < \lceil \alpha \rceil, \\[2ex] \dfrac{\Gamma(m+1)}{\Gamma(m-\alpha+1)}\, t^{m-\alpha}, & m \ge \lceil \alpha \rceil \text{ or } \big(m \notin \mathbb{N} \text{ and } m > \lfloor \alpha \rfloor\big). \end{cases} \tag{2.5}$$

Note 2.4. *For $m-1<\alpha\leq m$, the Caputo fractional derivative is defined by*

$$^{C}D^{\alpha}\omega(t)=I^{m-\alpha}\left(\frac{\mathrm{d}^m}{\mathrm{d}t^m}\omega(t)\right). \tag{2.6}$$

Note 2.5. *A key relation linking the Riemann–Liouville fractional integral and the Caputo derivative is the Newton–Leibniz identity:*

$$I^{\alpha}\left(^{C}D^{\alpha}\omega(t)\right)=\omega(t)-\sum_{r=0}^{\lceil\alpha\rceil-1}\omega^{(r)}(0)\frac{t^r}{r!}. \tag{2.7}$$

2.3. **Cardinal B-spline functions.**

Definition 2.6. *The nth-order cardinal B-spline $B_n(x)$ for $n\geq 2$ it can be defined as follows:*

$$B_n(x)=(B_{n-1}*B_1)(x)=\int_{-\infty}^{\infty}B_{n-1}(x-t)B_1(t)dt=\int_0^1 B_{n-1}(x-t)dt, \tag{2.8}$$

here $B_1(x)=\chi_{[0,1]}(x)$ is the characteristic function on $[0,1]$.

Lemma 2.7. *The following formula can be obtained recursively through calculations:*

$$B_n(x)=\frac{x}{n-1}B_{n-1}(x)+\frac{n-x}{n-1}B_{n-1}(x-1), \tag{2.9}$$

where $Supp[B_n(x)]=[0,n]$.

Definition 2.8. *In explicit form, linear B-spline function $B_2(x)$ can be defined as follows:*

$$B_2(x)=\begin{cases}x & x\in[0,1]\\ 2-x & x\in[1,2]\\ 0 & o.w\end{cases}$$

Definition 2.9. *The two-scale relationship for linear B-splines functions $B_2(x)$ for $x\geq 0$ is defined as follows:*

$$B_{p,q}(x)=B_2(2^Jx-q), \tag{2.10}$$

where $J,q\in\mathbb{Z}$.

Define $V_J=\{q:[2^{-J}q,2^{-J}(2+q)]\cap(0,1)\neq\oslash\}$ for $J\in\mathbb{Z}$. It can be derived that $min\{V_J\}=-1$ and $max\{V_J\}=2^J-1$ where $J\in\mathbb{Z}$. Since the support of $B_{J,q}(x)$ may not be in range of $[0,1]$, to take the scaling functions into the interval $[0,1]$, we let

$$\varphi_{J,q}(x)=B_{J,q}(x).\chi_{[0,1]}(x),\qquad J\in\mathbb{Z},q\in V_J. \tag{2.11}$$

As a result, one can write

$$\varphi_{J,q}(x)=\begin{cases}\begin{cases}1-2^Jx, & x\in[0,2^{-J}),\\ 0, & o.w,\end{cases} & q=-1,\\[2ex] \begin{cases}2^Jx-q, & x\in[2^{-J}q,2^{-J}(q+1)),\\ 2-2^Jx+q, & x\in[2^{-J}(q+1),2^{-J}(q+2)),\\ 0, & o.w,\end{cases} & q=0,1,\cdots,2^J-2,\\[2ex] \begin{cases}2^Jx-2^J+1, & x\in[1-2^{-J},1),\\ 0, & o.w,\end{cases} & q=2^J-1.\end{cases} \tag{2.12}$$

Let us consider the set of cardinal B-spline functions of J-th degree:

$$\phi_J(t)=\left[\varphi_{J,-1}(t),\varphi_{J,0}(t),...,\varphi_{J,2^J-1}(t)\right]^T\subset L^2[0,1], \tag{2.13}$$

and assume that

$$W = \mathrm{span}\{\varphi_{J,-1}(t), \varphi_{J,0}(t), ..., \varphi_{J,2^J-1}(t)\}.$$

It is well-known that the orthogonal projection onto the finite-dimensional space W yields the unique best L^2-approximation of ω. However, in this work we do not employ the orthogonal projection. Instead, we construct an *interpolant* $\omega_J \in W$ defined by the nodal values of ω at the dyadic points. It is crucial to emphasize that this interpolation operator is generally *not* the L^2-orthogonal projection; therefore, the inequality $\|\omega - \omega_J\|_2 \leq \|\omega - h\|_2$ for all $h \in W$ does not hold in general. Nevertheless, according to classical approximation theory, for sufficiently smooth functions ω, this interpolation operator is stable in L^2 and satisfies the following quasi-optimal error bound:

$$\|\omega - \omega_J\|_2 \leq C \inf_{h\in W} \|\omega - h\|_2, \tag{2.14}$$

where $C > 0$ is a constant independent of J and ω.

Since the interpolant $\omega_J \in W$, $\mathcal{L} = [l_{-1}, l_0, \cdots, l_{2^J-1}]^T$ will be a unique vector such that:

$$\omega(t) \simeq \omega_J(t) = \sum_{q=-1}^{2^J-1} l_q \varphi_{J,q}(t) = \mathcal{L}^T \phi_J(t). \tag{2.15}$$

The function $\varphi_{J,q}(t)$ can be expressed as follows:

$$\varphi_{J,q}\left(\frac{m+1}{2^J}\right) = \delta_{q,m} = \begin{cases} 1, & q = m, \\ 0, & q \neq m, \end{cases} \qquad m = -1, 0, \ldots, 2^J - 1. \tag{2.16}$$

For a continuous function ω on $[0, 1]$, the coefficients l_q in Eq. (2.15) are determined as:

$$l_q = \omega\left(\frac{q+1}{2^J}\right), \qquad q = -1, 0, \cdots, 2^J - 1. \tag{2.17}$$

Let $\mathbf{e}_m$ be the $(m + 2)$-th column of the identity matrix of order $2^J + 1$. Then it is easy to verify that

$$\phi_J(\mu_m) = \mathbf{e}_m, \qquad \mu_m = \frac{m+1}{2^J}, \qquad m = -1, 0, \ldots, 2^J - 1. \tag{2.18}$$

3. Description of the method

In this section, we first construct the integer integral and the stochastic integral of linear cardinal B-spline functions on $[0, 1]$. Then, a numerical approach based on matrix operators will be presented.

3.1. **Operational matrices.** Let $\phi_J(x)$ be the vector of linear cardinal B-spline basis functions defined in (2.13) on the domain $x \in [0, 1]$. By setting $N = 2^J + 1$, and in the sense of interpolation at the collocation points $\mu_m = \frac{(m+1)}{2^J}$ for $m = -1, 0, \ldots, 2^J - 1$ the operational matrix of integration based on the linear cardinal B-spline vector can be defined as follows:

$$\int_0^x \phi_J(t)\, dt := \mathbf{M}_{\text{int}}\, \phi_J(x),\ x \in [0, 1],\ J \in \mathbb{N}. \tag{3.1}$$

Theorem 3.1. *Based on the assumptions of (3.1), the equality holds exactly at these points where the matrix be defined by*

$$(\mathbf{M}_{int})_{i,q} = \int_0^{\mu_i} \varphi_{J,q}(t)\, dt,\ i, q = -1, 0, \ldots, 2^J - 1,$$

when $\mathbf{M}_{int} \in \mathbb{R}^{N\times N}$ *is as follows:*

$$\mathbf{M}_{int} = \begin{pmatrix} 0 & \zeta & \zeta & \zeta & \dots & \zeta \\ 0 & \zeta & 2\zeta & 2\zeta & \dots & 2\zeta \\ 0 & 0 & \zeta & 2\zeta & \dots & 2\zeta \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \zeta & 2\zeta \\ 0 & 0 & 0 & 0 & 0 & \zeta \end{pmatrix}, \qquad \textit{and} \qquad \zeta = 2^{-J-1}. \tag{3.2}$$

Proof. Let $x \in [0,1]$, then define

$$\mathcal{I}\varphi_{J,q}(x) = \int_0^x \varphi_{J,q}(t)dt. \tag{3.3}$$

For each basis function $\varphi_{J,q}(x)$, we compute its integral explicitly. The piecewise linear form of $\varphi_{J,q}$ allows direct integration. In order to calculate the aforementioned integral, three cases are analyzed and evaluated in detail:

Case 1: if $q = -1$

In this case, if $x \in [0, 2^{-J}]$ then Eq. (2.12) is calculated as follows

$$\mathcal{I}\varphi_{J,-1}(x) = \int_0^x \varphi_{J,-1}(t)dt = \int_0^x (1 - 2^J t)dt = x - 2^{J-1}x^2. \tag{3.4}$$

If $x > 2^{-J}$ then Eq. (2.12) can be obtained as

$$\mathcal{I}\varphi_{J,-1}(x) = \int_0^{2^{-J}} \varphi_{J,-1}(t)dt = \int_0^{2^{-J}} (1 - 2^J t)dt = 2^{-J-1}. \tag{3.5}$$

Case 2: if $q = 0, 1, \cdots, 2^J - 2$

In this case, if $x \in [0, 2^{-J}q]$ then:

$$\mathcal{I}\varphi_{J,q}(x) = 0. \tag{3.6}$$

If $x \in [2^{-J}q, 2^{-J}(q+1)]$ then:

$$\mathcal{I}\varphi_{J,q}(x) = \int_{2^{-J}q}^x (2^J t - q)dt = 2^{J-1}x^2 - qx + 2^{-J-1}q^2. \tag{3.7}$$

If $x \in [2^{-J}(q+1), 2^{-J}(q+2)]$ then:

$$\begin{aligned} \mathcal{I}\varphi_{J,q}(x) &= \int_{2^{-J}q}^{2^{-J}(q+1)} (2^J t - q)dt + \int_{2^{-J}(q+1)}^x (2 - (2^J t - q))dt \\ &= (2+q)x - 2^{J-1}x^2 - 2^{-J-1}(q^2 + 4q + 2). \end{aligned} \tag{3.8}$$

Else if $x > 2^{-J}(q+2)$ then:

$$\begin{aligned} \mathcal{I}\varphi_{J,q}(x) &= \int_{2^{-J}q}^{2^{-J}(q+1)} (2^J t - q)dt + \int_{2^{-J}(q+1)}^{2^{-J}(q+2)} (2 - (2^J t - q))dt \\ &= 2^{-J}. \end{aligned} \tag{3.9}$$

Case 3: $q = 2^J - 1$.

In this case, if $x \in [0, 2^{-J}q]$ then:

$$\mathcal{I}\varphi_{J,q}(x) = 0. \tag{3.10}$$

If $x \in [1 - 2^{-J}, 1]$ then:

$$\mathcal{I}\varphi_{J,2^J-1}(x) = \int_{1-2^{-J}}^x (2^J t - 2^J + 1)dt = 2^{J-1}x^2 - qx + 2^{-J-1}q^2. \tag{3.11}$$

Else if $x > 2^{-J}(q+2)$ then:

$$\mathcal{I}\varphi_{J,2^J-1}(x) = \int_{1-2^{-J}}^{1} (2^J t - 2^J + 1)dt = 2^{-J-1}. \tag{3.12}$$

Now, by using Eqs. (3.4)-(3.12), one can write

$$\mathcal{I}\varphi_{J,q}(x) = \begin{cases} \begin{cases} 1 - 2^{J-1}x^2, & x \in [0, 2^{-J}), \\ 2^{-J-1}, & x \geqslant 2^{-J}, \\ 0, & o.w, \end{cases} & q = -1, \\ \\ \begin{cases} 2^{J-1}x^2 - qx + 2^{-J-1}q^2, & x \in [2^{-J}q, 2^{-J}(q+1)), \\ 2^{-J-1}, & x \in [2^{-J}(q+1), 2^{-J}(q+2)), \\ 2^{-J}, & x \geqslant 2^{-J}(q+2), \\ 0, & o.w, \end{cases} & q = 0, 1, \cdots, 2^J - 2, \\ \\ \begin{cases} 2^{J-1}x^2 - (2^J - 1)x + 2^{-J-1}(2^J - 1)^2, & x \in [1 - 2^{-J}, 1), \\ 2^{-J-1}, & x \geqslant 1, \\ 0, & o.w, \end{cases} & q = 2^J - 1. \end{cases} \tag{3.13}$$

By using Eq. (2.13) and Eq. (3.13), it can be obtained:

$$\mathcal{I}\phi_J(x) = \mathbf{M}_{\text{int}}\phi_J(x), \tag{3.14}$$

where

$$(M_{\text{int}})_{i,q} = \mathcal{I}(\varphi_q(\mu_i)), \qquad \mu_i = \frac{i+1}{2^J}, \qquad i, q = -1, 0, ..., 2^J - 1. \tag{3.15}$$

Therefore, it can be proven that the operational matrix of integration based on the linear cardinal B-spline vector, $\mathbf{M}_{\text{int}}$ of order $(2^J+1) \times (2^J+1)$ is defined as follows:

$$\mathbf{M}_{\text{int}} = \begin{pmatrix} 0 & \zeta & \zeta & \zeta & \dots & \zeta \\ 0 & \zeta & 2\zeta & 2\zeta & \dots & 2\zeta \\ 0 & 0 & \zeta & 2\zeta & \dots & 2\zeta \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \zeta & 2\zeta \\ 0 & 0 & 0 & 0 & 0 & \zeta \end{pmatrix}, \tag{3.16}$$

where $\zeta = 2^{-J-1}$. □

Remark 3.2. *Evaluating these expressions at the collocation points $\mu_i = (i+1)/2^J$ for $i = -1, 0, \dots, 2^J - 1$ gives the entries $(\mathbf{M}_{int})_{i,q}$. The resulting matrix is exactly the upper triangular form stated in the theorem. Since the integrals are piecewise quadratic polynomials and we evaluate them at the collocation points, the equality $\int_0^x \phi_J(t)dt = \mathbf{M}_{int}\phi_J(x)$ holds at these points; this is sufficient for the collocation method, as all subsequent operations are performed at the collocation points. The index range -1 to $2^J - 1$ ensures that both boundaries $x = 0$ and $x = 1$ are included.*

Theorem 3.3. *Let $\phi_J(x)$ be the linear cardinal B-spline vector defined in Eq. (2.13). Then, the fractional Riemann–Liouville integral operational matrix of order $\nu \in (0,1)$ based on the linear cardinal B-spline vector is defined as follows:*

$$_0\mathcal{J}_x^{\nu}\phi_J(x) = \mathbf{M}_{\nu}\,\phi_J(x), \tag{3.17}$$

where $\mathbf{M}_\nu$ *is an upper triangular matrix of order* $(2^J+1)\times(2^J+1)$ *given by*

$$\mathbf{M}_\nu = \begin{pmatrix} 0 & \omega_0 & \omega_1 & \omega_2 & \cdots & \omega_{2^J-1} \\ & \gamma_0 & \gamma_1 & \gamma_2 & \cdots & \gamma_{2^J-1} \\ & & \gamma_0 & \gamma_1 & \cdots & \gamma_{2^J-2} \\ & & & \ddots & \ddots & \vdots \\ & & & & \gamma_0 & \gamma_1 \\ & & & & & \gamma_0 \end{pmatrix}, \tag{3.18}$$

with

$$\begin{aligned} \omega_0 &= \nu\lambda, \\ \omega_q &= \big[(\nu+1)(q+1)^\nu - (q+1)^{\nu+1} + q^{\nu+1}\big]\lambda, \quad q = 1,\dots,2^J-1, \\ \gamma_0 &= \lambda, \\ \gamma_1 &= \big[2^{\nu+1}-2\big]\lambda, \\ \gamma_q &= \big[(1+q)^{\nu+1} - 2q^{\nu+1} + (q-1)^{\nu+1}\big]\lambda, \quad q = 2,\dots,2^J-1, \end{aligned}$$

and

$$\lambda = \frac{1}{2^{J\nu}\,\Gamma(\nu+2)}.$$

Proof. Let $x\in[0,1]$ and define

$$\mathcal{J}^\nu \varphi_{J,q}(x) = {}_0\mathcal{J}_x^\nu \varphi_{J,q}(x) = \frac{1}{\Gamma(\nu)}\int_0^x (x-t)^{\nu-1}\varphi_{J,q}(t)\,dt. \tag{3.19}$$

To evaluate the integral in (3.19), three cases are examined in detail.
Case 1: $q=-1$.
If $x\in[0,2^{-J}]$, then

$$\begin{aligned} \mathcal{J}^\nu \varphi_{J,-1}(x) &= \frac{1}{\Gamma(\nu)}\int_0^x (x-t)^{\nu-1}(1-2^Jt)\,dt \\ &= \frac{1}{\Gamma(\nu)}\left(\frac{x^\nu}{\nu} - \frac{2^Jx^{\nu+1}}{\nu(\nu+1)}\right). \end{aligned} \tag{3.20}$$

If $x>2^{-J}$, then

$$\begin{aligned} \mathcal{J}^\nu \varphi_{J,-1}(x) &= \frac{1}{\Gamma(\nu)}\int_0^{2^{-J}} (x-t)^{\nu-1}(1-2^Jt)\,dt \\ &= \frac{1}{\Gamma(\nu)}\left(\frac{x^\nu}{\nu} - \frac{2^Jx^{\nu+1}}{\nu(\nu+1)} + \frac{(2^Jx-1)^{\nu+1}}{\nu(\nu+1)2^{J\nu}}\right). \end{aligned} \tag{3.21}$$

Case 2: $q=0,1,\dots,2^J-2$.
If $x\in[0,2^{-J}q]$, then $\mathcal{J}^\nu\varphi_{J,q}(x)=0$.
If $x\in[2^{-J}q,2^{-J}(q+1)]$, then

$$\begin{aligned} \mathcal{J}^\nu \varphi_{J,q}(x) &= \frac{1}{\Gamma(\nu)}\int_{2^{-J}q}^x (x-t)^{\nu-1}(2^Jt-q)\,dt \\ &= \frac{1}{2^{J\nu}\Gamma(\nu+2)}\big[(2^Jx-q)^{\nu+1}\big]. \end{aligned} \tag{3.22}$$

If $x \in [2^{-J}(q+1), 2^{-J}(q+2)]$, then

$$\mathcal{J}^{\nu}\varphi_{J,q}(x) = \frac{1}{\Gamma(\nu)}\left[\int_{2^{-J}q}^{2^{-J}(q+1)}(x-t)^{\nu-1}(2^Jt-q)\,dt + \int_{2^{-J}(q+1)}^{x}(x-t)^{\nu-1}\big(2-(2^Jt-q)\big)\,dt\right]$$
$$= \frac{1}{2^{J\nu}\Gamma(\nu+2)}\big[(2^Jx-q)^{\nu+1} - 2(2^Jx-q-1)^{\nu+1}\big]. \tag{3.23}$$

If $x > 2^{-J}(q+2)$, then

$$\mathcal{J}^{\nu}\varphi_{J,q}(x) = \frac{1}{\Gamma(\nu)}\left[\int_{2^{-J}q}^{2^{-J}(q+1)}(x-t)^{\nu-1}(2^Jt-q)\,dt + \int_{2^{-J}(q+1)}^{2^{-J}(q+2)}(x-t)^{\nu-1}\big(2-(2^Jt-q)\big)\,dt\right]$$
$$= \frac{1}{2^{J\nu}\Gamma(\nu+2)}\big[(2^Jx-q)^{\nu+1} - 2(2^Jx-q-1)^{\nu+1} + (2^Jx-q-2)^{\nu+1}\big]. \tag{3.24}$$

Case 3: $q = 2^J - 1$.
If $x \in [0, 1-2^{-J}]$, then $\mathcal{J}^{\nu}\varphi_{J,2^J-1}(x) = 0$.
If $x \in [1-2^{-J}, 1]$, then

$$\mathcal{J}^{\nu}\varphi_{J,2^J-1}(x) = \frac{1}{\Gamma(\nu)}\int_{1-2^{-J}}^{x}(x-t)^{\nu-1}(2^Jt-2^J+1)\,dt$$
$$= \frac{1}{2^{J\nu}\Gamma(\nu+2)}\big[(2^Jx-2^J+1)^{\nu+1}\big]. \tag{3.25}$$

If $x > 1$, then

$$\mathcal{J}^{\nu}\varphi_{J,2^J-1}(x) = \frac{1}{\Gamma(\nu)}\int_{1-2^{-J}}^{1}(x-t)^{\nu-1}(2^Jt-2^J+1)\,dt$$
$$= \frac{-(2^Jx-2^J)^{\nu}(\nu+1) - (2^Jx-2^J)^{\nu+1} + (2^Jx-2^J+1)^{\nu+1}}{2^{J\nu}\Gamma(\nu+2)}. \tag{3.26}$$

Collecting the results of Eqs. (3.20)–(3.26), we can summarize

$$\mathcal{J}^{\nu}\varphi_{J,q}(x) = \begin{cases} \dfrac{\left(\frac{x^{\nu}}{\nu} - \frac{2^Jx^{\nu+1}}{\nu(\nu+1)}\right)}{\Gamma(\nu)}, & x \in [0, \frac{1}{2^J}),\ q=-1,\\[2ex] \dfrac{\left(\frac{x^{\nu}}{\nu} - \frac{2^Jx^{\nu+1}}{\nu(\nu+1)} + \frac{(2^Jx-1)^{\nu+1}}{\nu(\nu+1)2^{J\nu}}\right)}{\Gamma(\nu)}, & x \geq \frac{1}{2^J},\ q=-1,\\[2ex] \dfrac{(2^Jx-q)^{\nu+1}}{2^{J\nu}\Gamma(\nu+2)}, & x \in \big[\frac{q}{2^J}, \frac{q+1}{2^J}\big],\ q=0,\dots,2^J-2,\\[2ex] \dfrac{(2^Jx-q)^{\nu+1} - 2(2^Jx-q-1)^{\nu+1}}{2^{J\nu}\Gamma(\nu+2)}, & x \in \big[\frac{q+1}{2^J}, \frac{q+2}{2^J}\big],\ q=0,\dots,2^J-2,\\[2ex] \dfrac{(2^Jx-q)^{\nu+1} - 2(2^Jx-q-1)^{\nu+1} + (2^Jx-q-2)^{\nu+1}}{2^{J\nu}\Gamma(\nu+2)}, & x \geq \frac{q+2}{2^J},\ q=0,\dots,2^J-2,\\[2ex] \dfrac{(2^Jx-2^J+1)^{\nu+1}}{2^{J\nu}\Gamma(\nu+2)}, & x \in \big[1-\frac{1}{2^J}, 1\big],\ q=2^J-1,\\[2ex] \dfrac{-(2^Jx-2^J)^{\nu} - (2^Jx-2^J)^{\nu+1} + (2^Jx-2^J+1)^{\nu+1}}{2^{J\nu}\Gamma(\nu+2)}, & x > 1,\ q=2^J-1,\\[2ex] 0, & \text{otherwise.} \end{cases} \tag{3.27}$$

Now, using the vector form $\phi_J(x) = [\varphi_{J,-1}(x), \varphi_{J,0}(x), \dots, \varphi_{J,2^J-1}(x)]^T$ and evaluating at the collocation points $\mu_i = \frac{i+1}{2^J}$ for $i = -1, 0, \dots, 2^J-1$, we obtain

$$\mathcal{J}^{\nu}\phi_J(\mu_i) = \mathbf{M}_{\nu}\,\phi_J(\mu_i), \tag{3.28}$$

where the entries of $\mathbf{M}_\nu$ are

$$(\mathbf{M}_\nu)_{i,q} = \mathcal{J}^\nu \varphi_{J,q}(\mu_i), \qquad i, q = -1, 0, \ldots, 2^J - 1. \tag{3.29}$$

Direct computation using (3.27) and (3.29) yields the explicit upper-triangular structure given in the theorem, with the coefficients ω_q and γ_q as defined above. Hence the fractional integral operational matrix satisfies

$${}_0\mathcal{J}_x^\nu \phi_J(x) = \mathbf{M}_\nu \, \phi_J(x), \tag{3.30}$$

which completes the proof.

□

Theorem 3.4. *Let $\phi_J(x)$ be the linear cardinal B-spline vector defined in Eq. (2.13). Let $c > 0$ and $\tau \geq 0$ be constants such that $0 \leq cx - \tau \leq 1$ for all $x \in [0, 1]$. Then there exists a unique matrix $\mathbf{D}_{delay}$ of order $(2^J + 1) \times (2^J + 1)$, called the delay operational matrix, satisfying*

$$\phi_J(cx - \tau) = \mathbf{D}_{delay} \, \phi_J(x), \qquad \forall x \in [0, 1]. \tag{3.31}$$

The entries of $\mathbf{D}_{delay}$ are given by

$$d_{i,q} = \varphi_{J,i}(c\mu_q - \tau), \qquad i, q = -1, 0, \ldots, 2^J - 1.$$

Proof. Let $x \in [0, 1]$ and fix $i \in \{-1, 0, \ldots, 2^J - 1\}$. Consider the composite function $\varphi_{J,i}(cx - \tau)$. Under the hypothesis $0 \leq cx - \tau \leq 1$, this function is well-defined and is a piecewise linear polynomial on $[0, 1]$, with possible breakpoints at the points where $cx - \tau = 2^{-J}k$ for integer k. Since the space

$$W = \mathrm{span}\{\varphi_{J,-1}, \varphi_{J,0}, \ldots, \varphi_{J,2^J-1}\}$$

consists exactly of all continuous piecewise linear functions on $[0, 1]$ with breakpoints at $2^{-J}k$, it follows that $\varphi_{J,i}(cx - \tau) \in W$. Therefore, there exist unique coefficients $d_{i,q}$, $q = -1, 0, \ldots, 2^J - 1$, such that

$$\varphi_{J,i}(cx - \tau) = \sum_{q=-1}^{2^J-1} d_{i,q} \, \varphi_{J,q}(x), \qquad \forall x \in [0, 1]. \tag{3.32}$$

The uniqueness of the coefficients follows from the linear independence of the basis functions $\{\varphi_{J,q}\}_{q=-1}^{2^J-1}$, which is a standard property of B-splines (see, e.g., [17]).

To determine the coefficients $d_{i,q}$, we evaluate both sides of (3.32) at the collocation points $x = \mu_m$, where $\mu_m = \frac{m+1}{2^J}$, for $m = -1, 0, \ldots, 2^J - 1$. Using the Kronecker delta property $\varphi_{J,q}(\mu_m) = \delta_{q,m}$, the right-hand side becomes

$$\sum_{q=-1}^{2^J-1} d_{i,q} \, \varphi_{J,q}(\mu_m) = \sum_{q=-1}^{2^J-1} d_{i,q} \, \delta_{q,m} = d_{i,m}.$$

The left-hand side evaluated at μ_m is $\varphi_{J,i}(c\mu_m - \tau)$. Hence,

$$d_{i,m} = \varphi_{J,i}(c\mu_m - \tau), \qquad m = -1, 0, \ldots, 2^J - 1. \tag{3.33}$$

This explicit formula shows that the entries of the delay matrix are simply the values of the source basis functions at the shifted collocation points. Since m is an arbitrary index, we can rewrite this as

$$d_{i,q} = \varphi_{J,i}(c\mu_q - \tau), \qquad i, q = -1, 0, \ldots, 2^J - 1.$$

Now, writing (3.32) in vector form for all i simultaneously gives

$$\phi_J(cx - \tau) = \mathbf{D}_{\text{delay}} \, \phi_J(x),$$

where $\mathbf{D}_{\text{delay}} = [d_{i,q}]$ is the $(2^J + 1) \times (2^J + 1)$ matrix with entries defined above.

Finally, we note that the matrix is sparse. Indeed, for a fixed column q, the value $\varphi_{J,i}(c\mu_q - \tau)$ is nonzero only for those indices i such that the point $c\mu_q - \tau$ lies in the support of $\varphi_{J,i}$. Since each basis function has compact support of length 2^{-J+1} and the supports overlap in at most two adjacent intervals, there are at most two such indices i for each q.

□

Corollary 3.5 (Sparsity of the delay matrix)**.** *The delay operational matrix $\mathbf{D}_{delay}$ is sparse. Specifically, each column of $\mathbf{D}_{delay}$ contains at most two nonzero entries.*

Proof. Recall that each linear cardinal B-spline function $\varphi_{J,i}(x)$ has compact support:

$$\operatorname{supp}(\varphi_{J,i}) = \left[2^{-J}i,\ 2^{-J}(i+2)\right] \cap [0,1].$$

For a fixed q, consider the point $c\mu_q - \tau$. This point lies in the interval $[0,1]$. The function $\varphi_{J,i}(c\mu_q - \tau)$ is nonzero only if $c\mu_q - \tau$ belongs to the interior of the support of $\varphi_{J,i}$. Since the support intervals have length 2^{-J} and are shifted by multiples of 2^{-J}, the point $c\mu_q - \tau$ can lie in at most two overlapping support intervals. Hence, for each fixed q, there are at most two indices i such that $d_{i,q} = \varphi_{J,i}(c\mu_q - \tau) \neq 0$. Therefore, each column of $\mathbf{D}_{delay}$ has at most two nonzero entries.

□

Note 3.6. *For illustration, consider the case $J = 2$ (hence $2^J = 4$), $c = 1$, and $\tau = 0.2$. The collocation points are $\xi_i = \frac{i+1}{4}$ for $i = -1, 0, 1, 2, 3$. The delay operational matrix $\mathbf{D}_{delay}$ of order 5×5 is computed explicitly as*

$$\mathbf{D}_{delay} = \begin{pmatrix} 0.8 & 0.2 & 0 & 0 & 0 \\ 0 & 0.6 & 0.4 & 0 & 0 \\ 0 & 0 & 0.4 & 0.6 & 0 \\ 0 & 0 & 0 & 0.2 & 0.8 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

This matrix exhibits the sparsity pattern described in Theorem 3.4: each row contains at most two nonzero entries, corresponding to the indices of the basis functions whose supports contain the shifted point $c\xi_i - \tau$.

Theorem 3.7. *Let $\phi_J(x)$ be linear cardinal B-spline vector defined in Eq. (2.13) and $C = [c_{-1}, c_0, \cdots, c_{2^J-1}]^T$ is an arbitrary vector. The operational matrix of product based on the linear cardinal B-spline vector is defined as follows:*

$$\phi_J(x)\phi_J^T(x)C = \widehat{C}\phi_J(x), \tag{3.34}$$

where

$$\widehat{C} = diag[c_{-1}, c_0, \cdots, c_{2^J-1}]. \tag{3.35}$$

Proof. It easily verified that $\phi_J(x)\phi_J^T(x)C$ is equal to

$$\begin{pmatrix} \varphi_{-1}(x)\varphi_{-1}(x) & \varphi_{-1}(x)\varphi_0(x) & \varphi_{-1}(x)\varphi_1(x) & \dots & \varphi_{-1}(x)\varphi_{2^J-1}(x) \\ \varphi_0(x)\varphi_{-1}(x) & \varphi_0(x)\varphi_0(x) & \varphi_0(x)\varphi_1(x) & \dots & \varphi_0(x)\varphi_{2^J-1}(x) \\ \varphi_1(x)\varphi_{-1}(x) & \varphi_1(x)\varphi_0(x) & \varphi_1(x)\varphi_1(x) & \dots & \varphi_1(x)\varphi_{2^J-1}(x) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varphi_{2^J-1}(x)\varphi_{-1}(x) & \varphi_{2^J-1}(x)\varphi_0(x) & \varphi_{2^J-1}(x)\varphi_1(x) & \dots & \varphi_{2^J-1}(x)\varphi_{2^J-1}(x) \end{pmatrix} \begin{pmatrix} c_{-1} \\ c_0 \\ c_1 \\ \vdots \\ c_{2^J-1} \end{pmatrix}. \tag{3.36}$$

Now, we approximate all functions $\varphi_i(x)\varphi_q(x)$ for $i, q = -1, 0, \cdots, 2^J - 1$ by the linear cardinal B-splines. Therefore,

$$\varphi_i(x)\varphi_q(x) = \sum_{k=-1}^{2^J-1} l_k\varphi_k(x), \tag{3.37}$$

where

$$l_k = \varphi_i(\mu_k)\varphi_q(\mu_k), \qquad \mu_k = \frac{k+1}{2^J}, \tag{3.38}$$

and

$$\varphi_i(x)\varphi_q(x) = \sum_{k=-1}^{2^J-1} \varphi_i(\mu_k)\varphi_q(\mu_k)\varphi_k(x) = \sum_{k=-1}^{2^J-1} \delta_{ik}\delta_{qk}\varphi_k(x)$$
$$= \begin{pmatrix} \varphi_{-1}(x) & 0 & 0 & \cdots & 0 \\ 0 & \varphi_0(x) & 0 & \cdots & 0 \\ 0 & 0 & \varphi_1(x) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \varphi_{2^J-1}(x) \end{pmatrix}. \tag{3.39}$$

Therefore,

$$\phi_J(x)\phi_J^T(x)C = \begin{pmatrix} \varphi_{-1}(x) & 0 & 0 & \cdots & 0 \\ 0 & \varphi_0(x) & 0 & \cdots & 0 \\ 0 & 0 & \varphi_1(x) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \varphi_{2^J-1}(x) \end{pmatrix} \begin{pmatrix} c_{-1} \\ c_0 \\ c_1 \\ \vdots \\ c_{2^J-1} \end{pmatrix} \tag{3.40}$$
$$= \widehat{C}\phi_J(x).$$

□

Lemma 3.8. *For linear cardinal B-spline scaling functions at level J, the matrix*

$$\mathbf{E} = \int_0^1 \phi_J'(t)\phi_J'(t)^\top dt,$$

is given explicitly by

$$\mathbf{E} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & & & & \\ \frac{1}{2} & 0 & -\frac{1}{2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \frac{1}{2} & 0 & -\frac{1}{2} \\ & & & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Proof. The entries of $\mathbf{E}$ are computed directly from the inner products of the derivatives of the linear B-spline scaling functions $\varphi_{J,q}(t)$. Because the support of each scaling function is limited to two adjacent intervals, the inner product is non-zero only for neighboring basis functions. The specific values follow from straightforward integration of piecewise-linear derivatives. See [17] for details.

□

Theorem 3.9. *The derivative of the scaling function vector $\phi_J(t)$ satisfies*

$$\phi_J'(t) = \mathbf{D}_{der}\phi_J(t), \tag{3.1}$$

where $\mathbf{D}_{der}$ is a $(2^J+1)\times(2^J+1)$ matrix given by

$$\mathbf{D}_{der} = \mathbf{E}\,\mathbf{P}^{-1}, \tag{3.2}$$

with

$$\mathbf{P} = \int_0^1 \phi_J(t)\phi_J(t)^\top dt, \qquad \mathbf{E} = \int_0^1 \phi_J'(t)\phi_J'(t)^\top dt,$$

and the elements of $\mathbf{P}$ and $\mathbf{E}$ are explicitly known.

Proof. We start from the definition of the dual basis $\widetilde{\phi}_J(t)$ of $\phi_J(t)$. Since the scaling function basis is not orthogonal, we define the dual basis via the Gramian matrix $\mathbf{P}$:

$$\widetilde{\phi}_J(t) = \mathbf{P}^{-1}\phi_J(t), \tag{3.3}$$

where

$$\mathbf{P} = \int_0^1 \phi_J(t)\phi_J(t)^\top dt = 2^{-J}\begin{bmatrix} \frac{1}{6} & \frac{1}{12} & & & \\ \frac{1}{12} & \frac{1}{3} & \frac{1}{12} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{12} & \frac{1}{3} & \frac{1}{12} \\ & & & \frac{1}{12} & \frac{1}{6} \end{bmatrix}. \tag{3.41}$$

The operational matrix $\mathbf{D}_{\text{der}}$ is defined by the relation

$$\phi'_J(t) = \mathbf{D}_{\text{der}}\,\phi_J(t). \tag{3.1}$$

Multiplying both sides from the left by $\widetilde{\phi}_J(t)^\top$ and integrating over $[0,1]$ yields

$$\int_0^1 \widetilde{\phi}_J(t)^\top \phi'_J(t)\,dt = \mathbf{D}_{\text{der}} \int_0^1 \widetilde{\phi}_J(t)^\top \phi_J(t)\,dt.$$

Using the duality relation $\displaystyle\int_0^1 \widetilde{\phi}_J(t)\phi_J(t)^\top dt = \mathbf{Id}$ (the identity matrix), the right-hand side simplifies to $\mathbf{D}_{\text{der}}$. Hence,

$$\mathbf{D}_{\text{der}} = \int_0^1 \widetilde{\phi}_J(t)^\top \phi'_J(t)\,dt. \tag{3.4}$$

Substituting the expression for $\widetilde{\phi}_J(t)$ from (3.3) into (3.4) gives

$$\mathbf{D}_{\text{der}} = \int_0^1 \big(\mathbf{P}^{-1}\phi_J(t)\big)^\top \phi'_J(t)\,dt = \mathbf{P}^{-1}\int_0^1 \phi_J(t)\phi'_J(t)^\top dt.$$

Notice that the integral $\int_0^1 \phi_J(t)\phi'_J(t)^\top dt$ is exactly the matrix $\mathbf{E}^\top$. However, from the explicit calculation of the derivatives of linear B-splines (see Lemma 3.8), it turns out that $\mathbf{E}$ is symmetric. Therefore,

$$\mathbf{D}_{\text{der}} = \mathbf{E}\,\mathbf{P}^{-1}, \tag{3.2}$$

which completes the proof. □

3.2. **Numerical algorithm for solving FDDEs.** The complete numerical procedure for solving the multi-term Caputo fractional delay differential equation (1.1) with initial conditions (1.2) and history function (1.3) is summarized in the following step by step algorithm.

(1) Input parameters and basis setup
- Choose resolution level J (number of basis functions: $N = 2^J + 1$).
- Input fractional orders α, β, γ, delay parameters c, τ, domain $[0, R]$, initial values λ_r, history function $g(t)$, and right-hand side f.
- If $R \neq 1$, perform scaling $t = Rx$ to transform the problem to $[0, 1]$.
- Construct the linear cardinal B-spline vector $\phi_J(x) = [\varphi_{J,-1}(x), \ldots, \varphi_{J,2^J-1}(x)]^\top$.
- Define collocation points $\mu_m = \frac{m+1}{2^J}$ for $m = -1, 0, \ldots, 2^J - 1$.

(2) Construct operational matrices
- Compute the ordinary integration matrix $\mathbf{M}_{\text{int}}$ using Theorem 3.1.
- Compute the fractional integration matrix $\mathbf{M}_\nu$ for required orders ν using Theorem 3.3.
- Compute the delay matrix $\mathbf{D}_{\text{delay}}$ for $cx - \tau$ using Theorem 3.4.
- Compute the derivative matrix $\mathbf{D}_{\text{der}}$ using Theorem 3.9.

- (For nonlinear problems) Prepare the product matrix rule from Theorem 3.7.

(3) Build Caputo fractional derivative matrices
- Let $m_\alpha = \lceil \alpha \rceil$, $m_\beta = \lceil \beta \rceil$, $m_\gamma = \lceil \gamma \rceil$.
- Construct integer-order derivative matrices:

$$\mathbf{D}_{m_\alpha} = (\mathbf{D}_{\text{der}})^{m_\alpha}, \quad \mathbf{D}_{m_\beta} = (\mathbf{D}_{\text{der}})^{m_\beta}, \quad \mathbf{D}_{m_\gamma} = (\mathbf{D}_{\text{der}})^{m_\gamma}.$$

- Form Caputo fractional derivative matrices:

$$\mathbf{D}^{(\alpha)} = \mathbf{D}_{m_\alpha} \mathbf{M}_{m_\alpha-\alpha},\ \mathbf{D}^{(\beta)} = \mathbf{D}_{m_\beta} \mathbf{M}_{m_\beta-\beta},\ \mathbf{D}^{(\gamma)} = \mathbf{D}_{m_\gamma} \mathbf{M}_{m_\gamma-\gamma}.$$

(4) Approximate the solution and its terms
- Express the unknown solution as $w_J(x) = \boldsymbol{\mathcal{L}}^\top \phi_J(x)$ where $\boldsymbol{\mathcal{L}} = [\ell_{-1}, \ell_0, \ldots, \ell_{2^J-1}]^\top$ is the coefficient vector.
- Approximate all terms appearing in (1.1):

$$\begin{aligned}
{}^C D^\alpha w_J(x) &\approx \boldsymbol{\mathcal{L}}^\top \mathbf{D}^{(\alpha)} \phi_J(x),\\
w_J(x) &\approx \boldsymbol{\mathcal{L}}^\top \phi_J(x),\\
{}^C D^\beta w_J(x) &\approx \boldsymbol{\mathcal{L}}^\top \mathbf{D}^{(\beta)} \phi_J(x),\\
w_J(cx-\tau) &\approx \boldsymbol{\mathcal{L}}^\top \mathbf{D}_{\text{delay}} \phi_J(x),\\
{}^C D^\gamma w_J(cx-\tau) &\approx \boldsymbol{\mathcal{L}}^\top \mathbf{D}^{(\gamma)} \mathbf{D}_{\text{delay}} \phi_J(x).
\end{aligned}$$

(5) Collocation and system formation
- Substitute the approximations into equation (1.1).
- Evaluate the resulting equation at each collocation point $x = \mu_m$, $m = -1, 0, \ldots, 2^J - 1$.
- Using the Kronecker property $\phi_J(\mu_m) = \mathbf{e}_m$, obtain a system of N equations:

$$\begin{aligned}
\boldsymbol{\mathcal{L}}^\top \mathbf{D}^{(\alpha)} \mathbf{e}_m = f\big(&\mu_m,\ \boldsymbol{\mathcal{L}}^\top \mathbf{e}_m,\ \boldsymbol{\mathcal{L}}^\top \mathbf{D}^{(\beta)} \mathbf{e}_m,\\
&\boldsymbol{\mathcal{L}}^\top \mathbf{D}_{\text{delay}} \mathbf{e}_m,\ \boldsymbol{\mathcal{L}}^\top \mathbf{D}^{(\gamma)} \mathbf{D}_{\text{delay}} \mathbf{e}_m\big).
\end{aligned}$$

(6) Enforce initial conditions
- For each $r = 0, 1, \ldots, \lceil \alpha \rceil - 1$, impose

$$w^{(r)}(0) = \lambda_r \implies \boldsymbol{\mathcal{L}}^\top (\mathbf{D}_{\text{der}})^r \phi_J(0) = \lambda_r.$$

- Replace the first $\lceil \alpha \rceil$ equations of the collocation system with these initial condition equations.

(7) Solve the algebraic system
- **Linear FDDE:** After collocation and imposition of initial conditions, the system reduces to

$$\mathbf{A}\boldsymbol{\mathcal{L}} = \mathbf{b},$$

where $\mathbf{A}$ is a sparse matrix due to the banded structure of the operational matrices. The sparsity pattern of $\mathbf{A}$ can be exploited by using a direct sparse solver such as MATLAB's `sparse` backslash operator or UMFPACK. For moderate J (e.g., $J \leq 10$), Gaussian elimination on the full matrix is also feasible.
- **Nonlinear FDDE:** The system becomes

$$\mathbf{F}(\boldsymbol{\mathcal{L}}) = \mathbf{0},$$

where $\mathbf{F}$ is a nonlinear vector-valued function. We solve this using Newton's method:

$$\boldsymbol{\mathcal{L}}^{(k+1)} = \boldsymbol{\mathcal{L}}^{(k)} - \left[\mathbf{J}_F(\boldsymbol{\mathcal{L}}^{(k)})\right]^{-1} \mathbf{F}(\boldsymbol{\mathcal{L}}^{(k)}),$$

where $\mathbf{J}_F$ is the Jacobian matrix of $\mathbf{F}$. The Jacobian can be computed analytically using the product matrix rule (Theorem 3.7), or approximated via finite differences. The initial guess is chosen as $\boldsymbol{\mathcal{L}}^{(0)} = [g(\mu_{-1}), \ldots, g(\mu_{2^J-1})]^\top$ using the history function, or simply the zero vector. The iteration is terminated when $\|\mathbf{F}(\boldsymbol{\mathcal{L}}^{(k)})\|_\infty < 10^{-12}$.

(8) Reconstruct the solution
   - Once $\boldsymbol{\mathcal{L}}$ is found, construct the approximate solution on $[0,1]$:

   $$w_J(x) = \boldsymbol{\mathcal{L}}^\top \phi_J(x).$$

   - If scaling $t = Rx$ was applied, transform back to obtain $w_J(t)$ on $[0,R]$.
   - For $t < 0$, set $w_J(t) = g(t)$ using the history function.

(9) Error analysis (optional)
   - Compute the residual by substituting $w_J(t)$ into the original equation.
   - If an exact solution is available, evaluate the absolute/relative error at the collocation points.

Remark 3.10 (Index ranges and boundary treatment)**.** *It is crucial to specify the index ranges used throughout the paper. The collocation points are defined as*

$$\mu_m = \frac{m+1}{2^J}, \qquad m = -1, 0, 1, \ldots, 2^J - 1.$$

Consequently, the total number of collocation points (and basis functions) is $N = 2^J + 1$. The choice of starting index $m = -1$ ensures that the interval boundaries are included as collocation points, since

$$\mu_{-1} = \frac{-1+1}{2^J} = 0, \qquad \mu_{2^J-1} = \frac{2^J - 1 + 1}{2^J} = 1.$$

Thus, both boundaries of the domain $[0,1]$ are captured exactly by the collocation grid.

The Kronecker delta property $\varphi_{J,q}(\mu_m) = \delta_{q,m}$ implies that the expansion coefficients are exactly the nodal values:

$$\ell_q = \omega_J(\mu_q) = \omega(\mu_q).$$

In particular, at the left boundary we have

$$\omega_J(0) = \omega_J(\mu_{-1}) = \ell_{-1}.$$

Thus, the initial condition $\omega(0) = \lambda_0$ is imposed simply by setting $\ell_{-1} = \lambda_0$.

Remark 3.11. *The algorithm exploits the sparsity of all operational matrices:*

- *Matrices $\mathbf{M}_{int}$, $\mathbf{M}_\nu$, and $\mathbf{D}_{der}$ are banded (upper-triangular or tridiagonal).*
- *The delay matrix $\mathbf{D}_{delay}$ has at most two nonzero entries per column (Corollary 3.5).*
- *Consequently, the algebraic system involves sparse matrices, enabling efficient solvers even for large J.*

Remark 3.12. *The following points should be noted:*

- *All operational matrices depend only on J, α, β, γ, c, and τ; they can be precomputed and stored.*
- *For variable delays or more complex arguments, $\mathbf{D}_{delay}$ must be recomputed accordingly.*
- *The method extends directly to systems of FDDEs by applying the same procedure componentwise.*

4. Some results and convergence analysis

Theorem 4.1. *Let $\omega(t) \in C^2[0,1]$ and $\omega_J(t)$ is approximate solutions of ω be the linear cardinal B-spline. Then the upper bound error for the method is*

$$E_J(t) = \|\omega(t) - \omega_J(t)\|_2 \leq 2^{-2J-1} \|\omega''(t)\|_2. \tag{4.1}$$

Proof. Let $D_q = \{t | 2^{-J} q \leq t \leq 2^{-J}(q+1)\}$ for $q = 0, 1, \cdots, 2^J - 1$ and define

$$E_q(t) = \begin{cases} \omega(t) - \omega_J(t) & t \in D_q \\ 0 & o.w. \end{cases} \tag{4.2}$$

Therefore,

$$\begin{aligned}
E_q(t) &= \omega(t) - \omega_J(t)\\
&= \omega(t) - \sum_{p=-1}^{2^J-1} \omega(2^{-J}(p+1))\varphi_p(t)\\
&= \omega(t) - \left[\omega(2^{-J}q)\varphi_{q-1}(t) + \omega(2^{-J}(q+1))\varphi_q(t)\right]\\
&= \omega(t) - \left[\omega(2^{-J}q)(-2^Jt+q+1) + \omega(2^{-J}(q+1))(2^Jt-q)\right]\\
&= \omega(t) - \left[\omega(2^{-J}q) - \omega(2^{-J}q)(2^Jt-q) + \omega(2^{-J}(q+1))(2^Jt-q)\right]\\
&= \omega(t) - \left[\omega(2^{-J}q) + 2^J(t-2^{-J}q)\left(\omega(2^{-J}(q+1)) - \omega(2^{-J}q)\right)\right]\\
&= \omega(t) - \left[\omega(2^{-J}q) + (t-2^{-J}q)\left(\frac{\omega(2^{-J}(q+1)) - \omega(2^{-J}q)}{2^{-J}}\right)\right]\\
&\simeq \omega(t) - \left[\omega(2^{-J}q) + (t-2^{-J}q)\omega^{'}(2^{-J}q)\right].
\end{aligned} \tag{4.3}$$

By writing the Taylor expansion for function $\omega(t)$ in Eq. (4.3) in terms of $(t-2^{-J}q)$, we have:

$$E_q(t) \simeq \frac{(t-2^{-J}q)^2}{2}\omega^{''}(\zeta_q), \qquad \zeta_q \in (2^{-J}q, 2^{-J}(q+1)). \tag{4.4}$$

Since $|t-2^{-J}q| \leq 2^{-J}$:

$$|E_q(t)| \simeq 2^{-2J-1}|\omega^{''}(\zeta_q)|, \tag{4.5}$$

Thus

$$\|E(t)\| = \underset{t\in D_q}{sup}|E_q| \simeq 2^{-2J-1}\|\omega^{''}(\zeta_q)\|. \tag{4.6}$$

□

Lemma 4.2. *Suppose $\omega_J(t) = \mathcal{L}^\top\phi_J(t)$ is the best approximation for the function $\omega(t) \in C^2[0,1]$ by the linear cardinal B-spline functions. In this case, we have*

$$\|E(t)\| = \|\omega(t) - \omega_J(t)\| = \mathbb{O}\left(\frac{1}{2^{2J}}\right). \tag{4.7}$$

Proof. By applying Theorem 4.1, the result is easily achieved.

□

Theorem 4.3. *(Convergence of the operational matrix method)*
Consider the FDDE (1.1) with initial conditions (1.2) and history function (1.3). Assume that:

1. *The exact solution $w(t) \in C^2[0,1]$;*
2. *The function f in (1.1) is Lipschitz continuous with respect to all its arguments, with Lipschitz constant L_f;*
3. *The operational matrices $\mathbf{D}^{(\alpha)}$, $\mathbf{D}^{(\beta)}$, $\mathbf{D}^{(\gamma)}$, and $\mathbf{D}_{delay}$ are constructed as in Theorems 3.4 and 3.9.*

Then the approximate solution $w_J(t)$ obtained by the algorithm in Section 3 converges to the exact solution $w(t)$ as $J \to \infty$, and the convergence rate is $O(2^{-2J})$.

Proof. Let $w_J(t) = \boldsymbol{\mathcal{L}}^T\phi_J(t)$ be the approximate solution obtained by solving the collocation system. Denote by $\widetilde{w}_J(t)$ the best approximation to $w(t)$ in the B-spline space, i.e., the projection of $w(t)$ onto span$\{\varphi_{J,q}\}$. By Theorem 4.1, we have

$$\|w - \widetilde{w}_J\|_{L^2} \leq C_1 \cdot 2^{-2J}\|w''\|_{L^2}.$$

The collocation method enforces the FDDE at the points μ_m. Since f is Lipschitz, the discrepancy between w_J and $\widetilde{w}_J$ can be controlled by the residual of the projection. Standard results on collocation methods for functional equations yield

$$\|w_J - \widetilde{w}_J\|_{L^2} \leq C_2 \cdot \|w - \widetilde{w}_J\|_{L^2},$$

where C_2 depends on L_f and the stability constants of the operational matrices. Combining the two estimates via the triangle inequality,

$$\|w - w_J\|_{L^2} \leq \|w - \widetilde{w}_J\|_{L^2} + \|\widetilde{w}_J - w_J\|_{L^2} \leq (1 + C_2)C_1 \cdot 2^{-2J}\|w''\|_{L^2},$$

which establishes the $O(2^{-2J})$ convergence rate.

□

Remark 4.4. *It is important to emphasize that the convergence rate established in Theorem 4.3 is algebraic, not exponential (spectral). Since the number of basis functions is $N = 2^J + 1$, the bound $O(2^{-2J})$ is equivalent to $O(N^{-2})$, which is the optimal algebraic convergence rate for piecewise linear finite-element approximations in the L^2-norm. Therefore, while the method does not exhibit spectral (exponential) convergence in the classical sense, it combined with the sparsity of the operational matrices, ensures high accuracy and computational efficiency.*

Remark 4.5. *The sparsity of the operational matrices is a key factor in numerical stability. Dense matrices (such as those arising in global spectral methods) often suffer from severe ill-conditioning as the number of basis functions increases. In contrast, the matrices used in our method are sparse:*

- $\mathbf{M}_{int}$ *and* $\mathbf{M}_\nu$ *are upper triangular with a fixed bandwidth.*
- $\mathbf{D}_{delay}$ *has at most two nonzeros per column.*
- $\mathbf{P}$ *is tridiagonal.*

This sparsity limits the accumulation of round-off errors during matrix–vector multiplications and linear system solves. Furthermore, the diagonal dominance of $\mathbf{P}$ *and the boundedness of the operational matrices ensure that the overall method is numerically stable for the refinement levels considered in our experiments ($J \leq 10$).*

5. Numerical examples

In this section, we present the numerical results obtained from the implementation of the proposed method based on linear B-spline basis functions. Three diverse examples, including linear, nonlinear, and multi-term FDDEs, are solved to evaluate the efficiency of the method. The numerical implementation is carried out in the MATLAB environment, and the accuracy of the method is examined using error norms such as the L^2 norm and the maximum error. The results are provided in the form of error tables and convergence graphs, and in some cases, comparisons are made with existing methods. These analyses demonstrate the high efficiency and accuracy of the proposed method in solving this class of problems.

Example 5.1. *Consider the following fractional delay differential equation with constant delay [2, 25]:*

$$^{C}D^{\alpha}w(t) + w(t) - w(t-1) = 2t + \frac{\Gamma(3)}{\Gamma(\frac{5}{2})}t^{3/2} - 1, \quad t \in [0,1], \tag{5.1}$$

with the history function

$$w(t) = t^2, \quad -1 \leq t < 0, \tag{5.2}$$

and initial condition

$$w(0) = 0. \tag{5.3}$$

The exact solution for $\alpha = \frac{1}{2}$ is $w(t) = t^2$.

TABLE 2. Detailed numerical results for Example 5.1 with resolution level $J = 3$ (9 basis functions).

t	$w(t)$ (exact)	$w_J(t)$ (approximate)	$\lvert w(t) - w_J(t)\rvert$
0.0	0.00000	0.00000	0.00000×10^{0}
0.1	0.01000	0.01000	3.02000×10^{-16}
0.2	0.04000	0.04000	2.29000×10^{-16}
0.3	0.09000	0.09000	1.25000×10^{-16}
0.4	0.16000	0.16000	8.33000×10^{-17}
0.5	0.25000	0.25000	5.55000×10^{-17}
0.6	0.36000	0.36000	1.11000×10^{-16}
0.7	0.49000	0.49000	1.11000×10^{-16}
0.8	0.64000	0.64000	3.33000×10^{-16}
0.9	0.81000	0.81000	2.22000×10^{-16}
1.0	1.00000	1.00000	4.44000×10^{-16}

TABLE 3. Comparison of maximum absolute errors for Example 5.1.

Method	Parameters	Maximum absolute error
Present method (B-spline linear)	$J = 3$ $(N = 9)$	3.40×10^{-16}
Haar wavelet collocation [2]	$N = 2048$	8.41×10^{-7}
Fractional backward difference [25]	$N = 640$	9.81×10^{-4}

Figure 1 presents a comprehensive visualization of the numerical results obtained using the proposed method. The figure consists of two subplots that collectively demonstrate both the accuracy and error characteristics of the approximation. The combination of these two plots provides a complete picture of the method's performance: the left plot demonstrates qualitative agreement with the exact solution, while the right plot quantifies the numerical error, confirming that the approximation achieves high accuracy throughout the computational domain.

Table 2 confirms the high accuracy of the method, with maximum absolute errors at machine precision levels. Table 3 compares the maximum absolute error of our method with other methods from the literature. The results demonstrate that our linear cardinal B-spline method achieves extremely high accuracy ($\sim 10^{-16}$) with only 9 basis functions. This is significantly better than the Haar wavelet method (which requires 2048 basis functions) and the fractional backward difference method (640 points).

Figure 2 presents the numerical results for different fractional orders α. Subplot (a): Comparison between the approximate solutions obtained using the proposed method and the exact solution $\omega(t) = t^2$ for different fractional orders α. The exact solution is shown as a solid black line, while approximate solutions are displayed with different line styles corresponding to each α value. Subplot (b): Absolute errors plotted on a logarithmic scale against t for different fractional orders. This subplot demonstrates how the error varies across the domain and with different α values. Subplot (c): Maximum absolute error as a function of fractional order α. This plot shows how the maximum error evolves with changing fractional order, with numerical values annotated at each data point. Subplot (d): Mean absolute error as a function of fractional order α. This provides insight into the average performance of the method across the domain for different fractional orders. Subplot (e): Relative errors plotted on a logarithmic scale against t for different fractional orders α. Subplot (f): Convergence behavior showing both maximum and mean errors versus fractional order α on a log-log scale, illustrating the error trends as α varies.

Example 5.2. *Consider the following nonlinear fractional delay differential equation [10]:*

$$ {}^{C}D^{\alpha}w(t) = 1 - 2\left[w\left(\frac{1}{2}t\right)\right]^2, \quad t \in [0, 1], \tag{5.4}$$

where $0 < \alpha \le 1$ *and initial condition*

$$ w(0) = 0, \tag{5.5}$$

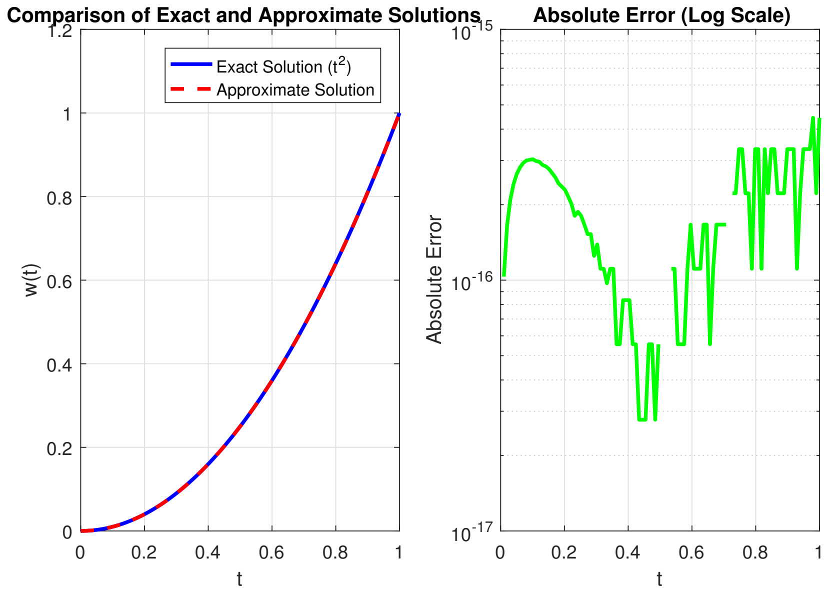

FIGURE 1. Approximate solutions, exact solutions and Absolute Error (Log Scale) of Example 5.1 for $J = 3$.

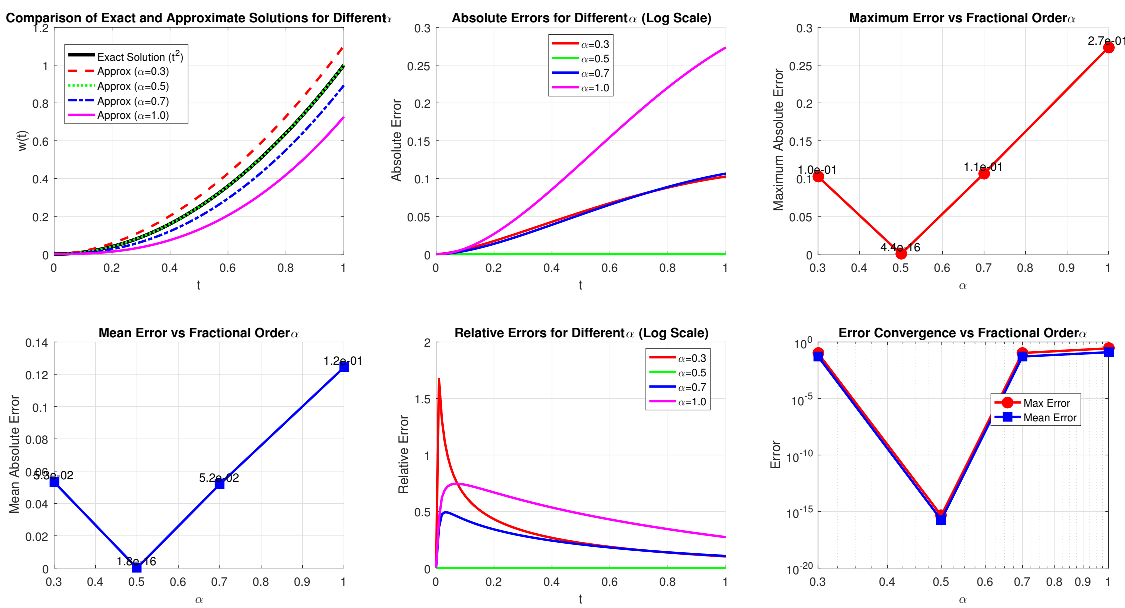

FIGURE 2. The numerical results for different fractional orders $\alpha \in \{0.3, 0.5, 0.7, 1.0\}$ of Example 5.1 for $J = 3$.

TABLE 4. Detailed numerical results for Example 5.2 with resolution level $J = 4$.

t	$w(t)$ (exact)	$w_J(t)$ (approximate)	$\lvert w(t) - w_J(t)\rvert$
0.0	0.00000000	0.00000000	0.00
0.1	0.09983342	0.09983342	87.20×10^{-13}
0.2	0.19866933	0.19866933	82.70×10^{-13}
0.3	0.29552021	0.29552021	78.60×10^{-13}
0.4	0.38941834	0.38941834	72.70×10^{-13}
0.5	0.47942554	0.47942554	65.40×10^{-13}
0.6	0.56464247	0.56464247	56.60×10^{-13}
0.7	0.64421769	0.64421769	46.50×10^{-13}
0.8	0.71735609	0.71735609	35.60×10^{-13}
0.9	0.78332691	0.78332691	21.50×10^{-13}
1.0	0.84147098	0.84147098	92.10×10^{-13}

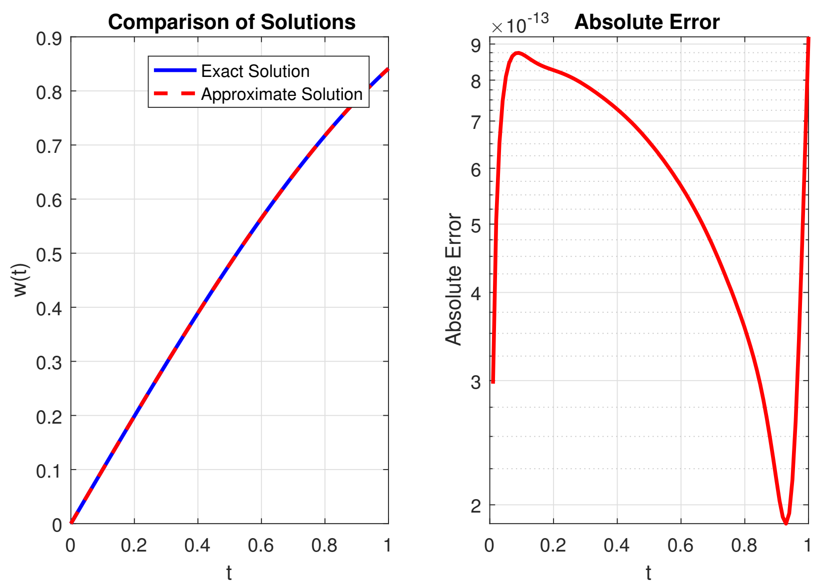

FIGURE 3. Approximate solutions, exact solutions and absolute error of Example 5.2 for $J = 4$.

where the exact solution for $\alpha = 1$ *is* $w(t) = \sin(t)$.

For $J = 4$, a comprehensive comparison between the approximate solution and the exact solution of the problem is performed, which is shown in Figure 3. These figure help to better understand the efficiency of the method. In addition, Figure 4 shows the approximate solution for different values of $\alpha \in 0.7, 0.8, 0.9, 1.0, 1.1, 1.2$.

In Table 4 includes the absolute error for different values of t with for $J = 4$ and $\alpha = 1$. The comparison presented in Table 5, shows the fundamental difference between the proposed method and some existing methods such as the semi analytical approach in Hilbert function space [10].

Comparisons have been made within the specific conditions of each method and by considering their specific parameters. By analyzing the data in this table, one can see a significant difference in the accuracy and efficiency of the method compared to other methods.

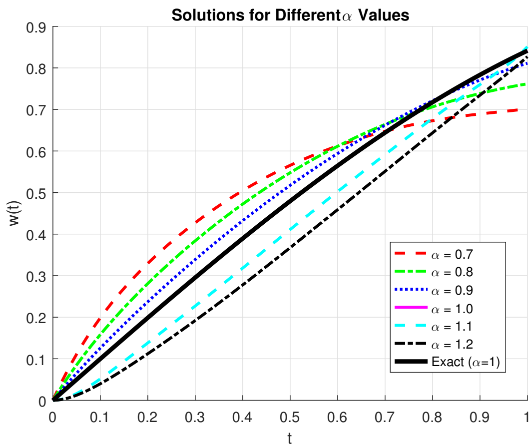

FIGURE 4. The numerical results for different fractional orders $\alpha \in \{0.7, 0.8, 0.9, 1.0, 1.1, 1.2\}$ of Example 5.2 for $J = 4$.

TABLE 5. Comparison of maximum absolute errors for Example 5.2.

Method	Parameters	Maximum absolute error
Present method (B-spline linear)	$J = 4$	9.2148×10^{-13}
Semi-analytical approach in Hilbert function space [10]	$N = 10$	5.91×10^{-4}

Example 5.3. *Consider a fractional-order neutral functional differential equation with proportional delay [11]:*

$$ {}^{C}D^{5/2}\omega(t) = \omega(t) + {}^{C}D^{1/2}\omega\left(\frac{t}{3}\right) + {}^{C}D^{3/2}\omega\left(\frac{t}{4}\right) + {}^{C}D^{5/2}\omega\left(\frac{t}{5}\right) + f(t), \tag{5.6} $$

with initial conditions:

$$ \omega(0) = \omega'(0) = \omega''(0) = 0, \tag{5.7} $$

where $t \in [0, R]$ *and*

$$ f(t) = \frac{32}{\sqrt{\pi}} t^{\frac{3}{2}} - t^4 - \frac{128}{945\sqrt{3\pi}} t^{\frac{7}{2}} - \frac{2}{5\sqrt{\pi}} t^{\frac{5}{2}} - \frac{32}{5\sqrt{5\pi}} t^{\frac{3}{2}}. \tag{5.8} $$

The exact solution is given by:

$$ \omega(t) = t^4. \tag{5.9} $$

In Table 6 includes the absolute error for different values of t with for $J = 6$ and $[0, 1]$. By analyzing the data in this table, we can see the accuracy and efficiency of the method. Figures 5 and 6 provide a comprehensive visualization of the numerical results obtained using the proposed method in different ranges. These figures include two sub-plots that generally show both the accuracy and the approximation error characteristics. The data in this example also shows the effectiveness of the method in larger ranges. Table 7 compares the absolute errors obtained by the present method and the shifted Gegenbauer-Gauss collocation method (SGGC) [11]. The results indicate that the proposed operational matrix method provides highly accurate solutions even with a small number of basis elements ($J = 6$) and for wide intervals. The method significantly outperforms the SGGC in terms of accuracy.

Example 5.4. *Consider the following fractional delay differential equations [20]:*

$$ {}^{C}D^{\alpha}w(t) = w(t) - 2w\left(t - \frac{1}{4}\right) + 2f\left(t - \frac{1}{4}\right), \tag{5.10} $$

Table 6. Detailed numerical results for Example 5.3 with resolution level $J = 6$ in $[0, 1]$.

t	$w(t)$ (exact)	$w_J(t)$ (approximate)	$\lvert w(t) - w_J(t)\rvert$
0.0	0.000000	-0.000000	3.64×10^{-15}
0.1	0.000100	0.000100	3.64×10^{-15}
0.2	0.001600	0.001600	3.62×10^{-15}
0.3	0.008100	0.008100	3.61×10^{-15}
0.4	0.025600	0.025600	3.61×10^{-15}
0.5	0.062500	0.062500	3.54×10^{-15}
0.6	0.129600	0.129600	3.61×10^{-15}
0.7	0.240100	0.240100	3.61×10^{-15}
0.8	0.409600	0.409600	3.33×10^{-15}
0.9	0.656100	0.656100	3.33×10^{-15}
1.0	1.000000	1.000000	3.33×10^{-15}

Table 7. Comparison of maximum absolute errors for Example 5.3.

Method	Parameters	Maximum absolute error
Present method (B-spline linear)	$J = 6$	1.289×10^{-15}
shifted Gegenbauer-Gauss collocation (SGGC) [11]	$N = 16$	3.591×10^{-4}

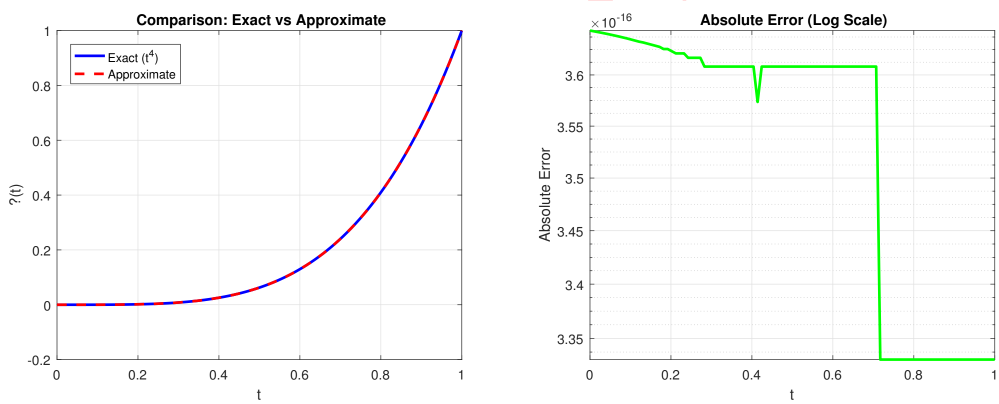

Figure 5. Approximate solutions, exact solutions and absolute error (log scale) of Example 5.3 for $J = 6$ in $[0, 1]$.

with the history function

$$w(t) = 0, -\frac{1}{4} \leq t \leq 0, \tag{5.11}$$

and

$$\begin{cases} f(t) = 0, & -\frac{1}{4} \leq t \leq 0, \\ f(t) = 1, & t > 0. \end{cases} \tag{5.12}$$

The analytic solution of this problem for $\alpha = 1$ is:

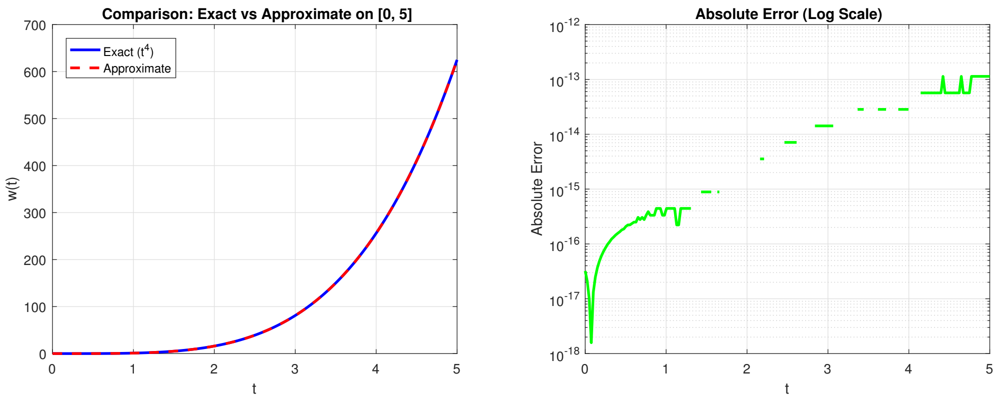

FIGURE 6. Approximate solutions, exact solutions and absolute error (log scale) of Example 5.3 for $J = 6$ in $[0, 5]$.

TABLE 8. Comparison of the numerical solution obtained using our method with the other methods in Example 5.4.

t	Block-pulse-Chebyshev $N = 4, M = 7$	Block-pulse-Legendre $N = 4, M = 7$	Block-pulse-Taylor $N = 4, M = 7$	Our method $J = 6$	Exact solution
0.25	0	0	0	0	0
0.40	0.27858404	0.27858404	0.278584046	0.2785840471	0.2785840471
0.55	0.51352714	0.51352714	0.513527142	0.513527141	0.513527141
0.70	0.65465130	0.65465130	0.654651249	0.654651311	0.654651311
0.85	0.70892964	0.70892964	0.708929644	0.708929635	0.708929636
1.00	0.71174280	0.71174280	0.711741219	0.711742824	0.711742826

$$w(t) = \begin{cases} 0, & 0 \le t < \dfrac{1}{4}, \\ 2 - 2e^{\frac{1}{4}-t}, & \dfrac{1}{4} \le t < \dfrac{1}{2}, \\ -2 - 2e^{\frac{1}{4}-t} + (2+4t)e^{\frac{1}{2}-t}, & \dfrac{1}{2} \le t < \dfrac{3}{4}, \\ 6 - 2e^{\frac{1}{4}-t} + (2+4t)e^{\frac{1}{2}-t} - \left(\dfrac{17}{4} + 2t + 4t^2\right) e^{\frac{3}{4}-t}, & \dfrac{3}{4} \le t \le 1. \end{cases} \tag{5.13}$$

Table 8 displays the numerical solution derived using the Block-pulse-Chebyshev polynomials [20], the Block-pulse-Legendre polynomials [21] and the Block-pulse-Taylor polynomials [22] together with present method in $J = 6$ and the exact solution for $\frac{1}{4} \le t \le 1$.

In Table 8, it can be seen that our method is more accurate than some existing methods for solving equation of Example 5.4.

Example 5.5. *Consider the following system of fractional delay differential equations [13]:*

$$\begin{cases} {}^{C}D^{\alpha}w(t) & = w\left(t - \frac{1}{3}\right) + 2u\left(t - \frac{2}{3}\right) + H_1(t), \quad t \in [0, 1], \\ {}^{C}D^{\alpha}u(t) & = 2tw\left(t - \frac{2}{3}\right) + tu\left(t - \frac{1}{3}\right) + H_2(t), \quad t \in [0, 1], \end{cases} \tag{5.14}$$

TABLE 9. Detailed numerical results for Example 5.5 with resolution level $J = 4$.

t	w_{exact}	w_{approx}	w_{error}	u_{exact}	u_{approx}	u_{error}
0.0	1.000000	1.000000	0.00×10^{0}	0.000000	0.000000	0.00×10^{0}
0.1	1.001000	1.001000	8.88×10^{-8}	0.010000	0.010000	2.20×10^{-8}
0.2	1.008000	1.008000	6.66×10^{-8}	0.040000	0.040000	2.08×10^{-8}
0.3	1.027000	1.027000	8.88×10^{-8}	0.090000	0.090000	2.08×10^{-8}
0.4	1.064000	1.064000	6.66×10^{-8}	0.160000	0.160000	2.22×10^{-8}
0.5	1.125000	1.125000	8.88×10^{-8}	0.250000	0.250000	2.22×10^{-8}
0.6	1.216000	1.216000	8.88×10^{-8}	0.360000	0.360000	2.22×10^{-8}
0.7	1.343000	1.343000	8.88×10^{-8}	0.490000	0.490000	2.78×10^{-8}
0.8	1.512000	1.512000	8.88×10^{-8}	0.640000	0.640000	4.44×10^{-8}
0.9	1.729000	1.729000	8.88×10^{-8}	0.810000	0.810000	5.55×10^{-8}
1.0	2.000000	2.000000	4.44×10^{-8}	1.000000	1.000000	0.00×10^{0}

TABLE 10. Comparison of maximum absolute errors for Example 5.5.

Method	Parameters	Maximum absolute of $w(t)$	Maximum absolute of $u(t)$
Present method (B-spline linear)	$J = 8$	2.9137×10^{-10}	1.3322×10^{-12}
Generalized Laguerre polynomials [13]	$N = 10$	5.91×10^{-11}	3.49×10^{-11}

with initial conditions:

$$u(t) = 0, \quad w(t) = 1, \quad z \in \left[-\frac{2}{3}, 0\right], \tag{5.15}$$

where $H_1(t)$ and $H_2(t)$ are chosen such that the exact solution is:

$$w(t) = t^3 + 1, \qquad u(t) = t^2. \tag{5.16}$$

For $J = 4$ and $J = 8$ a comprehensive comparison between the approximate solutions and the exact solutions of the problem is performed, which are shown in Figures 7 and 8. These figures help to better understand the efficiency of the method and also demonstrate the ability of the method to solve system of fractional delay differential equations. In Table 9 includes the absolute errors for different values of t with for $J = 4$ and $\alpha = 1$. The comparison presented in Table 10, shows the fundamental difference between the proposed method and some existing methods such as the collocation method with generalized Laguerre polynomials basis [13]. Comparisons have been made within the specific conditions of each method and by considering their specific parameters. By analyzing the data in this table, one can see a significant difference in the accuracy and efficiency of the method compared to other methods.

6. Conclusion

In this paper, a novel and efficient numerical method based on modified linear B-spline basis functions has been developed for solving multi-term Caputo fractional delay differential equations (FDDEs). The proposed technique employs operational matrices for fractional integration and delay transformation, converting the original FDDE into an equivalent algebraic system (linear or nonlinear) that can be solved by standard linear solvers or iterative schemes. For linear cardinal B-splines, the condition number of $\mathbf{P}$ is bounded independently of J, ensuring that the algebraic system derived from collocation remains numerically solvable as J increases. This property, together with the sparsity of the operational matrices, guarantees the robustness of the algorithm. A rigorous convergence analysis has been provided, showing that the method converges with order $O(2^{-2J})$, which guarantees high accuracy even with a small number of basis functions. The proposed method achieves the optimal algebraic convergence rate for piecewise linear approximations, as rigorously established in Theorem 4.3 and discussed in Remark 4.4. This provides a balance between high accuracy and computational efficiency, without relying on unproven spectral (exponential) convergence assumptions. Several numerical examples-Including linear, nonlinear, system and multi-term FDDEs-have been solved

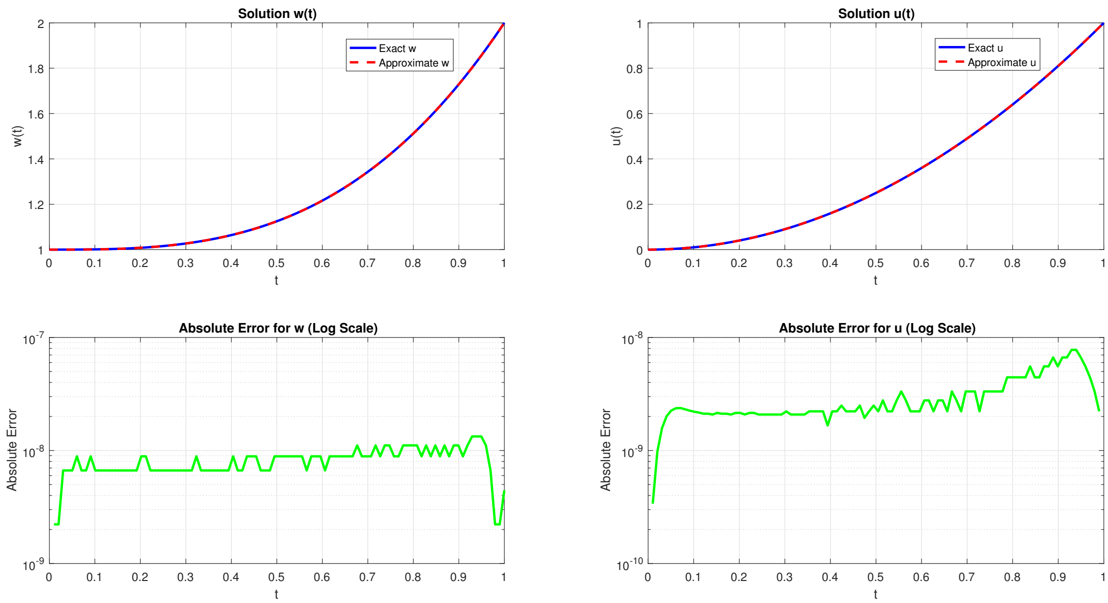

FIGURE 7. Approximate solutions, exact solutions and absolute errors of Example 5.5 for $J = 4$.

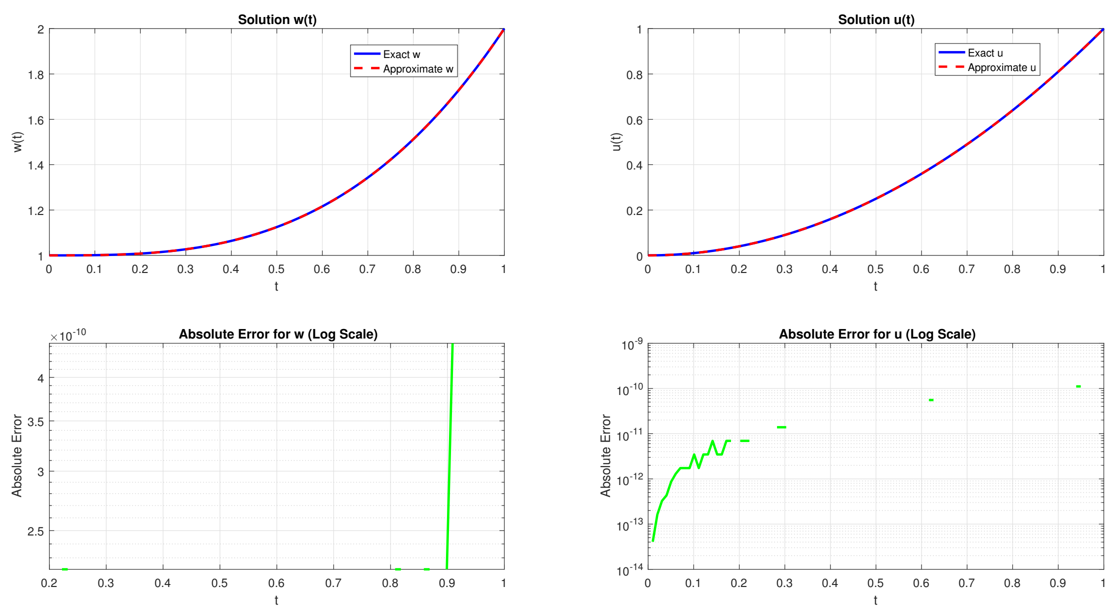

FIGURE 8. Approximate solutions, exact solutions and absolute errors of Example 5.5 for $J = 8$.

to illustrate the efficiency and reliability of the approach.The results demonstrate excellent agreement with exact solutions and, for the considered smooth test problems, attain errors close to machine precision with a relatively small number of basis functions, consistent with the theoretical algebraic rate. Comparisons with existing methods (such as Haar wavelet collocation, fractional backward difference, and semi-analytic Hilbert-space techniques) confirm that the present scheme is both more accurate and computationally economical.

The superior performance can be attributed to:

- The piecewise linear nature of B-spline basis functions provides optimal second-order approximation for solutions in $C^2[0,1]$, which is confirmed by the numerical experiments.
- The operational matrices for B-splines are sparse (e.g., $\mathbf{M}_{\text{int}}$ is upper-triangular, $\mathbf{D}_{\text{delay}}$ has at most two nonzero entries per column), leading to better numerical stability.
- The collocation points $\mu_m = (m+1)/2^J$ are optimally chosen at the midpoints of the support intervals, ensuring a well-conditioned system.

In summary, the combination of linear B-spline bases, analytically constructed operational matrices, and a straightforward collocation framework yields a robust, high-precision solver for a broad class of multi-term Caputo fractional delay differential equations. The method is straightforward to implement, numerically stable, and backed by rigorous error estimates. Future work may extend the approach to variable-delay problems, fractional partial differential equations with delay, and systems with more complex nonlinearities.

Declarations

Ethical Statement: Not Applicable.

Consent for Publication: Not Applicable.

Funding: This research received no funding.

Availability of Data and Materials: Ph.D. thesis dissertations are public documents, but at the moment are not internet-accessible. If one is interested in revising the investigation, information can be requested from the authors.

Conflict of interest :The authors declare no competing interests.

Authors' Contributions: Conceptualization, K. S. and I. M.; Formal analysis, K. S., I. M. and P. M; Investigation, K. S., I. M. and P. M.; Methodology, K. S., I. M. and P. M.; Validation, K. S., I. M.; Visualization, K. S.; Writing-original draft, K. S., I. M. and P. M.; Writing-review-editing, K. S.; I. M.; All authors contributed equally to this paper.

Institutional Review Board Statement: Not Applicable.

Acknowledgment

The authors wish to express their cordial thanks to the anonymous referees and editor of the journal for useful suggestions and comments.

References

[1] A. S. Alshomrani, M. Z. Ullah, and D. Baleanu, *Caputo SIR model for COVID-19 under optimized fractional order*, Advances in Difference Equations, *2021*(1) (2021), 185.

[2] R. Amin, K. Shah, M. Asif, and I. Khan, *A computational algorithm for the numerical solution of fractional order delay differential equations*, Applied Mathematics and Computation, *402* (2021), 125863.

[3] D. Aksim and D. Pavlov, *On the extension of Adams-Bashforth-Moulton methods for numerical integration of delay differential equations and application to the moon's orbit*, Mathematics in Computer Science, *14*(1) (2020), 103-109.

[4] M. Behroozifar and S. A. Yousefi, *Numerical solution of delay differential equations via operational matrices of hybrid of block-pulse functions and Bernstein polynomials*, Computational Methods for Differential Equations, *1*(2) (2013), 78-95.

[5] A. Coronel-Escamilla and J. F. Gomez-Aguilar, *A novel predictor-corrector scheme for solving variable-order fractional delay differential equations involving operators with Mittag-Leffler kernel*, Discrete and Continuous Dynamical Systems-S, *13*(3) (2020), 561-574.

[6] M. Dehghan and M. Abbaszadeh, *A Legendre spectral element method (SEM) based on the modified bases for solving neutral delay distributed-order fractional damped diffusion-wave equation*, Mathematical Methods in the Applied Sciences, *41*(9) (2018), 3476-3494.

[7] M. Dehghan, J. Manafian, and A. Saadatmandi, *Solving nonlinear fractional partial differential equations using the homotopy analysis method*, Numerical Methods for Partial Differential Equations: An International Journal, *26*(2) (2010), 448-479.

[8] A. Ebrahimijahan, M. Abbaszadeh, and M. Dehghan, *Efficient simulation of shallow water dynamics via integrated radial basis function-Proper Orthogonal Decomposition reduced-order modeling*, Engineering Analysis with Boundary Elements, *187* (2026), 106691.

[9] M. R. Eslahchi and M. Dehghan, *Application of Taylor series in obtaining the orthogonal operational matrix*, Computers and Mathematics with Applications, *61*(9) (2011), 2596-2604.

[10] M. Ghasemi, M. Fardi, and R. K. Ghaziani, *Numerical solution of nonlinear delay differential equations of fractional order in reproducing kernel Hilbert space*, Applied Mathematics and Computation, *268* (2015), 815-831.

[11] R. M. Hafez and Y. H. Youssri, *Shifted Gegenbauer-Gauss collocation method for solving fractional neutral functional-differential equations with proportional delays*, Kragujevac Journal of Mathematics, *46*(6) (2022), 981-996.

[12] I. Hashim, M. Sharadga, M. I. Syam, and M. Al-Refai, *A reliable approach for solving delay fractional differential equations*, Fractal and Fractional, *6* (2) (2022), 124.

[13] H. S. Hussien, *Efficient collocation operational matrix method for delay differential equations of fractional order*, Iranian Journal of Science and Technology, *43*(4) (2019), 1841-1850.

[14] A. Ghoreyshi, M. A. Zaky, M. Abbaszadeh, and M. Dehghan, *An efficient numerical scheme for time-fractional partial integro-differential equations with time delay on rectangular and non-rectangular domains*, Computational and Applied Mathematics, *45*(10) (2026), 417.

[15] A. Ghoreyshi, M. Abbaszadeh, and M. Dehghan, *Numerical methods for the time-fractional diffusion equation: A review*, AUT Journal of Mathematics and Computing, *7*(2) (2026), 213-269.

[16] M. Lakestani, M. Dehghan, and S. Irandoust-Pakchin, *The construction of operational matrix of fractional derivatives using B-spline functions*, Communications in nonlinear science and numerical simulation, *17*(3) (2012), 1149-1162.

[17] M. Lakestani, M. Razzaghi, and M. Dehghan, *Solution of nonlinear Fredholm-Hammerstein integral equations by using semiorthogonal spline wavelets*, Mathematical Problems in Engineering, *1* (2005), 113-121.

[18] M. Lakestani, R. Tuntaş, and M. Dehghan, *Constructing efficient bases from B-Spline functions and solving fractional optimal control problems*, Journal of Applied Mathematics and Computing, *72*(3) (2026), 120, https://doi.org/10.1007/s12190-025-02756-x

[19] M. Mahdavi-Parsa, K. Sayevand, H. Jafari, and I. Masti, *Analyzing the Black-Scholes equation with fractional coordinate derivatives using an implicit discrete design*, An International Journal of Optimization and Control: Theories and Applications, *15*(2) (2025), 225–244.

[20] H. R. Marzban and M. Razzaghi, *A hybrid domain analysis for systems with delays in state and control*, Mathematical Problems in Engineering, *7*(4) (2001), 337-353.

[21] H. R. Marzban and M. Razzaghi, *Solution of time-varying delay systems by hybrid functions*, Mathematics and Computers in Simulation, *64*(6) (2004), 597-607.

[22] H. R. Marzban and M. Razzaghi, *Analysis of time-delay systems via hybrid of block-pulse functions and Taylor series*, Journal of Vibration and Control, *11*(12) (2005), 1455-1468.

[23] I. Masti and K. Sayevand, *On collocation-Galerkin method and fractional B-spline functions for a class of stochastic fractional integro-differential equations*, Mathematics and Computers in Simulation, *216* (2024), 263-287.

[24] I. Masti, K. Sayevand, and H. Jafari, *On analyzing two dimensional fractional order brain tumor model based on orthonormal Bernoulli polynomials and Newton's method*, An International Journal of Optimization and Control:

Theories and Applications, *14*(1) (2024), 12-19.

[25] M. L. Morgado, N. J. Ford, and P. M.Lima, *Analysis and numerical methods for fractional differential equations with delay*, Journal of Computational and Applied Mathematics, *252* (2013), 159-168.

[26] S. Sabermahani, Y. Ordokhani, and S. A. Yousefi, *Fibonacci wavelets and their applications for solving two classes of time-varying delay problems*, Optimal Control Applications and Methods, *41*(2) (2020), 395-416.

[27] P. Sunthrayuth, H. M. Dutt, F. Ghani, and M. A. Arefin, *The analysis of fractional-order system delay differential equations using a numerical method*, Complexity, *1* (2022), 3570667.

[28] A. Saadatmandi and M. Dehghan, *A new operational matrix for solving fractional-order differential equations*, Computers and Mathematics with Applications, *59*(3) (2010), 1326-1336.

[29] M. Shakourifar and M. Dehghan, *On the numerical solution of nonlinear systems of Volterra integro-differential equations with delay arguments*, Computing, *82*(4) (2008), 241-260.

[30] S. Shahmorad, M. H. Ostadzad, and D. Baleanu, *A Tau-like numerical method for solving fractional delay integro-differential equations*, Applied Numerical Mathematics, *151* (2020), 322-336.

[31] S. Tyagi, S. Abbas, M. Pinto, and D. Sepulveda, *Approximation of solutions of fractional-order delayed cellular neural network on* $[0, \infty)$, Mediterranean Journal of Mathematics, *14*(1) (2017), 23.

[32] M. Q. Xu and Y. Z. Lin, *Simplified reproducing kernel method for fractional differential equations with delay*, Applied Mathematics Letters, *52* (2016), 156-161.

[33] Z. Zarvan, K. Sayevand, R. M. Ganji, and H. Jafari, *A reliable numerical algorithm mixed with hypergeometric function for analyzing fractional variational problems*, Numerical Algorithms, *98*(4) (2025), 2081-2112.