AIMS Mathematics, 10(3): 6195–6208.
DOI: 10.3934/math.2025282
Received: 30 January 2025
Revised: 08 March 2025
Accepted: 12 March 2025
Published: 19 March 2025

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

Research article

A normalized Caputo–Fabrizio fractional diffusion equation

Junseok Kim*

Department of Mathematics, Korea University, Seoul 02841, Republic of Korea

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

Abstract: We propose a normalized Caputo–Fabrizio (CF) fractional diffusion equation. The CF fractional derivative replaces the power-law kernel in the Caputo derivative with an exponential kernel, which avoids singularities. Compared to the Caputo derivative, the CF derivative is better suited for systems where memory effects decay smoothly rather than following a power law. However, the kernel is not normalized in the sense that its weighting function does not integrate to unity. To resolve this limitation, we develop a modified formulation that ensures proper normalization. To investigate the fractional order's effect on evolution dynamics, we perform computational tests that highlight memory effects.

Keywords: normalized Caputo–Fabrizio fractional derivative; Caputo derivative; diffusion equation
Mathematics Subject Classification:35R11, 80M20, 39A14

1. Introduction

In this article, we present a normalized Caputo–Fabrizio (CF) fractional diffusion equation and perform computational experiments to analyze the impact of the fractional order on the evolution dynamics. The classical fractional CF derivative [1] is as follows:

$$\frac{\partial^{\alpha} u(x,t)}{\partial t^{\alpha}} = \frac{M(\alpha)}{1-\alpha}\int_0^t \frac{\partial u(x,s)}{\partial s}\exp\left[-\frac{\alpha(t-s)}{1-\alpha}\right]ds, \quad 0<\alpha<1, \tag{1.1}$$

where $M(\alpha)$ is a normalization function such that $M(0) = M(1) = 1$. We note that the corresponding weighting function for the original CF fractional derivative, Eq (1.1), is

$$w_{\alpha}^{t}(s) = \frac{M(\alpha)}{1-\alpha}\exp\left[-\frac{\alpha(t-s)}{1-\alpha}\right]. \tag{1.2}$$

The fractional CF derivative is a modified version of the following Caputo derivative [2], defined as follows:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial u(x,s)}{\partial s} \frac{ds}{(t-s)^\alpha}, \quad 0 < \alpha < 1, \tag{1.3}$$

where $\Gamma(z) = \int_0^\infty \tau^{z-1} e^{-\tau} d\tau$ is the gamma function. The CF fractional derivative can be obtained by replacing the kernel $(t-s)^{-\alpha}$ with $\exp\left[-\frac{\alpha(t-s)}{1-\alpha}\right]$. Another famous time-fractional derivative is the Riemann–Liouville derivative [3]:

$${}_0^{RL}D_t^\alpha u(x,t) = \frac{1}{\Gamma(n-\alpha)} \frac{\partial}{\partial t} \int_0^t u(x,s)(t-s)^{-\alpha}\, ds, \quad 0 < \alpha < 1. \tag{1.4}$$

We note that each method has its own advantages and disadvantages; hence, it is essential to apply them according to their characteristics. The Riemann–Liouville derivative, with its power-law kernel, is well-suited for theoretical analysis and mathematical formulations but may face challenges in handling initial conditions. On the other hand, the Caputo–Fabrizio derivative, with its exponential kernel, offers better stability and preserves initial conditions; therefore, it is more practical for applications in engineering and physics. By understanding their respective strengths, one can select the appropriate derivative to achieve optimal results in various fields.

The classical fractional CF derivative [1] has been extensively applied in various applications. For instance, Rahman et al. [4] analyzed a fractional-order mathematical model for COVID-19 transmission using the CF derivative, established the existence and uniqueness of the solution via fixed-point theory, applied the Laplace Adomian decomposition algorithm to derive an infinite series solution, and demonstrated that the new derivative yields improved results compared to classical order derivatives. Chauhan et al. [5] studied the dynamics of a fractional financial model by examining interactions among interest rates, investment demand, the price index, and savings amounts. The existence and uniqueness of the solution were established using Banach's fixed point theory, and numerical simulations based on the CF derivative illustrated how parameter variations influence the model's behavior, revealing potential chaotic dynamics. Sivashankar et al. [6] developed a fractional-order mathematical model for SARS-CoV-2 transmission using the CF derivative, established existence and uniqueness results via Banach's fixed point theorem, analyzed Hyers–Ulam stability, and demonstrated the model's efficiency and accuracy. Arshad et al. [7] proposed a novel numerical method based on the fractional-order CF derivative in the Caputo sense and analyzed its convergence, stability, and effectiveness through numerical experiments. Dehingia et al. [8] explored the dynamics of a nutrient–plankton system by incorporating Caputo and CF fractional operators and analyzed their influence on stability. They studied the existence, uniqueness, and non-negativity of solutions, identified equilibria, and analyzed stability conditions, with numerical simulations showing that the CF fractional derivative yields dynamics closer to the integer-order system than the Caputo fractional derivative. Kumar et al. [9] investigated intraspecific infectious rivalry using a nonlinear dynamic model with multiple individual categories and applied the CF fractional derivative to analyze system complexity and parameter sensitivities. Existence, uniqueness, and stability were examined through fixed point theory and the Ulam-Hyers criterion. Alsidrani et al. [10] reviewed recent analytical and numerical approaches for fractional partial differential equations (FPDEs) and considered various

fractional derivatives, including those in the senses of Riemann–Liouville, Caputo, and Atangana–Baleanu, while also discussing exact solutions and highlighting numerical techniques to enhance accuracy and efficiency.

However, the kernel of the CF fractional derivative is not properly normalized because its weighting function does not integrate to unity. Specifically, the weight function (1.2) satisfies

$$\int_0^t w_\alpha^t(s)ds = \frac{M(\alpha)}{\alpha}\left[1 - \exp\left(\frac{\alpha t}{\alpha - 1}\right)\right], \quad \text{for} \quad 0 < \alpha < 1, \quad t > 0, \tag{1.5}$$

which indicates that the total weight of the function is not equal to unity and varies with both the fractional order α and time t.

To resolve this issue, we introduce a revised formulation that preserves the advantageous properties of the CF derivative while ensuring normalization. The contribution of the normalized fractional derivative lies in its ability to maintain these desirable characteristics while providing a properly normalized framework. We propose a normalized CF fractional diffusion equation:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{\partial^2 u(x,t)}{\partial x^2}, \tag{1.6}$$

$$u(x,0) = u_0(x), \quad x \in \Omega = (0,\ 1), \tag{1.7}$$

$$u(0,t) = u(1,t) = 0, \quad t \geq 0, \tag{1.8}$$

where $u(x,t)$ is the concentration. Let us propose the following normalized fractional CF derivative:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \int_0^t N(\alpha,t)\exp\left(\frac{\alpha s}{1-\alpha}\right)\frac{\partial u(x,s)}{\partial s}ds, \quad 0 < \alpha < 1, \tag{1.9}$$

where $N(\alpha,t)$ is a function of α and t that satisfies

$$\int_0^t N(\alpha,t)\exp\left(\frac{\alpha s}{1-\alpha}\right)ds = 1, \quad 0 < \alpha < 1,\ t > 0. \tag{1.10}$$

By solving Eq (1.10), we obtain $N(\alpha,t) = \alpha/\left\{(1-\alpha)\left[\exp\left(\frac{\alpha t}{1-\alpha}\right) - 1\right]\right\}$. Therefore, we can rewrite Eq (1.9) as follows:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \int_0^t w_\alpha^t(s)\frac{\partial u(x,s)}{\partial s}ds, \quad 0 < \alpha < 1, \tag{1.11}$$

where $w_\alpha^t(s)$ is a weight function

$$w_\alpha^t(s) = N(\alpha,t)\exp\left(\frac{\alpha s}{1-\alpha}\right) = \frac{\alpha\exp\left(\frac{\alpha s}{1-\alpha}\right)}{(1-\alpha)\left[\exp\left(\frac{\alpha t}{1-\alpha}\right) - 1\right]}, \tag{1.12}$$

which satisfies $\int_0^t w_\alpha^t(s)ds = 1$, for all $0 < \alpha < 1$, $t > 0$. Figure 1 shows weight functions $w_\alpha^t(s)$ for different values of α, where (a) corresponds to $t = 1$ and (b) corresponds to $t = 2$. The graphs illustrate the effect of $\alpha = 0.1$, 0.8, and 0.9. As s increases, the functions exhibit rapid growth for larger α, particularly when $t = 2$. The red dashed, blue dotted, and green dashed curves indicate

$\alpha = 0.1$, 0.8, and 0.9, respectively, and each case demonstrates distinct behavior in the weight function. For a normalized Caputo fractional diffusion derivative, see [11–13]. In summary, the normalized CF fractional diffusion equation allows for a comprehensive study of the effects of the fractional parameter on diffusion dynamics and ensures a fair comparison.

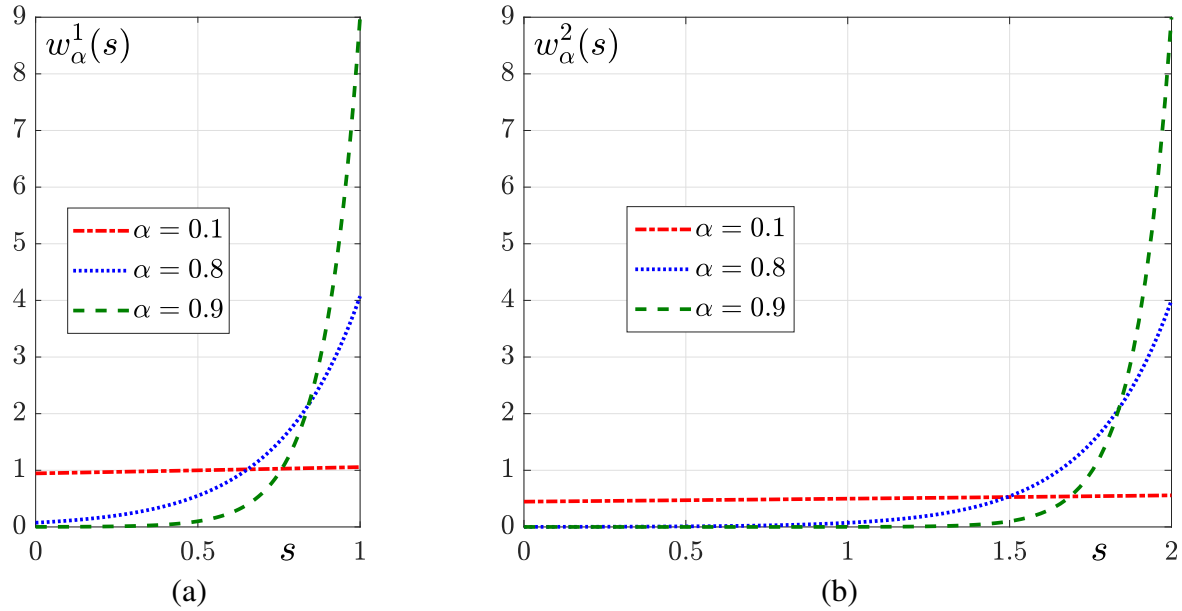

Figure 1. (a) and (b) are weight functions $w_\alpha^t(s)$ for various α values with $t = 1$ and $t = 2$, respectively. Here, $\alpha = 0.1,\ 0.8$, and 0.9 are used.

We note that the nature of diffusion can be classified based on the order of the derivative in the governing equation. Classical diffusion follows Fick's law and is modeled by the standard diffusion equation, which involves a second-order spatial derivative and a first-order time derivative. This corresponds to normal diffusion, where the mean square displacement (MSD) scales linearly with time, $\langle x^2 \rangle \sim t$. When the order of the time derivative is fractional (e.g., $0 < \alpha < 1$), the process is governed by the time-fractional diffusion equation, representing subdiffusion. Here, particles experience trapping or memory effects, leading to a sublinear MSD scaling, $\langle x^2 \rangle \sim t^\alpha$. In contrast, if the spatial derivative is of fractional order (e.g., Lévy flights), superdiffusion occurs, where particles undergo long jumps, resulting in a superlinear MSD, $\langle x^2 \rangle \sim t^\beta$ with $\beta > 1$. Thus, normal diffusion is associated with integer-order derivatives, subdiffusion with time-fractional derivatives, and superdiffusion with space-fractional derivatives. Analyzing the variation of the derivative order in diffusion equations allows us to categorize and predict the diffusion type in complex systems [14, 15].

The organization of this paper is presented as follows. Section 2 provides a detailed description of the numerical discretization scheme developed for the proposed normalized CF fractional diffusion equation. Section 3 presents computational tests that investigate the influence of the fractional order on the evolution dynamics, which offers insights into its effects on the system's behavior. Finally, Section 4 summarizes the key findings of the study and discusses potential directions for future research. The MATLAB source code is provided in the appendix for interested readers.

2. Discretization

Let $\Omega = (0,\ 1)$ and $\Omega_h = \{x_i | x_i = (i-1)h,\ i = 1, \ldots, N_x\}$, where $h = 1/(N_x - 1)$ for some positive integer N_x, see Figure 2. Let us assume $u_i^n = u(x_i,\ t_n)$, where $t_n = \Delta t(n-1)$.

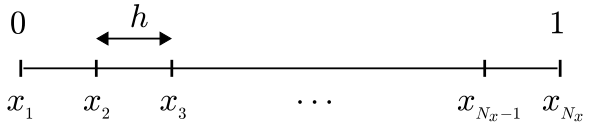

Figure 2. Discrete computational spatial domain.

We discretize Eq (1.11) as follows:

$$\begin{aligned}\frac{\partial^\alpha u(x_i, t_{n+1})}{\partial t^\alpha} &= \int_0^{t_{n+1}} w_\alpha^{t_{n+1}}(s)\frac{\partial u(x_i, s)}{\partial s}ds = \sum_{p=1}^{n}\int_{t_p}^{t_{p+1}} w_\alpha^{t_{n+1}}(s)\frac{\partial u(x_i, s)}{\partial s}ds \\ &\approx \sum_{p=1}^{n}\frac{u_i^{p+1}-u_i^p}{\Delta t}\int_{t_p}^{t_{p+1}} w_\alpha^{t_{n+1}}(s)ds = \sum_{p=1}^{n}\frac{u_i^{p+1}-u_i^p}{\Delta t}\frac{\exp\left(\frac{\alpha t_{p+1}}{1-\alpha}\right)-\exp\left(\frac{\alpha t_p}{1-\alpha}\right)}{\exp\left(\frac{\alpha t_{n+1}}{1-\alpha}\right)-1}. \end{aligned} \tag{2.1}$$

Then, we obtain a system of discrete equations from Eq (1.6):

$$\sum_{p=1}^{n} W_p^n \frac{u_i^{p+1}-u_i^p}{\Delta t} = \frac{u_{i-1}^{n+1}-2u_i^{n+1}+u_{i+1}^{n+1}}{h^2}, \quad \text{for } i = 2, \ldots, N_x - 1, \tag{2.2}$$

where

$$W_p^n = \frac{\exp\left(\frac{\alpha t_{p+1}}{1-\alpha}\right)-\exp\left(\frac{\alpha t_p}{1-\alpha}\right)}{\exp\left(\frac{\alpha t_{n+1}}{1-\alpha}\right)-1}, \tag{2.3}$$

which satisfies $\sum_{p=1}^{n} W_p^n = 1$. Hence, Eq (2.2) can be reformulated as follows:

$$-\frac{1}{h^2}u_{i-1}^{n+1} + \left(\frac{W_n^n}{\Delta t}+\frac{2}{h^2}\right)u_i^{n+1} - \frac{1}{h^2}u_{i+1}^{n+1} = \frac{W_n^n}{\Delta t}u_i^n - \sum_{p=1}^{n-1} W_p^n \frac{u_i^{p+1}-u_i^p}{\Delta t}, \tag{2.4}$$

which can be expressed as

$$A\mathbf{u}^{n+1} = \mathbf{f}, \tag{2.5}$$

where

$$A = \begin{pmatrix} \frac{W_n^n}{\Delta t}+\frac{2}{h^2} & -\frac{1}{h^2} & 0 & \cdots & 0 & 0 & 0 \\ -\frac{1}{h^2} & \frac{W_n^n}{\Delta t}+\frac{2}{h^2} & -\frac{1}{h^2} & \cdots & 0 & 0 & 0 \\ 0 & -\frac{1}{h^2} & \frac{W_n^n}{\Delta t}+\frac{2}{h^2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{1}{h^2} & \frac{W_n^n}{\Delta t}+\frac{2}{h^2} & -\frac{1}{h^2} \\ 0 & 0 & 0 & \cdots & 0 & -\frac{1}{h^2} & \frac{W_n^n}{\Delta t}+\frac{2}{h^2} \end{pmatrix},$$

$$\mathbf{u}^{n+1} = \begin{pmatrix} u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_{N_x-1}^{n+1} \end{pmatrix}, \text{ and } \mathbf{f} = \begin{pmatrix} W_n^n u_2^n/\Delta t - F \\ W_n^n u_3^n/\Delta t - F \\ \vdots \\ W_n^n u_{N_x-1}^n/\Delta t - F \end{pmatrix}.$$

Here, $F = \sum_{p=1}^{n-1} W_p^n (u_i^{p+1} - u_i^p)/\Delta t$ and $u_1^{n+1} = u_{N_x}^{n+1} = 0$ are used. Equation (2.5) is solved by using the Thomas algorithm, which is an efficient method for solving tridiagonal systems of equations. It is a specialized form of Gaussian elimination with a computational complexity of $O(N_x)$. The matrix A has nonzero elements only on the main diagonal and the first sub- and super-diagonals. The Thomas algorithm consists of two steps: *Step 1*. Forward elimination: Modify the coefficients to transform the system into an upper triangular form. *Step 2*. Backward substitution: Solve for $u_{N_x-1}^{n+1}$ first and then compute previous values sequentially. This method is widely used in numerical solutions of differential equations, particularly in finite difference methods [16, 17].

3. Computational tests

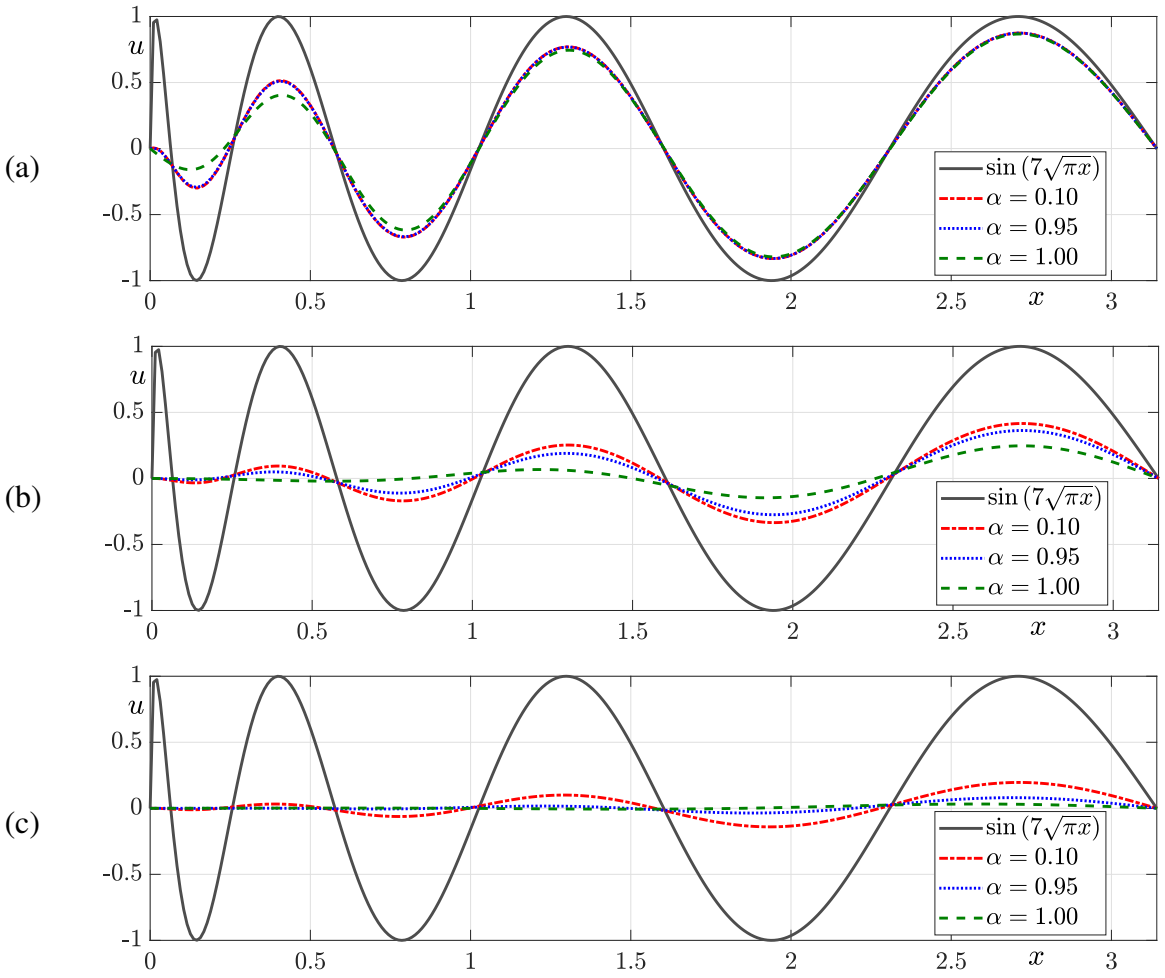

Figure 3. (a)–(c) are the computational results at three time points, $t = 0.01$, 0.1, and 0.3, respectively, for $\alpha = 0.1, 0.95$, and 1.

This section presents several numerical experiments to study the influence of α on the dynamic evolution of the normalized CF fractional diffusion equation. We consider an initial condition on $\Omega = (0, \pi)$:

$$u(x, 0) = \sin\left(7\sqrt{\pi x}\right), \tag{3.1}$$

which consists of multiple modes and exhibits a progressively varying profile. Here, $N_x = 300$, $h = \pi/N_x$, $\Delta t = 10^{-3}$ are used. Figure 3 shows computational results for different values of α at three distinct time points: $t = 0.01$, $t = 0.1$, and $t = 0.3$. Subplots (a)–(c) correspond to these time instances, respectively. The curves compare solutions for $\alpha = 0.1$, 0.95, and 1 and show their evolution over time. For larger α, the solution decays more rapidly, while for smaller α, it decays more slowly. The difference becomes more pronounced in high-frequency regions, where the effect of α on the solution's decay is more significant. These results highlight the impact of fractional parameters on diffusion behavior.

Next, we conduct tests to examine the influence of the fractional order on memory dynamics. To observe this, we include a source term that depends on both time and space:

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} = \frac{\partial^2 u(x,t)}{\partial x^2} + s(x,t), \quad \text{for } (x,t) \in (0,\ 3) \times (0,\ 0.1], \tag{3.2}$$

$$u(x, 0) = \sin(\pi x/3), \quad x \in (0,\ 3), \tag{3.3}$$

$$u(0, t) = u(3, t) = 0, \quad t \geq 0. \tag{3.4}$$

Here, we consider three types of source terms:

$$s_1(x,t) = g(x)\left\{1 - \tanh\left[300\left(t - \frac{1}{30}\right)\right]\right\}, \tag{3.5}$$

$$s_2(x,t) = g(x)\left\{\tanh\left[300\left(t - \frac{1}{30}\right)\right] - \tanh\left[300\left(t - \frac{2}{30}\right)\right]\right\}, \tag{3.6}$$

$$s_3(x,t) = g(x)\left\{1 + \tanh\left[300\left(t - \frac{2}{30}\right)\right]\right\}, \tag{3.7}$$

where

$$g(x) = \frac{10}{1 + \exp\left[50\,(x-2)^2\right]} - \frac{10}{1 + \exp\left[50\,(x-1)^2\right]}. \tag{3.8}$$

Figure 4(a,c,e) displays mesh plots of the three different source terms $s_1(x,t)$, $s_2(x,t)$, and $s_3(x,t)$, respectively. They show their variation over the spatial domain x and temporal domain t. Figure 4(b,d,f) illustrates the computational results of $u(x,t)$ at $t = 0.1$ corresponding to the respective source terms in Figure 4(a,c,e). The influence of the fractional order α on $u(x,t)$ is analyzed using different values: $\alpha = 0.1$, $\alpha = 0.95$, and $\alpha = 1$.

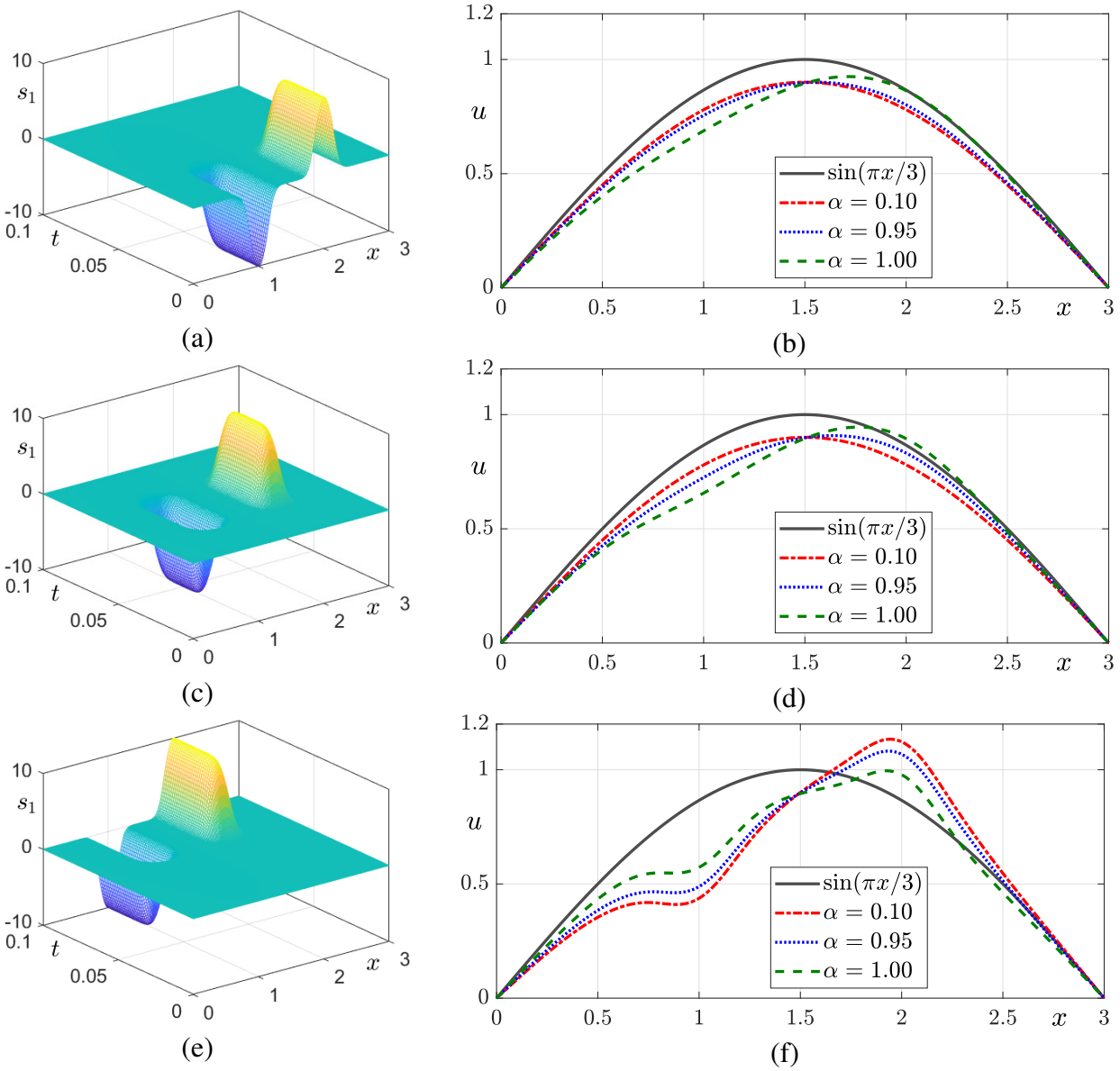

Figure 4. (a), (c), and (e) are the mesh plots of the source terms $s_1(x,t)$, $s_2(x,t)$, and $s_3(x,t)$, respectively. (b), (d), and (f) are the computational results of $u(x,t)$ at time $t = 0.1$ corresponding to the source terms shown in (a), (c), and (e), respectively.

The solid black curve represents the initial condition $\sin(\pi x/3)$. As α varies, the solution $u(x,t)$ changes, which illustrates the impact of fractional-order dynamics on the evolution of $u(x,t)$. As shown in Figure 3, the temporal evolution accelerates with increasing fractional order α in the absence of a source term. When the source term is influential at early times, as in the case of $s_1(x,t)$ (see Figure 4(a)), it significantly affects the dynamics for higher fractional order α values, as shown in Figure 4(b). Similarly, when the source term becomes effective at intermediate times, as in the case of $s_2(x,t)$ (see Figure 4(c)), it also has a greater impact on the dynamics for higher α values, as illustrated in Figure 4(d).

However, when the source term dominates at later times, as in the case of $s_3(x,t)$ (see Figure 4(e)), its influence on the dynamics is more pronounced for lower α values, as shown in Figure 4(f). This phenomenon can be explained by the definition of the normalized CF fractional derivative. When α is small, the derivative approximates a uniform average of historical first-order derivatives. To compensate for nonzero source term values at later times, the solution must exhibit significant variations to ensure that the average aligns with the source values. The effect of various source terms is evident, as each source function leads to a distinct profile in $u(x,t)$. The computational results demonstrate the sensitivity of the solution to both the fractional order α and the choice of source terms, which emphasizes the importance of these parameters in modeling diffusion processes.

Figure 5 presents the effects of various source terms and fractional orders on the solution $u(x,t)$ of the given normalized CF fractional diffusion equation. The top row (a)–(c) shows the 3D mesh plots of the source terms $s_1(x,t)$, $s_2(x,t)$, and $s_3(x,t)$, respectively. Each source term represents a different temporal and spatial influence on the system. The second, third, and fourth rows (d)–(l) show the computational results of $u(x,t)$ for different values of the fractional order α. Each column corresponds to the results for a specific source term from the top row. The choice of source term significantly impacts the solution profile, which affects both amplitude and evolution over time. These computational results effectively demonstrate how fractional-order derivatives and source terms influence the behavior of time-dependent diffusion processes.

The most notable results are observed in the third column. Unlike the other source terms, such as $s_1(x,t)$ and $s_2(x,t)$, the third source term, $s_3(x,t)$, which becomes nonzero at later times, particularly emphasizes the impact of the time-fractional order α on the evolution dynamics. As shown in the third column, when α is small, the normalized time-fractional derivative approximates an average of the first derivatives, which leads to significant changes in $u(x,t)$ at later times to align with the nonzero source values.

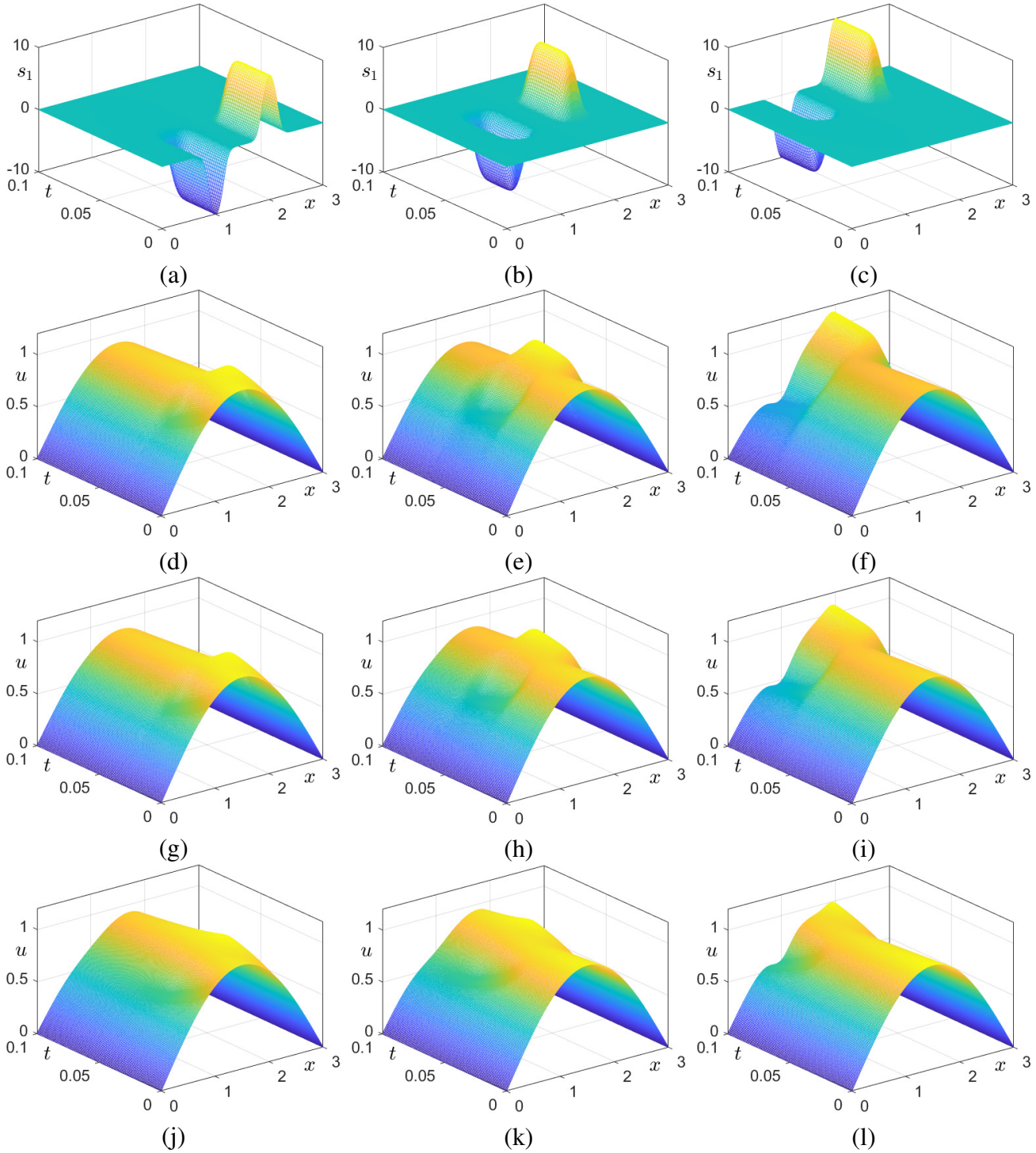

Figure 5. (a)–(c) are the mesh plots of the source terms $s_1(x,t)$, $s_2(x,t)$, and $s_3(x,t)$, respectively. Each column presents computational results corresponding to the source term shown in the top row. The second, third, and fourth rows are the computational results of $u(x,t)$ with $\alpha = 0.1$, $\alpha = 0.95$, and $\alpha = 1$, respectively.

4. Conclusions

In this article, we presented a normalized CF fractional diffusion equation to resolve the lack of proper normalization in the classical CF derivative. By introducing a modified formulation that ensures normalization while preserving the desirable properties of the original derivative, we provided a more consistent framework for modeling diffusion processes with memory effects. To analyze the influence of fractional order on the evolution dynamics, we implemented a numerical discretization scheme and conducted computational experiments. The computational results demonstrated that the fractional order α significantly influences the temporal evolution of the solution, with higher values of α leading to faster dynamics. Additionally, the presence and timing of source terms played a crucial role in developing the solution profile, particularly for lower values of α, where stronger memory effects induced larger variations in the solution at later times. Our findings highlight the importance of fractional-order derivatives in capturing anomalous diffusion behavior and suggest that the proposed normalized CF derivative can serve as a reliable tool for modeling diffusion processes such as shape transformation [18] and phase separation [19]. This study focused on proposing a new normalized Caputo–Fabrizio fractional diffusion equation and numerically analyzing its evolution dynamics. To this end, standard first-order temporal and second-order spatial numerical schemes were adopted. Therefore, developing temporally high-order numerical schemes for the proposed model would be a valuable future direction.

Use of Generative-AI tools declaration

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

Acknowledgments

This work was supported by the National Research Foundation(NRF), Korea, under project BK21 FOUR. I sincerely appreciate the reviewers for their valuable comments and suggestions, which have greatly improved the quality of this paper.

Conflict of interest

The author declares no conflict of interest in this paper.

References

1. M. Caputo, M. Fabrizio, A new definition of fractional derivative without singular kernel, *Progr. Fract. Differ. Appl.*, **1** (2015), 73–85.
2. R. B. Albadarneh, I. Batiha, A. K. Alomari, N. Tahat, Numerical approach for approximating the Caputo fractional-order derivative operator, *AIMS Math.*, **6** (2021), 12743–12756. https://doi.org/10.3934/math.2021735 .

3. A. Atangana, J. R. Gómez-Aguilar, Numerical approximation of Riemann-Liouville definition of fractional derivative: from Riemann-Liouville to Atangana-Baleanu, *Numer. Methods Partial Differ. Equ.*, **34** (2018), 1502–1523. https://doi.org/10.1002/num.22195 .
4. M. ur Rahman, S. Ahmad, R. T. Matoog, N. A. Alshehri, T. Khan, Study on the mathematical modelling of COVID-19 with Caputo-Fabrizio operator, *Chaos Soliton. Fract.*, **150** (2021), 111121. https://doi.org/10.1016/j.chaos.2021.111121 .
5. R. P. Chauhan, S. Kumar, B. S. T. Alkahtani, S. S. Alzaid, A study on fractional order financial model by using Caputo–Fabrizio derivative, *Results Phys.*, **57** (2024), 107335. https://doi.org/10.1016/j.rinp.2024.107335 .
6. M. Sivashankar, S. Sabarinathan, V. Govindan, U. Fernandez-Gamiz, S. Noeiaghdam, Stability analysis of COVID-19 outbreak using Caputo-Fabrizio fractional differential equation, *AIMS Math.*, **8** (2023), 2720–2735. https://doi.org/10.3934/math.2023143 .
7. S. Arshad, I. Saleem, A. Akgül, J. Huang, Y. Tang, S. M. Eldin, A novel numerical method for solving the Caputo–Fabrizio fractional differential equation, *AIMS Math.*, **8** (2023), 9535–9556. https://doi.org/10.3934/math.2023481 .
8. K. Dehingia, S. Boulaaras, S. Gogoi, On the dynamics of a nutrient–plankton system with Caputo and Caputo–Fabrizio fractional operators, *J. Comput. Sci.*, **76** (2024), 102232. https://doi.org/10.1016/j.jocs.2024.102232 .
9. P. Kumar, S. Kumar, B. S. T. Alkahtani, S. S. Alzaid, A mathematical model for simulating the spread of infectious disease using the Caputo-Fabrizio fractional-order operator, *AIMS Math.*, **9** (2024), 30864–30897. https://doi.org/10.3934/math.20241490 .
10. F. Alsidrani, A. Kılıçman, N. Senu, A comprehensive review of the recent numerical methods for solving FPDEs, *Open Math.*, **22** (2024), 20240036. https://doi.org/10.1515/math-2024-0036
11. 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
12. M. Jornet, J. J. Nieto, Power-series solution of the L-fractional logistic equation, *Appl. Math. Lett.*, **154** (2024), 109085. https://doi.org/10.1016/j.aml.2024.109085
13. K. A. Lazopoulos, A. K. Lazopoulos, Fractional vector calculus and fractional continuum mechanics, *Progr. Fract. Differ. Appl.*, **2** (2016), 67–86.
14. J. L. Suzuki, M. Gulian, M. Zayernouri, M. D'Elia, Fractional modeling in action: a survey of nonlocal models for subsurface transport, turbulent flows, and anomalous materials, *J. Peridyn. Nonlocal Model.*, **5** (2023), 392–459. https://doi.org/10.1007/s42102-022-00085-2
15. C. Li, Z. Zhao, Y. Chen, Numerical approximation of nonlinear fractional differential equations with subdiffusion and superdiffusion, *Comput. Math. Appl.*, **62** (2011), 855–875. https://doi.org/10.1016/j.camwa.2011.02.045
16. E. W. Cheney, D. R. Kincaid, *Numerical mathematics and computing*, 7 Eds., Thomson Brooks/Cole, 2012.
17. J. W. Thomas, *Numerical partial differential equations: finite difference methods*, Springer, New York, 1995. https://doi.org/10.1007/978-1-4899-7278-1

18. H. Kim, C. Lee, S. Yoon, Y. Choi, J. Kim, A fast shape transformation using a phase-field model, *Extreme Mech. Lett.*, **52** (2022), 101633. https://doi.org/10.1016/j.eml.2022.101633

19. S. Ham, Y. Li, D. Jeong, C. Lee, S. Kwak, Y. Hwang, et al., An explicit adaptive finite difference method for the Cahn–Hilliard equation, *J. Nonlinear Sci.*, **32** (2022), 80. https://doi.org/10.1007/s00332-022-09844-3

Appendix

The following listing 1 is a MATLAB code for the normalized CF fractional diffusion equation.

Listing 1. MATLAB code for the normalized CF fractional diffusion equation.

```
clear;
NX=301; h=pi/NX; x=linspace(0,pi,NX); T=0.1; dt=1.0e-3; NT=round(T/dt);
t=linspace(0,T,NT+1); u=zeros(NX,NT+1); u(:,1)=sin(7*sqrt(pi*x));
a=(-1/h^2)*ones(NX,1); c=a; figure(1); clf; hold on; box on;
plot(x, u(:, 1), 'color',[0.3, 0.3, 0.3], 'LineWidth', 2);
Alpha_set = [0.1 0.95 1]; line_styles = {'r-.', 'b:', '--'};
legendLabels{1}=sprintf('$\\sin\\left(7 \\sqrt{\\pi x} \\right)$');

for bb = 1:length(Alpha_set)
Alpha = Alpha_set(bb);
for m = 1:NT
DENO = exp(Alpha*t(m+1)/(1-Alpha))-1;
for q = 1:m
w(q) = (exp(Alpha*t(q+1)/(1-Alpha))-exp(Alpha*t(q)/(1-Alpha)))/DENO;
end
F = zeros(NX, 1);
if m > 1
for q = 1:m-1
F = F+w(q)*(u(:,q+1)-u(:,q))/dt;
end
end

if Alpha==1
for i = 1:NX
d(i) = 1.0/dt+2.0/h^2;
end
f = 1.0/dt*u(:,m);
else
for i = 1:NX
d(i) = w(m)/dt+2.0/h^2;
end
f = w(m)/dt*u(:,m)-F;
```

```
end
u(2:NX-1, m+1) = Thomas(a(2:NX-1), d(2:NX-1), c(2:NX-1), f(2:NX-1));
end

if bb == 3
plot(x,u(:, end),line_styles{bb},'Color',[0 0.5 0],'LineWidth',2);
else
plot(x,u(:, end),line_styles{bb},'LineWidth',2);
end
legendLabels{bb+1}=sprintf('$\\alpha = %.2f$',Alpha);
end

lgd=legend(legendLabels,'Interpreter','latex','Location', ...
'northwest','fontsize',17);
xlabel('x'); ylabel('u'); axis([x(1) x(end) -1 1]); grid on

function x = Thomas(A, B, G, f)
n = length(f);
for i = 2:n
mul = A(i) / B(i - 1);
B(i) = B(i) - mul * G(i - 1);
f(i) = f(i) - mul * f(i - 1);
end
x(n) = f(n) / B(n);
for i = n-1:-1:1
x(i) = (f(i) - G(i) * x(i + 1)) / B(i);
end
x = x';
end
```

AIMS Press