Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2025-0253

Vol. **41**, No. 3, pp. 921-943
March 2027

A Time-Fractional Diffusion Model with Normalization and Short Memory Effect

Junxiang Yang[1], Seokjun Ham[2], Soobin Kwak[2], Yunjae Nam[3], Zhengang Li[2], Xinpei Wu[2], Juho Ma[2] and Junseok Kim[2,*]

[1] *Faculty of Innovation Engineering, Macau University of Science and Technology, School of Computer Science and Engineering, Macau 999078, Macao Special Administrative Region of China.*

[2] *Department of Mathematics, Korea University, Seoul 02841, Republic of Korea.*

[3] *Program in Actuarial Science and Financial Engineering, Korea University, Seoul 02841, Republic of Korea.*

Received 20 September 2025; Accepted (in revised version) 16 April 2026

Abstract. We propose a time-fractional (TF) diffusion model that incorporates normalization and a short memory effect, which allows fast and efficient computations. The normalized TF diffusion equation enables fair comparison of the influence of the fractional order on evolution dynamics by using a weighting function whose total integral is uniformly equal to one across all times and fractional orders. However, simulating long-term evolution using the TF equations typically requires substantial computational memory and processing time. To resolve these issues, the proposed model applies a short memory scheme that limits the integral to a fixed time interval, which improves computational efficiency. We redefine the normalized weighting function so that its total integral over the fixed time interval is always equal to one. To highlight the improved efficiency of the short memory normalized TF diffusion equation, we perform a series of numerical experiments. We investigate dynamics with a given initial condition under different fractional orders and truncated time intervals. We apply the equation to more complex initial conditions to examine the evolution over time under varying fractional orders and memory intervals. Furthermore, to comprehensively evaluate the performance and applicability of the proposed method in higher-dimensional settings, we conduct three-dimensional (3D) numerical experiments.

AMS subject classifications: 35R11, 65M06, 65M22

Key words: Short memory, Normalized time-fractional diffusion equation, Operator splitting method.

*Corresponding author. *Email addresses:* cfdkim@korea.ac.kr (J. Kim), jxyang@must.edu.mo (J. Yang), seokjun@korea.ac.kr (S. Ham), soobin23@korea.ac.kr (S. Kwak), yjnam@korea.ac.kr (Y. Nam), 2024020532@korea.ac.kr (Z. Li), xpwu012@korea.ac.kr (X. Wu), mjh3406@korea.ac.kr (J. Ma)

 ©2027 Global-Science Press

1 Introduction

The time-fractional (TF) diffusion equation has received attention due to its capability to capture non-local temporal behavior and memory effects, which are relevant in a wide range of scientific and engineering fields involving anomalous diffusion in complex media [1,2]. In disciplines such as viscoelasticity [3,4], anomalous diffusion [5–12], and phase transition phenomena [13], the TF equation offers a more realistic description of dynamics by incorporating non-local effects in time. By introducing fractional time derivatives, this equation generalizes the classical diffusion equation, which allows for more accurate modeling of systems with memory effects. Several studies on the TF diffusion equation based on the Caputo differentiation [14–16] have been conducted. Tan [17] proposed a new and efficient two-grid finite element method (FEM) combined with a nonuniform L1 scheme to solve semilinear TF diffusion equations that have variable coefficients and initial singularities. The proposed method significantly reduced computational cost and achieved high accuracy by avoiding repeated convolutions in Caputo derivative approximation and by introducing a fast algorithm for coefficient computation, with numerical results confirming its stability and efficiency. An and Zhang [18] proposed a finite difference method (FDM) for the TF diffusion-wave model in the Caputo–Fabrizio sense. Jiang et al. [19] developed a linear FDM for the 2D nonlinear TF diffusion equation, using the L1 formula and backward approximation, and proved its stability and pointwise-in-time convergence. Nguyen and Yang [20] studied a time-space fractional diffusion equation, and established both local-in-time well-posedness in $L^p(\mathbb{R}^N)\cap L^\infty(\mathbb{R}^N)$ and global existence for small Orlicz-data. However, Lee et al. [21] highlighted that the sum of the weights in the classical TF differentiation is not constant, which prevents a fair comparison across different fractional orders, $0<\alpha<1$. They introduced a normalized TF diffusion equation in which the sum of the weights is constant and equals 1. Comprehensive studies have further established the link between these macroscopic fractional equations and microscopic stochastic processes, such as continuous time random walks (CTRW) and Langevin equations, which are essential for describing anomalous behaviors in biological and physical systems [22].

The TF diffusion equation requires significant computational memory and processing time to simulate long-term evolution. In traditional formulations, computing a fractional derivative at a given time requires integrating the entire past history of the solution. As simulation time increases, the memory requirements grow significantly, which renders long-term simulations computationally prohibitive. There has been a lot of research based on the short memory principle to address these issues and accelerate computation. Deng [23] proposed the short memory principle to address the numerical computation of fractional differential equations. By considering only recent history of the solution, the short memory principle significantly reduces storage and computational requirements. Wu et al. [24] presented a novel short memory fractional differential equation to reduce computational complexity in long-memory systems, and successfully applied it to the modeling and stability analysis of memristors and neural networks with

variable-order dynamics. Hai et al. [25] proposed an efficient fractional derivative with a short-term memory effect and established rigorous stability conditions by enabling accurate and computationally tractable modeling of dynamical systems with short memory effects. Wei et al. [26] systematically explored the short memory principle based on the Grünwald–Letnikov definition and extended it to the Riemann–Liouville and Caputo definitions. The authors also proposed variants with fixed memory length and fixed memory step size and analyzed the mathematical properties of these principles.

We consider a two-dimensional (2D) diffusion equation, which truncates the integral using a fixed time interval:

$$\frac{\partial^\alpha u(x,y,t)}{\partial t^\alpha} = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2}, \qquad (x,y)\in\Omega,\ t>0, \tag{1.1}$$

$$u(x,y,0) = u_0(x,y), \qquad (x,y)\in\Omega, \tag{1.2}$$

$$u(x,y,t) = 0, \qquad (x,y)\in\partial\Omega,\ t\geq 0, \tag{1.3}$$

where $u(x,y,t)$ is the concentration field of the diffusing quantity and

$$\frac{\partial^\alpha u(x,y,t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)}\int_{\max(0,t-M)}^{t} \frac{\partial u(x,y,s)}{\partial s}\frac{ds}{(t-s)^\alpha}, \quad 0<\alpha<1, \tag{1.4}$$

where $\Gamma(z)=\int_0^\infty \tau^{z-1}e^{-\tau}d\tau$ is the gamma function and M denotes a memory length [25]. Let a weighting function $w_\alpha^t(s)$ be defined as follows:

$$w_\alpha^t(s) := \frac{1}{\Gamma(1-\alpha)(t-s)^\alpha}. \tag{1.5}$$

Then, Eq. (1.4) can be rewritten as

$$\frac{\partial^\alpha u(x,y,t)}{\partial t^\alpha} = \int_{\max(0,t-M)}^{t} w_\alpha^t(s)\frac{\partial u(x,y,s)}{\partial s}ds, \quad 0<\alpha<1. \tag{1.6}$$

We have $W_\alpha(t)=\int_{\max(0,t-M)}^{t} w_\alpha^t(s)ds$, for all fractional orders of $0<\alpha<1$. In other words,

$$W_\alpha(t) = \int_{\max(0,t-M)}^{t} w_\alpha^t(s)ds = \frac{[t-\max(0,t-M)]^{1-\alpha}}{\Gamma(2-\alpha)}. \tag{1.7}$$

In this work, we introduce a normalized diffusion equation with short memory in 2D space, in which the integral is truncated over a fixed time interval:

$$\frac{\partial^\beta u(x,y,t)}{\partial t^\beta} = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2}, \qquad (x,y)\in\Omega,\ t>0, \tag{1.8}$$

$$u(x,y,0) = u_0(x,y), \qquad (x,y)\in\Omega, \tag{1.9}$$

$$u(x,y,t) = 0, \qquad (x,y)\in\partial\Omega,\ t\geq 0, \tag{1.10}$$

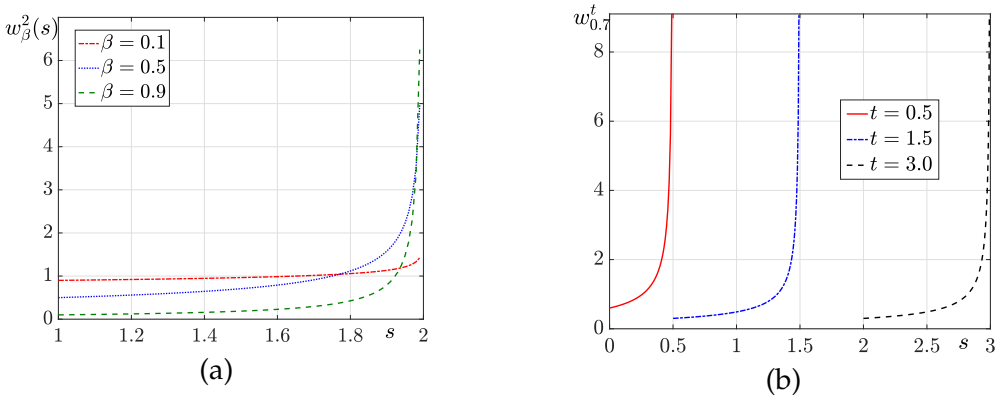


Figure 1: (a) $w_\beta^2(s)$ for different β with t=2 and M=1. (b) $w_{0.7}^t(s)$ for different t with β=0.7 and M=1. The values of t used are 0.5, 1.5, and 3.

where

$$\frac{\partial^\beta u(x,y,t)}{\partial t^\beta} = \frac{1-\beta}{[t-\max(0,t-M)]^{1-\beta}} \int_{\max(0,t-M)}^{t} \frac{\partial u(x,y,s)}{\partial s} \frac{ds}{(t-s)^\beta}, \quad 0<\beta<1, \tag{1.11}$$

where $[t-\max(0,t-M)]^{1-\beta}$ serves as the normalizing factor, which ensures that the right-hand side of Eq. (1.11) $\partial u(x,y,s)/\partial s$ when $\partial u(x,y,s)/\partial s$ is constant with respect to variable s [21,27]. That is,

$$\frac{1-\beta}{[t-\max(0,t-M)]^{1-\beta}} \int_{\max(0,t-M)}^{t} \frac{ds}{(t-s)^\beta} = 1, \quad 0<\beta<1. \tag{1.12}$$

Let a weighting function $w_\beta^t(s)$ be defined as follows:

$$w_\beta^t(s) := \frac{1-\beta}{[t-\max(0,t-M)]^{1-\beta}(t-s)^\beta}. \tag{1.13}$$

From Eq. (1.12), it can be inferred that $W_\beta(t) = \int_{\max(0,t-M)}^{t} w_\beta^t(s)ds = 1$. The weight functions $w_\beta^t(s)$ for β=0.1, 0.5, and 0.9 at t=2, for M=1, are shown in Fig. 1(a). It can be noted that $w_\beta^t(s)$ stay constant when β is near zero, while they exhibit a rapid change around time t as β approaches one. From Eq. (1.11), it is clear that the short memory normalized fractional derivative represents a form of weighted average of the traditional derivative over its historical values. This implies that the temporal derivative at the current time is influenced by the temporal derivatives from previous times.

Next, we show the normalized time-fractional diffusion equation with short memory presented in this work preserves the total mass under homogeneous Neumann boundary conditions, i.e., $\mathbf{n}\cdot\nabla u(x,y,t)=0$, for $(x,y)\in\partial\Omega$. Here, $\mathbf{n}$ is the normal vector to the boundary. We define the total quantity associated with the field $u(x,y,t)$ as the spatial integral

of u over the entire domain Ω. This quantity, denoted by $TM(t)$, acts as an indicator of whether the governing diffusion equation preserves the global amount of the modeled physical variable. Accordingly, we set

$$TM(t) := \int_\Omega u(x,y,t)dxdy. \tag{1.14}$$

Applying the normalized short memory time-fractional derivative to $TM(t)$ gives

$$\begin{aligned}\frac{\partial^\beta TM(t)}{\partial t^\beta} &= \frac{\partial^\beta \int_\Omega u(x,y,t)dxdy}{\partial t^\beta} = \int_\Omega \frac{\partial^\beta u(x,y,t)}{\partial t^\beta}dxdy \\ &= \int_\Omega \Delta u(x,y,t)dxdy = \int_{\partial\Omega} \mathbf{n}\cdot\nabla u(x,y,t)dS = 0,\end{aligned} \tag{1.15}$$

where the interchange of integration and differentiation and the use of the divergence theorem under the homogeneous Neumann boundary condition yield the above identity. Under homogeneous Neumann boundary conditions, we want to show that

$$TM(t) = TM(0) \qquad \text{for all } t>0.$$

The proof proceeds by mathematical induction on the intervals $[kM,(k+1)M]$ for $k=0,1,2,\ldots$. The proof is carried out in the following three steps.

Step 1. Base case: $t\in[0,M]$

For $0<t\le M$, the memory interval is $[0,t]$ and the short-memory normalized time-fractional derivative reduces to

$$\frac{\partial^\beta TM(t)}{\partial t^\beta} = \frac{1-\beta}{t^{1-\beta}}\int_0^t TM'(s)(t-s)^{-\beta}ds.$$

From Eq. (1.15), we have

$$\frac{\partial^\beta TM(t)}{\partial t^\beta} = 0 \qquad \text{for all } t\in[0,M].$$

Hence,

$$\int_0^t TM'(s)(t-s)^{-\beta}ds = 0, \qquad \forall t\in[0,M].$$

This is an Abel integral equation. Recall that the Abel integral equation [28, 29] is given by

$$F(t) = \int_0^t \frac{\varphi(s)}{(t-s)^\alpha}ds, \qquad 0<\alpha<1,$$

whose inversion formula reads

$$\varphi(t) = \frac{\sin(\pi\alpha)}{\pi}\frac{d}{dt}\int_0^t \frac{F(s)}{(t-s)^{1-\alpha}}ds.$$

It follows from $F(t)=0$ that $\varphi(t)=0$. Applying the inversion formula for Abel equations, it follows that

$$TM'(s)=0 \qquad \text{for a.e. } s\in[0,M],$$

and consequently,

$$TM(t)=TM(0), \qquad \forall t\in[0,M].$$

Step 2. Inductive hypothesis

Assume that for some integer $k\geq 1$, we have already proved

$$TM'(s)=0 \qquad \text{for a.e. } s\in[0,kM],$$

or equivalently,

$$TM(t)=TM(0) \qquad \text{for all } t\in[0,kM].$$

Step 3. Inductive step: $t\in[kM,(k+1)M]$

For $t\in[kM,(k+1)M]$, the memory interval is $[t-M,t]$. Since $t-M\in[(k-1)M,kM]$, the inductive hypothesis guarantees that

$$TM'(s)=0 \qquad \text{for all } s\in[t-M,kM].$$

Substituting into the fractional derivative expression yields

$$\begin{aligned} 0&=\frac{\partial^\beta TM(t)}{\partial t^\beta}=\frac{1-\beta}{M^{1-\beta}}\int_{t-M}^{t}TM'(s)(t-s)^{-\beta}ds \\ &=\frac{1-\beta}{M^{1-\beta}}\int_{t-M}^{kM}TM'(s)(t-s)^{-\beta}ds+\frac{1-\beta}{M^{1-\beta}}\int_{kM}^{t}TM'(s)(t-s)^{-\beta}ds. \end{aligned} \tag{1.16}$$

By the induction hypothesis, we have $TM'(s)=0$ for all $s\in[0,kM]$. Since $t\in[kM,(k+1)M]$, it follows that $t-M\in[(k-1)M,kM]$, and hence $[t-M,kM]\subset[0,kM]$. Therefore, $TM'(s)=0$ for all $s\in[t-M,kM]$, which implies

$$\int_{t-M}^{kM}TM'(s)(t-s)^{-\beta}ds=0. \tag{1.17}$$

From Eqs. (1.16) and (1.17), we obtain:

$$\int_{kM}^{t}TM'(s)(t-s)^{-\beta}ds=0, \qquad \forall t\in[kM,(k+1)M].$$

Let $\mu=t-kM$, $\tau=s-kM$ and define $g(\tau):=TM'(kM+\tau)$. Then the above equation becomes

$$\int_0^{\mu}g(\tau)(\mu-\tau)^{-\beta}d\tau=0, \qquad \forall \mu\in[0,M].$$

This is again an Abel integral equation. Applying the inversion formula, we obtain $g(\tau)=0$ for almost every $\tau\in[0,M]$, i.e.,

$$TM'(s)=0 \qquad \text{a.e. on } [kM,(k+1)M].$$

Therefore,

$$TM(t)=TM(kM)=TM(0), \qquad \forall t\in[kM,(k+1)M].$$

Thus, the continuous normalized time-fractional diffusion model with short memory preserves conservation analytically:

$$TM(t)=TM(0) \qquad \text{for all } t>0.$$

Although the classical time-fractional diffusion equation has been widely applied to the modeling of anomalous diffusion phenomena, their application to long-term simulations remains computationally demanding due to the need to account for the entire historical information of the solution. This motivates the development of more efficient fractional models that retain key physical characteristics while significantly reducing memory and computational burdens. The main contribution of this work lies in the formulation and numerical implementation of a short memory normalized time-fractional diffusion equation that achieves computational efficiency. By introducing a normalized weighting function with fixed memory truncation, we enable accurate simulations of fractional dynamics while substantially reducing computational costs, especially in high-dimensional domains. While previous studies have focused on either normalized fractional derivatives or short memory approximations, this study uniquely integrates both approaches into a unified method. This integration enables a fair comparison across different fractional orders and facilitates long-time simulations with reduced complexity. Furthermore, the applicability of this short memory scheme extends to a wide range of normalized time-fractional derivatives, including the Allen–Cahn equation [30], the Burgers equation [31], the SIR epidemic model [32], the Fokker–Planck equation [33], the Fisher equation [34], and the Cahn–Hilliard equation [35].

To validate the proposed short memory normalized TF diffusion model, the subsequent sections provide detailed numerical implementations and experiments. In Section 2, we describe the numerical schemes based on operator splitting and FDMs for both 2D and 3D domains. Section 3 presents extensive numerical experiments that investigate the effects of varying memory lengths and fractional orders on solution behavior and computational efficiency. Section 4 summarizes that the short memory model offers a practical and computationally efficient method for simulating anomalous diffusion phenomena across various spatial dimensions and temporal scales.

2 Numerical solutions

2.1 2D discretization

Let $\Omega=(L_x,\,R_x)\times(L_y,\,R_y)$ be the spatial domain, with discretization defined as follows: $\Omega=\{(x_i,y_j)|x_i=L_x+(i-1)h_x,\,i=1,\ldots,N_x \text{ and } y_j=L_y+(j-1)h_y,\,j=1,\ldots,N_y\}$, where $h_x=(R_x-L_x)/(N_x-1)$ and $h_y=(R_y-L_y)/(N_y-1)$, for some positive integers N_x and N_y. Let $u_{ij}^n=u\left(x_i,y_j,t_n\right)$ and $t_n=(n-1)\Delta t$ with the time step Δt. For simplicity, let $M=M_t\Delta t$ for some positive integer M_t and h be the grid size. Eq. (1.11) can be evaluated using the following computational integration formula:

$$
\begin{aligned}
\frac{\partial^\beta u(x,y,t)}{\partial t^\beta}&=\frac{1-\beta}{[t_{n+1}-\max(0,t_{n+1}-M)]^{1-\beta}}\sum_{p=\max(1,n+1-M_t)}^{n}\int_{t_p}^{t_{p+1}}\frac{\partial u(x,y,s)}{\partial s}\frac{ds}{(t_{n+1}-s)^\beta}\\
&\approx\sum_{p=\max(1,n+1-M_t)}^{n}\frac{1-\beta}{[t_{n+1}-\max(0,t_{n+1}-M)]^{1-\beta}}\int_{t_p}^{t_{p+1}}\frac{ds}{(t_{n+1}-s)^\beta}\frac{u_{ij}^{p+1}-u_{ij}^{p}}{\Delta t}\\
&=\sum_{p=\max(1,n+1-M_t)}^{n}\frac{(n+1-p)^{1-\beta}-(n-p)^{1-\beta}}{[n+1-\max(1,n+1-M_t)]^{1-\beta}}\frac{u_{ij}^{p+1}-u_{ij}^{p}}{\Delta t}\\
&=\sum_{p=\max(1,n+1-M_t)}^{n}w_p^n\frac{u_{ij}^{p+1}-u_{ij}^{p}}{\Delta t}\\
&=w_n^n\frac{u_{ij}^{n+1}-u_{ij}^{n}}{\Delta t}+\sum_{p=\max(1,n+1-M_t)}^{n-1}w_p^n\frac{u_{ij}^{p+1}-u_{ij}^{p}}{\Delta t}\\
&\approx w_n^n\frac{\partial u(x,y,t)}{\partial t}+\sum_{p=\max(1,n+1-M_t)}^{n-1}w_p^n\frac{u_{ij}^{p+1}-u_{ij}^{p}}{\Delta t},\quad \text{at } (x,y,t)=(x_i,y_j,t_{n+1}),
\end{aligned}
\tag{2.1}
$$

where $w_p^n:=[(n+1-p)^{1-\beta}-(n-p)^{1-\beta}]/[n+1-\max(1,n+1-M_t)]^{1-\beta}$. It is noted that for every value of n, the weighting parameter w_p^n satisfies the following condition:

$$
\sum_{p=\max(1,n+1-M_t)}^{n}w_p^n=1. \tag{2.2}
$$

We then rewrite the normalized diffusion equation as the following approximation:

$$
\begin{aligned}
\frac{\partial u(x,y,t)}{\partial t}=&\frac{1}{w_n^n}\frac{\partial^2 u(x,y,t)}{\partial x^2}+\frac{1}{w_n^n}\frac{\partial^2 u(x,y,t)}{\partial y^2}\\
&-\frac{1}{w_n^n}\sum_{p=\max(1,n+1-M_t)}^{n-1}w_p^n\frac{u_{ij}^{p+1}-u_{ij}^{p}}{\Delta t},\quad \text{at } (x,y,t)=(x_i,y_j,t_{n+1}).
\end{aligned}
$$

To efficiently solve this equation, we apply an operator splitting strategy [36, 37] that decomposes the computation into two sequential one-dimensional problems. This approach enables the use of dimensionwise discretization and simplifies the numerical implementation. Specifically, we separate the evolution process into the following two steps:

$$\begin{aligned} &Step\ 1\text{:} \qquad \frac{\partial u(x,y,t)}{\partial t}=\frac{1}{w_n^n}\frac{\partial^2 u(x,y,t)}{\partial x^2}+\frac{1}{2}s_{ij}^n.\\ &Step\ 2\text{:} \qquad \frac{\partial u(x,y,t)}{\partial t}=\frac{1}{w_n^n}\frac{\partial^2 u(x,y,t)}{\partial y^2}+\frac{1}{2}s_{ij}^n, \quad \text{at } (x,y,t)=(x_i,y_j,t_{n+1}). \end{aligned}$$

Here, $s_{ij}^n := -\sum_{p=\max(1,n+1-M_t)}^{n-1} w_p^n(u_{ij}^{p+1}-u_{ij}^p)/(w_n^n\Delta t)$. We solve the equations presented in Steps 1 and 2 using the FDM as follows:

$$\begin{aligned} &Step\ 1\text{:} \qquad \frac{u_{ij}^*-u_{ij}^n}{\Delta t}=\frac{u_{i-1,j}^*-2u_{ij}^*+u_{i+1,j}^*}{w_n^n h_x^2}+\frac{1}{2}s_{ij}^n, \qquad \text{for } i=2,\ldots,N_x-1.\\ &Step\ 2\text{:} \qquad \frac{u_{ij}^{n+1}-u_{ij}^*}{\Delta t}=\frac{u_{i,j-1}^{n+1}-2u_{ij}^{n+1}+u_{i,j+1}^{n+1}}{w_n^n h_y^2}+\frac{1}{2}s_{ij}^n, \quad \text{for } j=2,\ldots,N_y-1. \end{aligned}$$

Rearranging the discrete equation in *Step* 1, the implicit scheme leads to a tridiagonal linear system. For a fixed j, the equation for u_{ij}^* can be written as:

$$\zeta_i u_{i-1,j}^*+\eta_i u_{ij}^*+\theta_i u_{i+1,j}^*=\mathcal{G}_i, \qquad i=1,\ldots,N_x, \tag{2.3}$$

where the coefficients are defined as:

$$\zeta_i=-\frac{1}{w_n^n h_x^2}, \quad \eta_i=\frac{1}{\Delta t}+\frac{2}{w_n^n h_x^2}, \quad \theta_i=-\frac{1}{w_n^n h_x^2}, \quad \text{and} \quad \mathcal{G}_i=\frac{u_{ij}^n}{\Delta t}+\frac{1}{2}s_{ij}^n.$$

Under the Dirichlet boundary conditions, we have the following setting: $u_{0j}^*=0$ and $u_{N_x+1,j}^*=0$. The system (2.3) is solved efficiently using the Thomas algorithm. This corresponds to solving the linear system:

$$\mathcal{T}_x \mathbf{U}_{1:N_x,j}^* = \mathbf{G}_{1:N_x,j}, \tag{2.4}$$

where the tridiagonal matrix $\mathcal{T}_x$ takes the form:

$$\mathcal{T}_x=\begin{pmatrix} \eta_1 & \theta_1 & 0 & \cdots & 0 & 0\\ \zeta_2 & \eta_2 & \theta_2 & \cdots & 0 & 0\\ 0 & \zeta_3 & \eta_3 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & \eta_{N_x-1} & \theta_{N_x-1}\\ 0 & 0 & 0 & \cdots & \zeta_{N_x} & \eta_{N_x} \end{pmatrix}, \tag{2.5}$$

with vector solutions and the right-hand side vector

$$\mathbf{U}^*_{1:N_x,j}=[u^*_{1j},u^*_{2j},\ldots,u^*_{N_xj}]^T,\quad \mathbf{G}_{1:N_x,j}=[\mathcal{G}_1,\mathcal{G}_2,\ldots,\mathcal{G}_{N_x}]^T.$$

Step 2 can be solved by applying the same strategy used in *Step* 1, because the underlying structure of the discrete equation remains identical and the resulting linear system admits the same type of decoupling and sequential solution procedure.

2.2 Conservation of mass

To verify the conservation law, we use the zero Neumann boundary condition and a cell-centered discretization. Let $\Omega=(L_x,\ R_x)\times(L_y,\ R_y)$ be the spatial domain, with discretization defined as follows: $\Omega=\{(x_i,y_j)|x_i=L_x+(i-0.5)h_x,\ i=1,\ldots,N_x$ and $y_j=L_y+(j-0.5)h_y,\ j=1,\ldots,N_y\}$, where $h_x=(R_x-L_x)/N_x$ and $h_y=(R_y-L_y)/N_y$, for some positive integers N_x and N_y. We use h to represent the grid size. Based on Eq. (1.14), we define the discrete total quantity by

$$TM^n:=h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}u^n_{ij},$$

for all time steps $n\geq 1$. First, let us consider the case $n=1$. We have

$$\frac{u^*_{ij}-u^1_{ij}}{\Delta t}=\frac{u^*_{i-1,j}-2u^*_{ij}+u^*_{i+1,j}}{w^1_1h_x^2},\tag{2.6}$$

$$\frac{u^2_{ij}-u^*_{ij}}{\Delta t}=\frac{u^2_{i,j-1}-2u^2_{ij}+u^2_{i,j+1}}{w^1_1h_y^2}.\tag{2.7}$$

According to the definition of s^n_{ij}, we conclude that the sum is empty when $n=1$, i.e., $s^1_{ij}=0$. Summing over all discretized grid points (i,j), we obtain

$$\begin{aligned}h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{u^*_{ij}-u^1_{ij}}{\Delta t}&=h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{u^*_{i-1,j}-2u^*_{ij}+u^*_{i+1,j}}{w^1_1h_x^2}\\&=h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{(u^*_{i+1,j}-u^*_{ij})-(u^*_{ij}-u^*_{i-1,j})}{w^1_1h_x^2}\\&=h_xh_y\sum_{j=1}^{N_y}\frac{(u^*_{N_x+1,j}-u^*_{N_xj})-(u^*_{1j}-u^*_{0j})}{w^1_1h_x^2}=0,\end{aligned}\tag{2.8}$$

where we used the homogeneous Neumann boundary condition $u^*_{0j}=u^*_{1j}$ and $u^*_{N_x+1,j}=u^*_{N_x,j}$. Thus, we have

$$h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}u^*_{ij}=h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}u^1_{ij},$$

that is,

$$TM^* = TM^1. \tag{2.9}$$

For the second step, we apply the same procedure to Eq. (2.7), and we obtain

$$\begin{aligned} h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \frac{u_{ij}^2 - u_{ij}^*}{\Delta t} &= h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \frac{u_{i,j-1}^2 - 2u_{ij}^2 + u_{i,j+1}^2}{w_1^1 h_y^2} \\ &= h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \frac{(u_{i,j+1}^2 - u_{ij}^2) - (u_{ij}^2 - u_{i,j-1}^2)}{w_1^1 h_y^2} \\ &= h_x h_y \sum_{i=1}^{N_x} \frac{(u_{i,N_y+1}^2 - u_{iN_y}^2) - (u_{i1}^2 - u_{i0}^2)}{w_1^1 h_y^2} = 0, \end{aligned} \tag{2.10}$$

where we use the homogeneous Neumann boundary condition $u_{i0}^2 = u_{i1}^2$ and $u_{i,N_y+1}^2 = u_{i,N_y}^2$. Again, we have

$$h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} u_{ij}^2 = h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} u_{ij}^*,$$

that is,

$$TM^2 = TM^*. \tag{2.11}$$

From Eqs. (2.9) and (2.11), we obtain

$$TM^2 = TM^1. \tag{2.12}$$

Then, we assume that for some $n \geq 2$, the conservation law holds for all previous steps:

$$TM^k = TM^1, \quad 1 \leq k \leq n. \tag{2.13}$$

Next, for step n under the homogeneous Neumann boundary condition $u_{0j}^* = u_{1j}^*$, $u_{N_x+1,j}^* = u_{N_x j}^*$, $u_{i0}^{n+1} = u_{i1}^{n+1}$, and $u_{i,N_y+1}^{n+1} = u_{iN_y}^{n+1}$, we have

$$\begin{aligned} h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \frac{u_{ij}^* - u_{ij}^n}{\Delta t} &= h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \frac{u_{i-1,j}^* - 2u_{ij}^* + u_{i+1,j}^*}{w_n^n h_x^2} + \frac{1}{2} h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} s_{ij}^n \\ &= h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \frac{(u_{i+1,j}^* - u_{ij}^*) - (u_{ij}^* - u_{i-1,j}^*)}{w_n^n h_x^2} + \frac{1}{2} h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} s_{ij}^n \\ &= h_x h_y \sum_{j=1}^{N_y} \frac{(u_{N_x+1,j}^* - u_{N_x j}^*) - (u_{1j}^* - u_{0j}^*)}{w_n^n h_x^2} + \frac{1}{2} h_x h_y \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} s_{ij}^n \end{aligned}$$

$$=\frac{1}{2}h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}s_{ij}^n, \tag{2.14}$$

$$\begin{aligned}
h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{u_{ij}^{n+1}-u_{ij}^*}{\Delta t}&=h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{u_{i,j-1}^{n+1}-2u_{ij}^{n+1}+u_{i,j+1}^{n+1}}{w_n^nh_y^2}+\frac{1}{2}h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}s_{ij}^n\\
&=h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{(u_{i,j+1}^{n+1}-u_{ij}^{n+1})-(u_{ij}^{n+1}-u_{i,j-1}^{n+1})}{w_n^nh_y^2}+\frac{1}{2}h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}s_{ij}^n\\
&=h_xh_y\sum_{i=1}^{N_x}\frac{(u_{i,N_y+1}^{n+1}-u_{iN_y}^{n+1})-(u_{i1}^{n+1}-u_{i0}^{n+1})}{w_n^nh_y^2}+\frac{1}{2}h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}s_{ij}^n\\
&=\frac{1}{2}h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}s_{ij}^n.
\end{aligned} \tag{2.15}$$

Adding Eqs. (2.14) and (2.15), it reduces to

$$h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\frac{u_{ij}^{n+1}-u_{ij}^n}{\Delta t}=h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}s_{ij}^n. \tag{2.16}$$

Using the definition of s_{ij}^n, we obtain

$$\Delta t\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}s_{ij}^n=-\frac{1}{w_n^n}\sum_{p=\max(1,n+1-M_t)}^{n-1}w_p^n\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left(u_{ij}^{p+1}-u_{ij}^p\right). \tag{2.17}$$

According to our previous assumption in Eq. (2.13), we have $TM^k=h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}u_{ij}^k$ is conserved for all $k\le n$, equivalently,

$$TM^{q+1}=TM^q,\qquad \text{for all } q\le n-1.$$

Then, for each q appearing in the above sum, we have

$$\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}(u_{ij}^{q+1}-u_{ij}^q)=0.$$

Hence, according to Eq. (2.17):

$$\Delta t\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}s_{ij}^n=0.$$

Substituting this result into (2.16) gives

$$h_xh_y\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}(u_{ij}^{n+1}-u_{ij}^n)=0,\quad \text{i.e.,}\quad TM^{n+1}=TM^n.$$

Since TM^1 is conserved at the first step and we have shown that $TM^{n+1}=TM^n$, mathematical induction concludes

$$TM^n=TM^1, \qquad \text{for all } n\geq 1.$$

Therefore, the scheme preserves the total discrete conservation at every time level. The 3D case also satisfies the conservative law and can be proven through similar approach as the 2D case.

2.3 3D discretization

Let us straightforwardly extend the dimension and consider a 3D domain. The 3D case can similarly be discretized as $\Omega=\{(x_i,y_j,z_k)|x_i=L_x+(i-1)h_x,\ i=1,\ldots,N_x,\ y_j=L_y+(j-1)h_y,\ j=1,\ldots,N_y,$ and $z_k=L_z+(k-1)h_z,\ k=1,\ldots,N_z\}$, where $h_x=(R_x-L_x)/(N_x-1)$, $h_y=(R_y-L_y)/(N_y-1)$, and $h_z=(R_z-L_z)/(N_z-1)$ is the uniform grid size for positive integers N_x, N_y, and N_z. Let $u^n_{ijk}=u(x_i,y_j,z_k,t_n)$. The 3D short memory normalized TF derivative can be approximated as follows:

$$\frac{\partial^\beta u(x,y,z,t)}{\partial t^\beta}\approx w^n_n\frac{\partial u(x,y,z,t)}{\partial t}+\sum_{p=\max(1,n+1-M_t)}^{n-1} w^n_p\frac{u^{p+1}_{ijk}-u^p_{ijk}}{\Delta t},$$
$$\text{at } (x,y,z,t)=(x_i,y_j,z_k,t_{n+1}),$$

where w^n_p shares the same definition in Eq. (2.1) for a short memory time $M=M_t\Delta t$. Let us consider the following short memory normalized diffusion equation with zero Dirichlet boundary condition in 3D space:

$$\frac{\partial u(x,y,z,t)}{\partial t}=\frac{1}{w^n_n}\frac{\partial^2 u(x,y,z,t)}{\partial x^2}+\frac{1}{w^n_n}\frac{\partial^2 u(x,y,z,t)}{\partial y^2}+\frac{1}{w^n_n}\frac{\partial^2 u(x,y,z,t)}{\partial z^2}$$
$$-\frac{1}{w^n_n}\sum_{p=\max(1,n+1-M_t)}^{n-1} w^n_p\frac{u^{p+1}_{ijk}-u^p_{ijk}}{\Delta t},\quad \text{at } (x,y,z,t)=(x_i,y_j,z_k,t_{n+1}).$$

In order to solve the equation efficiently, we adopt an operator splitting technique that divides the original problem into two successive one-dimensional sub-problems. The temporal evolution is accordingly handled through the following three-step procedure:

$$\textit{Step } 1: \quad \frac{\partial u(x,y,z,t)}{\partial t}=\frac{1}{w^n_n}\frac{\partial^2 u(x,y,z,t)}{\partial x^2}+\frac{1}{3}s^n_{ijk}.$$
$$\textit{Step } 2: \quad \frac{\partial u(x,y,z,t)}{\partial t}=\frac{1}{w^n_n}\frac{\partial^2 u(x,y,z,t)}{\partial y^2}+\frac{1}{3}s^n_{ijk}.$$
$$\textit{Step } 3: \quad \frac{\partial u(x,y,z,t)}{\partial t}=\frac{1}{w^n_n}\frac{\partial^2 u(x,y,z,t)}{\partial z^2}+\frac{1}{3}s^n_{ijk},\quad \text{at } (x,y,z,t)=(x_i,y_j,z_k,t_{n+1}).$$

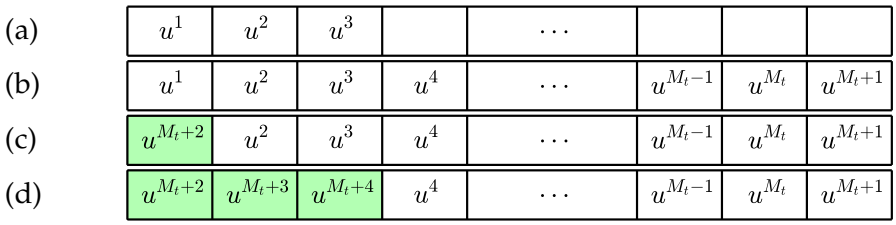


Figure 2: The stored matrix for computing the short memory normalized TF diffusion equation.

Here, $s_{ijk}^n := -\sum_{p=\max(1,n+1-M_t)}^{n-1} w_p^n (u_{ijk}^{p+1} - u_{ijk}^p)/(w_n^n \Delta t)$. We solve the equations presented in Steps 1, 2, and 3 using the finite difference method as follows:

$$\textit{Step } 1\text{:} \quad \frac{u_{ijk}^* - u_{ijk}^n}{\Delta t} = \frac{u_{i-1,j,k}^* - 2u_{ijk}^* + u_{i+1,j,k}^*}{w_n^n h_x^2} + \frac{1}{3} s_{ijk}^n, \qquad \text{for } i = 2,\ldots,N_x - 1.$$

$$\textit{Step } 2\text{:} \quad \frac{u_{ijk}^{**} - u_{ijk}^*}{\Delta t} = \frac{u_{i,j-1,k}^{**} - 2u_{ijk}^{**} + u_{i,j+1,k}^{**}}{w_n^n h_y^2} + \frac{1}{3} s_{ijk}^n, \qquad \text{for } j = 2,\ldots,N_y - 1.$$

$$\textit{Step } 3\text{:} \quad \frac{u_{ijk}^{n+1} - u_{ijk}^{**}}{\Delta t} = \frac{u_{i,j,k-1}^{n+1} - 2u_{ijk}^{n+1} + u_{i,j,k+1}^{n+1}}{w_n^n h_z^2} + \frac{1}{3} s_{ijk}^n, \qquad \text{for } k = 2,\ldots,N_z - 1.$$

These discrete equations are solved by using the Thomas algorithm [38] analogously as the 2D case presented in Section 2.1, with homogeneous Dirichlet boundary conditions:

$$u_{1jk} = 0, \quad u_{N_x jk} = 0, \qquad \text{for } j = 1,\ldots,N_y \text{ and } k = 1,\ldots,N_z, \tag{2.18}$$

$$u_{i1k} = 0, \quad u_{iN_y k} = 0, \qquad \text{for } i = 1,\ldots,N_x \text{ and } k = 1,\ldots,N_z, \tag{2.19}$$

$$u_{ij1} = 0, \quad u_{ijN_z} = 0, \qquad \text{for } i = 1,\ldots,N_x \text{ and } j = 1,\ldots,N_y. \tag{2.20}$$

3 Numerical experiments

In this section, all computational experiments were conducted on a MacBook Pro equipped with an Apple M1 Pro processor (8-core) and 16 GB of RAM. The algorithms were implemented using MATLAB R2024b. We examine the relationship between the memory length M and the fractional order β in the short memory TF method in 2D and 3D. For a fixed number of memory length steps M_t, when $n \le M_t + 1$, the size of matrix stored for calculation increases linearly, as in the conventional TF method (see Fig. 2(a) and (b)). Afterwards, when u^{M_t+2} has been computed, it is stored in the original u^1 location in order to maintain a fixed memory length (see Fig. 2(c)). Next, as in Fig. 2(d), the previously stored u is replaced with a new value.

For simplicity, we use a uniform spatial step size $h = h_x = h_y = h_z$ throughout the numerical tests.

Table 1: The temporal errors and convergence rates.

	$0.05/2^8$	rate	$0.05/2^9$	rate	$0.05/2^{10}$	rate	$0.05/2^{11}$
l_2-error	2.7648*e*-5	1.00	1.3798*e*-5	1.16	6.1820*e*-6	1.54	2.1216*e*-6

3.1 Convergence test

In this subsection, we analyze the temporal accuracy with the fixed parameters $\beta=0.5$, $T=0.05$ and $M=T/16$. The computation domain is set to $\Omega=(0,1)\times(0,1)$. The initial condition is given as:

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

Here, we use the zero Dirichlet boundary conditions.

We define the l_2-norm error $\mathcal{E}(\Delta t)=\|u^n-u^n_{ref}\|_2/\|u^n_{ref}\|_2$, where u_{ref} denotes the reference solution, and $\|u^n\|_2=\sqrt{\sum_{j=1}^{N_y}\sum_{i=1}^{N_x}(u^n_{ij})^2/(N_xN_y)}$. The temporal accuracy is examined by decreasing the time step through several refined values, taken as $\Delta t=0.05/2^8$, $0.05/2^9$, $0.05/2^{10}$, and $0.05/2^{11}$. The spatial grid remains fixed at 101×101 points, while the reference solution u_{ref} is computed using a finer time step of $\Delta t=0.05/2^{12}$. The temporal errors and the corresponding convergence rates for different time steps are listed in Table 1.

3.2 Total mass conservation

In this subsection, we demonstrate from a numerical perspective that the short-memory time-fractional formulation preserves this discrete quantity under homogeneous Neumann boundary conditions. The number of spatial grid points is $N_x=N_y=100$, the spatial step size is $h=1/N_x$, the time step size is $\Delta t=0.00005$, the number of time steps is $N_t=20000$, and final time is $T=N_t\Delta t$. Here, we use the parameters $M=0.001$ and $\beta=0.9$. The initial condition is set as follows:

$$u(x,y,0)=\cos(\pi x)\cos(\pi y)+1,\qquad (x,y)\in\Omega=(0,1)\times(0,1). \tag{3.1}$$

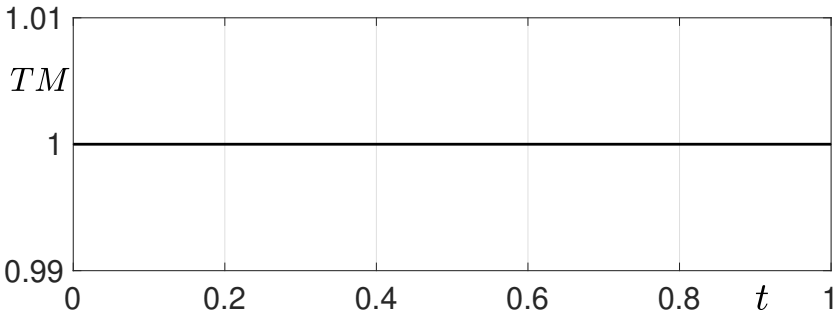


Figure 3: The temporal evolution of the discrete quantity.

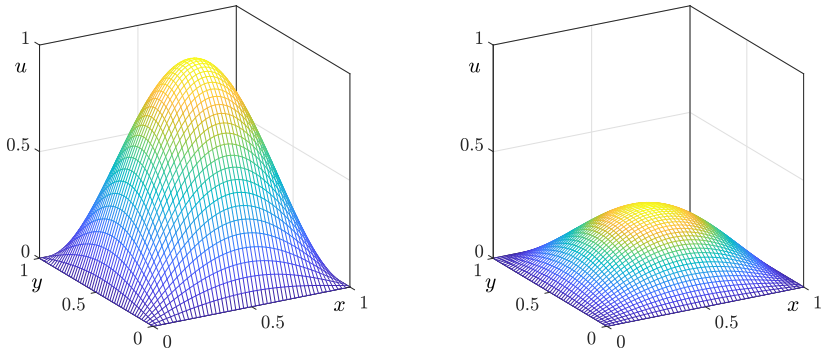


Figure 4: Numerical solution of the diffusion equation in (a) the initial condition and (b) $\beta=1$ and $t=0.05$.

Here, we impose homogeneous Neumann boundary conditions. Fig. 3 shows the evolution of the discrete quantity over time, where we observe a clear conservation behavior. Since the exact value of the mass is given by $\int_0^1\int_0^1(\cos(\pi x)\cos(\pi y)+1)\,dxdy=1$, the numerical results in Fig. 3 show that the discrete quantity stays equal to the exact value, which verifies the conservation property of the scheme.

3.3 2D experiment

In this experiment, we observe the numerical solutions for the diffusion equation in 2D field. The number of spatial grid points is $N_x=N_y=51$, the spatial step size is $h=1/(N_x-1)$, the time step size is $\Delta t=0.00005$, the number of time steps is $N_t=1000$, and final time is $T=N_t\Delta t$. The initial condition is set as follows:

$$u(x,y,0)=\sin(\pi x)\sin(\pi y),\qquad (x,y)\in\Omega=(0,1)\times(0,1). \tag{3.2}$$

Here, for all $t\geq 0$, the solution $u(x,y,t)$ satisfies zero Dirichlet boundary conditions on the boundary of the domain. Fig. 4(a) and (b) show the initial condition and the numerical solution when $t=0.05$ while β is set to 1, respectively. It is noted that $\beta=1$ corresponds to the standard time derivative.

Next, we consider the time-fractional derivative with a short memory. The memory lengths are set to $M=10\Delta t,100\Delta t$, and $1000\Delta t$. Fig. 5 exhibits numerical solutions with different β values and varying memory lengths M. More specifically, we select $\beta=$ 0.01 and 0.99 and for each β value, we choose memory lengths M=0.0005, 0.005, and 0.05 for comparison.

When the fractional order β is fixed, the numerical solution with a smaller memory length M exhibits more rapid decay compared to those with larger memory lengths. When the memory length M is fixed, the numerical solution decays less when adopting smaller fractional order β. Namely, the smaller the fractional order β is and the longer the memory length M is, the more historical data is carried during computation.

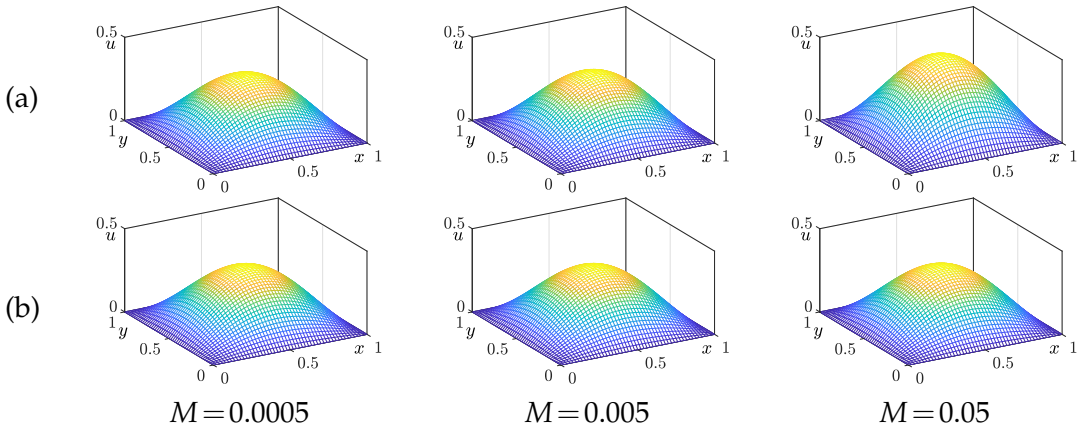


Figure 5: Numerical solutions of the diffusion equation at $t=0.05$ for various values of M, where (a) and (b) correspond to $\beta=0.01$ and $\beta=0.99$, respectively.

Table 2: The maximum values of the numerical solutions for the short memory normalized TF 2D diffusion equation under different parameters.

$\beta\backslash M$	0.0005	0.005	0.01	0.02	0.03	0.04	0.05
0.01	0.3745	0.3900	0.4060	0.4291	0.4500	0.4665	0.5026
0.50	0.3740	0.3843	0.3948	0.4131	0.4271	0.4422	0.4551
0.99	0.3729	0.3733	0.3736	0.3742	0.3746	0.3749	0.3751

In Fig. 6, we further visualize the maximum values of the computational solutions of the 2D diffusion equation. Fig. 6 illustrates how the maximum value of the solution changes with memory length M for different fractional orders β in 2D simulations. As M increases, the maximum value grows significantly for small β (e.g., $\beta=0.01$), moderately for $\beta=0.50$, and remains nearly constant for $\beta=0.99$. This indicates that longer memory effects (large M) are more sensitive to variations in β, while solutions with short memory (small M) are relatively unaffected.

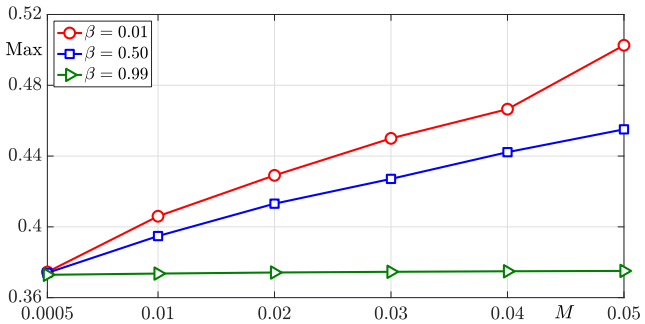


Figure 6: The maximum values with respect to memory length M for various β in 2D experiments.

Let us define the maximum value of the solution as:

$$\text{Max} := \max_{1\le i\le N_x, 1\le j\le N_y} u(x_i, y_j, T). \tag{3.3}$$

Table 2 shows that the maximum value of the numerical solution increases as the memory length M grows, particularly for small fractional order $\beta = 0.01$. For $\beta = 0.50$, the increase is moderate, while for $\beta = 0.99$, the maximum values remain nearly constant. This demonstrates that solutions with longer memory are more sensitive to β, whereas those with short memory are relatively stable regardless of M.

In addition to the two extreme fractional orders $\beta = 0.01$ and $\beta = 0.99$ discussed above, we include the intermediate case $\beta = 0.5$ in Table 2. As shown in Fig. 7, this intermediate value demonstrates that both the decay rate and CPU time exhibit smooth and monotonic dependence on M. Fig. 7 further displays the CPU time for solving Eq. (1.8) with the initial condition (3.2) as a function of the memory length M. The CPU time is compared to a reference value corresponding to the full memory case $M = T$, and the ratio is plotted accordingly. Apart from varying the memory length M and the number of memory intervals M_t, all other parameters are kept identical to those used in the experiment of Fig. 4. The results show that although the CPU time increases with memory length, it remains substantially lower than the full memory cost. Hence, reducing M improves computational efficiency without fully sacrificing memory effects.

3.4 3D experiment

We set the parameters uniformly as follows: the number of spatial grid points is $N_x = N_y = N_z = 51$, the spatial step size is $h = 1/(N_x - 1)$, the time step size is $\Delta t = 0.0005$, the number of time steps is $N_t = 100$, and the final time is $T = N_t \Delta t$. The initial condition is set as follows:

$$u(x,y,z,0) = \sin(\pi x)\sin(\pi y)\sin(\pi z), \quad (x,y,z) \in \Omega = (0,1)\times(0,1)\times(0,1). \tag{3.4}$$

Here, for all $t \ge 0$, the solution $u(x,y,z,t)$ satisfies zero Dirichlet boundary conditions on the boundary of the domain. Fig. 8 presents snapshots of the 0.2 isosurface of the solution

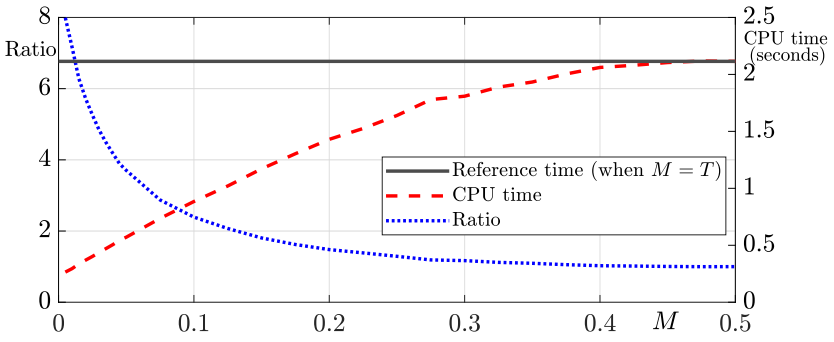


Figure 7: CPU time and Ratio with respect to memory length M for $\beta = 0.5$ in 2D.

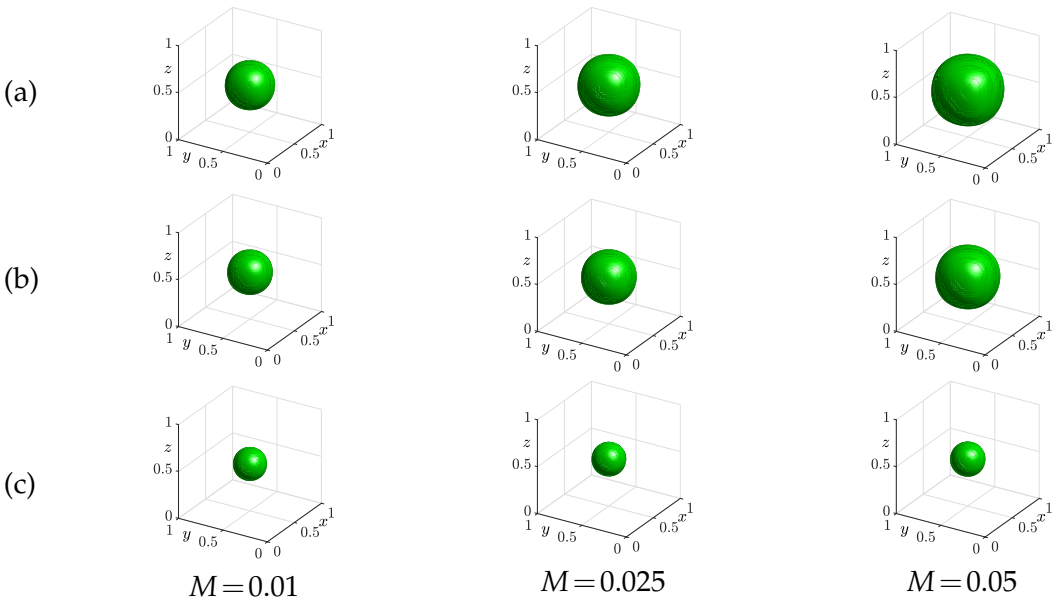


Figure 8: Snapshots of the 0.2 isosurface of the solution in 3D space at $t=0.05$ for various values of β and memory length M: (a) $\beta=0.01$ with $M=0.01$, 0.025, and 0.05; (b) $\beta=0.5$ with $M=0.01$, 0.025, and 0.05; (c) $\beta=0.99$ with $M=0.01$, 0.025, and 0.05.

in 3D space at $t=T$ under different values of β and M. In Fig. 8, the values of M are the same within each column, corresponding to $M=20\Delta t$, $50\Delta t$, and $100\Delta t$ from left to right. The values of β are the same within each row, corresponding to β=0.01, 0.5, and 0.99 from top to bottom. It can be observed that, for a fixed memory length M, a larger value of β leads to a faster temporal evolution. On the other hand, for a fixed β, a smaller memory length M also results in a faster temporal evolution.

We investigate maximum values and CPU time for increasing the memory length M for various β. We use the same set of parameters as those in the experiment shown in Fig. 8. To quantify the degree of variation, we define the maximum value of the solution as:

$$\text{Max} := \max_{1\le i\le N_x, 1\le j\le N_y, 1\le k\le N_z} u(x_i, y_j, z_k, T). \tag{3.5}$$

In Fig. 9 and Table 3, the Max increases as the memory length M becomes larger. Smaller values of the fractional order β result in more rapid growth of the Max. This suggests that lower-order models, which exhibit stronger memory effects, preserve more historical information. We observe CPU time with respect to memory length M for $\beta=0.5$. The CPU time increases as the memory length M becomes larger. This indicates that the proposed short memory model is considerably more advantageous in terms of computational efficiency.

In Fig. 10, we observe CPU time for increasing the memory length M for Eq. (3.4). We then compare the CPU time and the reference time, which is defined as the CPU

Table 3: The maximum values of the numerical solutions for the short memory normalized TF 3D diffusion equation under different parameters.

$\beta\backslash M$	0.005	0.01	0.02	0.03	0.04	0.05
0.01	0.2499	0.2720	0.3032	0.3315	0.3521	0.4019
0.50	0.2423	0.2568	0.2820	0.3012	0.3219	0.3421
0.99	0.2289	0.2293	0.2302	0.2309	0.2314	0.2316

time when $M = T$. We present the ratio of each case to the reference time. Except for increasing the memory length M, the number of memory intervals M_t, we use the same set of parameters as those in the experiment shown in Fig. 8. As the memory length M decreases, the CPU time is significantly reduced, leading to a higher ratio compared to the full memory model. However, as M increases and approaches T, the CPU time converges to the reference value, and the ratio gradually approaches one. These results demonstrate that using a shorter memory length substantially improves computational efficiency, whereas longer memory lengths diminish the computational advantage.

4 Conclusions

In this study, we introduced a short memory normalized TF diffusion equation to resolve the high computational cost associated with traditional TF models. By truncating

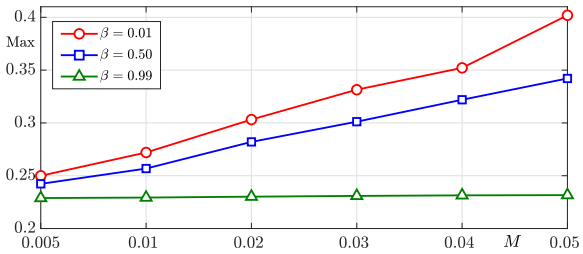


Figure 9: The maximum values with respect to memory length M for various β in 3D experiments.

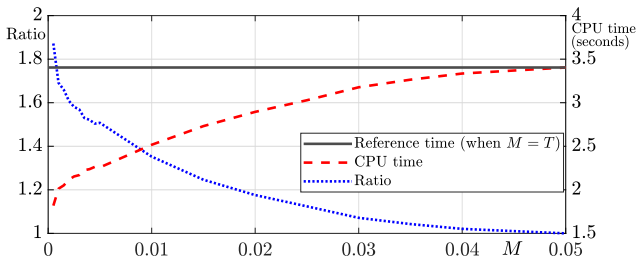


Figure 10: CPU time and Ratio with respect to memory length M for $\beta = 0.5$ in 3D.

the integral over a fixed memory interval and normalizing the weighting function, the proposed model preserves the essential characteristics of fractional dynamics while significantly improving computational efficiency. Through 2D and 3D numerical experiments, we demonstrated that the memory length and the fractional order jointly influence the temporal evolution of the system. In particular, smaller memory lengths and larger fractional orders lead to an evolution similar to that of the full memory model with larger fractional orders, while smaller fractional orders preserve more historical information, which results in slower decay. The numerical results further verified that the short memory approach drastically reduces computational time without compromising the accuracy of the solution under larger fractional orders. These computational results suggest that the proposed short memory normalized model is an effective and practical method for simulating anomalous diffusion phenomena over long time spans and in high-dimensional domains such as 2D and 3D spaces.

Acknowledgments

The first author Junxiang Yang is supported by the Macau Science and Technology Development Fund (FDCT) (No. FDCT-24-080-SCSE, No. FDCT-25-076-SCSE). The corresponding author (J.S. Kim) was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A2C1003844). The authors would like to express sincere appreciation to the reviewer for their constructive feedback. The authors are particularly grateful for the detailed proof of the conservation property for the proposed normalized time-fractional model.

References

[1] D. Vignesh, S. He, S. Banerjee, Modelling discrete time fractional Rucklidge system with complex state variables and its synchronization, *Applied Mathematics and Computation* 455 (2023) 128111.

[2] O. O. Zhokh, P. E. Strizhak, Time-fractional fabric to quantify non-Fickian diffusion in porous media: New vision from previous studies, *Communications in Nonlinear Science and Numerical Simulation* 137 (2024) 108122.

[3] A. Muliana, A fractional model of nonlinear multiaxial viscoelastic behaviors, *Mechanics of Time-Dependent Materials* 27 (2023) 1187–1207.

[4] Y. Qiao, X. Wang, H. Xu, H. Qi, Numerical analysis for viscoelastic fluid flow with distributed/variable order time fractional Maxwell constitutive models, *Applied Mathematics and Mechanics-English Edition* 42 (2021) 1771–1786.

[5] L. Liu, S. Chen, L. Feng, J. Zhu, J. Zhang, L. Zheng, C. Xie, A novel distributed order time fractional model for heat conduction, anomalous diffusion, and viscoelastic flow problems, *Computers & Fluids* 265 (2023) 105991.

[6] R. K. Gazizov, S. Y. Lukashchuk, Higher-order symmetries of a time-fractional anomalous diffusion equation, *Mathematics* 9 (2021) 216.

[7] P. Zhou, H. Jafari, R. M. Ganji, S. M. Narsale, Numerical study for a class of time fractional diffusion equations using operational matrices based on Hosoya polynomial, *Electronic Research Archive* 31 (2023) 4530–4548.
[8] L. Wei, X. Wei, B. Tang, Numerical analysis of variable-order fractional KdV–Burgers–Kuramoto equation, *Electronic Research Archive* 30 (2022) 1263–1281.
[9] J. Li, Y. Cheng, Barycentric rational interpolation method for solving time-dependent fractional convection-diffusion equation, *Electronic Research Archive* 31 (2023) 4034–4056.
[10] L. Qiao, F. Yang, X. Li, Simultaneous identification of the unknown source term and initial value for the time fractional diffusion equation with local and nonlocal operators, *Chaos, Solitons & Fractals* 189 (2024) 115601.
[11] Q. Xu, X. Ma, Z. Cheng, X. Xiao, Z. Ma, Numerical simulation of conduction problem with evaporation based on a SPH model improved by a fractional order convection-diffusion equation, *Engineering Analysis with Boundary Elements* 155 (2023) 668–681.
[12] Y. Ju, Z. Liu, J. Yang, Q. Xu, Meshfree methods for the nonlinear variable-order fractional advection-diffusion equation, *Engineering Analysis with Boundary Elements* 156 (2023) 126–143.
[13] A. Lapin, Mesh scheme for a phase transition problem with time-fractional derivative, *Russian Journal of Numerical Analysis and Mathematical Modelling* 37 (2022) 149–158.
[14] A. Charkaoui, A. Ben-Loghfyry, A novel multi-frame image super-resolution model based on regularized nonlinear diffusion with Caputo time fractional derivative, *Communications in Nonlinear Science and Numerical Simulation* 139 (2024) 108280.
[15] R. Y. Cai, L. Cheng, H. C. Zhou, Sliding mode control for the stabilization of fractional heat equations subject to boundary uncertainty, *Chaos, Solitons & Fractals* 181 (2024) 114718.
[16] Y. Zhao, Y. Tang, Critical behavior of a semilinear time fractional diffusion equation with forcing term depending on time and space, *Chaos, Solitons & Fractals* 178 (2024) 114309.
[17] Z. Tan, α-robust analysis of fast and novel two-grid FEM with nonuniform L1 scheme for semilinear time-fractional variable coefficient diffusion equations, *Communications in Nonlinear Science and Numerical Simulation* 131 (2024) 107830.
[18] W. An, X. Zhang, An implicit fully discrete compact finite difference scheme for time fractional diffusion-wave equation, *Electronic Research Archive* 32 (2024) 354–369.
[19] Y. Jiang, H. Chen, T. Sun, C. Huang, Efficient L1-ADI finite difference method for the two-dimensional nonlinear time-fractional diffusion equation, *Applied Mathematics and Computation* 471 (2024) 128609.
[20] A. T. Nguyen, C. Yang, On a time-space fractional diffusion equation with a semilinear source of exponential type, *Electronic Research Archive* 30 (2022) 1354–1373.
[21] C. Lee, Y. Nam, M. Bang, S. Ham, J. Kim, Numerical investigation of the dynamics for a normalized time-fractional diffusion equation, *AIMS Mathematics* 9 (2024) 26671–26687.
[22] H. Wang, X. Li, L. Zhao, W. Deng, Multiscale modeling and simulation for anomalous and nonergodic dynamics: From statistics to mathematics, *Fundamental Research* 6 (2026) 659–671.
[23] W. Deng, Short memory principle and a predictor-corrector approach for fractional differential equations, *Journal of Computational and Applied Mathematics* 206 (2007) 174–188.
[24] G. C. Wu, M. Luo, L. L. Huang, S. Banerjee, Short memory fractional differential equations for new memristor and neural network design, *Nonlinear Dynamics* 100 (2020) 3611–3623.
[25] X. Hai, Y. Yu, C. Xu, G. Ren, Stability analysis of fractional differential equations with the short-term memory property, *Fractional Calculus and Applied Analysis* 25 (2022) 962–994.
[26] Y. Wei, Y. Chen, S. Cheng, Y. Wang, A note on short memory principle of fractional calculus,

Fractional Calculus and Applied Analysis 20 (2017) 1382–1404.
[27] M. Jornet, J. J. Nieto, Power-series solution of the L-fractional logistic equation, *Applied Mathematics Letters* 154 (2024) 109085.
[28] L. Huang, Y. Huang, X. F. Li, Approximate solution of Abel integral equation, *Computers & Mathematics with Applications* 56 (2008) 1748–1757.
[29] Y. Luchko, Fractional derivatives and the fundamental theorem of fractional calculus, *Fractional Calculus and Applied Analysis* 23 (2020) 939–966.
[30] J. Wang, Q. Liu, K. Chen, J. Yang, Z. Han, S. Kwak, Y. Nam, S. Ham, J. Kim, An efficient operator splitting method for a normalized time-fractional Allen–Cahn equation, *Fractals* 33 (2025) 2550085.
[31] J. Kim, Nonlinear dynamic evolution of a novel normalized time-fractional Burgers equation, *Partial Differential Equations and Applications* 13 (2025) 101096.
[32] J. Kim, Influence of fractional order on the behavior of a normalized time-fractional SIR model, *Mathematics* 12 (2024) 3081.
[33] J. Wang, K. Chen, J. Kim, Computational analysis of a normalized time-fractional Fokker–Planck equation, *Physica A: Statistical Mechanics and its Applications* 665 (2025) 130500.
[34] S. Kwak, Y. Nam, S. Kang, J. Kim, Computational analysis of a normalized time-fractional Fisher equation, *Applied Mathematics Letters* 166 (2025) 109542.
[35] H. G. Lee, S. Kwak, S. Ham, Y. Hwang, J. Kim, The normalized time-fractional Cahn–Hilliard equation, *Chaos, Solitons & Fractals* 198 (2025) 116450.
[36] L. Tan, M. Huang, W. Ying, A GPU-accelerated Cartesian grid method for the heat, wave and Schrödinger equations on irregular domains, *Communications in Computational Physics* 36 (2024) 485–520.
[37] W. Li, P. Huang, Y. He, A time variable filter approach for the fluid-fluid interaction, *Communications in Computational Physics* 37 (2025) 1417–1451.
[38] S. Kwak, Practical implementation of boundary conditions in the Thomas algorithm, *Journal of the Korean Society for Industrial and Applied Mathematics* 29 (2) (2025) 171–183.