Electronic Research Archive

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

ERA, 34(3): 1939–1956.
DOI: 10.3934/era.2026087
Received: 25 December 2025
Revised: 13 February 2026
Accepted: 14 February 2026
Published: 04 March 2026

Research article

A Fourier spectral method for normalized time-fractional diffusion equations in two and three dimensions

Ke Zhang[1], **Seungjae Lee**[2], **Xinpei Wu**[2], **Meiyun Nan**[2] **and Junseok Kim**[2,*]

[1] College of Mathematics and System Sciences, Xinjiang University, Urumqi 830046, China
[2] Department of Mathematics, Korea University, Seoul 02841, Republic of Korea

* **Correspondence:** Email: cfdkim@korea.ac.kr.

Abstract: In this paper, we propose normalized time-fractional diffusion equations in two and three spatial dimensions, where the normalization guarantees that the total memory weight remains equal to one and allows a consistent interpretation of memory effects for different fractional orders. An efficient Fourier spectral method in space combined with a finite difference approximation in time is used to solve the governing equations in both two- and three-dimensional (2D and 3D) settings. A rigorous error analysis shows that the proposed scheme achieves a temporal convergence rate of order $O(\Delta t^{2-\alpha})$, and numerical experiments confirm exponential accuracy in space. Extensive 2D and 3D numerical tests demonstrate the robustness of the method, and the numerical results show that smaller fractional orders slow down the evolution dynamics due to stronger memory effects. A comparison with the standard Caputo time-fractional diffusion model indicates that the normalized formulation provides a more uniform temporal decay behavior while preserving the essential dynamics. These computational results suggest that the proposed method offers an accurate and reliable numerical algorithm for simulating multidimensional diffusion processes with memory effects.

Keywords: normalized time-fractional derivative; high-order diffusion equations; fourier spectral method

1. Introduction

In the past decades, diffusion models with time-fractional derivatives have been extensively studied in various disciplines. By incorporating memory effects, time-fractional derivatives [1–5] generalize classical models, provide a better representation of real-world processes with non-constant rates of change, describe non-standard diffusion behavior that leads to nonlocal temporal effects, and capture long-memory systems accurately. As a result of these properties, time-fractional derivatives have been applied to partial differential equations used in various fields. The time-fractional Allen–Cahn

equation [6, 7] has been applied to the description of phase separation in binary alloys and spinodal decomposition. Recently, Sun et al. [8] developed a second-order approximation scheme for the time Caputo–Hadamard fractional derivative and applied it to the Allen–Cahn equation. They established rigorous L^2-norm error estimates together with H^1 superconvergence results and demonstrated that different fractional operators may lead to distinct evolution behaviors. This work highlights the importance of accurately treating fractional memory kernels in phase-field models.

We consider the following normalized time-fractional diffusion equation:

$$\frac{\partial^\alpha u(\mathbf{x},t)}{\partial t^\alpha} = \Delta u(\mathbf{x},t), \quad \mathbf{x}\in\Omega,\ t\geq 0, \tag{1.1}$$

where $u(\mathbf{x},t)$ is the concentration at spatial position $\mathbf{x}\in\mathbb{R}^d$ and time t for $d = 2, 3$. The normalized time-fractional derivative is defined as

$$\frac{\partial^\alpha u(\mathbf{x},t)}{\partial t^\alpha} = \frac{1-\alpha}{t^{1-\alpha}}\int_0^t \frac{\partial u(\mathbf{x},s)}{\partial s}\frac{ds}{(t-s)^\alpha},$$

where the fractional order $0<\alpha<1$, and the following identity holds:

$$\frac{1-\alpha}{t^{1-\alpha}}\int_0^t \frac{ds}{(t-s)^\alpha} = 1.$$

The diffusion equation is one of the most fundamental equations and has been widely used as a foundation for many models. Moreover, various equations based on the diffusion equation have been extensively studied in forms with normalized time-fractional derivatives. For example, Wang et al. [9] proposed a normalized time-fractional Fokker–Planck equation in one-dimensional (1D) space:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{\partial^2 u(x,t)}{\partial x^2} - P(x)\frac{\partial u(x,t)}{\partial x} - \frac{dP(x)}{dx}u(x,t) + f(x,t), \quad x\in\Omega,\ t>0,$$

which is based on the diffusion equation and includes convection and external force terms. It is applied to the modeling of stochastic processes and probability density evolution. The normalized time-fractional Fisher equation proposed in [10] is based on the classical diffusion model and takes the following form:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = D\frac{\partial^2 u(x,t)}{\partial x^2} + p(1-u)u, \quad x\in\Omega=(L_x,R_x), \quad 0<\alpha<1.$$

However, although real-world problems are inherently three-dimensional (3D), the extension of time-fractional models to higher dimensions has been difficult due to limitations in memory capacity. Therefore, studies of the diffusion equation in two and three dimensions support the analysis of these problems and allow a comprehensive description of volumetric diffusion that cannot be captured by 1D models. Although the theory is well established, for example, rigorous existence and uniqueness results have been derived for initial value problems involving the Caputo–Hadamard derivative with delay and nonlocal conditions using measures of non-compactness and fixed point theorems, and such analytical approaches have also been used in the study of time–space fractional diffusion equations [11], it is difficult to find exact analytical solutions for these equations. Therefore, numerical methods are

essential, and many methods have been proposed and developed. However, the development of efficient schemes for multidimensional problems remains an important research area. In 2D problems, several approaches have emerged. With respect to Fourier spectral approaches, Lin and Xu [12] analyzed both finite difference and spectral schemes for time-fractional diffusion equations. This approach improved numerical stability and reduced computational cost. By combining Chebyshev collocation and the Laplace transform, Shah et al. [13] introduced an effective computational method for the time-fractional diffusion–wave equation. These studies demonstrate the utility of spectral and high-order methods in resolving spatial derivatives with high accuracy.

In contrast, relatively fewer studies have focused on 3D time-fractional diffusion equations. Nevertheless, many diffusion-dominated processes in physics, engineering, and materials science are inherently 3D. For example, a truly meshless numerical method combined with high-order temporal discretization has been proposed to efficiently and accurately solve 3D variable-order time-fractional diffusion equations [14]. It significantly reduces storage and computational cost by using a fast high-order L^2 scheme accelerated with the sum-of-exponentials technique [15], while preserving stability and accuracy. An unconditionally stable alternating direction implicit (ADI) method with second-order spatial accuracy and third-order temporal accuracy increases computational efficiency and accelerates convergence through extrapolation and the PageRank algorithm [16].

The main purpose of this study is to extend the 1D normalized time-fractional diffusion model to higher-dimensional spaces, specifically 2D and 3D cases.

The remainder of this article is organized as follows. Section 2 presents the computational solutions and the derivation of the Fourier spectral method for 2D and 3D normalized time-fractional diffusion equations. Section 3 presents numerical tests to verify the convergence and accuracy of the method. Finally, Section 4 provides the conclusions.

2. Computational solutions

We present a Fourier spectral method to solve the normalized time-fractional diffusion equation. Although the method is not restricted to periodic boundary conditions, we adopt periodic boundary conditions in each spatial direction in this study, and all numerical simulations are carried out under this setting.

2.1. 2D normalized time-fractional diffusion equation

In the 2D case, we consider the computational domain $\Omega = (L_x,\ R_x) \times (L_y,\ R_y)$. The discretized domain is $\Omega_h = \{(x_a, y_b) | x_a = L_x + (a - 0.5)h,\ 1 \le a \le N_x \text{ and } y_b = L_y + (b - 0.5)h,\ 1 \le b \le N_y\}$, where $h = (R_x - L_x)/N_x$. To present the computational solution algorithm based on the Fourier spectral method in a self-contained manner, we summarize the 2D procedure. Let $u^k_{ab} = u(x_a, y_b, t_k)$, where $t_k = (k - 1)\Delta t$. The set $\{u^k_{ab} \mid 1 \le a \le N_x,\ 1 \le b \le N_y\}$ represents the given data, and we define the discrete Fourier transform as

$$\hat{u}^k_{pv} = \sum_{a=1}^{N_x} \sum_{b=1}^{N_y} u^k_{ab} e^{-i(\xi_p x_a + \eta_v y_b)}, \qquad -\frac{N_x}{2} + 1 \le p \le \frac{N_x}{2},\ -\frac{N_y}{2} + 1 \le v \le \frac{N_y}{2}, \tag{2.1}$$

where $\xi_p = 2\pi p/L_x$ and $\eta_v = 2\pi v/L_y$. The inverse discrete Fourier transform is defined as

$$u_{ab}^k = \frac{1}{N_x N_y} \sum_{p=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} \hat{u}_{pv}^k e^{i(\xi_p x_a + \eta_v y_b)}. \tag{2.2}$$

We define $u(x, y, t)$ to be a continuous extension of u_{ab}^k. The derivatives of $u(x, y, t)$ are given by

$$\begin{aligned} \frac{\partial}{\partial x}u(x,y,t) &= \frac{1}{N_x N_y} \sum_{p=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} (i\xi_p)\hat{u}(\xi_p,\eta_v,t)e^{i(\xi_p x_a+\eta_v y_b)}, \\ \frac{\partial}{\partial y}u(x,y,t) &= \frac{1}{N_x N_y} \sum_{p=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} (i\eta_v)\hat{u}(\xi_p,\eta_v,t)e^{i(\xi_p x_a+\eta_v y_b)}. \end{aligned}$$

The normalized time-fractional derivative at t_{k+1} can be approximated by

$$\begin{aligned} \frac{\partial^\alpha u(x_a,y_b,t_{k+1})}{\partial t^\alpha} &= \frac{1-\alpha}{t_{k+1}^{1-\alpha}} \sum_{r=1}^{k} \int_{t_r}^{t_{r+1}} \frac{\partial u(x_a,y_b,s)}{\partial s} \frac{ds}{(t_{k+1}-s)^\alpha} \\ &= \sum_{r=1}^{k} \frac{1-\alpha}{t_{k+1}^{1-\alpha}} \int_{t_r}^{t_{r+1}} \frac{ds}{(t_{k+1}-s)^\alpha} \frac{u_{ab}^{r+1}-u_{ab}^r}{\Delta t} + R_{\Delta t} \\ &= \sum_{r=1}^{k} \frac{(t_{k+1}-t_r)^{1-\alpha}-(t_{k+1}-t_{r+1})^{1-\alpha}}{t_{k+1}^{1-\alpha}} \frac{u_{ab}^{r+1}-u_{ab}^r}{\Delta t} + R_{\Delta t}. \end{aligned} \tag{2.3}$$

Equation (2.3) can be rewritten as follows by using the definition $t_k = (k-1)\Delta t$:

$$\begin{aligned} \sum_{r=1}^{k} \frac{(k+1-r)^{1-\alpha}-(k-r)^{1-\alpha}}{k^{1-\alpha}} \frac{u_{ab}^{r+1}-u_{ab}^r}{\Delta t} + R_{\Delta t} &= \sum_{r=1}^{k} w_r^k \frac{u_{ab}^{r+1}-u_{ab}^r}{\Delta t} + R_{\Delta t} \\ &= w_k^k \frac{u_{ab}^{k+1}-u_{ab}^k}{\Delta t} + \sum_{r=1}^{k-1} w_r^k \frac{u_{ab}^{r+1}-u_{ab}^r}{\Delta t} + R_{\Delta t}, \end{aligned}$$

where $w_r^k = [(k+1-r)^{1-\alpha} - (k-r)^{1-\alpha}]/k^{1-\alpha}$, these weights satisfy $\sum_{r=1}^{k} w_r^k = 1$, and $R_{\Delta t}$ is the truncation error.

To analyze the truncation error $R_{\Delta t}$, we consider the difference between the exact derivative and the discrete approximation. For brevity, we omit the spatial indices (x_a, y_b) in the following analysis. Following the analytical method of [12, 17], we obtain the following estimate for the truncation error:

$$|R_{\Delta t}| = \left| \frac{1-\alpha}{t_{k+1}^{1-\alpha}} \sum_{r=1}^{k} \int_{t_r}^{t_{r+1}} \left(\frac{\partial u(s)}{\partial s} - \frac{u^{r+1}-u^r}{\Delta t} \right) \frac{ds}{(t_{k+1}-s)^\alpha} \right|. \tag{2.4}$$

We use the Taylor expansion at the midpoint $t_{r+1/2} = (t_r + t_{r+1})/2$ to analyze the term inside the integral. Assuming that the solution satisfies sufficient temporal regularity, specifically $u \in C^3([\Delta t, T])$, we carry out a formal truncation error analysis to illustrate the expected temporal accuracy of the proposed

scheme. The exact derivative at any $s \in (t_r, t_{r+1})$ can be expanded. Note that since $|s - t_{r+1/2}| \leq \Delta t/2$, the quadratic remainder term is uniformly bounded by $O(\Delta t^2)$:

$$\frac{\partial u}{\partial s}(s) = \frac{\partial u}{\partial s}(t_{r+1/2}) + \frac{\partial^2 u}{\partial s^2}(t_{r+1/2})(s - t_{r+1/2}) + O(\Delta t^2).$$

Similarly, the central difference quotient approximates the derivative at the midpoint with second-order accuracy:

$$\frac{u^{r+1} - u^r}{\Delta t} = \frac{\partial u}{\partial s}(t_{r+1/2}) + O(\Delta t^2).$$

Subtracting the two expressions and noting that $s - t_{r+1/2} = \frac{1}{2}(2s - t_r - t_{r+1})$, we obtain

$$\begin{aligned} \frac{\partial u}{\partial s} - \frac{u^{r+1} - u^r}{\Delta t} &= \frac{\partial^2 u}{\partial s^2}(t_{r+1/2})(s - t_{r+1/2}) + O(\Delta t^2) \\ &= \frac{1}{2}\frac{\partial^2 u}{\partial s^2}(t_{r+1/2})(2s - t_r - t_{r+1}) + O(\Delta t^2). \end{aligned} \tag{2.5}$$

Substituting this approximation back into $|R_{\Delta t}|$ implies

$$|R_{\Delta t}| = \left| \frac{1-\alpha}{t_{k+1}^{1-\alpha}} \sum_{r=1}^{k} \int_{t_r}^{t_{r+1}} \left(\frac{1}{2}\frac{\partial^2 u}{\partial s^2}(t_{r+1/2})(2s - t_r - t_{r+1}) + O(\Delta t^2) \right) \frac{ds}{(t_{k+1} - s)^\alpha} \right|. \tag{2.6}$$

Since $u \in C^3([\Delta t, T])$, $|\partial^2 u/\partial s^2|$ is bounded on $[\Delta t, T]$. Let $M_u = \max_{t \in [\Delta t, T]} |\partial^2 u/\partial t^2|$. Factoring out this constant and noting that the $O(\Delta t^2)$ term contributes to a higher-order error after integration [12], we obtain

$$\begin{aligned} |R_{\Delta t}| &\leq \left(\frac{1-\alpha}{2t_{k+1}^{1-\alpha}} M_u \right) \left| \sum_{r=1}^{k} \int_{t_r}^{t_{r+1}} \frac{2s - t_r - t_{r+1}}{(t_{k+1} - s)^\alpha} ds \right| + O(\Delta t^2) \\ &\leq C_1 \left| \sum_{r=1}^{k} \int_{t_r}^{t_{r+1}} \frac{2s - t_r - t_{r+1}}{(t_{k+1} - s)^\alpha} ds \right| + O(\Delta t^2). \end{aligned} \tag{2.7}$$

To evaluate the integral term, we introduce the change of variable $s = t_r + \eta \Delta t$ for $\eta \in [0, 1]$. Consequently, $ds = \Delta t d\eta$, and the term $2s - t_r - t_{r+1}$ simplifies to $\Delta t(2\eta - 1)$. The denominator becomes $(t_{k+1} - s)^\alpha = \Delta t^\alpha (k + 1 - r - \eta)^\alpha$. Substituting these into the summation gives

$$\begin{aligned} \left| \sum_{r=1}^{k} \int_{t_r}^{t_{r+1}} \frac{2s - t_r - t_{r+1}}{(t_{k+1} - s)^\alpha} ds \right| &= \left| \sum_{r=1}^{k} \int_0^1 \frac{\Delta t(2\eta - 1)}{\Delta t^\alpha (k + 1 - r - \eta)^\alpha} \Delta t d\eta \right| \\ &= \Delta t^{2-\alpha} \left| \sum_{r=1}^{k} \int_0^1 \frac{2\eta - 1}{(k + 1 - r - \eta)^\alpha} d\eta \right|. \end{aligned} \tag{2.8}$$

The remaining summation of integrals converges to a finite constant for $0 < \alpha < 1$. Therefore, we conclude

$$|R_{\Delta t}| \leq C_2 \Delta t^{2-\alpha} + O(\Delta t^{2-\alpha}) \leq C \Delta t^{2-\alpha}, \tag{2.9}$$

where C is a positive constant and does not depend on Δt. Thus, the theoretical temporal convergence rate is $O(\Delta t^{2-\alpha})$. We remark that this error estimate is derived under sufficient smoothness assumptions and should be understood as a formal analysis. In practice, the numerical scheme is applied for $t \geq \Delta t$, where the solution is less affected by the initial weak singularity.

Furthermore, we have

$$\begin{aligned}\Delta u(x,y,t) &= \frac{1}{N_x N_y}\sum_{p=-N_x/2+1}^{N_x/2}\sum_{v=-N_y/2+1}^{N_y/2}\Delta\hat{u}(\xi_p,\eta_v,t)e^{i(\xi_p x_a+\eta_v y_b)}\\ &= \frac{1}{N_x N_y}\sum_{p=-N_x/2+1}^{N_x/2}\sum_{v=-N_y/2+1}^{N_y/2}-(\xi_p^2+\eta_v^2)\hat{u}(\xi_p,\eta_v,t)e^{i(\xi_p x_a+\eta_v y_b)}.\end{aligned}$$

This leads to the following numerical algorithm for the considered equation:

$$w_k^k\frac{u_{ab}^{k+1}-u_{ab}^k}{\Delta t} = \Delta u_{ab}^{k+1} - s_{ab}^k, \tag{2.10}$$

where $s_{ab}^k = \sum_{r=1}^{k-1} w_r^k(u_{ab}^{r+1}-u_{ab}^r)/\Delta t$. Therefore, Eq (2.10) can be written as

$$w_k^k\frac{\hat{u}_{pv}^{k+1}-\hat{u}_{pv}^k}{\Delta t} = -\left(\xi_p^2+\eta_v^2\right)\hat{u}_{pv}^{k+1} - \hat{s}_{pv}^k.$$

Rearranging the above equation yields

$$\hat{u}_{pv}^{k+1} = \frac{w_k^k\hat{u}_{pv}^k - \Delta t\hat{s}_{pv}^k}{w_k^k+\Delta t(\xi_p^2+\eta_v^2)}.$$

Finally, u_{ab}^{k+1} is obtained by using Eq (2.2):

$$u_{ab}^{k+1} = \frac{1}{N_x N_y}\sum_{p=-N_x/2+1}^{N_x/2}\sum_{v=-N_y/2+1}^{N_y/2}\hat{u}_{pv}^{k+1}e^{i(\xi_p x_a+\eta_v y_b)}.$$

2.2. 3D normalized time-fractional diffusion equation

We expand the 2D Fourier spectral method to 3D on the domain $\Omega = (L_x,\ R_x)\times(L_y,\ R_y)\times(L_z,\ R_z)$ to solve the normalized time-fractional diffusion equation. The discretized domain is defined as

$$\Omega_h = \{(x_a,y_b,z_c)|x_a = L_x+(a-0.5)h,\ y_b = L_y+(b-0.5)h,\ z_c = L_z+(c-0.5)h\},$$

where $1\leq a\leq N_x,\ 1\leq b\leq N_y$, and $1\leq c\leq N_z$. Let $u_{abc}^k = u(x_a,y_b,z_c,t_k)$, where $t_k = (k-1)\Delta t$. The discrete Fourier transform is defined as

$$\hat{u}_{pvr}^k = \sum_{a=1}^{N_x}\sum_{b=1}^{N_y}\sum_{c=1}^{N_z}u_{abc}^k e^{-i(\xi_p x_a+\eta_v y_b+\omega_r z_c)}, \tag{2.11}$$

where $-\frac{N_x}{2}+1 \le p \le \frac{N_x}{2}$, $-\frac{N_y}{2}+1 \le v \le \frac{N_y}{2}$, and $-\frac{N_z}{2}+1 \le r \le \frac{N_z}{2}$. Here, $\xi_p = 2\pi p/L_x$, $\eta_v = 2\pi v/L_y$, and $\omega_r = 2\pi r/L_z$. The inverse discrete Fourier transform is defined as

$$u^k_{abc} = \frac{1}{N_xN_yN_z} \sum_{p=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} \sum_{r=-N_z/2+1}^{N_z/2} \hat{u}^k_{pvr} e^{i(\xi_p x_a+\eta_v y_b+\omega_r z_c)}. \tag{2.12}$$

Analogous to the 2D case, we assume that the solution $u(x, y, z, t)$ is sufficiently smooth and represents a continuous extension of the discrete numerical solution u^k_{abc}. Under this assumption, the spatial derivatives of $u(x, y, z, t)$ can be expressed in the Fourier space as

$$\begin{aligned}
\frac{\partial}{\partial x}u(x,y,z,t) &= \frac{1}{N_xN_yN_z} \sum_{p=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} \sum_{r=-N_z/2+1}^{N_z/2} (i\xi_p)\hat{u}(\xi_p,\eta_v,\omega_r,t)e^{i(\xi_p x_a+\eta_v y_b+\omega_r z_c)},\\
\frac{\partial}{\partial y}u(x,y,z,t) &= \frac{1}{N_xN_yN_z} \sum_{p=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} \sum_{r=-N_z/2+1}^{N_z/2} (i\eta_v)\hat{u}(\xi_p,\eta_v,\omega_r,t)e^{i(\xi_p x_a+\eta_v y_b+\omega_r z_c)},\\
\frac{\partial}{\partial z}u(x,y,z,t) &= \frac{1}{N_xN_yN_z} \sum_{p=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} \sum_{r=-N_z/2+1}^{N_z/2} (i\omega_r)\hat{u}(\xi_p,\eta_v,\omega_r,t)e^{i(\xi_p x_a+\eta_v y_b+\omega_r z_c)}.
\end{aligned}$$

Consequently, the Laplacian operator in the physical space corresponds to a multiplicative coefficient in the Fourier space, namely,

$$\begin{aligned}
\Delta u(x,y,z,t) &= \frac{1}{N_xN_yN_z} \sum_{p=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} \sum_{r=-N_z/2+1}^{N_z/2} \Delta\hat{u}(\xi_p,\eta_v,\omega_r,t)e^{i(\xi_p x_a+\eta_v y_b+\omega_r z_c)}\\
&= \frac{1}{N_xN_yN_z} \sum_{p=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} \sum_{r=-N_z/2+1}^{N_z/2} -(\xi_p^2+\eta_v^2+\omega^2)\hat{u}(\xi_p,\eta_v,t)e^{i(\xi_p x_a+\eta_v y_b+\omega_r z_c)}.
\end{aligned}$$

By applying the same procedure as in the 2D case, we obtain

$$\hat{u}^{k+1}_{pvr} = \frac{w^k_k \hat{u}^k_{pvr} - \Delta t \hat{s}^k_{pvr}}{w^k_k + \Delta t(\xi_p^2+\eta_v^2+\omega_r^2)}.$$

Finally, u^{k+1}_{abc} is obtained by performing the inverse discrete Fourier transform:

$$u^{k+1}_{abc} = \frac{1}{N_xN_yN_z} \sum_{p=-N_x/2+1}^{N_x/2} \sum_{v=-N_y/2+1}^{N_y/2} \sum_{r=-N_z/2+1}^{N_z/2} \hat{u}^{k+1}_{pvr} e^{i(\xi_p x_a+\eta_v y_b+\omega_r z_c)}.$$

3. Numerical experiments

3.1. 2D convergence test

We first conduct a 2D convergence test in the domain $\Omega = (0, 10)^2$. The parameters are set to $\alpha = 0.3,\ 0.5,\ 0.7$ and $T = 0.2$. The initial condition is chosen as

$$u(x, y, 0) = \cos(4\pi x)\sin(4\pi y).$$

The L_2-norm is defined as $\|u^k\|_2 = \sqrt{\left(\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}(u_{ij}^k)^2\right)/(N_xN_y)}$. The relative error is defined by

$$Error = \frac{\|u^k - u_{ref}^k\|_2}{\|u_{ref}^k\|_2}, \tag{3.1}$$

where u_{ref}^k denotes the reference solution. Temporal errors are evaluated by successively reducing the time step by a factor of two, with $\Delta t = T/2^5$ to $\Delta t = T/2^8$. A fixed 256×256 spatial grid is used, and the reference solution u_{ref}^k is obtained with $\Delta t = T/2^{12}$. Table 1 summarizes the temporal discretization errors and the associated convergence rates for different time step sizes.

Table 1. Temporal discretization errors and convergence rates in 2D.

α	$\Delta t = 0.2/2^5$		$\Delta t = 0.2/2^6$		$\Delta t = 0.2/2^7$		$\Delta t = 0.2/2^8$
	Error	Order	Error	Order	Error	Order	Error
0.3	1.89×10^{-3}	1.58	6.30×10^{-4}	1.66	1.99×10^{-4}	1.71	6.06×10^{-5}
0.5	4.10×10^{-3}	1.43	1.52×10^{-3}	1.48	5.44×10^{-4}	1.52	1.89×10^{-4}
0.7	7.62×10^{-3}	1.27	3.15×10^{-3}	1.30	1.28×10^{-3}	1.33	5.07×10^{-4}

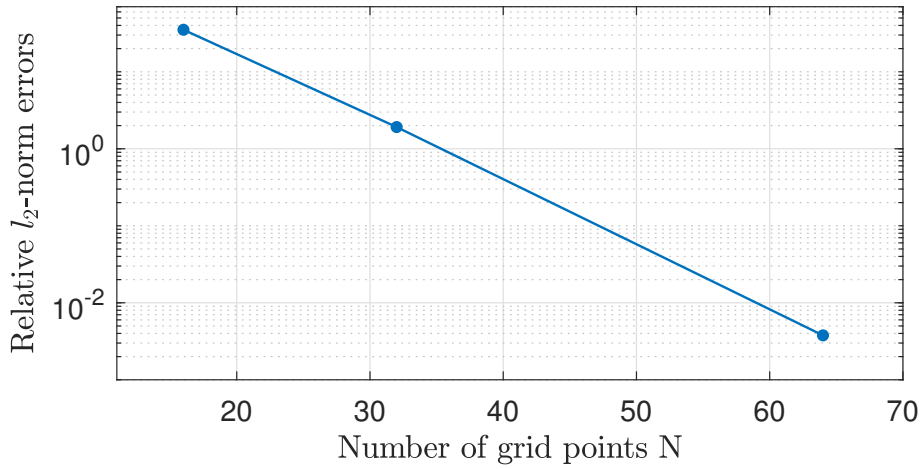


Figure 1. Exponentially decreasing spatial errors in 2D with the number of grid points $N = N_x = N_y$.

Next, the time step is fixed at $\Delta t = T/2^8$ to ensure sufficient temporal accuracy, which allows an isolated assessment of spatial discretization errors. The reference solution u_{ref}^k is computed on a $2^{10} \times 2^{10}$ grid. Furthermore, several spatial resolutions $N_x \times N_y = 2^\rho \times 2^\rho$ with $\rho = 4, 5, 6$ are tested, and the corresponding errors are reported in Figure 1. Spatial discretization is conducted using the Fourier spectral method, under which the discretization error decreases progressively as the mesh is refined.

3.2. 3D convergence test

We investigate the 3D convergence test in the domain $\Omega = (0, 1)^3$. We set the mesh size $h = 1/128$ and end time $T = 0.2$. The initial condition is given by

$$u(x, y, z, 0) = \exp(\sin(2\pi x)\cos(2\pi y)\sin(2\pi z)).$$

The L_2-norm is defined as

$$\|u^k\|_2 = \sqrt{\left(\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\sum_{l=1}^{N_z}(u_{ijl}^k)^2\right)/(N_xN_yN_z)},$$

and the relative error is given by Eq (3.1). We choose the time step sizes $T/2^2$, $T/2^3$, $T/2^4$, and $T/2^5$ to test the temporal accuracy. The reference solution is computed using the time step size $T/2^9$. The computational results are listed in Table 2.

Table 2. Temporal discretization errors and convergence rates in 3D.

α	$\Delta t = 0.2/2^2$		$\Delta t = 0.2/2^3$		$\Delta t = 0.2/2^4$		$\Delta t = 0.2/2^5$
	Error	Order	Error	Order	Error	Order	Error
0.3	3.04×10^{-4}	1.48	1.09×10^{-4}	1.55	3.71×10^{-5}	1.64	1.20×10^{-5}
0.5	4.55×10^{-4}	1.42	1.71×10^{-4}	1.43	6.33×10^{-5}	1.48	2.27×10^{-5}
0.7	5.12×10^{-4}	1.38	1.96×10^{-4}	1.31	7.90×10^{-5}	1.33	3.15×10^{-5}

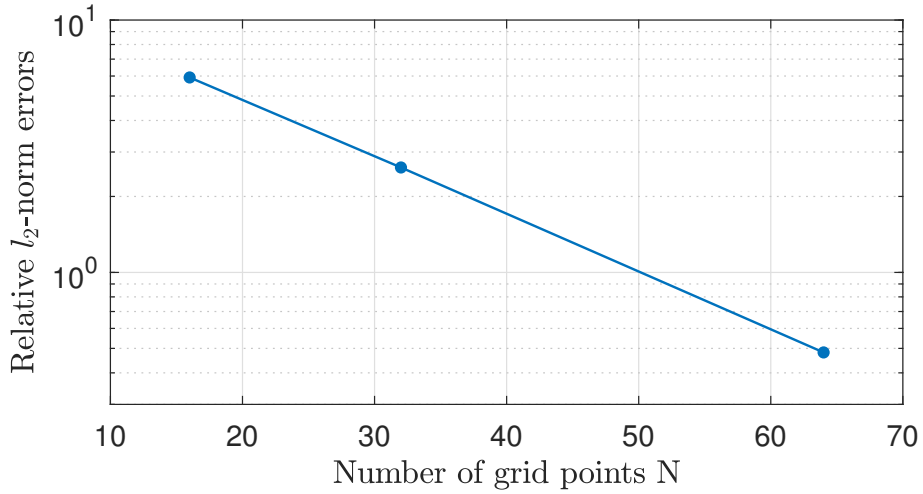


Figure 2. Exponentially decreasing spatial errors in 3D with the number of grid points $N = N_x = N_y = N_z$.

For the spatial convergence test, we consider the computational domain $(0, 10)^3$ and the initial condition as follows:

$$u(x, y, z, 0) = \sin(4\pi x)\sin(4\pi y)\sin(4\pi z).$$

We set the end time to $T = 0.005$, fix the time step size at $\Delta t = T/2^8$, $\alpha = 0.7$, and consider spatial resolutions $2^4 \times 2^4 \times 2^4$, $2^5 \times 2^5 \times 2^5$ and $2^6 \times 2^6 \times 2^6$ to examine the spatial discretization errors. The

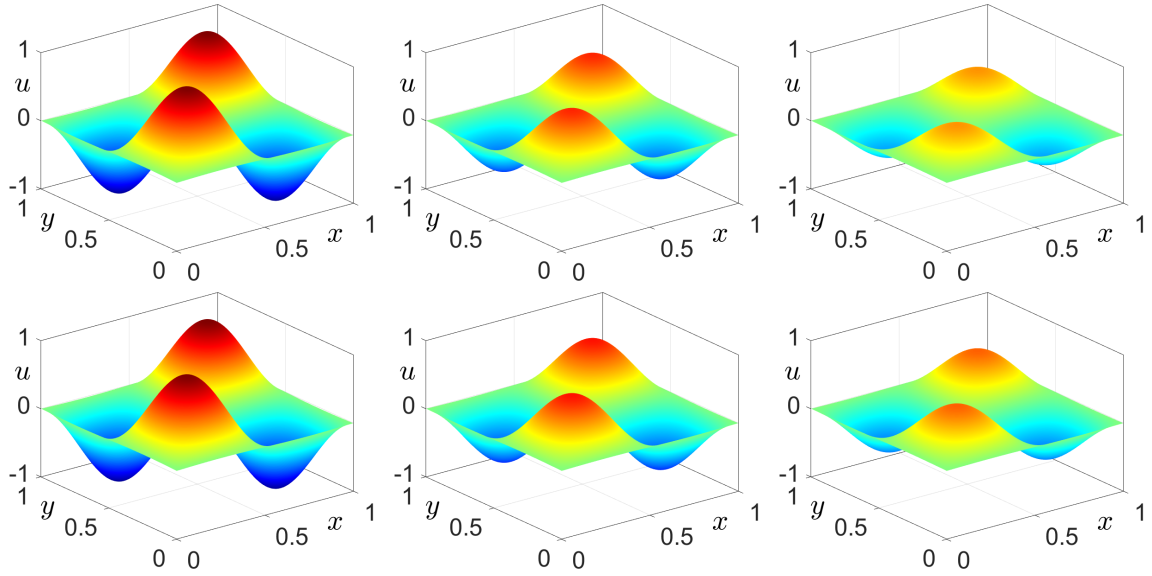
reference solution is computed on a mesh size of $2^7 \times 2^7 \times 2^7$. As shown in Figure 2, the numerical behavior is similar to that observed in the 2D case.

3.3. Effect of α on the temporal evolution

It is known that, for the 1D normalized time-fractional diffusion equation, the evolution dynamics are similar for different values of α [18]. In this section, we investigate the evolution dynamics in higher dimensions (2D and 3D) for different values of α. For the computational tests, we consider $\Omega = (0, 1)^2$, the number of grid points $N_x = N_y = 256$, the spatial step size $h = 1/N_x$, and the time step size $\Delta t = 0.0001$. The initial condition is defined as follows:

$$u(x, y, 0) = \sin(2\pi x) \sin(2\pi y).$$

Figure 3 presents the temporal evolution dynamics for $\alpha = 0.999$ and 0.001 at $t = 0$, 0.005, and 0.01. In 2D, similar evolution dynamics are observed for different values of α. However, the evolution becomes slower as α decreases. This suggests that the memory effect influences the evolution dynamics. Furthermore, we compare the temporal evolution of the amplitude height defined in Figure 4(a) for various values of α, where $h(t) = \max_{(x,y)\in\Omega} u(x, y, t)$. Figure 4(b) shows the temporal evolution of $h(t)$ for $\alpha = 0.9$, 0.7, 0.5, 0.3, and 0.1. This supports that the memory effect influences the evolution dynamics.



(a) $t = 0$ (b) $t = 0.005$ (c) $t = 0.01$

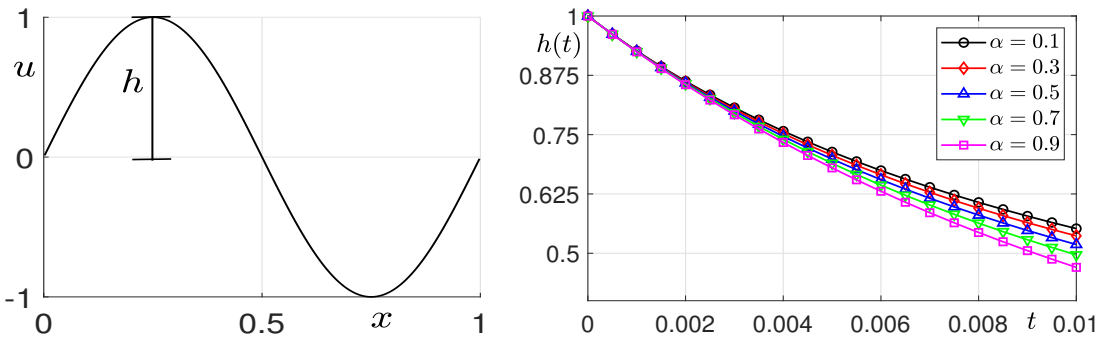
Figure 3. Temporal evolution dynamics for different values of α at $t = 0$, 0.005, and 0.01. From top to bottom rows, $\alpha = 0.999$ and $\alpha = 0.001$, respectively.

In addition, we investigate simulations with increased periodicity. The parameters are the same as those in Figure 3, and the initial condition is defined as follows:

$$u(x, y, 0) = \sin(4\pi x) \sin(4\pi y). \tag{3.2}$$

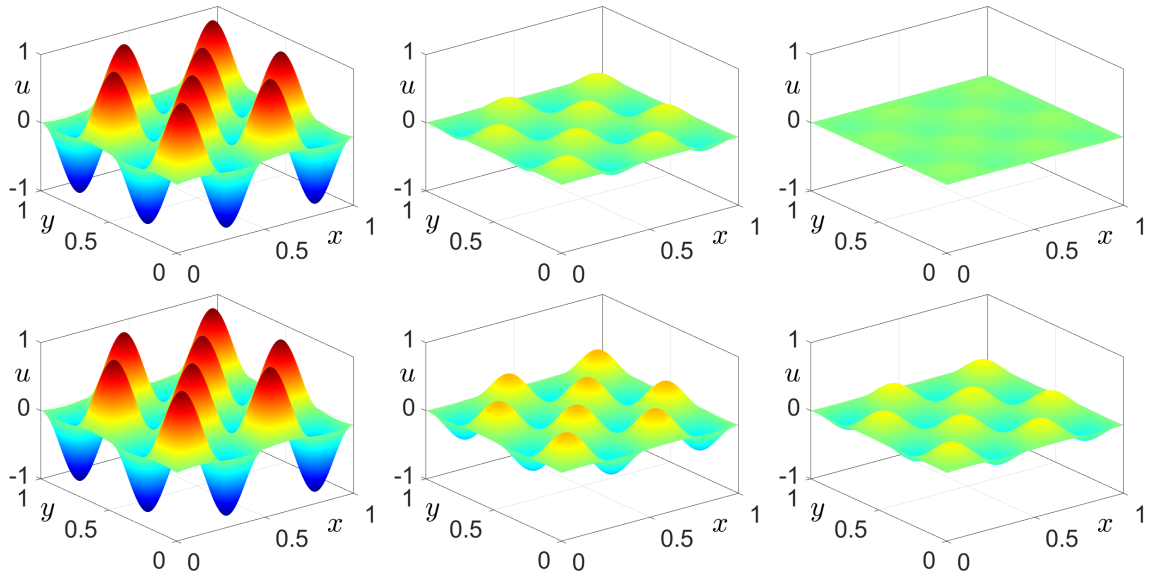
Figure 5 shows the temporal evolution of the solution to Eq (3.2) for $\alpha = 0.999$ and $\alpha = 0.001$, with snapshots taken at $t = 0$, 0.005, and 0.01. When the periodicity of the initial condition is high, the

temporal evolution becomes faster, and the reduced evolution speed due to memory effects is clearly observed.



(a) (b)

Figure 4. (a) Definition of the amplitude height h. (b) Temporal evolution of the height $h(t)$ for different values of α.



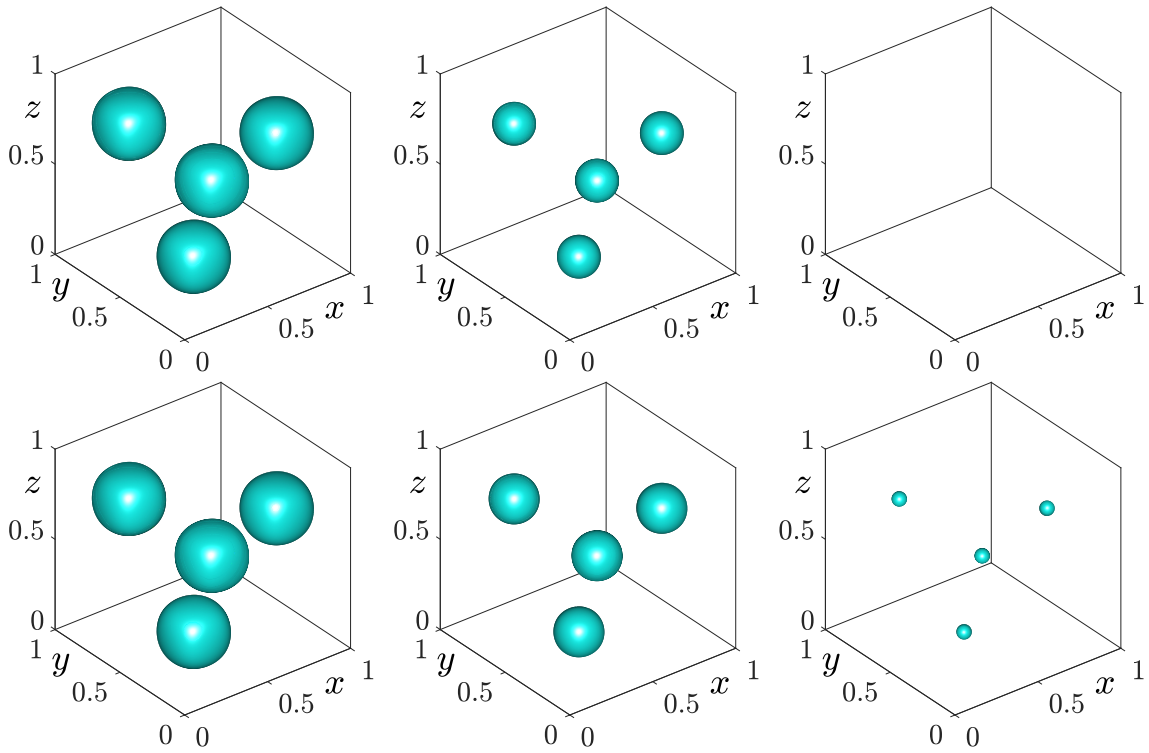
(a) $t = 0$ (b) $t = 0.005$ (c) $t = 0.01$

Figure 5. Temporal evolution dynamics with a higher-periodicity initial condition for different values of α at $t = 0$, 0.005, and 0.01. From top to bottom rows, $\alpha = 0.999$ and $\alpha = 0.001$, respectively.

In the 3D experiment, we consider $\Omega = (0, 1)^3$, the number of grid points $N_x = N_y = N_z = 200$, the spatial step size $h = 1/N_x$, and the time step size $\Delta t = 0.0001$. The initial condition is given as follows:

$$u(x, y, z, 0) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z).$$

Figure 6 shows the temporal evolution dynamics for $\alpha = 0.999$ and 0.001 at $t = 0$, 0.004, and 0.008. Similar to the 2D case, it is observed that in the 3D case, the evolution dynamics are slower as the fractional order α decreases.



(a) $t = 0$ (b) $t = 0.004$ (c) $t = 0.008$

Figure 6. Temporal evolution dynamics of 3D initial condition for different values of α at $t = 0$, 0.004, and 0.008. From top to bottom rows, $\alpha = 0.999$ and $\alpha = 0.001$, respectively.

3.4. Comparison with standard Caputo time-fractional model

In this section, we compare the proposed normalized time-fractional model with the conventional Caputo time-fractional model. The Caputo derivative operator differs from the normalized one and is defined as follows:

$$\frac{\partial^\beta u(x,t)}{\partial t^\beta} = \frac{1}{\Gamma(1-\beta)} \int_0^t \frac{\partial u(x,s)}{\partial s} \frac{ds}{(t-s)^\beta}, \quad 0 < \beta < 1, \tag{3.3}$$

where $\Gamma(z) = \int_0^\infty \tau^{z-1} e^{-\tau} d\tau$ is the gamma function and β is the fractional order of the Caputo derivative. The derivative reduces to the standard first-order derivative when $\beta = 1$. Here, we take the initial condition

$$u(x, y, 0) = \sin(0.4\pi x) \sin(0.4\pi y).$$

Numerical tests are performed on the square domain $\Omega = (0, 5)^2$ with 200×200 grid points. We take $h = 1/N_x$, $T = 3$, and $\Delta t = 0.003$ for the computations. The time evolution of the standard Caputo model with fractional orders $\beta = 0.1$ and 0.9 is shown in Figure 7. For smaller values of β, the system evolves rapidly. When β is large, the decay process is slower and more gradual. The temporal evolution of the normalized model for $\alpha = 0.1$ and 0.9 is presented in Figure 8. Although the parameter α still affects the evolution speed, the decay rate remains more uniform in time compared

with the conventional Caputo model. In particular, for $\beta = 0.1$, the solution decreases very rapidly, while as shown in Figure 8(a), when $\alpha = 0.1$, the decay behavior is more uniform in time. Although the Caputo derivative has been extensively studied and is mathematically well established, the normalized formulation provides a fair approach for comparing fractional-order effects.

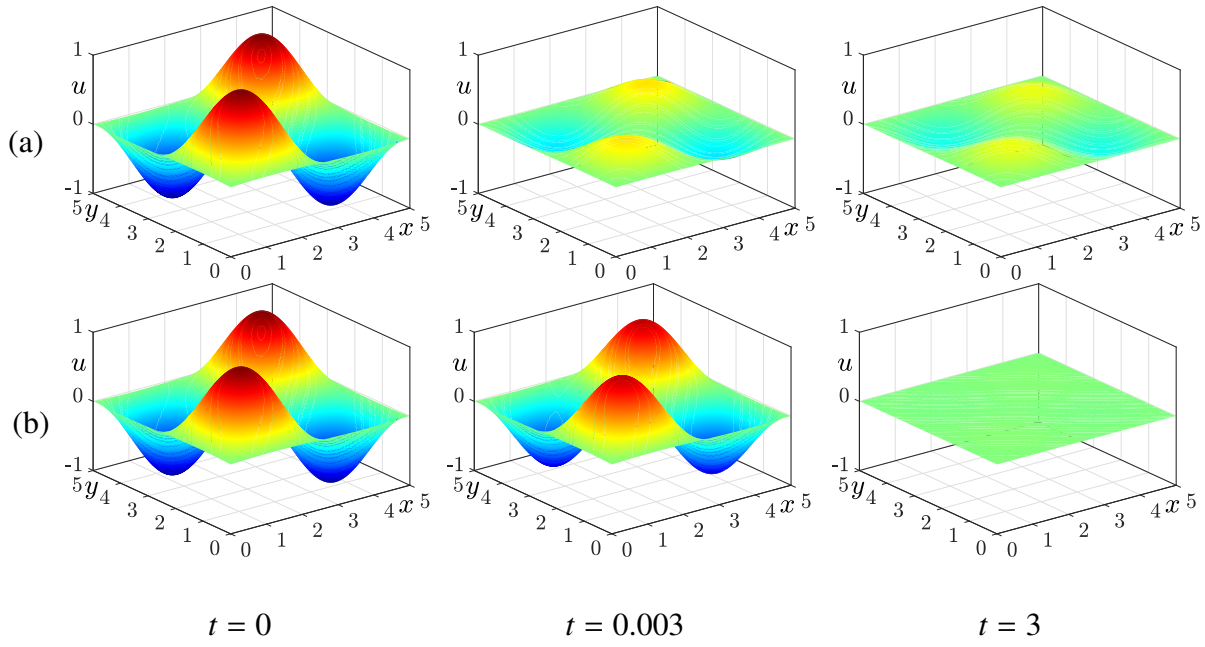


Figure 7. Time evolution for the conventional Caputo model with (a) $\beta = 0.1$ and (b) $\beta = 0.9$.

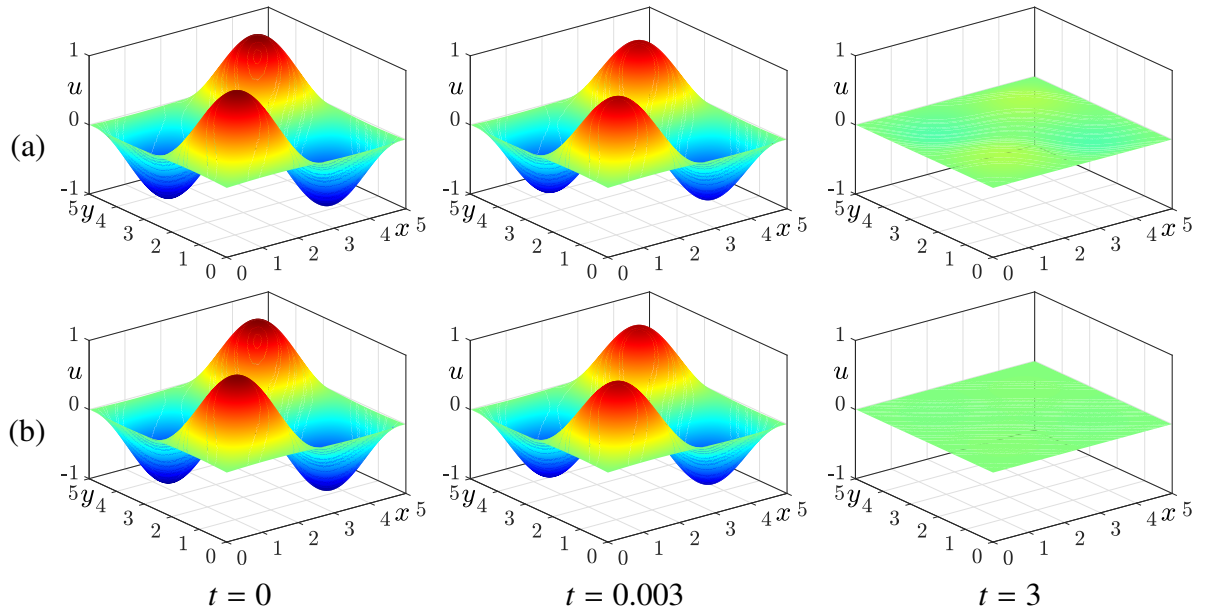


Figure 8. Time evolution for the normalized model with (a) $\alpha = 0.1$ and (b) $\alpha = 0.9$ at $t = 0$, 0.003, and 3.

3.5. 3D normalized time-fractional Fisher–Kolmogorov–Petrovsky–Piskunov (Fisher-KPP) model

The normalized time-fractional Fisher–KPP equation is given as

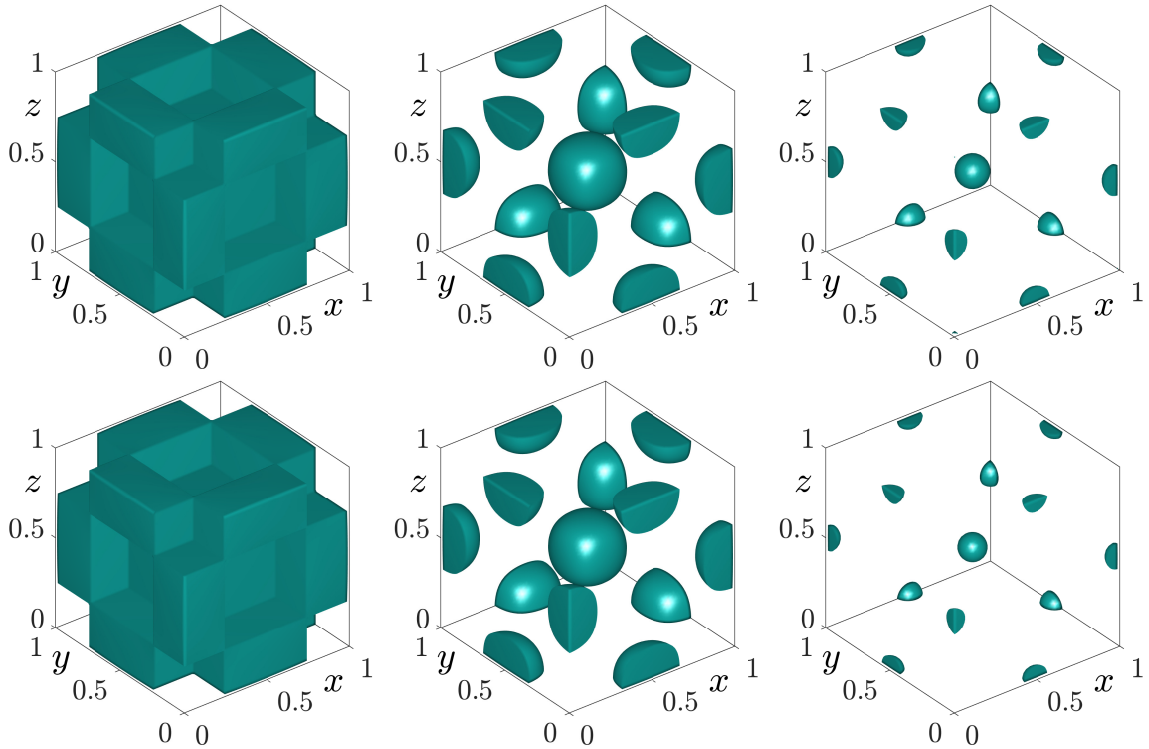
$$\frac{\partial^\alpha u(x,y,z,t)}{\partial t^\alpha} = D\Delta u(x,y,z,t) + K_{pq}u^p(x,y,z,t)[1-u(x,y,z,t)]^q, \quad (x,y,z) \in \Omega,$$

where D is the diffusion coefficient, K_{pq} is the nonlinear parameter, and p and q are the exponents that are all assumed to be positive. The nonlinear term $K_{pq}u^p(1-u)^q$ varies depending on the specific values of p and q. Specifically, $K_{pq} = \Gamma(p+q+2)/(\Gamma(p+1)\Gamma(q+1))$ and $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$.

In this section, we carry out numerical experiments with the following initial condition:

$$u(x,y,z,0) = 0.2\cos(2\pi x)\cos(2\pi y)\cos(2\pi z) + 0.5.$$

Here, the final time is set to $T = 0.12$ with time step $\Delta t = 0.003$. The spatial discretization uses $N_x = N_y = N_z = 64$, and we take $p = q = 1$ on the domain $\Omega = (0,1)\times(0,1)\times(0,1)$. Figure 9 shows the 0.5 level isosurfaces of temporal evolution dynamics for $\alpha = 0.999$ and $\alpha = 0.001$. It is observed that when the fractional order α is small, the evolution becomes faster and the interface shrinks more rapidly. Conversely, a larger α leads to slower change and more persistent structures over the same time interval, which indicates stronger memory effects in the system under identical conditions.



(a) $t = 0$ (b) $t = 0.06$ (c) $t = 0.12$

Figure 9. 0.5 level isosurfaces of temporal evolution dynamics for different values of α at $t = 0$, 0.06, and 0.12. From top to bottom rows, $\alpha = 0.999$ and $\alpha = 0.001$, respectively.

4. Conclusions

The main purpose of this study was to extend the 1D normalized time-fractional diffusion model to higher-dimensional settings, specifically 2D and 3D cases. This study used a Fourier spectral method combined with a finite difference approximation in time to solve normalized time-fractional diffusion equations in 2D and 3D spaces. The normalized formulation guaranteed that the memory kernel associated with the fractional derivative integrated to one, which allowed a consistent and fair interpretation of memory effects across different fractional orders. This property clearly distinguished the proposed model from conventional Caputo-type time-fractional derivatives and provided a reliable basis for comparative analysis. Numerical experiments confirmed a temporal convergence rate of order $O(\Delta t^{2-\alpha})$ and exponential accuracy in space.

The proposed higher-order normalized time-fractional diffusion equation can be extended to a wide class of important partial differential equations. In particular, normalized time-fractional reaction–diffusion models, as well as many other time-fractional systems, can be formulated on the basis of the present diffusion model. In general, when the fractional order is small, stability issues tend to arise, which pose additional challenges for numerical simulations. Moreover, time-fractional models inherently require high computational cost due to the nonlocal memory effect.

This computational burden can be reduced by using a short-memory strategy, which restricts the evaluation of the time-fractional derivative to a short temporal window. Such an approach proves effective when the time-fractional order is close to one, and it significantly reduces memory usage and computational cost. In addition, although the Fourier spectral method imposes periodic boundary conditions, these limitations can be alleviated by adopting alternative numerical approaches, such as multigrid methods, which allow greater flexibility in handling general boundary conditions and complex geometries.

The proposed normalized time-fractional diffusion models, together with the Fourier spectral discretization, provide accurate and reliable numerical methods for simulating multidimensional diffusion processes with memory effects. Future work will focus on extensions to nonlinear and coupled systems, variable-order normalized time-fractional derivatives, and alternative spatial discretizations that overcome periodic boundary constraints while maintaining high accuracy and efficiency. In addition, future studies will include 3D numerical experiments for the normalized time-fractional Fisher equation or other 3D nonlinear models. Furthermore, future work will focus on the development of a fully rigorous stability and convergence analysis for nonlinear extensions of the proposed scheme, including the establishment of uniform bounds for the numerical solution, as well as a rigorous error analysis accounting for the reduced temporal regularity near $t = 0$ for time-fractional diffusion equations.

Use of AI tools declaration

No Artificial Intelligence (AI) tools were used in the preparation of this work.

Acknowledgments

The first author (Ke Zhang) was supported by the Graduate Program of the China Scholarship Council (CSC) for Building High-Level Universities (No. 202507010017). The corresponding author

(J. S. Kim) received support from the Brain Korea 21 (BK 21) FOUR program, funded by the Ministry of Education. We thank the reviewers for their valuable comments and suggestions, which greatly improved this manuscript.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. Y. Zhu, Blow-up of solutions for a time fractional biharmonic equation with exponentional nonlinear memory, *Electron. Res. Arch.*, **32** (2024), 5988–6007. https://doi.org/10.3934/era.2024278
2. M. A. Ramadan, M. A. Abd El-Latif, M. Z. Alqarni, An integrated Laplace transform and accelerated Adomian decomposition approach for solving time-fractional nonlinear partial differential equations, *Electron. Res. Arch.*, **33** (2025), 4398–4434. https://doi.org/10.3934/era.2025201
3. W. M. Abd-Elhameed, M. A. Abdelkawy, N. M. A. Alsafri, A. G. Atta, Galerkin-based solution for the time-fractional diffusion-wave equation, *Electron. Res. Arch.*, **33** (2025), 5179–5206. https://doi.org/10.3934/era.2025232
4. M. A. Zaky, A linearized two-dimensional Galerkin–L1 spectral method with diagonalization for time-fractional diffusion equations with delay, *Appl. Numer. Math.*, **220** (2025), 1–12. https://doi.org/10.1016/j.apnum.2025.09.008
5. J. J. Liu, M. Yamamoto, A backward problem for the time-fractional diffusion equation, *Appl. Anal.*, **89** (2010), 1769–1788. https://doi.org/10.1080/00036810903479731
6. Z. Wang, L. Sun, The Allen–Cahn equation with a time Caputo–Hadamard derivative: mathematical and numerical analysis, *Commun. Anal. Mech.*, **15** (2023), 611–637. https://doi.org/10.3934/cam.2023031
7. J. Zhang, X. Li, W. Ma, Two-grid finite element method for the time-fractional Allen–Cahn equation with the logarithmic potential, *Math. Methods Appl. Sci.*, **48** (2025), 6654–6663. https://doi.org/10.1002/mma.10704
8. L. Sun, Z. Wang, Y. Wei, A second–order approximation scheme for Caputo–Hadamard derivative and its application in fractional Allen–Cahn equation, *Commun. Anal. Mech.*, **17** (2025), 630–661. https://doi.org/10.3934/cam.2025025
9. J. Wang, K. Chen, J. Kim, Computational analysis of a normalized time-fractional Fokker–Planck equation, *Physica A*, **665** (2025), 130500. https://doi.org/10.1016/j.physa.2025.130500
10. S. Kwak, Y. Nam, S. Kang, J. Kim, Computational analysis of a normalized time-fractional Fisher equation, *Appl. Math. Lett.*, **166** (2025), 109542. https://doi.org/10.1016/j.aml.2025.109542
11. H. T. Nguyen, V. T. Vo, Existence and uniqueness of the solution to Caputo–Hadamard differential equations with delay and nonlocal conditions, *Commun. Anal. Mech.*, **17** (2025), 725. https://doi.org/10.3934/cam.2025029
12. Y. Lin, C. Xu, Finite difference/spectral approximations for the time-fractional diffusion equation, *J. Comput. Phys.*, **225** (2007), 1533–1552. https://doi.org/10.1016/j.jcp.2007.02.001

13. F. A. Shah, Z. A. Khan, F. Azmi, N. Mlaiki, A hybrid collocation method for the approximation of 2D time-fractional diffusion-wave equation, *AIMS Math.*, **9** (2024), 27122–27149. https://doi.org/10.3934/math.20241319
14. Y. Gu, H. Sun, A meshless method for solving three-dimensional time fractional diffusion equation with variable-order derivatives, *Appl. Math. Model.*, **78** (2020), 539–549. https://doi.org/10.1016/j.apm.2019.09.055
15. H. Zhu, C. Xu, A fast high order method for the time-fractional diffusion equation, *SIAM J. Numer. Anal.*, **57** (2019), 2829–2849. https://doi.org/10.1137/18M1231225
16. M. Concezzi, R. Spigler, ADI methods for three-dimensional fractional diffusions, *Int. J. Comput. Commun.*, **16** (2022), 9–12. https://doi.org/10.46300/91013.2022.16.2
17. J. Zhang, X. Zhang, B. Yang, An approximation scheme for the time fractional convection-diffusion equation, *Appl. Math. Comput.*, **335** (2018), 305–312. https://doi.org/10.1016/j.amc.2018.04.019
18. C. Lee, Y. Nam, M. Bang, S. Ham, J. Kim, Numerical investigation of the dynamics for a normalized time-fractional diffusion equation, *AIMS Math.*, **9** (2024), 26671–26687. https://doi.org/10.3934/math.20241297

Appendix

Listing 1 is MATLAB code for the considered equation.

Listing 1. MATLAB code for the considered equation.

```
clear all;
Nx=256; Ny=256; Lx=0; Rx=1; Ly=0; Ry=1; h=(Rx-Lx)/Nx;
x=linspace(Lx+0.5*h,Rx-0.5*h,Nx); y=linspace(Ly+0.5*h,Ry-0.5*h,Ny);
dt=0.0001; T=0.01; Nt = round(T/dt); ns=round(Nt/20);
u0=sin(2*pi*x').*sin(2*pi*y);
k=2*pi/(Rx-Lx)*[0:Nx/2 -Nx/2+1:-1]; p=2*pi/(Ry-Ly)*[0:Ny/2 -Ny/2+1:-1];
k2=k.^2; p2=p.^2; [kk2, pp2]=meshgrid(k2,p2);
kp=kk2'+pp2'; alp=0.7; u = zeros(Nx,Ny,Nt+1); u(:,:,1)=u0;
for n = 1:Nt
if alp==1
w(n)=1;
F=0*u0(:,:);
else
deno = n^(1-alp);
for p = 1:n
w(p) = ((n+1-p)^(1-alp)-(n-p)^(1-alp))/deno;
end
F = 0*u0(:,:);
```

```
    if n > 1
    for p = 1:n-1
    F = F+w(p)*(u(:,:,p+1)-u(:,:,p))/dt;
    end
    end
    end
    v_hat = fft2(w(n)*u(:,:,n)-dt*F); %Converts to Fourier space
    v_hat = v_hat./(w(n)+dt*kp ); %Backwards Euler timestepping
    u(:,:,n+1) = real(ifft2(v_hat)); %Converts back to real space
    if mod(n,ns)==0
    figure(1); clf; view(3); grid on; colormap jet;
    surf(x,y,u(:,:,n+1)'); shading interp;
    axis([Lx,Rx,Ly,Ry,-1,1]); caxis([-1 1]);
    end
    end
```

AIMS Press