AIMS Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 11(5): 13777–13804.
DOI: 10.3934/math.2026567
Received: 03 February 2026
Revised: 02 April 2026
Accepted: 08 April 2026
Published: 16 May 2026

Research article

Wavelet−ALA fusion scheme for (2+1)D time−fractional mobile/immobile models

Fengying Zhou* and Jiakun Zhang

School of Mathematical Sciences, Jiangxi Science and Technology Normal University, Nanchang, Jiangxi, 330038, China

* **Correspondence:** Email: zhoufengying11@163.com.

Abstract: A numerical scheme is proposed, merging 3D eighth−kind fractional Chebyshev wavelets with the artificial lemming algorithm (ALA) to address (2+1)D time−fractional mobile/immobile models. Initially, we construct the 3D eighth−kind fractional Chebyshev wavelets and investigate their properties, including approximation capability and convergence analysis. Furthermore, by establishing a multivariate Caputo fractional Taylor formula, error bounds for the presented wavelets approximation are derived. Then, we provide a detailed algorithmic procedure for solving the (2+1)D time−fractional mobile/immobile models considered in this paper, incorporating the ALA. Finally, the performance of the proposed method is examined through selected numerical examples. Comparative studies with existing findings provide quantitative evidence of the method's accuracy and the enhanced computational efficiency resulting from ALA incorporation. Moreover, our method remains applicable to 3D space−fractional differential equations, and we validate this claim through a 3D space−fractional Poisson equation.

Keywords: 3D eighth−kind fractional Chebyshev wavelets; ALA; multivariate Caputo fractional Taylor formula; (2+1)D time−fractional mobile/immobile models; collocation method
Mathematics Subject Classification: 41A10, 65T60

1. Introduction

As a natural extension of integer-order differential equations, fractional differential equations have attracted considerable attention in both theoretical research and practical applications owing to their distinctive nonlocal properties and inherent memory effects. These characteristics not only substantially enrich the theoretical framework of conventional differential equation theory but also offer a more robust mathematical foundation for modeling complex phenomena across various engineering and scientific disciplines [1–3].

The inherent complexity of fractional differential equations coupled with their scientific utility makes numerical approximation not merely advantageous but essential. To date, a wide variety of numerical methods have been developed for solving fractional differential equations, such as the Adomian decomposition method [4], finite difference method [5], homotopy perturbation method [6], spectral method [7], and high precision interpolation method [8]. Recent years have witnessed remarkable advances in polynomial collocation methods for numerical solutions of fractional differential equations, among which Chebyshev polynomials are particularly notable, as shown in [9–11]. Over the past few years, wavelet−based numerical methods have made considerable progress in solving differential equations, including but not limited to Chebyshev wavelets [12, 13], Legendre wavelets [14], Euler wavelets [15], and Haar wavelets [16]. Research has primarily focused on improving wavelet construction techniques, optimizing numerical algorithms, and extending their use to diverse types of differential equations.

In this article, we consider the (2+1)D time−fractional mobile/immobile models as

$$\frac{\partial^\lambda u(x,y,t)}{\partial t^\lambda}+\frac{\partial u(x,y,t)}{\partial t}=\frac{\partial^2 u(x,y,t)}{\partial x^2}+\frac{\partial^2 u(x,y,t)}{\partial y^2}+g(u)+f(x,y,t) \tag{1.1}$$

for $(x,y,t)\in(0,1)^2\times(0,1]$, subject to the initial boundary conditions

$$u(x,y,0)=L(x,y),\ \ u(0,y,t)=G_1(y,t),\ \ u(1,y,t)=G_2(y,t),\ \ u(x,0,t)=H_1(x,t),\ \ u(x,1,t)=H_2(x,t),$$

where $\lambda\in(0,1)$, $g(u)$ is the nonlinear term, $f(x,y,t)$ and $L(x,y)$ are known functions, and $G_i(y,t)$ and $H_i(x,t)$ possess second−order continuous differentiability over the domain $[0,1]^2$. The fractional derivative is interpreted in accordance with the Caputo definition.

The time-fractional mobile/immobile model is commonly employed to describe anomalous diffusion phenomena in heterogeneous media, capturing key transport behaviors observed in real−world applications such as contaminant transport in porous media, solute transport in fractured rocks, and particle diffusion in biological tissues. This model accounts for the retention and delayed transport effects induced by heterogeneous media, which are essential for accurate prediction and control in practical settings. The (2+1)−dimensional extension of these equations generalizes the classical one−dimensional formulation by incorporating two spatial dimensions (x,y) alongside the temporal dimension (t). This enhanced formulation offers a more physically comprehensive description of transport processes in heterogeneous media, where the coupling between mobile and immobile phases coexists with anisotropic spatial diffusion and memory−dependent temporal evolution. The inclusion of an additional spatial dimension enables the modeling of more realistic scenarios involving directionally varying transport properties and complex boundary conditions that significantly influence system dynamics. Various numerical approaches have been employed to solve these equations, including the finite difference method [17], the double−parameter shifted convolution quadrature formula [18], and the alternating−direction implicit (ADI) compact difference scheme [19], among others.

The artificial lemming algorithm (ALA) is a nature−inspired metaheuristic optimization algorithm that mimics the collective behavior of lemmings, small rodents known for their periodic mass migrations, which sometimes include jumping off cliffs in large groups [20]. Although the real−life "lemming suicide" myth is a misconception, the ALA metaphorically utilizes this idea to model exploration and exploitation in optimization problems. Due to its stochastic exploration−exploitation balance and resilience to local optima, the ALA proves particularly effective for the high−dimensional, multimodal optimization problem investigated here.

Guided by previous findings, our research focuses on designing an efficient numerical method based on 3D eighth−kind fractional Chebyshev wavelets and the ALA for solving the time-fractional mobile/immobile systems. The synergy between the 3D eighth−kind fractional Chebyshev wavelet method and the ALA in this study is primarily reflected in two aspects:

(i) The wavelet method provides an approximation framework with parameters: The 3D eighth−kind fractional Chebyshev wavelets we constructed contain three preset fractional parameters (α, β, δ), which control the scaling characteristics of the wavelets and directly affect the approximation accuracy.

(ii) The ALA is responsible for automatically searching for optimal fractional parameters: The ALA possesses strong spatial search capabilities. Through iterative optimization, it searches within the wavelet parameter space for the parameter combination that minimizes the solution error.

The paper proceeds as follows. Section 2 outlines the basic mathematical framework of fractional calculus. In Section 3, we construct the 3D eighth−kind fractional Chebyshev wavelets and rigorously analyze their mathematical properties, with particular emphasis on convergence analysis and error estimation for three−dimensional case. Section 4 details our proposed algorithm for solving Eq (1.1). The effectiveness of our method is then demonstrated in Section 5 through several numerical experiments and comparative analysis with existing approaches. Section 6 concludes the paper with a summary of key findings and potential directions for future research.

2. Essential knowledge

In this section, we review the fundamental principles of fractional calculus. For clarity, we assume all functions satisfy the necessary conditions required for the subsequent definitions to be properly established. For further details on fractional calculus, see [21].

Definition 2.1. *The Riemann−Liouville fractional integral of a function f is defined as*

$$\mathcal{I}^{\lambda} f(t) = \begin{cases} \frac{1}{\Gamma(\lambda)} \int_0^t (t-s)^{\lambda-1} f(s) ds, & \lambda > 0, \\ f(t), & \lambda = 0, \end{cases}$$

where $\Gamma(\cdot)$ is the gamma function.

Definition 2.2. *The Caputo fractional derivative of a function f is defined as*

$$\mathcal{D}^{\lambda} f(t) = \begin{cases} \frac{1}{\Gamma(n-\lambda)} \int_0^t (t-s)^{n-\lambda-1} f^{(n)}(s) ds, & n-1 < \lambda < n, \\ \frac{d^n f(t)}{dt^n}, & \lambda = n. \end{cases}$$

Proposition 2.1. *The Riemann−Liouville fractional integral and Caputo fractional derivative hold the following properties:*

(i) $\mathcal{D}^{\lambda} t^n = \begin{cases} 0, & \lambda > n, \\ \frac{\Gamma(n+1)}{\Gamma(n-\lambda+1)} t^{n-\lambda}, & \lambda \le n; \end{cases}$

(ii) $\mathcal{I}^{\lambda}(\mathcal{D}^{\lambda}f(t)) = f(t) - \sum_{l=0}^{n-1} f^{(l)}(0^{+})\frac{t^{l}}{l!},\ n \geq \lambda > n-1,\ t > 0;$
(iii) $\mathcal{D}^{\lambda}\mathcal{I}^{\lambda}f(t) = f(t);$
(iv) $\mathcal{D}^{\lambda_1}\mathcal{I}^{\lambda_2}f(t) = \mathcal{I}^{\lambda_2-\lambda_1}f(t),\ \lambda_2 > \lambda_1.$

3. 3D eighth−kind fractional Chebyshev wavelets

This section presents a comprehensive study of 3D eighth−kind fractional Chebyshev wavelets, systematically investigating their fundamental properties, approximation technique, convergence analysis, and error estimation.

3.1. Construction of 3D eighth−kind fractional Chebyshev wavelets

Given that the 3D eighth−kind fractional Chebyshev wavelets are constructed from the eighth−kind Chebyshev polynomials, we first introduce the eighth−kind Chebyshev polynomials [11].

The eighth−kind Chebyshev polynomials of degree m on the interval $[-1, 1]$ take the form as

$$E_m(x) = \sum_{p=0}^{m}\sum_{i=0}^{p} \frac{a_{p,m}}{2^p} C_p^i x^i, \tag{3.1}$$

where $C_p^i = p!/(i!(p-i)!)$, and

$$a_{p,m} = \begin{cases} \frac{3}{(2p+1)!} \sum\limits_{j=\lfloor \frac{p+1}{2} \rfloor}^{\frac{m}{2}} \frac{(-1)^{\frac{m}{2}} 2^{2p-2}(\frac{m}{2}-j+1)(\frac{m}{2}+j+2)\Gamma(\frac{m+3}{2})(-1)^{j+p}(2j+p+1)!}{\Gamma(\frac{m+1}{2}+2)(2j-p)!} & if\ m\ is\ even; \\ \\ \frac{3}{(2p+1)!} \sum\limits_{j=\lfloor \frac{p}{2} \rfloor}^{\frac{m-1}{2}} \frac{(-1)^{\frac{m+1}{2}}(j+1)^{2p-3}(m-2j+1)(\frac{m+5}{2}+j)\Gamma(\frac{m}{2}+1)(-1)^{j+p}(2j+p+2)!}{\Gamma(\frac{m}{2}+3)(2j-p+1)!} & if\ m\ is\ odd. \end{cases}$$

These polynomials are orthogonal over the interval $[-1, 1]$ with respect to the weight function $\omega(x) = x^4\sqrt{1-x^2}$ as

$$\int_{-1}^{1} E_m(x)E_{m'}(x)\omega(x)dx = h_m\delta_{m,m'}, \tag{3.2}$$

where $\delta_{m,m'}$ is the Kronecker delta function, and

$$h_m = \begin{cases} \frac{9\pi}{128}\frac{(m+2)(m+4)}{(m+3)^2} & if\ m\ is\ even; \\ \\ \frac{9\pi}{128}\frac{(m+1)(m+5)}{(m+2)(m+4)} & if\ m\ is\ odd. \end{cases}$$

Now, we present the definition of the one−dimensional eighth-kind fractional Chebyshev wavelets.

Definition 3.1. *[22] The eighth−kind fractional Chebyshev wavelets defined over the interval* $[0, 1]$ *are constructed as*

$$\varphi_{n,m}^{\alpha}(x) = \begin{cases} \frac{2^{\frac{k}{2}}}{\sqrt{h_m}} E_m(2^k x^{\alpha} - 2n + 1),\ (\frac{n-1}{2^{k-1}})^{\frac{1}{\alpha}} \leq x \leq (\frac{n}{2^{k-1}})^{\frac{1}{\alpha}}, \\ \\ 0,\ \text{O.W.}, \end{cases}$$

where k is a tunable positive integer, $n = 1, 2, \cdots, 2^{k-1}$, $m = 0, 1, 2, \cdots$, and $\alpha > 0$. In particular, when $\alpha = 1$, it reduces to the so−called integer−order eighth-kind Chebyshev wavelets.

Next, we define the 3D eighth-kind fractional Chebyshev wavelets via the cross product operation.

Definition 3.2. *The 3D eighth−kind fractional Chebyshev wavelets over the domain $[0,1]^3$ are constructed as*

$$\psi^{\alpha,\beta,\delta}_{n,i,q,m,j,s}(x,y,t) = \varphi^{\alpha}_{n,m}(x)\varphi^{\beta}_{i,j}(y)\varphi^{\delta}_{q,s}(t),$$

where $n, i, q = 1, 2, \cdots, 2^{k-1}$, $m, j, s = 0, 1, 2, \cdots$, $\alpha > 0, \beta > 0$, and $\delta > 0$.

Definition 3.3. *For $f, g \in L^2([0,1]^3)$, the inner product and norm associated with the weight function $\omega(x) = x^4\sqrt{1-x^2}$ are defined as follows:*

$$\langle f, g\rangle_\omega = \sum_{n=1}^{2^{k-1}}\sum_{i=1}^{2^{k-1}}\sum_{q=1}^{2^{k-1}} \int_{D^{\alpha,\beta,\delta}_{n,i,q}} f(x,y,t)g(x,y,t)\omega^{\alpha,\beta,\delta}_{n,i,q}(x,y,t)dxdydt,$$

$$\|f\|_\omega = \Big(\sum_{n=1}^{2^{k-1}}\sum_{i=1}^{2^{k-1}}\sum_{q=1}^{2^{k-1}} \int_{D^{\alpha,\beta,\delta}_{n,i,q}} |f(x,y,t)|^2\omega^{\alpha,\beta,\delta}_{n,i,q}(x,y,t)dxdydt\Big)^{\frac{1}{2}},$$

with k being a configurable positive integer, and

$$\begin{aligned} D^{\alpha,\beta,\delta}_{n,i,q} = \{&(\frac{n-1}{2^{k-1}})^{\frac{1}{\alpha}} \le x \le (\frac{n}{2^{k-1}})^{\frac{1}{\alpha}}, (\frac{i-1}{2^{k-1}})^{\frac{1}{\beta}} \le y \le (\frac{i}{2^{k-1}})^{\frac{1}{\beta}}, (\frac{q-1}{2^{k-1}})^{\frac{1}{\delta}} \le t \le (\frac{q}{2^{k-1}})^{\frac{1}{\delta}} : \\ &n, i, q = 1, 2, \cdots, 2^{k-1}\}, \end{aligned}$$

and $\omega^{\alpha,\beta,\delta}_{n,i,q}(x,y,t) = \alpha\beta\delta x^{\alpha-1}y^{\beta-1}t^{\delta-1}\omega(2^kx^\alpha - 2n + 1)\omega(2^ky^\beta - 2i + 1)\omega(2^kt^\delta - 2q + 1)$.

According to the definition of 3D eighth−kind fractional Chebyshev wavelets and Proposition 3.1 in [22], we have the following properties.

Proposition 3.1. *The 3D eighth−kind fractional Chebyshev wavelets satisfy, for $n \ne n'$ or $i \ne i'$ or $q \ne q'$,*

$$\left\langle \psi^{\alpha,\beta,\delta}_{n,i,q,m,j,s}, \psi^{\alpha,\beta,\delta}_{n',i',q',m',j',s'} \right\rangle_\omega = 0,$$

and

$$\left\langle \psi^{\alpha,\beta,\delta}_{n,i,q,m,j,s}, \psi^{\alpha,\beta,\delta}_{n,i,q,m',j',s'} \right\rangle_\omega = \delta_{m,m'}\delta_{j,j'}\delta_{s,s'}.$$

Proposition 3.2. *For $0 \le m, j, s \le M - 1$, $\alpha > 0, \beta > 0$, and $\delta > 0$, it has*

$$\xi^{m\alpha}\eta^{j\beta}\zeta^{s\delta} = \sum_{n=1}^{2^{k-1}}\sum_{i=1}^{2^{k-1}}\sum_{q=1}^{2^{k-1}}\sum_{m'=0}^{M-1}\sum_{j'=0}^{M-1}\sum_{s'=0}^{M-1} A^{\alpha,\beta,\delta}_{n,i,q,m',j',s'}\varphi^{\alpha}_{n,m'}(\xi)\varphi^{\beta}_{i,j'}(\eta)\varphi^{\delta}_{q,s'}(\zeta). \tag{3.3}$$

Proof. Assuming the validity of Eq (3.3), the remaining task reduces to determining the coefficients within this equation. Fixing values of n_0, i_0, q_0, m_0, j_0, and s_0, by Proposition 3.1, one can obtain

$$\begin{aligned} &A^{\alpha,\beta,\delta}_{n_0,i_0,q_0,m_0,j_0,s_0} \\ &= \int_{D^{\alpha,\beta,\delta}_{n_0,i_0,q_0}} \xi^{m\alpha}\eta^{j\beta}\zeta^{s\delta}\varphi^{\alpha}_{n_0,m_0}(\xi)\varphi^{\beta}_{i_0,j_0}(\eta)\varphi^{\delta}_{q_0,s_0}(\zeta)\omega^{\alpha,\beta,\delta}_{n_0,i_0,q_0}(\xi,\eta,\zeta)\,d\xi d\eta d\zeta \end{aligned}$$

$$= \int_{(\frac{n_0-1}{2^{k-1}})^{\frac{1}{\alpha}}}^{(\frac{n_0}{2^{k-1}})^{\frac{1}{\alpha}}} \xi^{m\alpha} \frac{2^{\frac{k}{2}}}{\sqrt{h_{m_0}}} E_{m_0}(2^k\xi^\alpha - 2n_0 + 1)(2^k\xi^\alpha - 2n_0 + 1)^4 \sqrt{1-(2^k\xi^\alpha - 2n_0 + 1)^2}\, \alpha\xi^{\alpha-1} d\xi$$
$$\cdot \int_{(\frac{i_0-1}{2^{k-1}})^{\frac{1}{\beta}}}^{(\frac{i_0}{2^{k-1}})^{\frac{1}{\beta}}} \eta^{j\beta} \frac{2^{\frac{k}{2}}}{\sqrt{h_{j_0}}} E_{j_0}(2^k\eta^\beta - 2i_0 + 1)(2^k\eta^\beta - 2i_0 + 1)^4 \sqrt{1-(2^k\eta^\beta - 2i_0 + 1)^2}\, \beta\eta^{\beta-1} d\eta$$
$$\cdot \int_{(\frac{q_0-1}{2^{k-1}})^{\frac{1}{\delta}}}^{(\frac{q_0}{2^{k-1}})^{\frac{1}{\delta}}} \zeta^{s\delta} \frac{2^{\frac{k}{2}}}{\sqrt{h_{s_0}}} E_{s_0}(2^k\zeta^\delta - 2q_0 + 1)(2^k\zeta^\delta - 2q_0 + 1)^4 \sqrt{1-(2^k\zeta^\delta - 2q_0 + 1)^2}\, \delta\zeta^{\delta-1} d\zeta.$$

Letting $2^k\xi^\alpha - 2n_0 + 1 = \tau$ and incorporating Eq (3.1) yields

$$\int_{(\frac{n_0-1}{2^{k-1}})^{\frac{1}{\alpha}}}^{(\frac{n_0}{2^{k-1}})^{\frac{1}{\alpha}}} \xi^{m\alpha} \frac{2^{\frac{k}{2}}}{\sqrt{h_{m_0}}} E_{m_0}(2^k\xi^\alpha - 2n_0 + 1)(2^k\xi^\alpha - 2n_0 + 1)^4 \sqrt{1-(2^k\xi^\alpha - 2n_0 + 1)^2}\, \alpha\xi^{\alpha-1} d\xi$$
$$= \int_{-1}^{1} \left(\frac{\tau + 2n_0 - 1}{2^k}\right)^m \frac{2^{-\frac{k}{2}}}{\sqrt{h_{m_0}}} E_{m_0}(\tau)\tau^4 \sqrt{1-\tau^2}\, d\tau$$
$$= \int_{-1}^{1} \frac{1}{2^{mk}} \sum_{r=0}^{m} C_m^r \tau^r (2n_0 - 1)^{m-r} \frac{2^{-\frac{k}{2}}}{\sqrt{h_{m_0}}} \sum_{p=0}^{m_0}\sum_{l=0}^{p} \frac{a_{p,m_0}}{2^p} C_p^l \tau^l \tau^4 \sqrt{1-\tau^2}\, d\tau$$
$$= \sum_{r=0}^{m}\sum_{p=0}^{m_0}\sum_{l=0}^{p} \frac{2^{-\frac{k}{2}}}{2^{mk}\sqrt{h_{m_0}}} C_m^r (2n_0 - 1)^{m-r} \frac{a_{p,m_0}}{2^p} C_p^l \int_{-1}^{1} \tau^{r+l+4} \sqrt{1-\tau^2}\, d\tau$$
$$= \sum_{r=0}^{m}\sum_{p=0}^{m_0}\sum_{l=0}^{p} Q_{r,p,l}^{m,m_0} \int_{-1}^{1} \tau^{r+l+4} \sqrt{1-\tau^2}\, d\tau,$$

where $Q_{r,p,l}^{m,m_0} = 2^{-k/2}/(2^{mk}\sqrt{h_{m_0}})C_m^r C_p^l (2n_0 - 1)^{m-r} a_{p,m_0}/2^p$. Observe that, for $m, n \in \mathbb{N}$,

$$J(m,n) = \int_0^\pi \cos^m\theta \sin^n\theta\, d\theta = \begin{cases} 0 & \text{if } m \text{ is odd;} \\ 2I(m,n) & \text{if } m \text{ is even,} \end{cases}$$

where

$$I(m,n) = \begin{cases} \frac{\pi}{2}\frac{(m-1)!!(n-1)!!}{(m+n)!!} & \text{if } m \text{ and } n \text{ are both even;} \\ \frac{(m-1)!!(n-1)!!}{(m+n)!!} & \text{O.W.} \end{cases}$$

Therefore,

$$\int_{(\frac{n_0-1}{2^{k-1}})^{\frac{1}{\alpha}}}^{(\frac{n_0}{2^{k-1}})^{\frac{1}{\alpha}}} \xi^{m\alpha} \frac{2^{\frac{k}{2}}}{\sqrt{h_{m_0}}} E_{m_0}(2^k\xi^\alpha - 2n_0 + 1)(2^k\xi^\alpha - 2n_0 + 1)^4 \sqrt{1-(2^k\xi^\alpha - 2n_0 + 1)^2}\, \alpha\xi^{\alpha-1} d\xi$$
$$= \sum_{r=0}^{m}\sum_{p=0}^{m_0}\sum_{l=0}^{p} Q_{r,p,l}^{m,m_0} J(r + l + 4, 2).$$

Similarly,

$$\int_{(\frac{i_0-1}{2^{k-1}})^{\frac{1}{\beta}}}^{(\frac{i_0}{2^{k-1}})^{\frac{1}{\beta}}} \eta^{j\beta} \frac{2^{\frac{k}{2}}}{\sqrt{h_{j_0}}} E_{j_0}(2^k\eta^\beta - 2i_0 + 1)(2^k\eta^\alpha - 2i_0 + 1)^4 \sqrt{1 - (2^k\eta^\beta - 2i_0 + 1)^2}\, \beta\eta^{\beta-1} d\eta$$
$$= \sum_{r'=0}^{j} \sum_{p'=0}^{j_0} \sum_{l'=0}^{p'} R^{j,j_0}_{r',p',l'} J(r' + l' + 4, 2),$$

$$\int_{(\frac{q_0-1}{2^{k-1}})^{\frac{1}{\delta}}}^{(\frac{q_0}{2^{k-1}})^{\frac{1}{\delta}}} \zeta^{s\delta} \frac{2^{\frac{k}{2}}}{\sqrt{h_{s_0}}} E_{s_0}(2^k\zeta^\delta - 2q_0 + 1)(2^k\zeta^\delta - 2q_0 + 1)^4 \sqrt{1 - (2^k\zeta^\delta - 2q_0 + 1)^2}\, \delta\zeta^{\delta-1} d\zeta$$
$$= \sum_{r''=0}^{s} \sum_{p''=0}^{s_0} \sum_{l''=0}^{p''} S^{s,s_0}_{r'',p'',l''} J(r'' + l'' + 4, 2),$$

where

$$R^{j,j_0}_{r',p',l'} = \frac{2^{-\frac{k}{2}}}{2^{jk}\sqrt{h_{j_0}}} C^{r'}_j (2i_0 - 1)^{j-r'} \frac{a_{p',j_0}}{2^{p'}} C^{l'}_{p'},$$

$$S^{s,s_0}_{r'',p'',l''} = \frac{2^{-\frac{k}{2}}}{2^{sk}\sqrt{h_{s_0}}} C^{r''}_s (2q_0 - 1)^{s-r''} \frac{a_{p'',s_0}}{2^{p''}} C^{l''}_{p''}.$$

Therefore,

$$A^{\alpha,\beta,\delta}_{n_0,i_0,q_0,m_0,j_0,s_0} = \sum_{r=0}^{m} \sum_{p=0}^{m_0} \sum_{l=0}^{p} \sum_{r'=0}^{j} \sum_{p'=0}^{j_0} \sum_{l'=0}^{p'} \sum_{r''=0}^{s} \sum_{p''=0}^{s_0} \sum_{l''=0}^{p''} Q^{m,m_0}_{r,p,l} R^{j,j_0}_{r',p',l'} S^{s,s_0}_{r'',p'',l''}$$
$$\cdot J(r + l + 4, 2) J(r' + l' + 4, 2) J(r'' + l'' + 4, 2).$$

□

3.2. Function approximation

Assume that a function $f(x, y, t) \in L^2([0, 1]^3)$ is expanded by the 3D eighth−kind fractional Chebyshev wavelets as

$$f(x, y, t) = \sum_{n=1}^{2^{k-1}} \sum_{i=1}^{2^{k-1}} \sum_{q=1}^{2^{k-1}} \sum_{m=0}^{\infty} \sum_{j=0}^{\infty} \sum_{s=0}^{\infty} d^{m,j,s}_{n,i,q} \varphi^\alpha_{n,m}(x)\varphi^\beta_{i,j}(y)\varphi^\delta_{q,s}(t), \tag{3.4}$$

where $d^{m,j,s}_{n,i,q} = \int_{D^{\alpha,\beta,\delta}_{n,i,q}} f(x, y, t)\varphi^\alpha_{n,m}(x)\varphi^\beta_{i,j}(y)\varphi^\delta_{q,s}(t)\omega^{\alpha,\beta,\delta}_{n,i,q}(x, y, t)dxdydt$. The truncated series is given as

$$f(x, y, t) \approx \sum_{n=1}^{2^{k-1}} \sum_{i=1}^{2^{k-1}} \sum_{q=1}^{2^{k-1}} \sum_{m=0}^{M-1} \sum_{j=0}^{M-1} \sum_{s=0}^{M-1} d^{m,j,s}_{n,i,q} \varphi^\alpha_{n,m}(x)\varphi^\beta_{i,j}(y)\varphi^\delta_{q,s}(t).$$

Write

$$
\mathbf{D}_{q,s}=\begin{pmatrix}
d_{1,1,q}^{0,0,s} & d_{1,1,q}^{0,1,s} & \cdots & d_{1,1,q}^{0,M-1,s} & \cdots & d_{1,2^{k-1},q}^{0,0,s} & d_{1,2^{k-1},q}^{0,1,s} & \cdots & d_{1,2^{k-1},q}^{0,M-1,s}\\
d_{1,1,q}^{1,0,s} & d_{1,1,q}^{1,1,s} & \cdots & d_{1,1,q}^{1,M-1,s} & \cdots & d_{1,2^{k-1},q}^{1,0,s} & d_{1,2^{k-1},q}^{1,1,s} & \cdots & d_{1,2^{k-1},q}^{1,M-1,s}\\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
d_{1,1,q}^{M-1,0,s} & d_{1,1,q}^{M-1,1,s} & \cdots & d_{1,1,q}^{M-1,M-1,s} & \cdots & d_{1,2^{k-1},q}^{M-1,0,s} & d_{1,2^{k-1},q}^{M-1,1,s} & \cdots & d_{1,2^{k-1},q}^{M-1,M-1,s}\\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
d_{2^{k-1},1,q}^{0,0,s} & d_{2^{k-1},1,q}^{0,1,s} & \cdots & d_{2^{k-1},1,q}^{0,M-1,s} & \cdots & d_{2^{k-1},2^{k-1},q}^{0,0,s} & d_{2^{k-1},2^{k-1},q}^{0,1,s} & \cdots & d_{2^{k-1},2^{k-1},q}^{0,M-1,s}\\
d_{2^{k-1},1,q}^{1,0,s} & d_{2^{k-1},1,q}^{1,1,s} & \cdots & d_{2^{k-1},1,q}^{1,M-1,s} & \cdots & d_{2^{k-1},2^{k-1},q}^{1,0,s} & d_{2^{k-1},2^{k-1},q}^{1,1,s} & \cdots & d_{2^{k-1},2^{k-1},q}^{1,M-1,s}\\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots\\
d_{2^{k-1},1,q}^{M-1,0,s} & d_{2^{k-1},1,q}^{M-1,1,s} & \cdots & d_{2^{k-1},1,q}^{M-1,M-1,s} & \cdots & d_{2^{k-1},2^{k-1},q}^{M-1,0,s} & d_{2^{k-1},2^{k-1},q}^{M-1,1,s} & \cdots & d_{2^{k-1},2^{k-1},q}^{M-1,M-1,s}
\end{pmatrix},
$$

$$
\mathbf{D}=\Big(\mathbf{D}_{1,0},\mathbf{D}_{1,1},\cdots,\mathbf{D}_{1,M-1},\cdots,\mathbf{D}_{2^{k-1},0},\mathbf{D}_{2^{k-1},1},\cdots,\mathbf{D}_{2^{k-1},M-1}\Big),
$$

and

$$
\mathbf{\Psi}_{k,M}^{\tau}(\cdot)=(\varphi_{1,0}^{\tau}(\cdot),\varphi_{1,1}^{\tau}(\cdot),\cdots,\varphi_{1,M-1}^{\tau}(\cdot),\cdots,\varphi_{2^{k-1},0}^{\tau}(\cdot),\varphi_{2^{k-1},1}^{\tau}(\cdot),\cdots,\varphi_{2^{k-1},M-1}^{\tau}(\cdot))^{\top}.
$$

It yields

$$
f(x,y,t)\approx\left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^{\top}\mathbf{D}\left(\mathbf{\Psi}_{k,M}^{\delta}(t)\otimes\mathbf{\Psi}_{k,M}^{\beta}(y)\right), \tag{3.5}
$$

where $\otimes$ denotes the Kronecker product.

3.3. Convergence analysis and approximation error bounds

This section focuses on the convergence analysis and approximation error bounds of the 3D eighth−kind fractional Chebyshev wavelets formulated in Eq (3.5).

Building upon the theoretical foundation established in Proposition 3.2 and using the Weierstrass approximation theorem, we develop the following convergence result through an analytical approach analogous to that employed in our prior research [22].

Theorem 3.1. (Convergence analysis) *Assume that $f(x,y,t)$ is a real function defined over $[0,1]^3$. If $f(x^{1/\alpha},y^{1/\beta},t^{1/\delta})\in L^2([0,1]^3)$, then*

$$
\lim_{M\to\infty}\|f-\left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^{\top}\mathbf{D}\left(\mathbf{\Psi}_{k,M}^{\delta}(t)\otimes\mathbf{\Psi}_{k,M}^{\beta}(y)\right)\|_{\omega}=0,
$$

where $\left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^{\top}\mathbf{D}\left(\mathbf{\Psi}_{k,M}^{\delta}(t)\otimes\mathbf{\Psi}_{k,M}^{\beta}(y)\right)$ is given in Eq (3.5).

Furthermore, if $f(x^{1/\alpha},y^{1/\beta},t^{1/\delta})\in C([0,1]^3)$, it leads to

$$
\lim_{M\to\infty}\max_{(x,y,t)\in[0,1]^3}\left|\left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^{\top}\mathbf{D}\left(\mathbf{\Psi}_{k,M}^{\delta}(t)\otimes\mathbf{\Psi}_{k,M}^{\beta}(y)\right)-f(x,y,t)\right|=0.
$$

Now, we establish rigorous error bounds for the approximation using 3D eighth−kind fractional Chebyshev wavelets. In preparation for this, we first construct the Taylor expansion formula in the Caputo fractional derivative for multivariate functions via the single−variable generalized fractional

Taylor formula and variable−by−variable recursive expansion method. Before we proceed, let us introduce some notations. For $\mathbf{a} = (a_1, a_2, \cdots, a_n)$, $\mathbf{b} = (b_1, b_2, \cdots, b_n)$, $\mathbf{k} = (k_1, k_2, \cdots, k_n)$, $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_n)$, and $\mathbf{x} = (x_1, x_2, \cdots, x_n)$, write

$$[\mathbf{a}, \mathbf{b}] = [a_1, b_1] \times \cdots \times [a_n, b_n] \text{ and } \mathcal{D}^{\alpha\mathbf{k}} = \mathcal{D}_{x_1}^{k_1\alpha_1} \cdots \mathcal{D}_{x_n}^{k_n\alpha_n}.$$

On the multivariate Caputo fractional derivative, consult [1].

Lemma 3.1. (Univariate Caputo fractional Taylor formula) *[23] Suppose that $\mathcal{D}^{k\alpha} f(x) \in C[a, b]$ for $k = 0, 1, \cdots, n + 1$, where $0 < \alpha \le 1$. It then holds that*

$$f(x) = \sum_{i=0}^{n} \frac{\mathcal{D}^{i\alpha} f(a)}{\Gamma(i\alpha + 1)}(x - a)^{i\alpha} + \frac{\mathcal{D}^{(n+1)\alpha} f(\xi)}{\Gamma((n + 1)\alpha + 1)}(x - a)^{(n+1)\alpha}$$

with $a \le \xi \le x$, $\forall x \in (a, b]$, where $\mathcal{D}^{i\alpha} = \mathcal{D}^{\alpha} \cdot \mathcal{D}^{\alpha} \cdots \mathcal{D}^{\alpha}$ (i-times).

Theorem 3.2. (Multivariate Caputo fractional Taylor formula) *Assume that $\mathcal{D}^{\alpha\mathbf{k}} u(\mathbf{x}) \in C([\mathbf{a}, \mathbf{b}])$ for $k_i = 0, 1, 2, \cdots, m_i + 1$, $\alpha_i \in (0, 1]$, and $i = 1, 2, \cdots, n$. Then,*

$$u(\mathbf{x}) = \sum_{k_1=0}^{m_1} \cdots \sum_{k_n=0}^{m_n} \frac{\mathcal{D}_{x_1}^{k_1\alpha_1} \cdots \mathcal{D}_{x_n}^{k_n\alpha_n} u(\mathbf{a})}{\prod_{i=1}^{n} \Gamma(k_i\alpha_i + 1)} \prod_{i=1}^{n} (x_i - a_i)^{k_i\alpha_i} + \sum_{\substack{k_1 \in \{0,1,\cdots,m_1\}, \\ \vdots \\ k_n \in \{0,1,\cdots,m_n\}, \\ \textit{at least one } k_1=m_1+1 \textit{ or } \cdots \textit{or } k_n=m_n+1}} \frac{\mathcal{D}_{x_1}^{k_1\alpha_1} \cdots \mathcal{D}_{x_n}^{k_n\alpha_n} u(\xi_{\mathbf{k}})}{\prod_{i=1}^{n} \Gamma(k_i\alpha_i + 1)} \prod_{i=1}^{n} (x_i - a_i)^{k_i\alpha_i}$$

for some $\xi_{\mathbf{k}} = (\xi_1^{k_1}, \cdots, \xi_n^{k_n}) \in [\mathbf{a}, \mathbf{x}]$.

Proof. Fixing $x_1, \cdots, x_{n-1}$ and applying the univariate Caputo fractional Taylor formula to x_n according to Lemma 3.1, we obtain

$$\begin{aligned} u(\mathbf{x}) = &\sum_{k_n=0}^{m_n} \frac{\mathcal{D}_{x_n}^{k_n\alpha_n} u(x_1, x_2, \cdots, x_{n-1}, a_n)}{\Gamma(k_n\alpha_n + 1)}(x_n - a_n)^{k_n\alpha_n} \\ &+ \frac{\mathcal{D}_{x_n}^{(m_n+1)\alpha_n} u(x_1, x_2, \cdots, x_{n-1}, \xi_n)}{\Gamma((m_n + 1)\alpha_n + 1)}(x_n - a_n)^{(m_n+1)\alpha_n} \end{aligned} \tag{3.6}$$

for some $\xi_n \in [a_n, x_n]$.

Holding $x_1, \cdots, x_{n-2}$ fixed and using the univariate Caputo fractional Taylor formula to x_{n-1} for functions $\mathcal{D}_{x_n}^{k_n\alpha_n} u(x_1, x_2, \cdots, x_{n-1}, a_n)$ and $\mathcal{D}_{x_n}^{(m_n+1)\alpha_n} u(x_1, x_2, \cdots, x_{n-1}, \xi_n)$ in Eq (3.6) again, we get

$$\begin{aligned} &\mathcal{D}_{x_n}^{k_n\alpha_n} u(x_1, x_2, \cdots, x_{n-1}, a_n) \\ = &\sum_{k_{n-1}=0}^{m_{n-1}} \frac{\mathcal{D}_{x_{n-1}}^{k_{n-1}\alpha_{n-1}} \mathcal{D}_{x_n}^{k_n\alpha_n} u(x_1, \cdots, x_{n-2}, a_{n-1}, a_n)}{\Gamma(k_{n-1}\alpha_{n-1} + 1)}(x_{n-1} - a_{n-1})^{k_{n-1}\alpha_{n-1}} \\ &+ \frac{\mathcal{D}_{x_{n-1}}^{(m_{n-1}+1)\alpha_{n-1}} \mathcal{D}_{x_n}^{k_n\alpha_n} u(x_1, \cdots, x_{n-2}, \xi_{n-1}, a_n)}{\Gamma((m_{n-1} + 1)\alpha_{n-1} + 1)}(x_{n-1} - a_{n-1})^{(m_{n-1}+1)\alpha_{n-1}}, \end{aligned} \tag{3.7}$$

$$
\begin{aligned}
&\mathcal{D}_{x_n}^{(m_n+1)\alpha_n}u(x_1, x_2, \cdots, x_{n-1}, \xi_n) \\
&= \sum_{k_{n-1}=0}^{m_{n-1}} \frac{\mathcal{D}_{x_{n-1}}^{k_{n-1}\alpha_{n-1}}\mathcal{D}_{x_n}^{(m_n+1)\alpha_n}u(x_1, \cdots, x_{n-2}, a_{n-1}, \xi_n)}{\mathbf{\Gamma}(k_{n-1}\alpha_{n-1}+1)}(x_{n-1}-a_{n-1})^{k_{n-1}\alpha_{n-1}} \\
&\quad + \frac{\mathcal{D}_{x_{n-1}}^{(m_{n-1}+1)\alpha_{n-1}}\mathcal{D}_{x_n}^{(m_n+1)\alpha_n}u(x_1, \cdots, x_{n-2}, \xi'_{n-1}, \xi_n)}{\mathbf{\Gamma}((m_{n-1}+1)\alpha_{n-1}+1)}(x_{n-1}-a_{n-1})^{(m_{n-1}+1)\alpha_{n-1}}
\end{aligned}
\tag{3.8}
$$

for $\xi_{n-1}, \xi'_{n-1} \in [a_{n-1}, x_{n-1}]$.

Analogously, for fixed $x_1, \cdots, x_{n-3}$, we expand the functions in Eqs (3.7) and (3.8) via the univariate Caputo fractional Taylor formula to x_{n-2}. By iteratively applying this variable−by−variable expansion method, followed by substitution and simplification, we ultimately obtain the multivariate Caputo fractional Taylor expansion formula as

$$
u(\mathbf{x}) = \sum_{k_1=0}^{m_1}\cdots\sum_{k_n=0}^{m_n} \frac{\mathcal{D}_{x_1}^{k_1\alpha_1}\cdots\mathcal{D}_{x_n}^{k_n\alpha_n}u(\mathbf{a})}{\prod_{i=1}^n \mathbf{\Gamma}(k_i\alpha_i+1)}\prod_{i=1}^n (x_i-a_i)^{k_i\alpha_i} + \sum_{\substack{k_1\in\{0,1,\cdots,m_1\},\\ \vdots \\ k_n\in\{0,1,\cdots,m_n\},\\ \text{at least one } k_1=m_1+1 \text{ or } \cdots \text{or } k_n=m_n+1}} \frac{\mathcal{D}_{x_1}^{k_1\alpha_1}\cdots\mathcal{D}_{x_n}^{k_n\alpha_n}u(\xi_{\mathbf{k}})}{\prod_{i=1}^n \mathbf{\Gamma}(k_i\alpha_i+1)}\prod_{i=1}^n (x_i-a_i)^{k_i\alpha_i}
$$

for $\xi_{\mathbf{k}} \in [\mathbf{a}, \mathbf{x}]$. □

Theorem 3.3. (Approximation error bounds) *Suppose that* $\mathcal{D}_x^{m\alpha}\mathcal{D}_y^{j\beta}\mathcal{D}_t^{s\delta}f(x,y,t) \in C([0,1]^3)$, $m, j, s = 0, 1, 2, \cdots, M$, $\alpha, \beta, \delta \in (0,1]$, *and* $\left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^{\top}\mathbf{D}\left(\mathbf{\Psi}_{k,M}^{\delta}(t)\otimes\mathbf{\Psi}_{k,M}^{\beta}(y)\right)$ *is as in Eq (3.5). Then,*

$$
\begin{aligned}
&\|f - \left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^{\top}\mathbf{D}\left(\mathbf{\Psi}_{k,M}^{\delta}(t)\otimes\mathbf{\Psi}_{k,M}^{\beta}(y)\right)\|_{\omega}^2 \\
&\leq \Bigg(\sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} \left(\frac{B_{m,j,s}}{\mathbf{\Gamma}(m\alpha+1)\mathbf{\Gamma}(j\beta+1)\mathbf{\Gamma}(s\delta+1)}\right)^2\Bigg) \\
&\quad\cdot \sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} \Bigg(\sum_{n=1}^{2^{k-1}}\sum_{l=0}^{2m}\frac{C_{2m}^{l}(2n-1)^{2m-l}}{2^{k(2m+1)}}J(l+4,2)\Bigg)\Bigg(\sum_{i=1}^{2^{k-1}}\sum_{l'=0}^{2j}\frac{C_{2j}^{l'}(2i-1)^{2j-l'}}{2^{k(2j+1)}}J(l'+4,2)\Bigg) \\
&\quad\cdot\Bigg(\sum_{q=1}^{2^{k-1}}\sum_{l''=0}^{2s}\frac{C_{2s}^{l''}(2q-1)^{2s-l''}}{2^{k(2q+1)}}J(l''+4,2)\Bigg),
\end{aligned}
$$

where $B_{m,j,s} = \max\limits_{(x,y,t)\in[0,1]^3}\left|(\mathcal{D}_x^{m\alpha}\mathcal{D}_y^{j\beta}\mathcal{D}_t^{s\delta}f)(x,y,t)\right|$.

Proof. By Theorem 3.2, it follows that

$$
f(x,y,t) = \sum_{m=0}^{M-1}\sum_{j=0}^{M-1}\sum_{s=0}^{M-1}\frac{(\mathcal{D}_x^{m\alpha}\mathcal{D}_y^{j\beta}\mathcal{D}_t^{s\delta}f)(0,0,0)}{\mathbf{\Gamma}(m\alpha+1)\mathbf{\Gamma}(j\beta+1)\mathbf{\Gamma}(s\delta+1)}x^{m\alpha}y^{j\beta}t^{s\delta}
$$

$$+ \sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} \frac{(\mathcal{D}_x^{m\alpha}\mathcal{D}_y^{j\beta}\mathcal{D}_t^{s\delta} f)(\xi_m, \eta_j, \zeta_s)}{\mathbf{\Gamma}(m\alpha+1)\mathbf{\Gamma}(j\beta+1)\mathbf{\Gamma}(s\delta+1)} x^{m\alpha} y^{j\beta} t^{s\delta},$$

where $(\xi_m, \eta_j, \zeta_s) \in [0, x] \times [0, y] \times [0, t]$. Write

$$H_{k,M}^{\alpha,\beta,\delta} = span\{\varphi_{n,m}^{\alpha}(x)\varphi_{i,j}^{\beta}(y)\varphi_{q,s}^{\delta}(t) : n, i, q = 1, 2, \cdots, 2^{k-1}, m, j, s = 0, 1, 2, \cdots, M-1\}.$$

Proposition 3.2 states that

$$\sum_{m=0}^{M-1}\sum_{j=0}^{M-1}\sum_{s=0}^{M-1} \frac{(\mathcal{D}_x^{m\alpha}\mathcal{D}_y^{j\beta}\mathcal{D}_t^{s\delta} f)(0,0,0)}{\mathbf{\Gamma}(m\alpha+1)\mathbf{\Gamma}(j\beta+1)\mathbf{\Gamma}(s\delta+1)} x^{m\alpha} y^{j\beta} t^{s\delta} \in H_{k,M}^{\alpha,\beta,\delta}.$$

Applying the optimal approximation theorem yields

$$\begin{aligned}
&\|f - \left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^{\top} \mathbf{D} \left(\mathbf{\Psi}_{k,M}^{\delta}(t) \otimes \mathbf{\Psi}_{k,M}^{\beta}(y)\right) \|_{\omega}^2 \\
&\le \sum_{n=1}^{2^{k-1}}\sum_{i=1}^{2^{k-1}}\sum_{q=1}^{2^{k-1}} \int_{D_{n,i,q}^{\alpha,\beta,\delta}} \Bigg| \sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} \frac{(\mathcal{D}_x^{m\alpha}\mathcal{D}_y^{j\beta}\mathcal{D}_t^{s\delta} f)(\xi_m, \eta_j, \zeta_s)}{\mathbf{\Gamma}(m\alpha+1)\mathbf{\Gamma}(j\beta+1)\mathbf{\Gamma}(s\delta+1)} x^{m\alpha} y^{j\beta} t^{s\delta} \Bigg|^2 \\
&\qquad \cdot \omega_{n,i,q}^{\alpha,\beta,\delta}(x, y, t)\, dxdydt.
\end{aligned}$$

Observe that

$$\begin{aligned}
&\Bigg| \sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} \frac{(\mathcal{D}_x^{m\alpha}\mathcal{D}_y^{j\beta}\mathcal{D}_t^{s\delta} f)(\xi_m, \eta_j, \zeta_s)}{\mathbf{\Gamma}(m\alpha+1)\mathbf{\Gamma}(j\beta+1)\mathbf{\Gamma}(s\delta+1)} x^{m\alpha} y^{j\beta} t^{s\delta} \Bigg|^2 \\
&\le \Bigg(\sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} \Big(\frac{(\mathcal{D}_x^{m\alpha}\mathcal{D}_y^{j\beta}\mathcal{D}_t^{s\delta} f)(\xi_m, \eta_j, \zeta_s)}{\mathbf{\Gamma}(m\alpha+1)\mathbf{\Gamma}(j\beta+1)\mathbf{\Gamma}(s\delta+1)} \Big)^2 \Bigg) \Bigg(\sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} x^{2m\alpha} y^{2j\beta} t^{2s\delta} \Bigg).
\end{aligned}$$

Thus

$$\begin{aligned}
&\|f - \left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^{\top} \mathbf{D} \left(\mathbf{\Psi}_{k,M}^{\delta}(t) \otimes \mathbf{\Psi}_{k,M}^{\beta}(y)\right) \|_{\omega}^2 \\
&\le \Bigg(\sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} \Big(\frac{B_{m,j,s}}{\mathbf{\Gamma}(m\alpha+1)\mathbf{\Gamma}(j\beta+1)\mathbf{\Gamma}(s\delta+1)} \Big)^2 \Bigg) \\
&\quad \cdot \sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} \Bigg(\sum_{n=1}^{2^{k-1}}\sum_{i=1}^{2^{k-1}}\sum_{q=1}^{2^{k-1}} \int_{D_{n,i,q}^{\alpha,\beta,\delta}} x^{2m\alpha} y^{2j\beta} t^{2s\delta} \omega_{n,i,q}^{\alpha,\beta,\delta}(x, y, t) dxdydt \Bigg).
\end{aligned}$$

It is worth noting that

$$\sum_{n=1}^{2^{k-1}}\sum_{i=1}^{2^{k-1}}\sum_{q=1}^{2^{k-1}} \int_{D_{n,i,q}^{\alpha,\beta,\delta}} x^{2m\alpha} y^{2j\beta} t^{2s\delta} \omega_{n,i,q}^{\alpha,\beta,\delta}(x, y, t) dxdydt$$

$$
\begin{aligned}
&= \sum_{n=1}^{2^{k-1}} \int_{(\frac{n-1}{2^{k-1}})^{\frac{1}{\alpha}}}^{(\frac{n}{2^{k-1}})^{\frac{1}{\alpha}}} x^{2m\alpha}(2^k x^\alpha - 2n + 1)^4 \sqrt{1 - (2^k x^\alpha - 2n + 1)^2} \alpha x^{\alpha-1} dx \\
&\cdot \sum_{i=1}^{2^{k-1}} \int_{(\frac{i-1}{2^{k-1}})^{\frac{1}{\beta}}}^{(\frac{i}{2^{k-1}})^{\frac{1}{\beta}}} y^{2j\beta}(2^k y^\beta - 2i + 1)^4 \sqrt{1 - (2^k y^\beta - 2i + 1)^2} \beta y^{\beta-1} dy \\
&\cdot \sum_{q=1}^{2^{k-1}} \int_{(\frac{q-1}{2^{k-1}})^{\frac{1}{\delta}}}^{(\frac{q}{2^{k-1}})^{\frac{1}{\delta}}} t^{2s\delta}(2^k t^\delta - 2q + 1)^4 \sqrt{1 - (2^k t^\delta - 2q + 1)^2} \delta t^{\delta-1} dt.
\end{aligned}
$$

Similar to the calculations in the proof of Proposition 3.2, it is straightforward to obtain that

$$
\sum_{n=1}^{2^{k-1}} \int_{(\frac{n-1}{2^{k-1}})^{\frac{1}{\alpha}}}^{(\frac{n}{2^{k-1}})^{\frac{1}{\alpha}}} x^{2m\alpha}(2^k x^\alpha - 2n + 1)^4 \sqrt{1 - (2^k x^\alpha - 2n + 1)^2}\, \alpha x^{\alpha-1} dx = \sum_{n=1}^{2^{k-1}} \sum_{l=0}^{2m} \frac{C_{2m}^{l}(2n-1)^{2m-l}}{2^{k(2m+1)}} J(l+4, 2),
$$

$$
\sum_{i=1}^{2^{k-1}} \int_{(\frac{i-1}{2^{k-1}})^{\frac{1}{\beta}}}^{(\frac{i}{2^{k-1}})^{\frac{1}{\beta}}} y^{2j\beta}(2^k y^\beta - 2i + 1)^4 \sqrt{1 - (2^k y^\beta - 2i + 1)^2}\, \beta y^{\beta-1} dy = \sum_{i=1}^{2^{k-1}} \sum_{l'=0}^{2j} \frac{C_{2j}^{l'}(2i-1)^{2j-l'}}{2^{k(2j+1)}} J(l'+4, 2),
$$

$$
\sum_{q=1}^{2^{k-1}} \int_{(\frac{q-1}{2^{k-1}})^{\frac{1}{\delta}}}^{(\frac{q}{2^{k-1}})^{\frac{1}{\delta}}} t^{2s\delta}(2^k t^\delta - 2q + 1)^4 \sqrt{1 - (2^k t^\delta - 2q + 1)^2}\, \delta t^{\delta-1} dt = \sum_{q=1}^{2^{k-1}} \sum_{l''=0}^{2s} \frac{C_{2s}^{l''}(2q-1)^{2s-l''}}{2^{k(2q+1)}} J(l''+4, 2).
$$

Therefore,

$$
\begin{aligned}
&\|f - \left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^\top \mathbf{D} \left(\mathbf{\Psi}_{k,M}^{\delta}(t) \otimes \mathbf{\Psi}_{k,M}^{\beta}(y)\right)\|_\omega^2 \\
&\le \Bigg(\sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} \left(\frac{B_{m,j,s}}{\mathbf{\Gamma}(m\alpha+1)\mathbf{\Gamma}(j\beta+1)\mathbf{\Gamma}(s\delta+1)} \right)^2 \Bigg) \\
&\cdot \sum_{\substack{m,j,s\in\{0,1,\cdots,M-1\},\\ \text{at least one } m=M \text{ or } j=M \text{ or } s=M}} \Big(\sum_{n=1}^{2^{k-1}} \sum_{l=0}^{2m} \frac{C_{2m}^{l}(2n-1)^{2m-l}}{2^{k(2m+1)}} J(l+4, 2) \Big) \Big(\sum_{i=1}^{2^{k-1}} \sum_{l'=0}^{2j} \frac{C_{2j}^{l'}(2i-1)^{2j-l'}}{2^{k(2j+1)}} J(l'+4, 2) \Big) \\
&\cdot \Big(\sum_{q=1}^{2^{k-1}} \sum_{l''=0}^{2s} \frac{C_{2s}^{l''}(2q-1)^{2s-l''}}{2^{k(2q+1)}} J(l''+4, 2) \Big).
\end{aligned}
$$

□

4. Algorithm

This section introduces the algorithmic framework based on the 3D eighth−kind fractional Chebyshev wavelets and ALA technique to provide effective numerical solutions to the problem given by Eq (1.1). Our numerical algorithm is fundamentally grounded in the Riemann−Liouville fractional integration formulas of eighth−kind fractional Chebyshev wavelets, as rigorously established in our previous work [22].

4.1. Fractional integration formulas

Lemma 4.1. *[22] Given $\lambda > 0$, the fractional integration formulas of the eighth−kind fractional Chebyshev wavelets $\varphi^{\alpha}_{n,m}(x)(n = 1, 2, \cdots, 2^{k-1},\ m = 0, 1, 2, \cdots)$ admit the representation*

$$\mathcal{I}^{\lambda}\varphi^{\alpha}_{n,m}(x) = \begin{cases} 0,\ 0 \le x < (\frac{n-1}{2^{k-1}})^{\frac{1}{\alpha}}, \\ \mathrm{F}(x),\ (\frac{n-1}{2^{k-1}})^{\frac{1}{\alpha}} \le x < (\frac{n}{2^{k-1}})^{\frac{1}{\alpha}}, \\ \mathrm{F}(x) - \mathrm{G}(x),\ (\frac{n}{2^{k-1}})^{\frac{1}{\alpha}} \le x \le 1, \end{cases}$$

where

$$\mathrm{F}(x) = \sum_{p=0}^{m}\sum_{i=0}^{p}\sum_{j=0}^{i} Q^{n,m,p}_{i,j,k}\Big(\frac{\mathbf{\Gamma}(\alpha j+1)}{\mathbf{\Gamma}(\alpha j+\lambda+1)}x^{\alpha j+\lambda}\Big[1-\mathbb{I}(\frac{(\frac{n-1}{2^{k-1}})^{\frac{1}{\alpha}}}{x};\alpha j+1,\lambda)\Big]\mu_{(\frac{n-1}{2^{k-1}})^{\frac{1}{\alpha}}}(x)\Big),$$

$$\mathrm{G}(x) = \sum_{p=0}^{m}\sum_{i=0}^{p}\sum_{j=0}^{i} Q^{n,m,p}_{i,j,k}\Big(\frac{\mathbf{\Gamma}(\alpha j+1)}{\mathbf{\Gamma}(\alpha j+\lambda+1)}x^{\alpha j+\lambda}\Big[1-\mathbb{I}(\frac{(\frac{n}{2^{k-1}})^{\frac{1}{\alpha}}}{x};\alpha j+1,\lambda)\Big]\mu_{(\frac{n}{2^{k-1}})^{\frac{1}{\alpha}}}(x)\Big),$$

$Q^{n,m,p}_{i,j,k} = \frac{2^k}{\sqrt{h_m}}\frac{a_{p,m}}{2^p}C^i_pC^j_i2^{kj}(1-2n)^{i-j}$, $\mathbb{I}(x;a,b) = \frac{\mathbf{\Gamma}(a+b)}{\mathbf{\Gamma}(a)\mathbf{\Gamma}(b)}\int_0^x \tau^{a-1}(1-\tau)^{b-1}d\tau$, *and* $\mu_c(x) = \begin{cases} 1, x \ge c, \\ 0, \mathrm{O.W.} \end{cases}$

4.2. Approximation algorithm

• **Linear case** (*i.e.* $g(u) = 0$)**:**

Suppose that the function $\partial^5 u(x,y,t)/\partial t\partial y^2\partial x^2$ of $u(x,y,t)$ in Eq (1.1) is approximated by 3D eighth−kind fractional Chebyshev wavelets as

$$\frac{\partial^5 u(x,y,t)}{\partial t\partial y^2\partial x^2} = \left(\mathbf{\Psi}^{\alpha}_{k,M}(x)\right)^{\top}\mathbf{D}\left(\mathbf{\Psi}^{\delta}_{k,M}(t)\otimes\mathbf{\Psi}^{\beta}_{k,M}(y)\right), \tag{4.1}$$

where $\mathbf{D}$ represents the coefficient matrix with size $2^{k-1}M \times (2^{k-1}M)^2$, to be determined subsequently. By integrating Eq (4.1) once with respect to the time variable t and incorporating the initial boundary conditions, we obtain

$$\frac{\partial^4 u(x,y,t)}{\partial y^2\partial x^2} = \left(\mathbf{\Psi}^{\alpha}_{k,M}(x)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}^1_t\mathbf{\Psi}^{\delta}_{k,M}(t)\right)\otimes\mathbf{\Psi}^{\beta}_{k,M}(y)\right) + \frac{\partial^4 L(x,y)}{\partial y^2\partial x^2}. \tag{4.2}$$

Integrating Eq (4.2) once in y along with the initial boundary conditions yields

$$\frac{\partial^3 u(x,y,t)}{\partial y\partial x^2} = \left(\mathbf{\Psi}^{\alpha}_{k,M}(x)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}^1_t\mathbf{\Psi}^{\delta}_{k,M}(t)\right)\otimes\left(\mathcal{I}^1_y\mathbf{\Psi}^{\beta}_{k,M}(y)\right)\right) + \frac{\partial^3 L(x,y)}{\partial y\partial x^2} + \left.\frac{\partial^3 u(x,y,t)}{\partial y\partial x^2}\right|_{y=0} - \left.\frac{\partial^3 L(x,y)}{\partial y\partial x^2}\right|_{y=0}. \tag{4.3}$$

Integrating Eq (4.3) once more in y and imposing the initial boundary conditions leads to

$$\frac{\partial^2 u(x,y,t)}{\partial x^2} = \left(\mathbf{\Psi}^{\alpha}_{k,M}(x)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}^1_t\mathbf{\Psi}^{\delta}_{k,M}(t)\right)\otimes\left(\mathcal{I}^2_y\mathbf{\Psi}^{\beta}_{k,M}(y)\right)\right) + \frac{\partial^2 L(x,y)}{\partial x^2}$$

$$+\, y\left(\left.\frac{\partial^3 u(x,y,t)}{\partial y \partial x^2}\right|_{y=0} - \left.\frac{\partial^3 L(x,y)}{\partial y \partial x^2}\right|_{y=0}\right) + \frac{\partial^2 H_1(x,t)}{\partial x^2} - \left.\frac{\partial^2 L(x,y)}{\partial x^2}\right|_{y=0}. \tag{4.4}$$

Setting $y = 1$ in Eq (4.4) gives

$$\begin{aligned}\left.\frac{\partial^3 u(x,y,t)}{\partial y \partial x^2}\right|_{y=0} - \left.\frac{\partial^3 L(x,y)}{\partial y \partial x^2}\right|_{y=0} &= -\left(\mathbf{\Psi}^{\alpha}_{k,M}(x)\right)^{\top} \mathbf{D}\left(\left(\mathcal{I}^1_t \mathbf{\Psi}^{\delta}_{k,M}(t)\right) \otimes \left(\mathcal{I}^2_y \mathbf{\Psi}^{\beta}_{k,M}(1)\right)\right) + \frac{\partial^2 H_2(x,t)}{\partial x^2} \\ &\quad - \frac{\partial^2 H_1(x,t)}{\partial x^2} + \left.\frac{\partial^2 L(x,y)}{\partial x^2}\right|_{y=0} - \left.\frac{\partial^2 L(x,y)}{\partial x^2}\right|_{y=1}. \end{aligned}\tag{4.5}$$

Substituting Eq (4.5) into Eq (4.4) results in

$$\frac{\partial^2 u(x,y,t)}{\partial x^2} = \left(\mathbf{\Psi}^{\alpha}_{k,M}(x)\right)^{\top} \mathbf{D}\left(\left(\mathcal{I}^1_t \mathbf{\Psi}^{\delta}_{k,M}(t)\right) \otimes \left(\mathcal{I}^2_y \mathbf{\Psi}^{\beta}_{k,M}(y) - y\mathcal{I}^2_y \mathbf{\Psi}^{\beta}_{k,M}(1)\right)\right) + K(x,y,t), \tag{4.6}$$

where

$$\begin{aligned}K(x,y,t) &= \frac{\partial^2 L(x,y)}{\partial x^2} + y\left(\frac{\partial^2 H_2(x,t)}{\partial x^2} - \frac{\partial^2 H_1(x,t)}{\partial x^2} + \left.\frac{\partial^2 L(x,y)}{\partial x^2}\right|_{y=0} - \left.\frac{\partial^2 L(x,y)}{\partial x^2}\right|_{y=1}\right) \\ &\quad + \frac{\partial^2 H_1(x,t)}{\partial x^2} - \left.\frac{\partial^2 L(x,y)}{\partial x^2}\right|_{y=0}.\end{aligned}$$

Integrating Eq (4.6) in x once under the initial boundary conditions, we derive

$$\begin{aligned}\frac{\partial u(x,y,t)}{\partial x} &= \left(\mathcal{I}^1_x \mathbf{\Psi}^{\alpha}_{k,M}(x)\right)^{\top} \mathbf{D}\left(\left(\mathcal{I}^1_t \mathbf{\Psi}^{\delta}_{k,M}(t)\right) \otimes \left(\mathcal{I}^2_y \mathbf{\Psi}^{\beta}_{k,M}(y) - y\mathcal{I}^2_y \mathbf{\Psi}^{\beta}_{k,M}(1)\right)\right) + V(x,y,t) \\ &\quad + \left.\frac{\partial u(x,y,t)}{\partial x}\right|_{x=0} - V(x,y,t)|_{x=0}, \end{aligned}\tag{4.7}$$

where

$$\begin{aligned}V(x,y,t) &= \frac{\partial L(x,y)}{\partial x} + y\left(\frac{\partial H_2(x,t)}{\partial x} - \frac{\partial H_1(x,t)}{\partial x} + \left.\frac{\partial L(x,y)}{\partial x}\right|_{y=0} - \left.\frac{\partial L(x,y)}{\partial x}\right|_{y=1}\right) \\ &\quad + \frac{\partial H_1(x,t)}{\partial x} - \left.\frac{\partial L(x,y)}{\partial x}\right|_{y=0}.\end{aligned}$$

Upon integrating Eq (4.7) in x one time and by the initial boundary conditions, it reduces to

$$\begin{aligned}u(x,y,t) &= \left(\mathcal{I}^2_x \mathbf{\Psi}^{\alpha}_{k,M}(x)\right)^{\top} \mathbf{D}\left(\left(\mathcal{I}^1_t \mathbf{\Psi}^{\delta}_{k,M}(t)\right) \otimes \left(\mathcal{I}^2_y \mathbf{\Psi}^{\beta}_{k,M}(y) - y\mathcal{I}^2_y \mathbf{\Psi}^{\beta}_{k,M}(1)\right)\right) + W(x,y,t) \\ &\quad + x\left(\left.\frac{\partial u(x,y,t)}{\partial x}\right|_{x=0} - V(x,y,t)|_{x=0}\right) + G_1(y,t) - W(x,y,t)|_{x=0}, \end{aligned}\tag{4.8}$$

where

$$W(x,y,t) = L(x,y) + y\left(H_2(x,t) - H_1(x,t) + L(x,y)|_{y=0} - L(x,y)|_{y=1}\right) + H_1(x,t) - L(x,y)|_{y=0}.$$

Taking $x = 1$ in Eq (4.8) yields

$$\left.\frac{\partial u(x,y,t)}{\partial x}\right|_{x=0} - V(x,y,t)|_{x=0} = -\left(\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(1)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}_t^1\mathbf{\Psi}_{k,M}^{\delta}(t)\right)\otimes\left(\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(y) - y\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(1)\right)\right) + G_2(y,t) - G_1(y,t) + W(x,y,t)|_{x=0} - W(x,y,t)|_{x=1}. \tag{4.9}$$

By substituting Eq (4.9) into Eq (4.8), it follows that

$$u(x,y,t) = \left(\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(x) - x\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(1)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}_t^1\mathbf{\Psi}_{k,M}^{\delta}(t)\right)\otimes\left(\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(y) - y\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(1)\right)\right) + S(x,y,t), \tag{4.10}$$

where

$$S(x,y,t) = W(x,y,t) + x\left(G_2(y,t) - G_1(y,t) + W(x,y,t)|_{x=0} - W(x,y,t)|_{x=1}\right) + G_1(y,t) - W(x,y,t)|_{x=0}.$$

Differentiating Eq (4.10) with respect to each independent variable gives the following equations:

$$\frac{\partial^{\lambda} u(x,y,t)}{\partial t^{\lambda}} = \left(\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(x) - x\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(1)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}_t^{1-\lambda}\mathbf{\Psi}_{k,M}^{\delta}(t)\right)\otimes\left(\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(y) - y\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(1)\right)\right) + \frac{\partial^{\lambda} S(x,y,t)}{\partial t^{\lambda}}, \tag{4.11}$$

$$\frac{\partial u(x,y,t)}{\partial t} = \left(\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(x) - x\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(1)\right)^{\top}\mathbf{D}\left(\mathbf{\Psi}_{k,M}^{\delta}(t)\otimes\left(\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(y) - y\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(1)\right)\right) + \frac{\partial S(x,y,t)}{\partial t}, \tag{4.12}$$

$$\frac{\partial^2 u(x,y,t)}{\partial x^2} = \left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}_t^1\mathbf{\Psi}_{k,M}^{\delta}(t)\right)\otimes\left(\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(y) - y\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(1)\right)\right) + \frac{\partial^2 S(x,y,t)}{\partial x^2}, \tag{4.13}$$

$$\frac{\partial^2 u(x,y,t)}{\partial y^2} = \left(\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(x) - x\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(1)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}_t^1\mathbf{\Psi}_{k,M}^{\delta}(t)\right)\otimes\mathbf{\Psi}_{k,M}^{\beta}(y)\right) + \frac{\partial^2 S(x,y,t)}{\partial y^2}. \tag{4.14}$$

By substituting Eqs (4.11–4.14) into Eq (1.1), we obtain

$$\begin{aligned}
0 = {} & \left(\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(x) - x\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(1)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}_t^{1-\lambda}\mathbf{\Psi}_{k,M}^{\delta}(t) + \mathbf{\Psi}_{k,M}^{\delta}(t)\right)\otimes\left(\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(y) - y\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(1)\right)\right) \\
& - \left(\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(x) - x\mathcal{I}_x^2\mathbf{\Psi}_{k,M}^{\alpha}(1)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}_t^1\mathbf{\Psi}_{k,M}^{\delta}(t)\right)\otimes\mathbf{\Psi}_{k,M}^{\beta}(y)\right) \\
& - \left(\mathbf{\Psi}_{k,M}^{\alpha}(x)\right)^{\top}\mathbf{D}\left(\left(\mathcal{I}_t^1\mathbf{\Psi}_{k,M}^{\delta}(t)\right)\otimes\left(\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(y) - y\mathcal{I}_y^2\mathbf{\Psi}_{k,M}^{\beta}(1)\right)\right) + F(x,y,t),
\end{aligned} \tag{4.15}$$

where $F(x,y,t) = \partial^{\lambda}S(x,y,t)/\partial t^{\lambda} + \partial S(x,y,t)/\partial t - \partial^2 S(x,y,t)/\partial x^2 - \partial^2 S(x,y,t)/\partial y^2 - f(x,y,t)$. Take the $(2^{k-1}M)^3$ collocation points

$$\mathtt{x}_i = \left(\frac{i-0.5}{2^{k-1}M}\right)^{\frac{1}{\alpha}}, \quad \mathtt{y}_j = \left(\frac{j-0.5}{2^{k-1}M}\right)^{\frac{1}{\beta}}, \quad \mathtt{t}_l = \left(\frac{l-0.5}{2^{k-1}M}\right)^{\frac{1}{\delta}}, \quad i,j,l = 1,2,\ldots,2^{k-1}M,$$

and write

$$\mathbf{P}=\left(\mathcal{I}_x^2\mathbf{\Psi}^{\alpha}_{k,M}(\mathtt{x}_i)-\mathtt{x}_i\mathcal{I}_x^2\mathbf{\Psi}^{\alpha}_{k,M}(1)\right)^{\top}_{2^{k-1}M\times 2^{k-1}M},\quad \mathbf{Q}=\left(\mathcal{I}_y^2\mathbf{\Psi}^{\beta}_{k,M}(\mathtt{y}_j)-\mathtt{y}_j\mathcal{I}_y^2\mathbf{\Psi}^{\beta}_{k,M}(1)\right)_{2^{k-1}M\times 2^{k-1}M},$$

$$\mathbf{E}=\left(\left(\mathcal{I}_t^{1-\lambda}\mathbf{\Psi}^{\delta}_{k,M}(\mathtt{t}_l)+\mathbf{\Psi}^{\delta}_{k,M}(\mathtt{t}_l)\right)\otimes\mathbf{Q}-\left(\left(\mathcal{I}_t^{1}\mathbf{\Psi}^{\delta}_{k,M}(\mathtt{t}_l)\right)\otimes\left(\mathbf{\Psi}^{\beta}_{k,M}(\mathtt{y}_j)\right)\right)\right)_{(2^{k-1}M)^2\times(2^{k-1}M)^2},$$

$$\mathbf{R}=\left(-\mathbf{\Psi}^{\alpha}_{k,M}(\mathtt{x}_i)\right)^{\top}_{2^{k-1}M\times 2^{k-1}M},$$

$$\mathbf{T}=\left(\left(\mathcal{I}_t^{1}\mathbf{\Psi}^{\delta}_{k,M}(\mathtt{t}_l)\right)\otimes\left(\mathcal{I}_y^2\mathbf{\Psi}^{\beta}_{k,M}(\mathtt{y}_j)-\mathtt{y}_j\mathcal{I}_y^2\mathbf{\Psi}^{\beta}_{k,M}(1)\right)\right)_{(2^{k-1}M)^2\times(2^{k-1}M)^2},$$

$$\mathbf{F}=\left(F(\mathtt{x}_i,\mathtt{y}_j,\mathtt{t}_l)\right)_{2^{k-1}M\times 2^{k-1}M\times 2^{k-1}M},\quad \mathbf{Z}=\left(\mathbf{F}\right)_{2^{k-1}M\times(2^{k-1}M)^2},$$

where the matrix $\mathbf{Z}$ is obtained by processing the matrix $\mathbf{F}$ using the method shown in Figure 1.

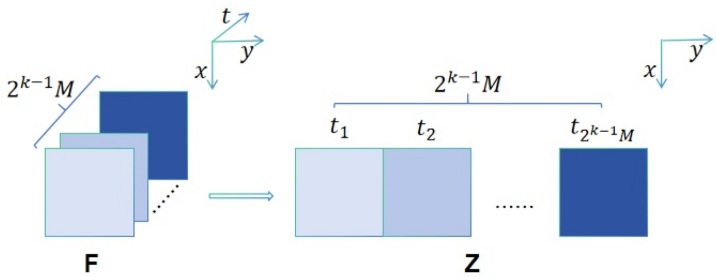

Figure 1. Method for obtaining the matrix $\mathbf{Z}$ by processing the matrix $\mathbf{F}$.

From Eq (4.15), we derive the following matrix equation:

$$\mathbf{PDE}+\mathbf{RDT}+\mathbf{Z}=\mathbf{0}.$$

By writing

$$\mathbf{A}=\mathbf{P}^{-1}\mathbf{R},\ \ \mathbf{B}=\mathbf{E}\mathbf{T}^{-1},\ \ \mathbf{C}=\mathbf{P}^{-1}\mathbf{Z}\mathbf{T}^{-1},$$

we arrive at the Sylvester equation

$$\mathbf{AD}+\mathbf{DB}+\mathbf{C}=\mathbf{0}.$$

The matrix $\mathbf{D}$ can be determined by solving this equation. Subsequently, the approximate solution is derived by inserting $\mathbf{D}$ into Eq (4.10).

• **Nonlinear case (***i.e.* $g(u)\neq 0$**):**

The Picard iteration method converts the nonlinear problem into a linear one. By iteratively applying the procedure established for the linear case, we ultimately obtain the coefficient matrix $\mathbf{D}$.

4.3. Artificial lemming algorithm

Furthermore, we enhanced the algorithm presented in Section 4.2 through the ALA. The ALA is employed to optimize the fractional orders α,β, and δ in 3D eighth−kind fractional Chebyshev wavelets, thereby significantly improving the algorithm's numerical accuracy.

Procedure of the ALA:
(1) Initialize the group;
(2) Calculate the value of the population objective function;
(3) Update the optimal function values of both the individual and the population;
(4) Population update and iteration:
  (i) Calculate the energy of each individual;
  (ii) Individuals with energy higher than the threshold enter the global exploration stage;
  (iii) Individuals whose energy is lower than the threshold enter the local exploitation stage.
(5) Repeat steps (2) to (4) before the termination condition is met;
(6) When the termination condition is met, output the optimal function value and the position.

5. Numerical examples

Several (2+1)D time−fractional mobile/immobile equations are investigated herein to demonstrate the proposed method's performance. As supplementary validation of our method's capability in handling 3D space−fractional equations, we conduct a numerical test on a 3D space−fractional Poisson equation. The presented numerical results are carried out using MATLAB R2022b.

This paper investigates the following two distinct types of numerical errors:
(i)
$$\mathrm{e}_2^l = \left(\frac{1}{pq}\sum_{i=1}^{p}\sum_{j=1}^{q}\left(u_{k,M}(x_i, y_j, t_l) - u(x_i, y_j, t_l)\right)^2\right)^{\frac{1}{2}},$$
$$\mathrm{e}_2 = \max_{1\le l\le r} \mathrm{e}_2^l;$$
(ii)
$$\mathrm{e}_\infty^l = \max_{1\le i\le p, 1\le j\le q}\left|u_{k,M}(x_i, y_j, t_l) - u(x_i, y_j, t_l)\right|,$$
$$\mathrm{e}_\infty = \max_{1\le l\le r} \mathrm{e}_\infty^l,$$
where $u(x, y, t)$ and $u_{k,M}(x, y, t)$ represent the exact and approximate solutions, respectively, and (x_i, y_j, t_l) are the sampling points in the unit cube $[0, 1]^3$. The fitness function considered in this paper is given as follows:
$$fitness = [e_2, e_\infty].$$

For the numerical results presented, the scale of the approximation is governed by the number of terms, $(2^{k-1}M)^3$, in the summation of Eq (3.5).

Example 1. Consider the (2+1)D time−fractional mobile/immobile equation
$$\frac{\partial^\lambda u(x, y, t)}{\partial t^\lambda} + \frac{\partial u(x, y, t)}{\partial t} = \frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2} + f(x, y, t),\ \ (x, y, t) \in (0, 1)^2 \times (0, 1],$$
where
$$\begin{aligned} f(x, y, t) = &-(1 + \frac{t^{1+\lambda}}{\boldsymbol{\Gamma}(2 + \lambda)})\left((12x^2 - 12x + 2)(y - y^2)^2 + (x - x^2)^2(12y^2 - 12y + 2)\right) \\ &+ (t + \frac{1 + \lambda}{\boldsymbol{\Gamma}(2 + \lambda)}t^\lambda)(x - x^2)^2(y - y^2)^2 \end{aligned}$$

Pseudo−code

for $iter = 1 : maxIterations$ **do**
  $E \leftarrow 4 * \arctan(1 - \frac{iter}{maxIterations}) \cdot \log(\frac{1}{rand(numLemmings, 1)})$
  for $i = 1 : numLemmings$ **do**
    if $E(i) > 1$ **then**
      if $rand() < 0.3$ **then**
        $a \leftarrow randi([1, numLemmings])$
        $R \leftarrow 2 * rand(1, dim) - 1$
        $BM \leftarrow normrnd(0, 1, 1, dim)$
        $FF \leftarrow sign(2 * rand() + 1 - 1.5)$
        $newLemmings(i, :) \leftarrow bestbestLemming + FF * BM \cdot (R \cdot (bestbestLemming - Lemmings(i, :)) + (1 - R) \cdot (Lemmings(i, :) - Lemmings(a, :)))$
      else
        $b \leftarrow randi([1, numLemmings])$
        $L \leftarrow rand() * (1 + \sin(\frac{iter}{2}))$
        $FF \leftarrow sign(2 * rand() + 1 - 1.5)$
        $newLemmings(i, :) \leftarrow Lemmings(i, :) + FF * L * (bestbestLemming - Lemmings(b, :))$
      end if
    else
      if $rand() < 0.5$ **then**
        $radius \leftarrow norm(bestbestLemming - Lemmings(i, :))$
        $spiral \leftarrow radius * (\sin(2\pi * rand()) + \cos(2\pi * rand()))$
        $FF \leftarrow sign(2 * rand() + 1 - 1.5)$
        $newLemmings(i, :) \leftarrow bestbestLemming + FF * spiral * rand() * Lemmings(i, :)$
      else
        $G \leftarrow 2 * (1 - \frac{iter}{maxIterations})$
        $u \leftarrow rand(1, dim)$
        $v \leftarrow rand(1, dim)$
        $\beta \leftarrow 1.5$
        $\sigma \leftarrow (\frac{\Gamma(1+\beta)\sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2})\beta 2^{\frac{\beta-1}{2}}})^{\frac{1}{\beta}}$
        $Levy \leftarrow 0.01 * u \cdot \sigma / (|v|^{\frac{1}{\beta}})$
        $FF \leftarrow sign(2 * rand() + 1 - 1.5)$
        $newLemmings(i, :) \leftarrow bestbestLemming + FF * G * Levy \cdot (bestbestLemming - Lemmings(i, :))$
      end if
    end if
    $newLemmings(i, :) \leftarrow \max(\min(newLemmings(i, :), upperBounds), lowerBounds)$
  end for
  $Lemmings \leftarrow newLemmings$
end for

with the initial boundary conditions

$$u(x,y,0) = (x-x^2)^2(y-y^2)^2,\ u(0,y,t) = u(1,y,t) = u(x,0,t) = u(x,1,t) = 0.$$

The exact solution is $u(x,y,t) = (1+t^{1+\lambda}/\boldsymbol{\Gamma}(2+\lambda))(x-x^2)^2(y-y^2)^2$.

Table 1 presents the $\mathfrak{e}_2^l$ and $\mathfrak{e}_\infty^l$ for α, β, δ, taking different values when $\lambda = 0.8$, $k = 2$, and $M = 8$. At this stage, the optimal parameters obtained through the 3D ALA are determined to be $\alpha = 0.333202425711421$, $\beta = 0.500151850052752$, and $\delta = 0.398933989676722$. From the optimization results, we choose the approximations as $\alpha = 1/3$, $\beta = 0.5$, and $\delta = 0.4$. As shown in Table 1, the 3D ALA demonstrates superior numerical accuracy. Table 2 compares the $\mathfrak{e}_2$ and $\mathfrak{e}_\infty$ with the method in [17] under fewer approximation scales when $\lambda = 0.6$, $\alpha = 0.5$, $\beta = 0.5$, and $\delta = 0.3$. The data indicates that the proposed method has higher numerical accuracy and computational efficiency than other frameworks. Under the same approximation scale, Tables 3 and 4 compare $\mathfrak{e}_2$ and $\mathfrak{e}_\infty$ when $(\lambda, k, M) = (0.4, 2, 8)$, and $(\lambda, k, M) = (0.2, 1, 8)$, respectively. Specifically, Table 4 takes the approximate values of optimal parameters obtained by ALA optimization. Figure 2 illustrates the errors and the evolution of parameters α, β and δ with respect to iteration number. As can be observed from this figure, the ALA typically requires only 5 to 10 iterations to stabilize the parameters, with the error curve rapidly decreasing and converging, indicating that the algorithm has a fast convergence rate. Figure 3 displays the slice plots of absolute errors of $x = y = t = 0.3, 0.7$ when the parameters α, β, δ, λ, k, and M take different values.

Table 1. Errors for different (α,β,δ) when $\lambda = 0.8$, $k = 2$, and $M = 8$ in Example 1.

	$\alpha = 0.333202425711421 \approx \frac{1}{3} \to \alpha = \frac{1}{3}$ $\beta = 0.500151850052752 \approx 0.5 \to \beta = 0.5$ $\delta = 0.398933989676722 \approx 0.4 \to \delta = 0.4$		$\alpha = 0.45$ $\beta = 0.5$ $\delta = 0.55$		$\alpha = 1$ $\beta = 1$ $\delta = 1$	
t_l	$\mathfrak{e}_2^l$	$\mathfrak{e}_\infty^l$	$\mathfrak{e}_2^l$	$\mathfrak{e}_\infty^l$	$\mathfrak{e}_2^l$	$\mathfrak{e}_\infty^l$
0.1	7.1281e-15	1.6198e-14	4.6943e-11	9.5591e-11	1.6798e-08	2.9790e-08
0.2	2.4221e-14	4.9161e-14	5.6707e-11	1.2487e-10	5.4075e-09	9.5261e-09
0.3	1.4277e-13	2.6553e-13	1.7314e-09	3.9822e-09	2.1033e-09	3.7344e-09
0.4	2.9159e-13	5.4225e-13	3.3061e-10	6.4979e-10	1.4722e-09	3.1629e-09
0.5	3.8272e-13	7.4596e-13	1.6246e-10	5.5489e-10	1.7226e-08	4.2343e-08
0.6	4.6328e-13	9.0433e-13	3.0093e-10	9.3727e-10	3.0613e-09	5.5513e-09
0.7	6.0206e-13	1.1228e-12	4.3924e-10	1.3036e-09	1.0470e-09	1.8896e-09
0.8	7.3197e-13	1.4646e-12	5.8345e-10	1.7002e-09	4.4985e-10	8.2229e-10
0.9	8.3583e-13	1.7867e-12	7.3761e-10	2.1192e-09	2.4815e-10	4.9597e-10
CPU(s)	15.721586		0.147582		0.158279	

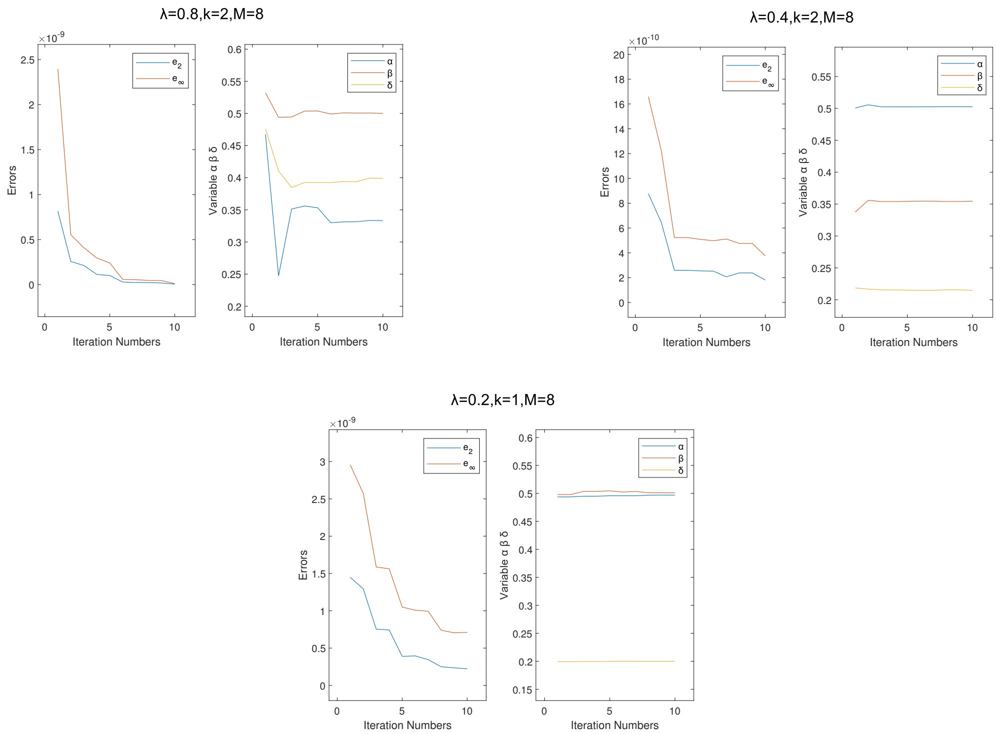

Figure 2. Errors and parameters α, β, δ regarding the iteration numbers for Example 1.

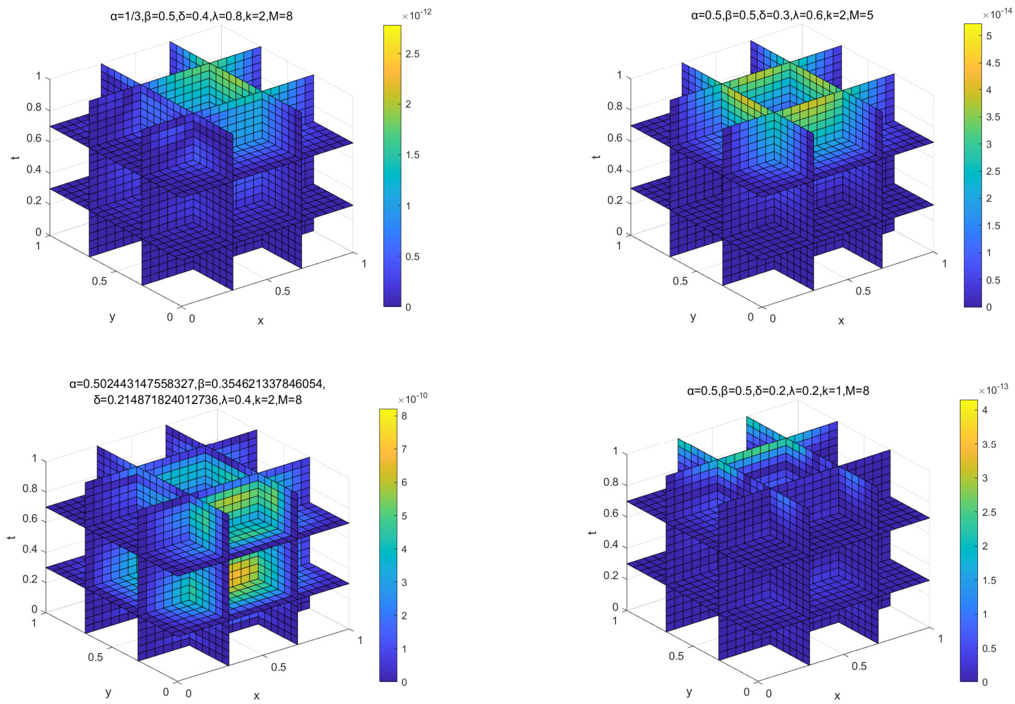

Figure 3. Absolute errors for Example 1.

Table 2. Errors for different (k, M) when $(\alpha,\beta,\delta) = (0.5, 0.5, 0.3)$ and $\lambda = 0.6$ in Example 1.

	Proposed method				Method in [17]			
(k, M)	Scale	e_2	e_∞	CPU(s)	Scale	e_2	e_∞	CPU(s)
(1,5)	$(2^0 \times 5)^3$	1.1224e-15	2.1129e-15	0.014941	$(2^3)^3$	1.7455e-04	3.3713e-04	
(1,6)	$(2^0 \times 6)^3$	3.5553e-15	6.5642e-15	0.021641	$(2^4)^3$	4.3664e-05	8.4106e-05	
(1,7)	$(2^0 \times 7)^3$	1.5794e-14	2.8847e-14	0.033796	$(2^5)^3$	1.0910e-05	2.1001e-05	
(2,5)	$(2^1 \times 5)^3$	2.2946e-14	4.3113e-14	0.03771	$(2^6)^3$	2.7242e-06	5.2422e-06	1039.2031
(2,6)	$(2^1 \times 6)^3$	1.6588e-14	3.2072e-14	0.059576	$(2^7)^3$	6.7969e-07	1.3076e-06	
(2,7)	$(2^1 \times 7)^3$	2.4019e-13	4.4486e-13	0.099327	$(2^8)^3$	1.6940e-07	3.2580e-07	
(2,8)	$(2^1 \times 8)^3$	2.5221e-13	4.7483e-13	0.155852	$(2^9)^3$	4.2153e-08	8.1033e-08	

Table 3. Errors for different (α,β,δ) when $\lambda = 0.4$, $k = 2$, and $M = 8$ in Example 1.

	Proposed method				Method in [17]
Errors	$\alpha = 0.502443147558327$ $\beta = 0.354621337846054$ $\delta = 0.214871824012736$	$\alpha = 0.75$ $\beta = 0.75$ $\delta = 0.3$	$\alpha = 0.65$ $\beta = 0.5$ $\delta = 0.25$	$\alpha = 1$ $\beta = 1$ $\delta = 1$	Scale $(2^4)^3$
e_2	1.8088e-10	4.0842e-08	1.2177e-08	1.6356e-07	4.6533e-05
e_∞	3.7485e-10	1.0733e-07	2.8425e-08	4.0790e-07	8.9758e-05
CPU(s)	16.95584	0.158793	0.153761	0.152508	-

Table 4. Errors for different (α,β,δ) when $\lambda = 0.2$, $k = 1$, and $M = 8$ in Example 1.

	Proposed method				Method in [17]
Errors	$\alpha = 0.496960032150132 \approx 0.5 \to \alpha = 0.5$ $\beta = 0.501021196931196 \approx 0.5 \to \beta = 0.5$ $\delta = 0.200088822556961 \approx 0.2 \to \delta = 0.2$	$\alpha = 0.95$ $\beta = 0.65$ $\delta = 0.55$	$\alpha = 0.45$ $\beta = 0.75$ $\delta = 0.2$	$\alpha = 1$ $\beta = 1$ $\delta = 1$	Scale $(2^3)^3$
e_2	7.1855e-14	6.7031e-08	1.5498e-08	4.3214e-07	1.9758e-04
e_∞	1.6657e-13	1.7038e-07	4.4931e-08	6.7583e-07	3.8283e-04
CPU(s)	5.361596	0.053533	0.052192	0.054127	-

Example 2. Consider the (2+1)D time−fractional mobile/immobile equation

$$\frac{\partial^\lambda u(x,y,t)}{\partial t^\lambda} + \frac{\partial u(x,y,t)}{\partial t} = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} + f(x,y,t),\ \ (x,y,t) \in (0,1)^2 \times (0,1],$$

where

$$\begin{aligned} f(x,y,t) = {} & (3t^2 + \frac{6t^{3-\lambda}}{\Gamma(4-\lambda)})(x-x^2)(1-\cos(y))(1-y) + 2(1+t^3)(1-\cos(y))(1-y) \\ & -(1+t^3)(x-x^2)\left(\cos(y)(1-y) - 2\sin(y)\right), \end{aligned}$$

with the initial boundary conditions

$$u(x,y,0) = (x-x^2)(1-\cos(y))(1-y),\ \ u(0,y,t) = u(1,y,t) = u(x,0,t) = u(x,1,t) = 0.$$

The exact solution is $u(x,y,t) = (1+t^3)(x-x^2)(1-\cos(y))(1-y)$.

Table 5 presents the e_2^l and e_∞^l for different values of α, β, δ when $\lambda = 0.4$, $k = 2$, and $M = 7$. In this case, the optimal parameters obtained through the 3D ALA are $\alpha = 0.700149874114528$, $\beta = 0.999999994488757$, and $\delta = 0.333848214897647$. According to the optimization results, the selected approximation parameters are $\alpha = 0.7$, $\beta = 1$, and $\delta = 1/3$. The data demonstrates that 3D ALA significantly enhances the numerical results. With fewer approximation scales, Table 6 compares the e_∞ with different values of λ as $(\alpha,\beta,\delta) = (0.5, 0.5, 0.5)$, $k = 3$, and $M = 8$. When $\lambda = 0.4$, $k = 2$, and $M = 7$, Figure 4 illustrates the errors and the evolution of parameters α,β, and δ regarding the iteration numbers and slice plots of absolute errors when $\alpha = 0.7, \beta = 1, \delta = 1/3$, and $x = y = t = 0.3, 0.7$.

Table 5. Errors for different (α,β,δ) when $\lambda = 0.4$, $k = 2$, and $M = 7$ in Example 2.

	$\alpha = 0.700149874114528 \approx 0.7 \to \alpha = 0.7$ $\beta = 0.999999994488757 \approx 1 \to \beta = 1$ $\delta = 0.333848214897647 \approx \frac{1}{3} \to \delta = \frac{1}{3}$		$\alpha = 0.5$ $\beta = \frac{1}{3}$ $\delta = 0.25$		$\alpha = 0.25$ $\beta = 0.35$ $\delta = 0.45$	
t_l	e_2^l	e_∞^l	e_2^l	e_∞^l	e_2^l	e_∞^l
0.1	2.8460e-16	7.9277e-16	4.2875e-10	8.7490e-10	1.2358e-11	2.2825e-11
0.2	2.4090e-15	6.7966e-15	4.8484e-10	9.0478e-10	1.0925e-10	2.5118e-10
0.3	9.2296e-15	2.5334e-14	3.5187e-10	6.3762e-10	1.3340e-10	3.8486e-10
0.4	2.4573e-14	6.5555e-14	2.5554e-09	4.3526e-09	3.6793e-10	8.2778e-10
0.5	5.1673e-14	1.3543e-13	2.4251e-09	3.9450e-09	6.4407e-10	1.7575e-09
0.6	9.3419e-14	2.4232e-13	2.0370e-09	5.6511e-09	1.1266e-09	3.2474e-09
0.7	1.5293e-13	3.9403e-13	7.8240e-09	1.4110e-08	1.8016e-09	5.2010e-09
0.8	2.3358e-13	5.9882e-13	2.3077e-08	4.2834e-08	2.6687e-09	7.5837e-09
0.9	3.3885e-13	8.6513e-13	9.0423e-09	2.3168e-08	3.9029e-09	1.0461e-08
CPU(s)	11.176786		0.101422		0.098303	

Table 6. e_∞ for different λ when $(\alpha,\beta,\delta) = (0.5, 0.5, 0.5)$, $k = 3$, and $M = 8$ in Example 2.

	Scale	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.7$	$\lambda = 0.9$
Method in [18]	$40^2 \times 1000$	3.5257e-07	1.6387e-06	2.6366e-06	3.3355e-06	3.7368e-06
Proposed method	$(2^2 \times 8)^3$	2.6510e-12	3.0014e-12	6.9066e-12	6.8256e-12	3.1052e-12
CPU(s)	-	0.815055	0.813864	0.811458	0.809454	0.814508

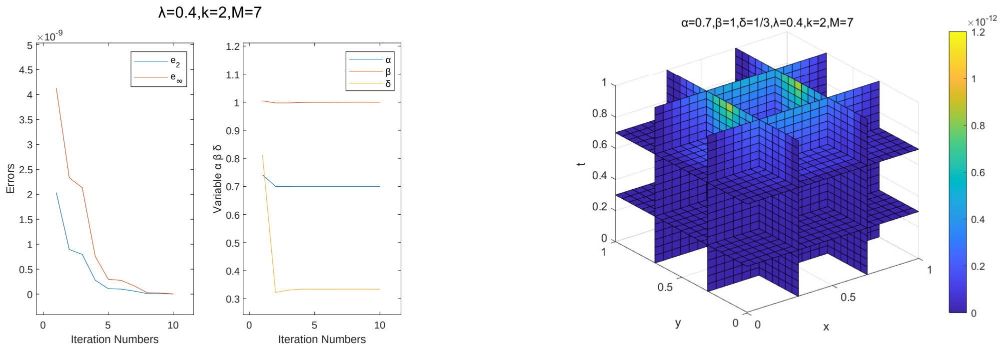

Figure 4. Errors and parameters α, β, δ regarding the iteration numbers and absolute errors for Example 2.

Example 3. Consider the (2+1)D time−fractional mobile/immobile equation

$$\frac{\partial^\lambda u(x,y,t)}{\partial t^\lambda} + \frac{\partial u(x,y,t)}{\partial t} = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} + g(u) + f(x,y,t),\ \ (x,y,t) \in (0,1)^2 \times (0,1],$$

where $g(u) = u - u^3$, and

$$\begin{aligned} f(x,y,t) = &\left((2+\lambda)t^{1+\lambda}\left(1 + \frac{\Gamma(2+\lambda)}{2}t^{1-\lambda}\right) + 2\pi^2(t^{2+\lambda}+1)\right)\sin(\pi x)\sin(\pi y) \\ &+ \left((t^{2+\lambda}+1)\sin(\pi x)\sin(\pi y)\right)^3 - (t^{2+\lambda}+1)\sin(\pi x)\sin(\pi y), \end{aligned}$$

subject to the initial boundary conditions

$$u(x,y,0) = \sin(\pi x)\sin(\pi y),\ \ u(0,y,t) = u(1,y,t) = u(x,0,t) = u(x,1,t) = 0.$$

The exact solution is $u(x,y,t) = (t^{2+\lambda}+1)\sin(\pi x)\sin(\pi y)$.

Table 7 shows the values of e_2^l and e_∞^l corresponding to different values of α, β, and δ under the condition of $\lambda = 0.5$, $k = 1$, and $M = 9$. Given this condition, the optimal parameters obtained via the 3D ALA are $\alpha = 0.874644773755138$, $\beta = 1.00173715273155$, and $\delta = 0.373728117226182$. Based on these optimal values, we select the approximations as $\alpha = 7/8$ and $\beta = 1$. The data demonstrates that the 3D ALA effectively improves the numerical results. With fewer approximation scales, Table 8 compares the e_∞ and CPU time(s) with different values of λ for $(\alpha,\beta,\delta) = (0.5, 0.95, 1/3)$, $k = 2$, and $M = 8$. When $\lambda = 0.5$, $k = 1$, and $M = 9$, Figure 5 depicts the errors and the variations of parameters α, β, and δ regarding the iteration numbers and gives the slice plots of absolute errors when $\alpha = 7/8$, $\beta = 1$, $\delta = 0.373728117226182$, and $x = y = t = 0.3, 0.7$.

Table 7. Errors for different (α, β, δ) when $\lambda = 0.5$, $k = 1$, and $M = 9$ in Example 3.

	$\alpha = 0.874644773755138 \approx \frac{7}{8} \to \alpha = \frac{7}{8}$, $\beta = 1.00173715273155 \approx 1 \to \beta = 1$, $\delta = 0.373728117226182$		$\alpha = 0.5$, $\beta = 0.85$, $\delta = 0.35$		$\alpha = 0.55$, $\beta = 0.5$, $\delta = 0.45$	
t_l	e_2^l	e_∞^l	e_2^l	e_∞^l	e_2^l	e_∞^l
0.1	3.0941e-10	8.9213e-10	1.3996e-08	5.0866e-08	1.4927e-08	5.1063e-08
0.2	1.7643e-09	4.2973e-09	8.4949e-08	3.1631e-07	9.1031e-08	3.1344e-07
0.3	4.9970e-09	1.0956e-08	2.4013e-07	9.0319e-07	2.5685e-07	8.9595e-07
0.4	1.0477e-08	2.1477e-08	4.9963e-07	1.8880e-06	5.3469e-07	1.8699e-06
0.5	1.8756e-08	3.8161e-08	8.8016e-07	3.3330e-06	9.4125e-07	3.3033e-06
0.6	2.9937e-08	6.1465e-08	1.3955e-06	5.2981e-06	1.4910e-06	5.2561e-06
0.7	4.3696e-08	9.0496e-08	2.0578e-06	7.8390e-06	2.2001e-06	7.7604e-06
0.8	6.0384e-08	1.2597e-07	2.8789e-06	1.0994e-05	3.0828e-06	1.0845e-05
0.9	8.2893e-08	1.7311e-07	3.8713e-06	1.4763e-05	4.1339e-06	1.4632e-05
CPU(s)	10.025517		0.099598		0.094484	

Table 8. e_∞ for different λ when $(\alpha, \beta, \delta) = (0.5, 0.95, 1/3)$, $k = 2$, and $M = 8$ in Example 3.

	Proposed method		Method in [19]		
λ	e_∞	CPU(s)	e_∞	CPU(s)	Scale
0.4	7.1281e-07	0.375854	3.8159e-05	3.73	
0.6	6.9121e-07	0.357433	2.8159e-04	3.54	
0.05	7.4936e-07	0.378612	3.7103e-05	3.54	16^4
0.95	6.5043e-07	0.374476	1.9853e-05	3.49	

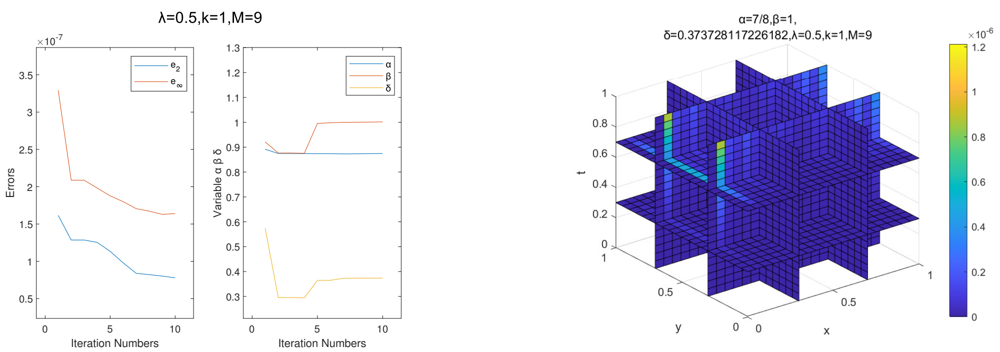

Figure 5. Errors and parameters α, β, δ regarding the iteration numbers and absolute errors for Example 3.

Example 4. Consider the 3D space−fractional Poisson equation

$$\frac{\partial^{1.75} u(x,y,z)}{\partial x^{1.75}} + \frac{\partial^{1.5} u(x,y,z)}{\partial y^{1.5}} + \frac{\partial^{1.25} u(x,y,z)}{\partial z^{1.25}} = f(x,y,z), \ \ (x,y,z) \in (0,1]^3,$$

where

$$g(x,y,z) = -\frac{2x^{0.25}}{\Gamma(1.25)}(2y-y^2)(2z-z^2) - \frac{2y^{0.5}}{\Gamma(1.5)}(2x-x^2)(2z-z^2) - \frac{2z^{0.75}}{\Gamma(1.75)}(2x-x^2)(2y-y^2),$$

subject to the initial boundary conditions

$$u(0,y,z) = u(x,0,z) = u(x,y,0) = \left.\frac{\partial u(x,y,z)}{\partial x}\right|_{x=1} = \left.\frac{\partial u(x,y,z)}{\partial y}\right|_{y=1} = \left.\frac{\partial u(x,y,z)}{\partial z}\right|_{z=1} = 0.$$

The exact solution is $u(x,y,z) = (2x-x^2)(2y-y^2)(2z-z^2)$.

In this example, we use the following approximation approach:

$$\frac{\partial^{1.75+1.5+1.25}u(x,y,z)}{\partial z^{1.25}\partial y^{1.5}\partial x^{1.75}} = \left(\mathbf{\Psi}^{\alpha}_{k,M}(x)\right)^{\top}\mathbf{D}\left(\mathbf{\Psi}^{\gamma}_{k,M}(z)\otimes\mathbf{\Psi}^{\beta}_{k,M}(y)\right).$$

Below, we provide the data analysis. When $k = 2$ and $M = 8$, Table 9 presents the numerical solutions, absolute errors, and the parameters α,β,γ optimized by the 3D ALA. It also makes a comparison with the method in [24]. The results demonstrate the superiority of the proposed method. When $k = 2$ and $M = 8$, Figure 6 depicts the errors and the variations of parameters α, β, and γ regarding the iteration numbers and shows the slice plots of absolute errors when $\alpha = 0.249770756762094$, $\beta = 0.255413806021099$, $\gamma = 0.246286028187843$, and $x = y = z = 0.3, 0.7$.

Table 9. Numerical effect for different (x,y,z) when $k = 2$, and $M = 8$ in Example 4.

		Proposed method $\alpha = 0.249770756762094$ $\beta = 0.255413806021099$ $\gamma = 0.246286028187843$		Method in [24] Scale 8^3	Scale 16^3	Scale 32^3
(x,y,z)	Exact solution	Numerical solution	Absolute error	Numerical solution	Numerical solution	Numerical solution
(0.1,0.1,0.1)	0.00685900	0.00685900	4.1142e-11	0.00826179	0.00712819	0.00686361
(0.2,0.2,0.2)	0.04665600	0.04665600	3.5623e-10	0.00474215	0.04668360	0.04667235
(0.3,0.3,0.3)	0.13265100	0.13265100	5.9298e-10	0.13537123	0.13269271	0.13265689
(0.4,0.4,0.4)	0.26214400	0.26214400	3.3800e-10	0.26618289	0.26219217	0.26214938
(0.5,0.5,0.5)	0.42187500	0.42187500	2.3310e-10	0.42728151	0.42196216	0.42188594
(0.6,0.6,0.6)	0.59270400	0.59270400	9.3710e-10	0.59821987	0.58276251	0.59271328
(0.7,0.7,0.7)	0.75357100	0.75357100	1.6299e-09	0.75927190	0.75364612	0.75357819
(0.8,0.8,0.8)	0.88473600	0.88473600	2.3952e-09	0.89128921	0.88480127	0.88474026
(0.9,0.9,0.9)	0.97029900	0.97029900	2.8121e-09	0.97823619	0.97037265	0.97030106
CPU(s)	-	16.176567		-	-	-

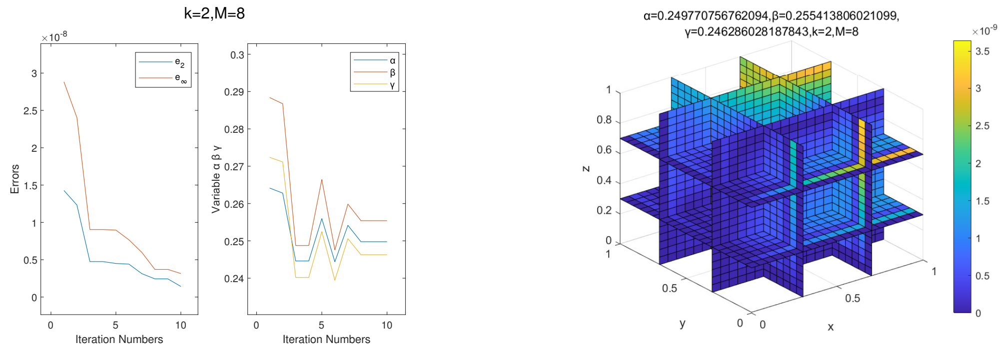

Figure 6. Errors α, β, γ regarding the iteration numbers and absolute errors for Example 4.

6. Conclusions

This study has innovatively developed a numerical algorithm that integrates 3D eighth−kind fractional Chebyshev wavelets with the ALA, which has been applied to solve (2+1)D time-fractional mobile/immobile equations. First, we established the theoretical construction of the model: Based on the eighth−kind Chebyshev polynomials, we established the 3D eighth−kind fractional Chebyshev wavelets and analyzed their mathematical properties, including function approximation and convergence analysis. Through the multivariate Caputo fractional Taylor formula established in this work, the error bounds of 3D eighth−kind fractional Chebyshev wavelets approximation were derived. Second, we set up algorithmic implementation: A complete algorithmic framework was designed for solving (2+1)D time−fractional mobile/immobile equations, which innovatively incorporates the ALA technique. Third, we established numerical validation: Through several numerical examples and comparative analyses with existing findings, not only were the high−precision characteristics of the proposed algorithm numerically verified, but the significant improvement in computational efficiency attributable to the ALA was also demonstrated. In particular, by solving the 3D space−fractional Poisson equation, the algorithm's applicability to broader problems was confirmed. Fourth, we provided extended applications: Based on the theoretical and technical foundations established in this study, the proposed method can be potentially extended to the numerical solution of (3+1)D time−fractional differential equations as well as 4D fractional differential equations, demonstrating the method's promising extensibility.

The research findings of this paper will not only enrich the theoretical framework of fractional wavelets, promoting the in−depth development of wavelet analysis into the fractional−order domain; it will also provide an efficient and scalable numerical solution for complex fractional−order systems in engineering science from an application perspective, thereby significantly enhancing their practical solving capability and engineering application value.

Author contributions

Fengying Zhou: Conceptualization, Methodology, Writing-original draft, Visualization, Supervision; Jiakun Zhang: Methodology, Software, Validation, Writing-review and editing, Visualization. All authors have read and agreed to the published version of the manuscript.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Jiangxi Science and Technology Normal University Doctoral Research Initiation Fund (Grant number [2023BSQD23]).

Conflict of interest

The authors declare that they have no conflicts of interest.

References

1. A. A. Kilbas, H. M. Srivastava, J. J. Trujillo, *Theory and applications of fractional differential equations*, Elsevier, 2006.
2. Y. C. Keluskar, N. G. Singhaniya, V. A. Vyawahare, C. S. Jage, P. Patil, G. Espinosa-Paredes, Solution of nonlinear fractional-order models of nuclear reactor with parallel computing: Implementation on GPU platform, *Ann. Nucl. Energy*, **195** (2024), 110134. https://doi.org/10.1016/j.anucene.2023.110134
3. D. Amilo, K. Sadri, B. Kaymakamzade, E. Hincal, A mathematical model with fractional-order dynamics for the combined treatment of metastatic colorectal cancer, *Commun. Nonlinear Sci. Numer. Simul.*, **130** (2024), 107756. https://doi.org/10.1016/j.cnsns.2023.107756
4. W. J. Li, Y. N. Pang, Application of Adomian decomposition method to nonlinear systems, *Adv. Difference Equ.*, **2020** (2020), 67. https://doi.org/10.1186/s13662-020-2529-y
5. X. Zhang, H. Wang, Z. Luo, L. Wei, A high-accuracy compact finite difference scheme for time-fractional diffusion equations, *Revista de la Unión Matemática Argentina*, **24** (2025), 1952–1968. https://doi.org/10.33044/revuma.4665
6. J. H. He, M. L. Jiao, K. A. Gepreel, Y. Khan, Homotopy perturbation method for strongly nonlinear oscillators, *Math. Comput. Simul.*, **204** (2023), 243–258. https://doi.org/10.1016/j.matcom.2022.08.005
7. M. Adel, M. M. Khader, S. Algelany, High-dimensional Chaotic Lorenz system: Numerical treatment using Changhee polynomials of the appell type, *Fractal Fract.*, **7** (2023), 398. https://doi.org/10.3390/fractalfract7050398
8. Y. Chen, X. Zhang, L. Wei, High precision interpolation method for mixed 2D time-space fractional convection-diffusion equation, *Int. J. Dynam. Control*, **13** (2025), 333. https://doi.org/10.1007/s40435-025-01842-z
9. J. C. Mason, D. C. Handscomb, *Chebyshev polynomials*, Chapman and Hall/CRC, 2002.
10. E. H. Doha, W. M. Abd-Elhameed, M. A. Bassuony, New algorithms for solving high even-order differential equations using third and fourth Chebyshev–Galerkin methods, *J. Comput. Phys.*, **236** (2013), 563–579. https://doi.org/10.1016/j.jcp.2012.11.009

11. W. M. Abd-Elhameed, Y. H. Youssri, A. K. Amin, A. G. Atta, Eighth-kind Chebyshev polynomials collocation algorithm for the nonlinear time-fractional generalized Kawahara equation, *Fractal Fract.*, **7** (2023), 652. https://doi.org/10.3390/fractalfract7090652
12. M. Ahmadinia, H. Afshariarjmand, M. Salehi, Numerical solution of multi-dimensional Itô Volterra integral equations by the second kind Chebyshev wavelets and parallel computing process, *Appl. Math. Comput.*, **450** (2023), 127988. https://doi.org/10.1016/j.amc.2023.127988
13. J. Shahni, R. Singh, A novel collocation approach using Chebyshev wavelets for solving fourth-order Emden-Fowler type equations, *J. Comput. Sci.*, **77** (2024), 102243. https://doi.org/10.1016/j.jocs.2024.102243
14. F. Mohammadi, C. Cattani, A generalized fractional-order Legendre wavelet Tau method for solving fractional differential equations, *J. Comput. Appl. Math.*, **339** (2018), 306–316. https://doi.org/10.1016/j.cam.2017.09.031
15. S. Behera, S. S. Ray, An efficient numerical method based on Euler wavelets for solving fractional order pantograph Volterra delay-integro-differential equations, *J. Comput. Appl. Math.*, **406** (2022), 113825. https://doi.org/10.1016/j.cam.2021.113825
16. B. Hazarika, G. Methi, R. Aggarwal, Application of generalized Haar wavelet technique on simultaneous delay differential equations, *J. Comput. Appl. Math.*, **449** (2024), 115977. https://doi.org/10.1016/j.cam.2024.115977
17. H. L. Qiao, A. J. Cheng, A fast finite difference method for 2D time fractional mobile/immobile equation with weakly singular solution, *Fractal Fract.*, **9** (2025), 204. https://doi.org/10.3390/fractalfract9040204
18. Z. H. Sheng, Y. Liu, Y. H. Li, A double-parameter shifted convolution quadrature formula and its application to fractional mobile/immobile transport equations, *Appl. Math. Lett.*, **162** (2025), 109388. https://doi.org/10.1016/j.aml.2024.109388
19. H. F. Jiang, D. Xu, W. L. Qiu, J. Zhou, An ADI compact difference scheme for the two-dimensional semilinear time-fractional mobile–immobile equation, *Comput. Appl. Math.*, **39** (2020), 287. https://doi.org/10.1007/s40314-020-01345-x
20. Y. N. Xiao, H. Cui, R. A. Khurma, P. A. Castillo, Artificial lemming algorithm: A novel bionic meta-heuristic technique for solving real-world engineering optimization problems, *Artif. Intell. Rev.*, **58** (2025), 84. https://doi.org/10.1007/s10462-024-11023-7
21. I. Podlubny, *Fractional Differential Equations*, Academic Press, 1999.
22. J. K. Zhang, F. Y. Zhou, N. W. Mao, Numerical optimization algorithm for solving time-fractional telegraph equations, *Phys. Scr.*, **100** (2025), 045237. https://doi.org/10.1088/1402-4896/adbf6f
23. Z. M. Odibat, N. T. Shawagfeh, Generalized Taylor's formula, *Appl. Math. Comput.*, **186** (2007), 286–293. https://doi.org/10.1016/j.amc.2006.07.102
24. J. Q. Xie, Z. B. Yao, R. R. Wu, X. F. Ding, J. Zhang, Block-pulse functions method for solving three-dimensional fractional Poisson type equations with Neumann boundary conditions, *Bound. Value Probl.*, **2018** (2018), 26. https://doi.org/10.1186/s13661-018-0945-7