Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. *, No. *, *, pp. 1-24
DOI:10.22034/cmde.2025.66393.3103

Trigonometric cubic b-spline collocation method for time fractional diffusion equation

Zainab Kammappa and Ashish Awasthi*

Department of Mathematics, National Institute of Technology Calicut, 673601, Kerala, India.

Abstract

This paper presents a numerical approach based on Cubic Trigonometric B-spline (CuTBS) interpolation for solving Time-Fractional Diffusion Equations (TFDEs) involving the Caputo-Fabrizio fractional time derivative. The CuTBS-based scheme effectively combines accurate spatial interpolations with a robust finite-difference discretization for the fractional derivative, ensuring high precision in both temporal and spatial domains. The method is unconditionally stable and demonstrates second-order convergence in time and space. Numerical experiments are conducted to validate the applicability and feasibility of the technique.

Keywords. Cubic Trigonometric B-spline Method, Time-Fractional Diffusion Equation, Caputo–Fabrizio Fractional Derivative, Spline Interpolation, Finite Difference Formulation, Stability.

1991 Mathematics Subject Classification.

1. Introduction

Classical diffusion equations, while successful in many applications, fail to adequately capture complex diffusion processes that exhibit non-local behavior, memory effects, and anomalous scaling properties. In particular, these traditional models assume Markovian processes in which the future state depends only on the present state, ignoring historical influences that may be crucial in many real-world phenomena [9].

Unlike classical diffusion where mean-squared displacement grows linearly with time ($r^2(t) \propto t$), anomalous diffusion follows a power law relationship ($r^2(t) \propto t^\alpha$), where α is not necessarily equal to 1 [21] where $\alpha < 1$ implies subdiffusion which occurs in crowded environments or porous media where particle movement is hindered, $\alpha = 1$ implies normal diffusion which corresponds to classical Brownian motion, $1 < \alpha < 2$ implies superdiffusion which is associated with systems where long jumps are possible and $\alpha = 2$ implies ballistic motion which represents constant velocity movement [4]. This fundamental difference requires the use of fractional-order derivatives to accurately model this behavior [20].

Unlike Riemann-Liouville or standard Caputo derivatives that use singular kernels, the Caputo - Fabrizio fractional derivative (CFFD) employs a non-singular exponential kernel that offers several benefits, such as better representation of heterogeneities in materials and media, more effective modeling of memory effects with exponential decay rather than power law, the ability to capture complex transition phenomena between different diffusion regimes and mathematical advantages in terms of analytical solutions and numerical implementation [1, 5, 13, 38].

The CFFD represents processes where the influence of past states diminishes exponentially over time. This characteristic makes it particularly suitable for modeling diffusion in heterogeneous porous media with complex geometric structures, transport processes in biological systems with cellular obstacles, heat transfer in materials with spatial memory, groundwater contamination and flow in geological formations and financial and economic systems exhibiting non-Markovian behavior [19, 27].

Received: 18 March 2025 ; Accepted: 28 July 2025.
∗ Corresponding author. Email: aawasthi@nitc.ac.in.

The one dimensional non - homogeneous time-fractional diffusion equation (TFDE) considered in this paper is given by,

$$ {}^{CF}_{a}D^{\gamma}_{t}u(x,t) - u_{xx} = f(x,t), \quad (x,t) \in \Omega, \quad 0 < \gamma < 1 \,, \tag{1.1} $$

where

$$ \Omega = [a,b] \times [0,T] \,, $$

with initial condition (IC),

$$ u(x,0) = g(x), \quad a \le x \le b \,, \tag{1.2} $$

and boundary conditions (BCs),

$$ u(a,t) = g_1(t), \quad u(b,t) = g_2(t), \quad 0 \le t \le T. \tag{1.3} $$

Here, the diffusion exponent is denoted by γ, which is a key parameter in characterizing the dynamics of diffusion processes. f, g, g_1, and g_2 are given functions of the variables assumed to be continuous and u is the unknown function, ${}^{CF}_{a}D^{\gamma}_{t}u(x,t)$ is the Caputo - Fabrizio fractional derivative (CFFD) of order γ given by [7]

$$ \begin{aligned} {}^{CF}_{a}D^{\gamma}_{t}u(x,t) &= \frac{\partial^{\gamma}}{\partial t^{\gamma}}u(x,t) \\ &= \frac{R(\gamma)}{1-\gamma}\int_0^t \left[\frac{\partial}{\partial k}u(x,k)\right] e^{\frac{\gamma(t-k)}{\gamma-1}}\,dk, \quad \gamma \in (0,1) \,, \end{aligned} \tag{1.4} $$

where the normalization function $R(\gamma)$ satisfies the condition,

$$ R(0) = R(1) = 1. $$

The mathematical framework for TFDEs is grounded in fractional calculus, which extends traditional calculus to non-integer order derivatives. The Riemann-Liouville and Caputo definitions are the most commonly used fractional derivatives in TFDEs. Oldham and Spanier [25] in 1973 provided foundational insights into fractional calculus, establishing a robust theoretical backdrop for subsequent studies. Following this, the works of Podlubny [31] in 1998 and Gorenflo et al. [11] in 2002 further elucidated the properties and applications of fractional derivatives, emphasizing their utility in modeling real-world phenomena characterized by memory and hereditary effects.

Several researchers have investigated the diffusion equation with the Caputo time fractional derivative in place of the first-order time derivative. In 2006, Zhuang and Liu [51] used a backward difference method to solve TFDE. Sweilam et al. [43] introduced the Crank-Nicolson method to numerically solve TFDEs in 2012. Sun et al. [44] applied a semi-structured finite element method (FEM) to address a range of TFDEs in 2013. Mustapha et al. [24] in 2014 created a discontinuous Petrov-Galerkin method to approximate solutions to TFDEs. Esmaeili and Garrappa [10] in 2015 formulated a pseudo-spectral method for numerical solutions of TFDEs. Tuan et al. [46] in 2019 investigated the inverse problem for 1D TFDEs using a modified regularization method in the frequency domain. An ideal Galerkin FEM in the absence of any regularity assumptions on its real solution was created by Liu et al. in [18] after studying the well-posedness and solution regularity of a variable-order multi-term TFDE in 2022. In 2023, Roul et al. provided a high-order computing technique for a TFDE's numerical solution [36], and in 2024, Poojitha and Awasthi [29, 30] presented a numerical approach based on an operational matrix of Legendre polynomials to solve the class of TFDEs and a spectral approach based on derivatives of orthogonal polynomials to solve the time fractional convection - diffusion - reaction equations.

In 2016, Atangana and Alqahtani explored the groundwater contamination equation and its application in [3], where they established a numerical method for the advection-dispersion equation having CFFD in space and time. In 2017, Mirza and Vieru [22] solved the 2D convection equation in the Caputo-Fabrizio sense using the Laplace and Fourier transforms. In 2019, Liu et al. [17] used a finite difference method (FDM) to analyze and design a quasilinear temporal fractional parabolic PDE having a non-singular kernel and using an iterative Laplace transform technique, Shaikh et al. [40] examined the estimated solutions of the Fisher and Fitzhugh-Nagumo equations that incorporate CFFD. To estimate the duration and extent of the spread, Panday et al. [28] in 2022 investigated the dynamics of the Coronavirus disease 2019 (COVID-19) virus in the population of the human race. Using the CFFD,

the suggested fractional mathematical model was numerically investigated using the Genocchi collocation approach. In [41], a non-singular derivative called the Caputo-Fabrizio fractal fractional derivative was used to examine the tumor growth model both statistically and numerically by Singh et al in 2023 and in 2024, Onitsuka and El-Fassi suggested a generalization of CFFD [26].

The ability of B-splines to handle complex BCs and irregular geometries enhances their applicability in solving fractional differential equations. As research in this area grows, B-spline interpolation is a valuable tool for advancing the numerical analysis of fractional differential equations, providing insight into various applications across scientific disciplines.

Recent advances in B-spline methodologies have significantly expanded their applicability to fractional differential equations. In 2017, a cubic trigonometric B-spline (CuTBS) collocation technique was introduced by Yaseen et al. [47] for the numerical solution of the non-integer order sub-diffusion problem, the authors of [48] put forth an effective numerical technique to approximate the solution of a time-fractional diffusion-wave equation that includes reaction components which are founded on cubic trigonometric basis functions and the nonlinear Burgers' equation was solved using cubic B-splines (CBS) after being linearized to the Heat equation through the Hopf-Cole transformation by Lakshmi and Awasthi [15]. Mohyud-Din et al. put together extended CBS in [23] in 2018 to create a difference scheme to approximate solutions of the time-fractional convection equation, and Lakshmi and Awasthi solved the one-dimensional nonlinear modified Burgers equation using a combination of quintic splines for spatial discretization and the Crank-Nicolson scheme for temporal discretization in [16]. The applications of 5th-degree non-polynomial spline functions for the numerical analysis of the 4th-order fractional ordinary differential equations with product terms were suggested by Khalid et al. [14]. An innovative method for the approximate solution of a class of 4th-order time-fractional PDEs was described by Amin et al. in [2] where time discretization was accomplished through the use of FDM. In contrast, spatial discretization was performed using the non-polynomial quintic spline approach. In 2020, Yaseen and Abbas [49] published a numerical method based on cubic trigonometric B-spline functions for the time fractional Bateman–Burgers problem. Later, Roul and Goura presented an efficient B-spline collocation technique to address the one-dimensional nonlinear Bratu problem [35] and developed a high-order numerical method to solve the time-fractional reaction–diffusion equation, where the fractional derivative was interpreted in the Caputo sense in which a quintic B-spline collocation scheme was used for spatial discretization [33]. In addition, Roul developed a higher-order numerical method for pricing Asian options with a fixed strike price [32]. The original two-dimensional PDE that governed the option value was first reduced to a one-dimensional PDE. The time domain was discretized using the Crank–Nicolson scheme, while a quartic B-spline collocation method was applied for spatial discretization. In 2022, Roul and Kumari addressed the numerical solution of a broad class of nonlinear singular boundary value problems (SBVPs). The approach began by reformulating the original problem to handle the singularity, followed by the development of a numerical scheme using quartic trigonometric B-spline functions to solve the modified equation [34].

Recently in 2023, Roul et al. [37] solved a class of TFDEs numerically effectively. The CFD discretization was performed using the L1 method, while the space discretization was performed using a collocation technique based on the sextic B-spline basis function. In 2024, the authors presented a novel numerical technique based on cubic spline interpolation in [12] to solve Caputo-type fractional differential equations.

This study introduces a CuTBS collocation technique to determine numerical solutions to TFDEs. $\theta-$ weighted scheme is employed to implement the proposed algorithm. This approach utilizes the Caputo–Fabrizio derivative in conjunction with a forward difference method for time discretisation and CuTBS functions to interpolate over the space grid. The analysis of convergence and stability is also addressed to demonstrate that the method does not increase errors. Numerical experiments are conducted to validate the applicability and feasibility of the procedure. The results are juxtaposed with [39], which is solved using the CBS collocation technique. The algorithm proposed is new and has not yet been documented in the literature to date. The errors are also much smaller compared to many other existing methods. Also, it is a very efficient method in terms of computation, as it is not much complex.

The subsequent sections of the paper are structured as follows. In Section. 2, the CuTBS numerical method is derived and formulated. The analysis of stability and convergence for this scheme is detailed in Sections. 3 and 4

respectively. In the last section, the numerical results are discussed and compared with [39] and the final section presents the results and conclusions of the study.

2. Formulation of Numerical Method - The CuTBS

Divide the domain of time, $[0,T]$ into N equal intervals of length $k=\frac{T}{N}$, using points, $0=t_0<t_1<...<t_N=T$, where $t_n=nk,\ n=0,1,2,...,N$. Similarly, divide the domain of space, $[a,b]$ into M equal intervals of length, $h=\frac{b-a}{M}$ using the points, $a=x_0<x_1<x_2<...<x_M=b$, where $x_i=a+ih,\ i=0,1,2,...,M$. Let $u(x,t)$ represent the analytical solution and $U(x,t)$ denote the numerical solution of the specified differential equation. Then for the solution $u(x,t)$, the approximation, $U(x,t)$ in terms of CuTBS functions can be represented as

$$u(x,t)\approx U(x,t)=\sum_{i=-1}^{M+1}p_i(t)T_i(x)\ , \tag{2.1}$$

where $p_i(t)$ are unknown control points to be determined by the collocation method employing initial and end conditions and $T_i(x)$, the CuTBS functions are specified as [50]

$$T_i(x)=\frac{1}{K}\begin{cases} f^3(x_i),\ x\in[x_i,x_{i+1}]\ ,\\ f(x_i)\left(f(x_i)h(x_{i+2})+h(x_{i+3})f(x_{i+1})\right)+h(x_{i+4})f^2(x_{i+1}),\ x\in[x_{i+1},x_{i+2}]\ ,\\ h(x_{i+4})\left(f(x_{i+1})h(x_{i+3})+h(x_{i+4})f(x_{i+2})\right)+f(x_i)h^2(x_{i+3}),\ x\in[x_{i+2},x_{i+3}]\ ,\\ h^3(x_{i+4}),\ x\in[x_{i+3},x_{i+4}]. \end{cases} \tag{2.2}$$

where

$$\begin{aligned} f(x_i)&=\sin\frac{x-x_i}{2}\ ,\\ h(x_i)&=\sin\frac{x_i-x}{2}=-f(x_i)\ ,\\ K&=\sin\frac{h}{2}\sin h\sin\frac{3h}{2}. \end{aligned}$$

The support of the CuTBS function, $T_i(x)$ is presumed to be $[x_i,x_{i+4}]$. Each T_i is non-zero and piece-wise cubic throughout four successive sub-intervals and zero elsewhere. Thus, each subinterval $[x_i,x_{i+1}]$ comprises three segments of $T_i(x)$. T_i is also characterized by geometric properties such as partition of unity and C^2 continuity. Furthermore, T_{-1}, T_0, T_1, ..., T_{M+1} have been constructed to function as a basis for the space interval $[a,b]$. Using Equations (2.1) and (2.2), the values of $U(x,t)$ and its required derivatives at nodes are ascertained with the parameters, p_i as follows:

$$\begin{aligned} U_i&=r_1\left[p_{i-1}(t)+p_{i+1}(t)\right]+r_2p_i(t)\ ,\\ (U_x)_i&=r_3\left[p_{i+1}(t)-p_{i-1}(t)\right]\ ,\\ (U_{xx})_i&=r_4\left[p_{i-1}(t)+p_{i+1}(t)\right]+r_5p_i(t)\ , \end{aligned} \tag{2.3}$$

where

$$\begin{aligned} r_1&=\frac{1-\cos h}{2\sin h\sin\frac{3h}{2}}\ ,\\ r_2&=\frac{2}{3-4\sin^2\frac{h}{2}}\ ,\\ r_3&=\frac{3}{4\sin\frac{3h}{2}}\ ,\\ r_4&=\frac{3}{4}\left(\frac{3\sin^2\frac{h}{2}-2}{\sin^2\frac{h}{4}-\sin^2\frac{5h}{4}}\right)\ ,\\ r_5&=-\frac{3}{2(1+2\cos h)\tan^2\frac{h}{2}}. \end{aligned}$$

By substituting Equation (2.1) into Equation (1.1),

$$ {}^{CF}_{a}D^{\gamma}_{t}U(x,t) - U_{xx} = f(x,t). \tag{2.4}$$

The Caputo-Fabrizio fractional-time derivative in the above equation is discretized at $t = t_{n+1}$ as follows :

$$\begin{aligned} {}^{CF}_{a}D^{\gamma}_{t}U(x,t_{n+1}) &= \frac{1}{1-\gamma}\int_{0}^{t_{n+1}} U_{\nu}(x,\nu)e^{\frac{\gamma(t_{n+1}-\nu)}{\gamma-1}}d\nu \\ &= \frac{1}{1-\gamma}\sum_{j=0}^{n}\int_{t_j}^{t_{j+1}} U_{\nu}(x,\nu)e^{\frac{\gamma(t_{n+1}-\nu)}{\gamma-1}}d\nu. \end{aligned} \tag{2.5}$$

The above equation is revised using the forward difference formulation as follows:

$$\begin{aligned} {}^{CF}_{a}D^{\gamma}_{t}U(x,t_{n+1}) &= \frac{1}{k(1-\gamma)}\sum_{j=0}^{n}\left[U(x,t_{n-j+1}) - U(x,t_{n-j})\right]\int_{t_j}^{t_{j+1}} e^{\frac{\gamma(t_{n+1}-\nu)}{\gamma-1}}d\nu + E_k^{n+1} \\ &= \frac{1-e^{\frac{\gamma k}{\gamma-1}}}{\gamma k}\sum_{j=0}^{n}\left[U(x,t_{j+1}) - U(x,t_j)\right]e^{\frac{\gamma jk}{\gamma-1}} + E_k^{n+1}. \end{aligned}$$

As a consequence,

$$\begin{aligned} {}^{CF}_{a}D^{\gamma}_{t}U(x,t_{n+1}) &= \frac{\mu}{k\gamma}\sum_{j=0}^{n} l_j\left[U(x,t_{n-j+1}) - U(x,t_{n-j})\right] + E_k^{n+1} \\ &= \frac{\mu}{k\gamma}\sum_{j=0}^{n} l_{n-j}\left[U(x,t_{j+1}) - U(x,t_j)\right] + E_k^{n+1}. \end{aligned} \tag{2.6}$$

where

$$\begin{aligned} \mu &= 1 - e^{\frac{\gamma k}{\gamma-1}} \\ l_j &= e^{\frac{\gamma jk}{\gamma-1}}. \end{aligned}$$

Furthermore, the truncation error is specified by [45],

$$\begin{aligned} |E_k^{n+1}| &\leq \frac{\gamma k^2}{2(1-\gamma)^2} \\ &= ck^2. \end{aligned} \tag{2.7}$$

Here, c represents a constant. Based on the characteristics of the exponential function, it is evident that

- $l_j > 0,\ j = 0, 1, 2, ..., n,$
- $1 = l_0 > l_1 > l_2 > ... > l_j,\ l_j \to 0$ as $j \to \infty,$
- $\sum_{j=0}^{n}(l_j - l_{j+1}) + l_{n+1} = 1.$

The θ - weighted discretization for Equation (2.4) using Equation (2.6) is articulated as

$$\frac{\mu}{k\gamma}\sum_{j=0}^{n} l_{n-j}\left[U(x,t_{j+1}) - U(x,t_j)\right] - \theta U_{xx}(x,t^{n+1}) - (1-\theta)U_{xx}(x,t^n) = f(x,t^{n+1}). \tag{2.8}$$

Discrete the above equation implicitly, that is, for $\theta = 1$,

$$\begin{aligned}
&\frac{\mu}{k\gamma}\sum_{j=0}^{n}\left(l_{n-j}\left[U(x,t_{j+1})-U(x,t_j)\right]\right)-U_{xx}(x,t^{n+1})=f(x,t^{n+1})\\
&\implies \frac{\mu}{k\gamma}\sum_{j=0}^{n-1}\left[l_{n-j}(U_i^{j+1}-U_i^j)\right]+\frac{\mu}{k\gamma}(U_i^{n+1}-U_i^n)=(U_{xx})_i^{n+1}+f_i^{n+1}\\
&\implies U_i^{n+1}-\beta(U_{xx})_i^{n+1}=U_i^n-\sum_{j=0}^{n-1}\left[l_{n-j}(U_i^{j+1}-U_i^j)\right]+\beta f_i^{n+1}
\end{aligned} \tag{2.9}$$

where

$$\beta=\frac{k\gamma}{\mu}\,,$$
$$U_i^n=U(x_i,t_n)$$

and

$$f_i^{n+1}=f(x_i,t_{n+1}).$$

Substituting Equation (2.3) into the above equation, we get

$$\begin{aligned}
&r_1\left[p_{i-1}(t_{n+1})+p_{i+1}(t_{n+1})\right]+r_2p_i(t_{n+1})-\beta\left[r_4p_{i-1}(t_{n+1})+r_5p_i(t_{n+1})+r_4p_{i+1}(t_{n+1})\right]\\
&=r_1\left[p_{i-1}(t_n)+p_{i+1}(t_n)\right]+r_2p_i(t_n)-\sum_{j=0}^{n-1}\left[l_{n-j}\left(r_1p_{i-1}(t_{j+1})+r_2p_i(t_{j+1})+r_1p_{i+1}(t_{j+1})\right.\right.\\
&\left.\left.-r_1p_{i-1}(t_j)-r_2p_i(t_j)-r_1p_{i+1}(t_j)\right)\right]+\beta f_i^{n+1}\\
&\implies r_1\left(p_{i-1}^{n+1}+p_{i+1}^{n+1}\right)+r_2p_i^{n+1}-\beta\left(r_4p_{i-1}^{n+1}+r_5p_i^{n+1}+r_4p_{i+1}^{n+1}\right)=r_1\left(p_{i-1}^n+p_{i+1}^n\right)+r_2p_i^n\\
&-\sum_{j=0}^{n-1}\left[l_{n-j}\left(r_1p_{i-1}^{j+1}+r_2p_i^{j+1}+r_1p_{i+1}^{j+1}-r_1p_{i-1}^j-r_2p_i^j-r_1p_{i+1}^j\right)\right]+\beta f_i^{n+1}\\
&\implies (r_1-\beta r_4)\left(p_{i-1}^{n+1}+p_{i+1}^{n+1}\right)+(r_2-\beta r_5)p_i^{n+1}=r_1\left(p_{i-1}^n+p_{i+1}^n\right)+r_2p_i^n\\
&-\sum_{j=0}^{n-1}\left[l_{n-j}\left(r_1\left[p_{i-1}^{j+1}+p_{i+1}^{j+1}-p_{i-1}^j-p_{i+1}^j\right]+r_2\left[p_i^{j+1}-p_i^j\right]\right)\right]+\beta f_i^{n+1},\ i=0,1,2,...,M\,,
\end{aligned} \tag{2.10}$$

where

$$p_i^n=p_i(t_n).$$

This system comprises $(M+1)$ linear equations in $(M+3)$ unknowns. BCs (Equation (1.3)) are used to obtain the other two additional equations to obtain the unique solution.

$$\begin{aligned}
U\left(a,t_{n+1}\right)&=U\left(p_0,t_{n+1}\right)\\
&=r_1\left(p_{-1}^{n+1}+p_1^{n+1}\right)+r_2p_0^{n+1}\,,
\end{aligned}$$

$$\begin{aligned}
U\left(b,t_{n+1}\right)&=U\left(p_M,t_{n+1}\right)\\
&=r_1\left(p_{M-1}^{n+1}+p_{M+1}^{n+1}\right)+r_2p_M^{n+1}.
\end{aligned}$$

As a result, a matrix system of $(M+3)\times(M+3)$ dimension is achieved,

$$Ap^{n+1}=B\left(\sum_{j=1}^{n}\left(\left[l_{n-j}-l_{n-j+1}\right]p^j\right)+l_np^0\right)+F\,, \tag{2.11}$$

where

$$A = \begin{bmatrix} r_1 & r_2 & r_1 & ... & 0 & 0 \\ r_1 - \beta r_4 & r_2 - \beta r_5 & r_1 - \beta r_4 & ... & 0 & 0 \\ 0 & r_1 - \beta r_4 & r_2 - \beta r_5 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & r_2 - \beta r_5 & r_1 - \beta r_4 \\ 0 & 0 & 0 & ... & r_2 & r_1 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & ... & 0 & 0 \\ r_1 & r_2 & r_1 & ... & 0 & 0 \\ 0 & r_1 & r_2 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & r_2 & r_1 \\ 0 & 0 & 0 & ... & 0 & 0 \end{bmatrix},$$

$$p^n = \begin{bmatrix} p_{-1} \\ p_0 \\ p_1 \\ ... \\ p_M \\ p_{M+1} \end{bmatrix}^n \quad \text{and} \quad F = \begin{bmatrix} g_1 \\ \beta f_0 \\ \beta f_1 \\ ... \\ \beta f_M \\ g_2 \end{bmatrix}^{n+1}.$$

To solve the above system of equations, the initial vector,

$$p^0 = \begin{bmatrix} p_{-1} \\ p_0 \\ p_1 \\ ... \\ p_M \\ p_{M+1} \end{bmatrix}^0$$

is required and it is obtained by using the ICs as,

$$\begin{cases} (U_x)_0^0 = g'(x_0)\,, \\ U_i^0 = g(x_i),\ i = 0, 1, 2, ..., M\,, \\ (U_x)_M^0 = g'(x_M)\,. \end{cases} \tag{2.12}$$

The matrix representation of the equation system described above is as follows:

$$Cp^0 = D\,, \tag{2.13}$$

where

$$C = \begin{bmatrix} -r_3 & 0 & r_3 & ... & 0 & 0 \\ r_1 & r_2 & r_1 & ... & 0 & 0 \\ 0 & r_1 & r_2 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & r_2 & r_1 \\ 0 & 0 & 0 & ... & 0 & r_3 \end{bmatrix},$$

and

$$D = \begin{bmatrix} g'(x_0) \\ g(x_0) \\ g(x_1) \\ ... \\ g(x_M) \\ g'(x_M) \end{bmatrix}.$$

Equation (2.13) is solved using MATLAB's mldivide operator for p^0. All the numerical calculations are performed with the help of MATLAB.

3. Numerical Stability of Proposed Scheme

Stability refers to the property that any errors introduced during the computational process diminish or remain bounded as the procedure progresses [6]. Duhamel's principle is a mathematical tool that simplifies the study of inhomogeneous linear partial differential equations (PDEs) by relating their behavior to the corresponding homogeneous equations [42]. For stability analysis, this principle states that the stability properties of the inhomogeneous problem are the same as those of the homogeneous problem. Consequently, it is sufficient to analyze the stability of the homogeneous equation ($f = 0$) to establish the stability of the complete inhomogeneous problem. For the TFDE (Equation (1.1)), which is linear, the stability of the numerical scheme can be effectively analyzed using the Fourier method. The error at a grid point, denoted by ϵ_i^n, is defined as

$$\epsilon_i^n = u_i^n - U_i^n,\ i = 1, 2, 3, ..., M-1,\ n = 0, 1, 2, ..., N\ , \tag{3.1}$$

and can be represented as a vector:

$$\epsilon^n = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ ... \\ \epsilon_{M-1} \end{bmatrix}^n . \tag{3.2}$$

Using this definition, the error equations of Equation (2.10) is derived,

$$\begin{aligned} &(r_1 - \beta r_4)\left(\epsilon_{i-1}^{n+1} + \epsilon_{i+1}^{n+1}\right) + (r_2 - \beta r_5)\epsilon_i^{n+1} = r_1\left(\epsilon_{i-1}^n + \epsilon_{i+1}^n\right) + r_2\epsilon_i^n \\ &- \sum_{j=0}^{n-1}\left[l_{n-j}\left(r_1\left[\epsilon_{i-1}^{j+1} + \epsilon_{i+1}^{j+1} - \epsilon_{i-1}^j - \epsilon_{i+1}^j\right] + r_2\left[\epsilon_i^{j+1} - \epsilon_i^j\right]\right)\right]. \end{aligned} \tag{3.3}$$

Boundary and initial conditions then become,

$$\epsilon_0^n = g_1(t_n), \quad \epsilon_M^n = g_2(t_n), \quad n = 0, 1, 2, ..., N\ , \tag{3.4}$$

and

$$\epsilon_i^0 = g(x_i), \quad i = 1, 2, 3, ..., M. \tag{3.5}$$

Establish grid functions via the Fourier approach as outlined:

$$\epsilon^n = \begin{cases} \epsilon_i^n, & x_i - \frac{h}{2} < x \le x_i + \frac{h}{2},\ i = 1, 2, 3, ..., M-1, \\ 0, & a \le x \le a + \frac{h}{2} \quad \text{or} \quad b - \frac{h}{2} \le x \le b. \end{cases} \tag{3.6}$$

The Fourier series may then be used to express ϵ^n as

$$\epsilon^n(x) = \sum_{m=-\infty}^{\infty} \eta^n(m) e^{\frac{2\pi m x i}{b-a}}, \quad n = 1, 2, 3, ..., N\ , \tag{3.7}$$

where

$$\eta^n(m) = \frac{1}{b-a}\int_a^b \epsilon^n(x) e^{\frac{2\pi m x i}{a-b}}\, dx. \tag{3.8}$$

Computing the $||.||_2$ norm,

$$\begin{aligned} ||\epsilon^n||_2 &= \sqrt{\sum_{i=1}^{M-1} h|\epsilon_i^n|^2} \\ &= \sqrt{\int_a^{a+\frac{h}{2}} |\epsilon^n|^2 dx + \sum_{i=1}^{M-1}\int_{x_i-\frac{h}{2}}^{x_i+\frac{h}{2}} |\epsilon^n|^2 dx + \int_{b-\frac{h}{2}}^{b} |\epsilon^n|^2 dx} \\ &= \sqrt{\int_a^b |\epsilon^n|^2 dx}\ , \end{aligned}$$

and utilizing Parseval's identity in the context of the discrete Fourier transform [8],

$$\int_a^b |\epsilon^n|^2 dx = \sum_{m=-\infty}^{\infty} |\eta^n(m)|^2,$$

we obtain

$$||\epsilon^n||_2^2 = \sum_{m=-\infty}^{\infty} |\eta^n(m)|^2. \tag{3.9}$$

The solution is expressed through Fourier series analysis in the following manner.

$$\epsilon_i^n = \eta^n e^{\sigma i h I}, \tag{3.10}$$

where $\sigma \in [-\pi, \pi]$ and

$$I = \sqrt{-1}.$$

Using the expression above in Equation (3.3),

$$\begin{aligned} &(r_1 - \beta r_4)\left(\eta^{n+1} e^{\sigma(i-1)hI} + \eta^{n+1} e^{\sigma(i+1)hI}\right) + (r_2 - \beta r_5)\eta^{n+1} e^{\sigma i h I} = r_1\left(\eta^n e^{\sigma(i-1)hI} + \eta^n e^{\sigma(i+1)hI}\right) \\ &+ r_2 \eta^n e^{\sigma i h I} - \sum_{j=0}^{n-1}\Bigg[l_{n-j}\Bigg(r_1\left[\eta^{j+1} e^{\sigma(i-1)hI} + \eta^{j+1} e^{\sigma(i+1)hI} - \eta^j e^{\sigma(i-1)hI} - \eta^j e^{\sigma(i+1)hI}\right] \\ &+ r_2\left[\eta^{j+1} e^{\sigma i h I} - \eta^j e^{\sigma i h I}\right]\Bigg)\Bigg]. \end{aligned} \tag{3.11}$$

Dividing the above equation by $e^{\sigma i h I}$ and using the trigonometric identity,

$$e^{\sigma h I} + \frac{1}{e^{\sigma h I}} = 2\cos\sigma h,$$

and consolidating similar expressions, we obtain the following.

$$\left(1 - \frac{\beta\left(2r_4\cos\sigma h + r_5\right)}{2r_1\cos\sigma h + r_2}\right)\eta^{n+1} = \eta^n - \sum_{j=0}^{n-1} l_{n-j}\left(\eta^{j+1} - \eta^j\right). \tag{3.12}$$

With no loss of generality, let

$$\sigma = 0.$$

Hence, Equation (3.12) simplifies to

$$\left(1-\frac{\beta\left(2r_4+r_5\right)}{2r_1+r_2}\right)\eta^{n+1}=\eta^n-\sum_{j=0}^{n-1}l_{n-j}\left(\eta^{j+1}-\eta^j\right). \tag{3.13}$$

Subsequently,

$$\eta^{n+1}=\frac{\eta^n-\sum_{j=0}^{n-1}l_{n-j}\left(\eta^{j+1}-\eta^j\right)}{\delta}, \tag{3.14}$$

where

$$\delta=1-\frac{\beta\left(2r_4+r_5\right)}{2r_1+r_2}.$$

It is important to note that

$$\begin{aligned}\frac{\beta\left(2r_4+r_5\right)}{2r_1+r_2}&=-\frac{3\beta\tan^2\frac{h}{4}}{4}\\&\leq 0\,.\end{aligned}$$

Then

$$\delta\geq 1\,.$$

Proposition 3.1. *If η^k $(k=0,1,2,...,N)$ represents the solution of Equation (3.14), then the magnitude, $|\eta^k|$ is bounded by the initial value $|\eta^0|$, i.e.*

$$|\eta^k|\leq|\eta^0|\;\forall\;k$$

Proof We give a proof by induction on k. For, $k=0$, the Equation (3.14) gives, $\eta^1=\frac{\eta^0}{\delta}$. Then,

$$\begin{aligned}|\eta^1|&=\frac{|\eta^0|}{\delta}\\&\leq|\eta^0|\end{aligned}$$

because $\delta\geq 1$. Assume $|\eta^j|\leq|\eta^0|$ for $j=1,2,3,...,k$. Then

$$\begin{aligned}|\eta^{k+1}|&\leq\frac{|\eta^k|-\sum_{j=0}^{n-1}l_{n-j}\left(|\eta^{j+1}|-|\eta^j|\right)}{\delta}\\&\leq\frac{|\eta^0|}{\delta}\\&\leq|\eta^0|\end{aligned}$$

Theorem 3.2. *The scheme (2.10) is unconditionally stable.*

Proof. Using the relation (3.9) and the above proposition, we get

$$||\epsilon^n||_2\leq||\epsilon^0||_2\quad n=0,1,2,...,N\,.$$

The proposed method is hence unconditionally stable. □

4. Convergence Analysis of Proposed Scheme

In this section, convergence estimates are provided for the time-discretized problem (2.9). Similarly to the approach taken in the analysis of stability, the convergence study is conducted exclusively for the homogeneous problem.

Theorem 4.1. *Consider the exact solution* $\{u(x,t_n)\}_{n=0}^{N-1}$ *of Equation (5.1) and the time-discrete approximation* $\{U^n\}_{n=0}^{N-1}$ *corresponding to Equation (2.9). Define the error in the* $(n+1)$*-th step by*

$$e^{n+1} = u(x,t_{n+1}) - U^{n+1}.$$

Then, there exists a constant c *such that the following error bound holds:*

$$\|e^{n+1}\| \leq D + ck^2,$$

where k *is the discrete time increment.*

Proof. As in the previous analysis, the case where the source term $f = 0$ is considered. Note that the exact solution $u(x,t_n)$ satisfies the same semi-discrete scheme as given in Equation (2.9). Therefore, the exact solution at time t_{n+1} is expressed as:

$$u(x,t_{n+1}) = \sum_{j=1}^{n}(l_{n-j} - l_{n-j+1})u(x,t_j) + l_n u(x,t_0) + \beta u_{xx}(x,t_{n+1}). \tag{4.1}$$

Similarly, the numerical approximation at the same time level is given by:

$$U^{n+1} = \sum_{j=1}^{n}(l_{n-j} - l_{n-j+1})U^j + l_n U^0 + \beta U_{xx}^{n+1}.$$

Subtracting the above equation from Equation (4.1) gives the error equation:

$$\begin{aligned} e^{n+1} &= \sum_{j=1}^{n}(l_{n-j} - l_{n-j+1})e^j + l_n e^0 + \beta e_{xx}^{n+1} + E_k^{n+1} \\ &= \sum_{j=1}^{n}(l_{n-j} - l_{n-j+1})e^j + \beta e_{xx}^{n+1} + E_k^{n+1} \end{aligned} \tag{4.2}$$

where

$$e^n = u(x,t_n) - U^n$$

and

$$e^0 = 0.$$

Here, E_k^{n+1} denotes the truncation error in step $n+1$. Now take the inner product of both sides of Equation (4.2) with e_{n+1}. Using the identity,

$$\langle x, x\rangle = \|x\|^2 \geq 0,$$

we get:

$$\|e^{n+1}\|^2 = \sum_{j=1}^{n}(l_{n-j} - l_{n-j+1})\langle e^j, e^{n+1}\rangle + \beta\langle e_{xx}^{n+1}, e^{n+1}\rangle + \langle E_k^{n+1}, e^{n+1}\rangle. \tag{4.3}$$

By applying the identity,

$$\langle u_{xx}, u\rangle = -\langle u_x, u_x\rangle,$$

Equation (4.3) is simplified as:

$$\begin{aligned}\|e^{n+1}\|^2 &= \sum_{j=1}^{n}(l_{n-j}-l_{n-j+1})\langle e^j, e^{n+1}\rangle - \beta\langle e_x^n, e_x^n\rangle + \langle E_k^{n+1}, e^{n+1}\rangle \\ &= \sum_{j=1}^{n}(l_{n-j}-l_{n-j+1})\langle e^j, e^{n+1}\rangle - \beta\|e_x^n\|^2 + \langle E_k^{n+1}, e^{n+1}\rangle \\ &\leq \sum_{j=1}^{n}(l_{n-j}-l_{n-j+1})\langle e^j, e^{n+1}\rangle + \langle E_k^{n+1}, e^{n+1}\rangle\end{aligned}$$

Finally, applying the Cauchy–Schwarz inequality,

$$\langle x, y\rangle \leq \|x\|\|y\|,$$

we attain

$$\|e^{n+1}\|^2 \leq \sum_{j=1}^{n}(l_{n-j}-l_{n-j+1})\|e^j\|\|e^{n+1}\| + \|E_k^{n+1}\|\|e^{n+1}\|$$

Dividing by $\|e_{n+1}\|$, a bound for the error at time step, $n+1$ is obtained as :

$$\|e_{n+1}\| \leq \sum_{j=1}^{n}(l_{n-j}-l_{n-j+1})\|e^j\| + \|E_k^{n+1}\|.$$

Introducing

$$D_n = \max_{1\leq j\leq n}\|e_j\|,$$

we acquire,

$$\begin{aligned}\|e_{n+1}\| &\leq D_n\sum_{j=1}^{n} l_{n-j} - l_{n-j+1} + \|E_k^{n+1}\| \\ &= D_n(1-l_n) + \|E_k^{n+1}\|\end{aligned}$$

Using the relation $1-l_n < 1$ and Equation (2.7),

$$\begin{aligned}\|e_{n+1}\| &\leq D_n + \|E_k^{n+1}\| \\ &\leq D + ck^2,\end{aligned}$$

where

$$D = \max_{0\leq n\leq N} D_n.$$

□

5. Numerical Results and Discussions

The section reports the numerical results of the problems solved using the method presented in this paper. The computational accuracy of the method is evaluated using the error norms, L_2 and L_∞ as follows:

$$\begin{aligned}L_2 &= \|U^{exact} - U^N\|_2 \\ &\approx \sqrt{h\sum_{i=0}^{M}(U_i^{exact} - U_i^N)^2}\,,\end{aligned}$$

TABLE 1. Absolute errors when $M = 80,\ t = 1,\ k = 0.01$ and $\gamma = 0.5$ at several space grid points of Example 5.1.

x	Approx Sol	Exact Sol	Absolute Error
0.1	3.823449453735425	3.823452746534693	3.292799×10^{-6}
0.2	3.939678565149819	3.939684586619215	6.02147×10^{-6}
0.3	4.068132491048232	4.068140636035048	8.14499×10^{-6}
0.4	4.210096914703770	4.210106526100315	9.611397×10^{-6}
0.5	4.366992742517661	4.367003099159174	1.03566×10^{-5}
0.6	4.540390325586540	4.540400628849554	1.03032×10^{-5}
0.7	4.732025176970544	4.732034535929522	9.35896×10^{-6}
0.8	4.943815341967077	4.943822756951513	7.41498×10^{-6}
0.9	5.177880595237690	5.177884939615995	4.34438×10^{-6}

and

$$L_\infty = \|U^{exact} - U^N\|_\infty \approx \max_{0\le i\le M} |U_i^{exact} - U_i^N| .$$

Additionally, the Order of Convergence (OC) in both temporal and spatial directions is calculated as $\log_2 \left| \frac{L_\infty(h,k)}{L_\infty(h,\frac{k}{2})} \right|$ and $\log_2 \left| \frac{L_\infty(h,k)}{L_\infty(\frac{h}{2},k)} \right|$.

Example 5.1. Consider the TFDE,

$$ {}_a^{CF}D_t^\gamma u(x,t) - u_{xx} = f(x,t), \quad (x,t) \in \Omega, \quad 0 < \gamma < 1 , \tag{5.1}$$

where

$$\Omega = [0,1] \times [0,1] ,$$

with IC,

$$u(x,0) = 1 + e^x, \quad 0 \le x \le 1 ,$$

and BCs,

$$u(0,t) = 1 + e^{\frac{\gamma t}{1-\gamma}}, \quad u(1,t) = e + e^{\frac{\gamma t}{1-\gamma}}, \quad 0 \le t \le 1 ,$$

for

$$f(x,t) = \frac{\sinh \frac{\gamma t}{1-\gamma}}{1-\gamma} - e^x ,$$

Equation (5.1) has the exact solution,

$$u(x,t) = e^x + e^{\frac{\gamma t}{1-\gamma}} .$$

The absolute errors and numerical results of this example at several spatial grid points with $M = 80$ and 100, $\gamma = 0.5$ and 0.3, and $k = 0.01$ at $t = 1$ are presented in Tables. 1 and 2 respectively. Table 3 presents the error norms for $M = 64,\ k = 0.01,\ v \in [0,1]$ and $\gamma = 0.6$ across various time levels. Tables. 4 and 5 compare convergence orders with those documented in [39] for both spatial and temporal directions. Figure. 1 compares the exact solution with the numerical solution for Example 5.1 at various points in the temporal grid, with $\gamma = 0.3$ and 0.5 and $M = 32$. Figure. 2 presents a space-time plot comparing the numerical and exact solutions for $M = 90,\ k = 0.01,\ \gamma = 0.5$, and $t = 1$. At $t = 1$, Figure 3 illustrates the 2D and 3D error profiles and Figure 4 illustrates the exact and approximate solutions for $k = 0.01,\ M = 16$, and $t = 1$, which correspond to a variety of γ values.

TABLE 2. Absolute errors when $M = 100$, $t = 1$, $k = 0.01$ and $\gamma = 0.3$ at several space grid points of Example 5.1.

x	Approx Sol	Exact Sol	Absolute Error
0.1	2.640231608151269	2.640233927330858	2.31918×10^{-6}
0.2	2.756461544159319	2.756465767415380	4.22326×10^{-6}
0.3	2.884916125935474	2.884921816831213	5.690896×10^{-6}
0.4	3.026881014425153	3.026887706896480	6.69247×10^{-6}
0.5	3.183777090616226	3.183784279955338	7.18934×10^{-5}
0.6	3.357174676657497	3.357181809645719	7.13299×10^{-5}
0.7	3.548809252674672	3.548815716725687	6.46405×10^{-6}
0.8	3.760598826587128	3.760603937747678	5.11116×10^{-6}
0.9	3.994663130773039	3.994666120412160	2.98964×10^{-6}

TABLE 3. Error norms for Example 5.1: $M = 64$, $k = 0.01$ and $\gamma = 0.6$.

t	L_∞	L_2
0.2	1.48068×10^{-5}	1.08393×10^{-5}
0.4	1.43861×10^{-5}	1.05125×10^{-5}
0.6	1.370502×10^{-5}	9.99784×10^{-6}
0.8	1.26992×10^{-5}	9.24544×10^{-6}
1	1.12784×10^{-5}	8.18496×10^{-6}

TABLE 4. Order of Convergence in the spatial direction for Example 5.1: $k = 0.01$ and $\gamma = 0.5$.

	CBS [39]		Proposed Method			
h	L_∞	Order	L_∞	L_2	Order	CPU Time (s)
0.25	1.01779×10^{-3}	...	5.10824×10^{-3}	3.74626×10^{-3}	...	0.031
0.125	2.51478×10^{-4}	2.01694	1.25518×10^{-3}	9.22614×10^{-4}	2.02493	0.047
0.0625	6.16166×10^{-5}	2.02904	3.11801×10^{-4}	2.28591×10^{-4}	2.009197	0.063
0.03125	1.381×10^{-5}	2.15761	7.64588×10^{-5}	5.59104×10^{-5}	2.028	0.094

TABLE 5. Order of Convergence in the temporal direction for Example 5.1: $h = 0.01$ and $\gamma = 0.5$.

	CBS [39]		Proposed Method			
k	L_∞	Order	L_∞	L_2	Order	CPU Time (s)
0.1	2.13309×10^{-4}	...	2.06945×10^{-4}	1.51713×10^{-4}	...	0.016
0.05	5.21507×10^{-5}	2.03219	4.57799×10^{-5}	3.35534×10^{-5}	2.17646	0.031
0.025	1.18261×10^{-5}	2.14071	8.50394×10^{-6}	5.99638×10^{-6}	2.42851	0.063
0.0125	1.75263×10^{-6}	2.75438	2.71519×10^{-6}	2.42629×10^{-6}	1.64708	0.078

Example 5.2. Consider the TFDE,

$$ {}_a^{CF}D_t^\gamma u(x,t) - u_{xx} = f(x,t), \quad (x,t) \in \Omega, \quad 0 < \gamma < 1 \,, \tag{5.2} $$

where

$$ \Omega = [0,1] \times [0,1] \,, $$

with IC,

$$ u(x,0) = e^{\gamma x}, \quad 0 \le x \le 1, $$

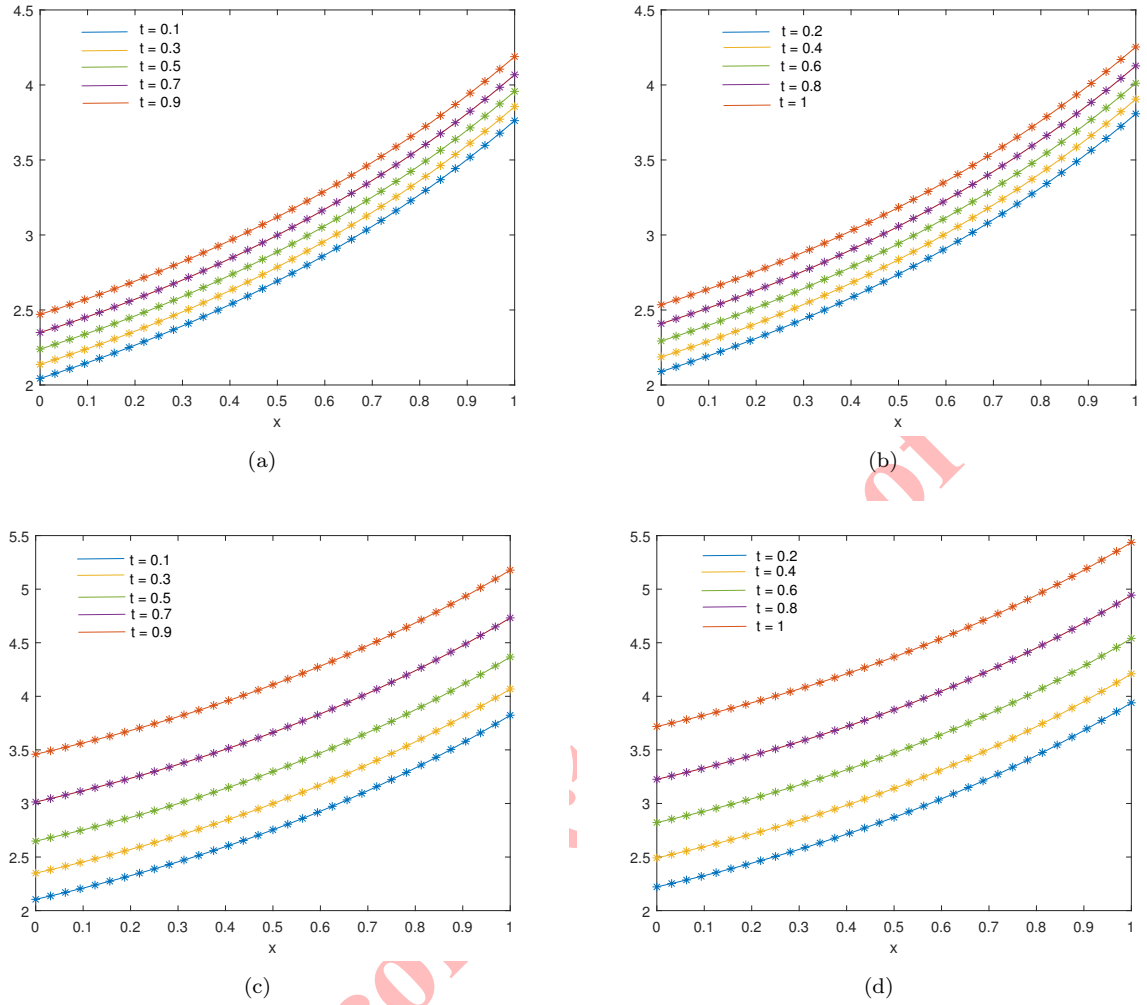


FIGURE 1. Approximate and exact solutions at different time points for Example 5.1 with $k = 0.01$ and $M = 32$ for (a) and (b) $\gamma = 0.3$ and (c) and (d) $\gamma = 0.5$.

and BCs,

$$u(0,t) = e^t, \quad u(1,t) = e^{\gamma+t}, \quad 0 \le t \le 1,$$

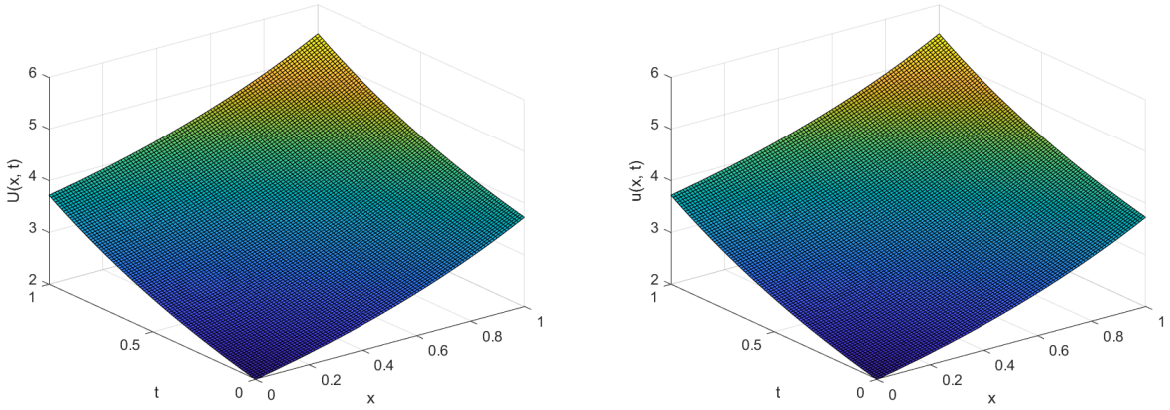
for

$$f(x,t) = e^{\gamma x+t}\left(1-\gamma^2-e^{\frac{t}{\gamma-1}}\right) .$$

The Equation (5.2) has exact solution,

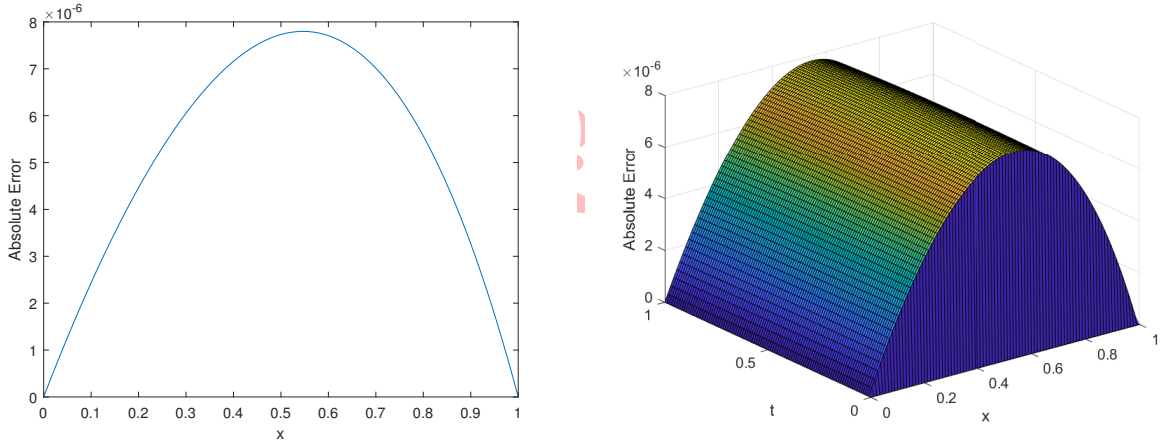
$$u(x,t) = e^{\gamma x+t} .$$

For Example 5.2, the absolute errors are reported in Tables 6 and 7 when $k = 0.01$, $t = 1$, $\gamma = 0.1$ band 0.3, and $M = 10$ and 20, respectively. Error norms are presented in Table 8 at various time intervals. In Table 9, the maximal errors of Example 5.2 are compared to the results in [39] at $t = 1$ for various γ choices. Table 10 juxtaposes the error and the order of convergence at $t = 1$ and $\gamma = 0.5$. The graphs of computational and exact solutions at different



(a) (b)

FIGURE 2. 3D approximate and exact solutions respectively for Example 5.1 with $k = 0.01$, $\gamma = 0.5$, $t = 1$, $x \in [0, 1]$ And $M = 90$.



(a) (b)

FIGURE 3. 2D and 3D error profiles respectively for Example 5.1 with $k = 0.01$, $\gamma = 0.5$, $t = 1$, $x \in [0, 1]$ and $M = 90$.

temporal phases for $\gamma = 0.5, M = 16$ and 100 and $k = 0.01$ are shown in Figure 5. 3D plots of numerical and exact solutions are illustrated in Figure 6 illustrating the accuracy of the proposed scheme. At the time $t = 1$, the error profiles in both 2D and 3D are presented in Figures 7 and 8 illustrates the exact and approximate solutions for $k = 0.01$, $M = 100$, and $t = 1$, which correspond to a variety of γ values.

Example 5.3. To further demonstrate the practicability of the CuTBS scheme on a problem without a known analytic solution, consider a one-dimensional sub-diffusive transport with a smoothly ramped, continuous source:

$$ {}^{CF}_{a}D_{t}^{0.5}u(x,t) - u_{xx} = \sin \pi x \left(1 - \frac{1}{e^{5t}}\right), \quad (x,t) \in \Omega, \tag{5.3}$$

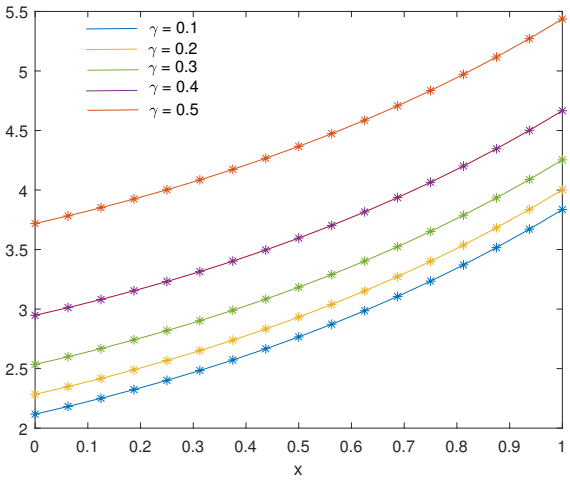


FIGURE 4. A comparison of the exact and approximate solutions at various γ values for Example 5.1, where $k = 0.01$, $t = 1$ and $M = 16$.

TABLE 6. Absolute errors when $M = 10$, $t = 1$, $k = 0.01$ and $\gamma = 0.1$ at several space grid points of Example 5.2.

x	Approx Sol	Exact Sol	Absolute Error
0.1	2.745544117810513	2.745601015016916	5.68972×10^{-5}
0.2	2.773093861516842	2.773194763964298	1.00903×10^{-4}
0.3	2.800933496246069	2.801065834699079	1.32339×10^{-4}
0.4	2.829065618645931	2.829217014351560	1.51396×10^{-4}
0.5	2.857492983658018	2.857651118063164	1.58134×10^{-4}
0.6	2.886218504189339	2.886370989267959	1.52485×10^{-4}
0.7	2.915245252135056	2.915379499976997	1.34248×10^{-4}
0.8	2.944576460760067	2.944679551065524	1.030903×10^{-4}
0.9	2.974215528461246	2.974274072563066	5.85441×10^{-5}

TABLE 7. Absolute errors when $M = 20$, $t = 1$, $k = 0.01$ and $\gamma = 0.3$ at several space grid points of Example 5.2.

x	Approx Sol	Exact Sol	Absolute Error
0.1	2.801045582428107	2.801065834699079	2.02523×10^{-5}
0.2	2.886334835886546	2.886370989267959	3.61534×10^{-5}
0.3	2.974226332598537	2.974274072563066	4.773997×10^{-5}
0.4	3.064799206604017	3.064854203293002	5.49967×10^{-5}
0.5	3.158135053759219	3.158192909689768	5.78559×10^{-5}
0.6	3.254318006015866	3.254374202889671	5.61969×10^{-5}
0.7	3.353434808515173	3.353484652549024	4.984403×10^{-5}
0.8	3.455574899580859	3.455613464762676	3.85652×10^{-5}
0.9	3.560830493703757	3.560852562355521	2.20687×10^{-5}

TABLE 8. Error norms for various γ choices with $k = 0.01$ and $M = 32$ for Example 5.2.

t	L_∞		L_2	
	$\gamma = 0.3$	$\gamma = 0.5$	$\gamma = 0.3$	$\gamma = 0.5$
0.2	9.97195×10^{-6}	1.66317×10^{-5}	7.31467×10^{-6}	1.219903×10^{-5}
0.4	1.21431×10^{-5}	2.03392×10^{-5}	8.90511×10^{-6}	1.49098×10^{-5}
0.6	1.47975×10^{-5}	2.48634×10^{-5}	1.08498×10^{-5}	1.82192×10^{-5}
0.8	1.80421×10^{-5}	3.03859×10^{-5}	1.32269×10^{-5}	2.22601×10^{-5}
1.0	3.71281×10^{-5}	7.86892×10^{-5}	1.61322×10^{-5}	2.71945×10^{-5}

TABLE 9. The comparison of the maximum error when $t = 1$ for Example 5.2.

h	k	CBS [39]		Proposed Method	
		$\gamma = 0.75$	$\gamma = 0.95$	$\gamma = 0.75$	$\gamma = 0.95$
0.05	0.05	2.41731×10^{-4}	1.88707×10^{-3}	5.72081×10^{-5}	1.58968×10^{-3}
	0.02	1.3786×10^{-5}	2.34313×10^{-4}	1.70919×10^{-6}	6.377628×10^{-5}
	0.01	1.88081×10^{-5}	5.83444×10^{-6}	1.03555×10^{-6}	3.04028×10^{-5}
0.02	0.05	2.66772×10^{-4}	1.9591×10^{-3}	2.37247×10^{-4}	1.91155×10^{-3}
	0.02	3.8726×10^{-5}	3.066×10^{-4}	9.17163×10^{-6}	2.58939×10^{-4}
	0.01	6.1196×10^{-6}	6.64896×10^{-5}	2.34416×10^{-6}	1.88124×10^{-5}
0.01	0.05	2.70322×10^{-4}	1.96935×10^{-3}	2.62942×10^{-4}	1.95746×10^{-3}
	0.02	4.22867×10^{-5}	3.16918×10^{-4}	3.48988×10^{-4}	3.05004×10^{-4}
	0.01	9.68207×10^{-6}	7.68187×10^{-5}	2.29261×10^{-6}	6.49009×10^{-5}

TABLE 10. Temporal order of convergence's comparison when $\gamma = 0.5$ and $t = 1$ for Example 5.2.

h	k	CBS [39]		Proposed Method			
		L_∞	Order	L_∞	L_2	Order	CPU Time (s)
0.05	0.02	6.01029×10^{-6}	...	9.09089×10^{-5}	6.67057×10^{-5}	...	0.063
	0.01	2.32171×10^{-6}	1.37225	2.925703×10^{-5}	2.28346×10^{-5}	1.63564	0.078
0.02	0.02	1.03118×10^{-5}	...	5.2161004×10^{-6}	3.81597×10^{-6}	...	0.086
	0.01	1.96501×10^{-6}	2.39169	1.35655×10^{-6}	9.93318×10^{-6}	1.94303	0.094
0.01	0.02	1.09244×10^{-5}	...	7.04276×10^{-6}	5.16136×10^{-6}	...	0.125
	0.01	2.57792×10^{-6}	2.08327	1.30432×10^{-6}	9.54199×10^{-7}	2.43284	0.141

where

$$\Omega = [0, 1] \times [0, 1] ,$$

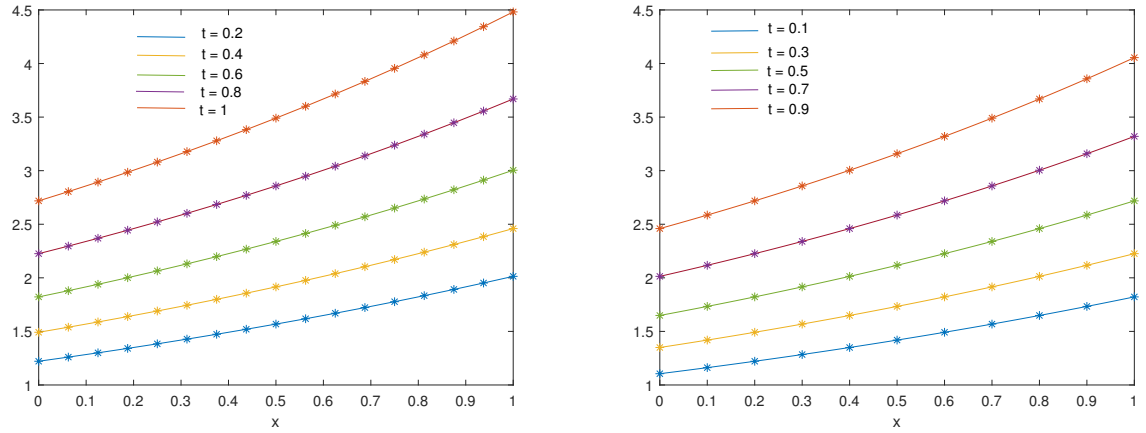
with IC,

$$u(x, 0) = 0, \quad 0 \le x \le 1,$$

and BCs,

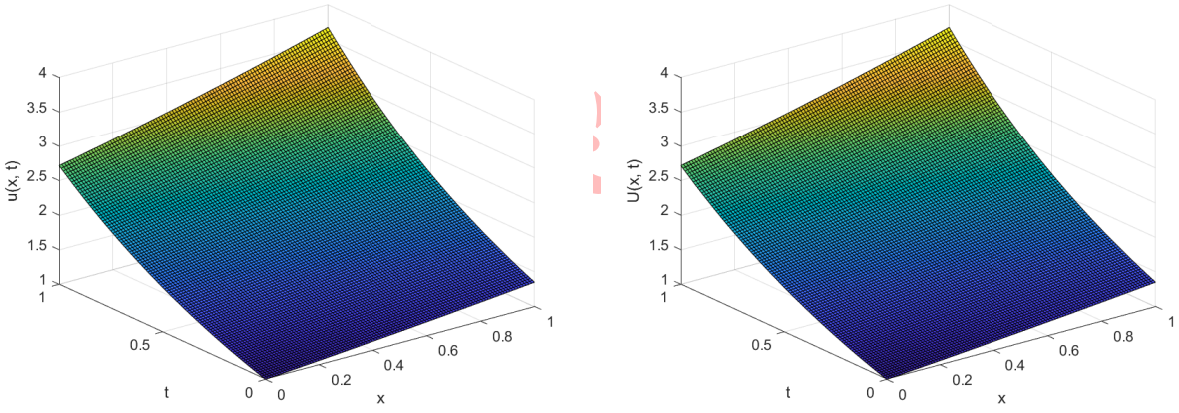
$$u(0, t) = u(1, t) = 0, \quad 0 \le t \le 1.$$

In this example, in order to verify the effectiveness of the method, assume that the exact solution of the equation is unknown and take the solution on the finer grid, i.e. M = N = 2000 as the corresponding exact solution. First, the proposed scheme will be used to test the accuracy in the direction of time. In this case, take M = 200. The errors, temporal OC, and CPU times are shown in Table 11. The data in Table 11 show that the temporal OC is about 2. Furthermore, the accuracy of the scheme for space is tested. Choose N = 1000 for different M (M = 10, 20, 40 and



(a) (b)

FIGURE 5. Approximate and exact solutions of Example 5.2 at different times when $k = 0.01$ And $\gamma = 0.5$ for (a) $M = 16$ And (b) $M = 100$.

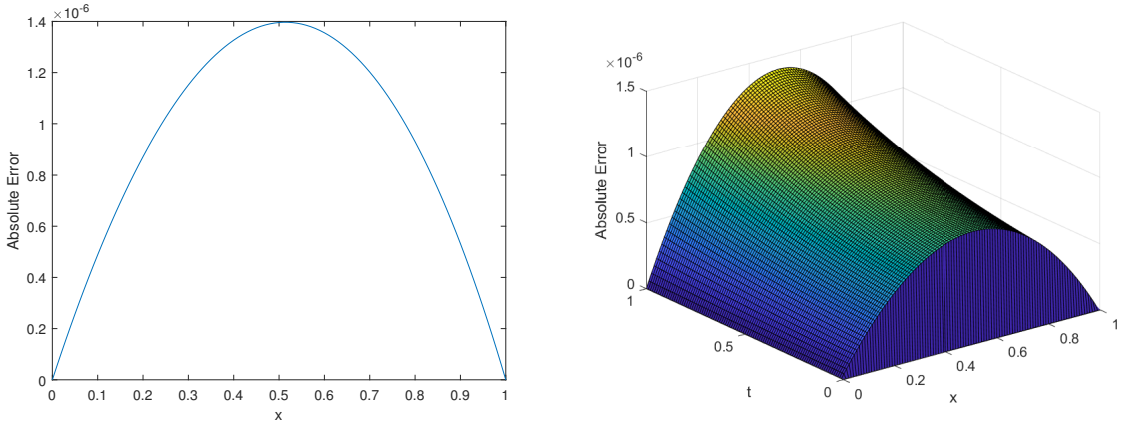


(a) (b)

FIGURE 6. 3D exact and approximate solutions respectively for Example 5.2 with $k = 0.01$, $\gamma = 0.3$, $t = 1$, $x \in [0, 1]$ And $M = 100$.

80). The errors, spatial OC, and CPU times are shown in Table 12. The solution on a finer grid, that is, M = N = 2000 and numerical solution (M = N = 200) is presented in Figure 9.

The CuTBS approach to TFDE with the Caputo-Fabrizio fractional derivative is new to the literature. Employing a finite difference method for the fractional derivative made the whole process simpler and faster. Across all benchmark tests, the CuTBS collocation scheme consistently delivers second-order convergence in time while exhibiting a markedly lower temporal error and reduced computational cost compared to established Caputo–Fabrizio solvers. By retaining the non-singular exponential kernel in its discrete convolution and employing a fully implicit $\theta = 1$ time discretization, CuTBS eliminates mixed-time truncation errors and achieves a sharper prefactor in the $O(k^2)$ error bound. Moreover, the C^2 spatial continuity of the trigonometric B-splines mitigates high-frequency error amplification through the



(a) (b)

FIGURE 7. 2D and 3D error profiles respectively for Example 5.2 with $k = 0.01$, $\gamma = 0.3$, $t = 1$, $x \in [0, 1]$ and $M = 100$.

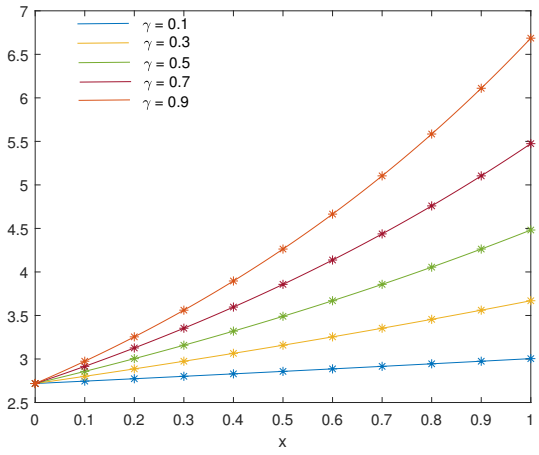


FIGURE 8. A comparison of the exact and approximate solutions at various γ values for Example 5.2, where $k = 0.01$, $t = 1$ and $M = 100$.

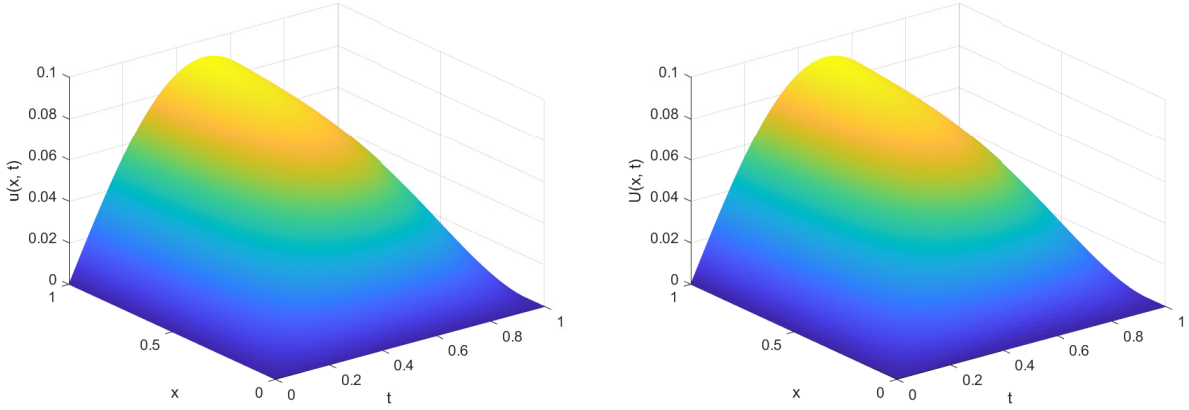
TABLE 11. Errors, Temporal OC and CPU times for Example 5.3 where M = 200

N	L_∞	Order	CPU Time (s)
25	5.241×10^{-6}	...	0.359
50	1.636×10^{-6}	1.68	0.688
100	4.094×10^{-7}	1.999	1.031
200	1.024×10^{-7}	1.999	1.813

convolution memory, strengthening temporal stability across a broad range of fractional orders. These combined features enable CuTBS to achieve the target accuracy with coarser time grids or significantly shorter run times than other methods, confirming its superior accuracy-cost balance in time.

TABLE 12. Errors, Spatial OC and CPU times for Example 5.3 where N = 1000

M	L_∞	Order	CPU Time (s)
10	0.001	...	0.875
20	2.452×10^{-4}	2.03	0.907
40	6.072×10^{-5}	2.01	0.938
80	1.512×10^{-5}	2.01	1.719



(a) (b)

FIGURE 9. Solution on finer grid and Numerical solution with t, $x \in [0, 1]$ and $M = N = 200$ respectively for Example 5.3.

6. CONCLUSIONS

This study introduced a numerical approach that uses CuTBS functions to solve TFDEs that involve the Caputo-Fabrizio fractional-time derivative. The CuTBS-based scheme effectively combines accurate spatial interpolation with a robust finite-difference discretization for the fractional derivative, ensuring high precision in both temporal and spatial domains. The stability and convergence of the scheme were analyzed to confirm that the method is unconditionally stable, the numerical error remains bounded, and it is validated with the numerical order of convergence. Numerical experiments confirm its reliability and efficiency in handling TFDE, demonstrating its potential as a powerful tool to solve mathematical models with fractional derivatives of Caputo-Fabrizio in various scientific and engineering applications.

REFERENCES

[1] A. Alshabanat, M. Jleli, S. Kumar, and B. Samet, *Generalization of Caputo-Fabrizio fractional derivative and applications to electrical circuits*, Frontiers in Physics, *8* (2020), 64.

[2] M. Amin, M. Abbas, M. K. Iqbal, and D. Baleanu, *Non-polynomial quintic spline for numerical solution of fourth-order time fractional partial differential equations*, Advances in Difference Equations, *2019* (2019), 1–22.

[3] A. Atangana and R. T. Alqahtani, *Numerical approximation of the space-time Caputo-Fabrizio fractional derivative and application to groundwater pollution equation*, Advances in Difference Equations, *2016* (2016), 1–13.

[4] B. Baeumer, S. Kurita, and M. M. Meerschaert, *Inhomogeneous fractional diffusion equations*, Fractional Calculus and Applied Analysis, *8(4)* (2005), 371–386.

[5] A. Benzahi, N. Abada, N. Arar, S. A. Idris, M. S. Abdo, and W. Shatanawi, *Caputo-Fabrizio type fractional differential equations with non-instantaneous impulses: Existence and stability results*, Alexandria Engineering Journal, *87* (2024), 186–200.

[6] W. E. Boyce, R. C. DiPrima, and D. B. Meade, *Elementary Differential Equations*, John Wiley & Sons, USA, 2017.

[7] M. Caputo and M. Fabrizio, *A new definition of fractional derivative without singular kernel*, Progress in Fractional Differentiation & Applications, *1*(2) (2015), 73–85.

[8] M. Cui, *A high-order compact exponential scheme for the fractional convection–diffusion equation*, Journal of Computational and Applied Mathematics, *255* (2014), 404–416.

[9] M. Du, Z. Wang, and H. Hu, *Measuring memory with the order of fractional derivative*, Scientific reports, *3* (2013), 3431.

[10] S. Esmaeili and R. Garrappa, *A pseudo-spectral scheme for the approximate solution of a time-fractional diffusion equation*, International Journal of Computer Mathematics, *92*(5) (2015), 980–994.

[11] R. Gorenflo, F. Mainardi, D. Moretti, and P. Paradisi, *Time fractional diffusion: a discrete random walk approach*, Nonlinear Dynamics, *29* (2002), 129–143.

[12] L. He, Y. Zhu, and Z. Lu, *A cubic spline interpolation based numerical method for fractional differential equations*, Journal of Industrial and Management Optimization, *20*(8) (2024), 2770–2794.

[13] J. Hristov, *Derivatives with non-singular kernels from the Caputo-Fabrizio definition and beyond: Appraising analysis with emphasis on diffusion models*, Frontiers in Fractional Calculus, Bentham Science Publishers, (2018), 269–341.

[14] N. Khalid, M. Abbas, and M. K. Iqbal, *Non-polynomial quintic spline for solving fourth-order fractional boundary value problems involving product terms*, Applied Mathematics and Computation, *349* (2019), 393–407.

[15] C. Lakshmi and A. Awasthi, *Numerical simulation of Burgers' equation using cubic B-splines*, Nonlinear Engineering, *6*(1) (2017), 61–77.

[16] C. Lakshmi and A. Awasthi, *Robust numerical scheme for nonlinear modified Burgers equation*, International Journal of Computer Mathematics, *95*(9) (2018), 1910–1926.

[17] H. Liu, A. Cheng, H. Yan, Z. Liu, and H. Wang, *A fast compact finite difference method for quasilinear time fractional parabolic equation without singular kernel*, International Journal of Computer Mathematics, *96*(7) (2019), 1444–1460.

[18] H. Liu, X. C. Zheng, and H. F. Fu, *Analysis of a multi-term variable-order time-fractional diffusion equation and its Galerkin finite element approximation*, Journal of Computational Mathematics, *40* (2022), 814–834.

[19] F. Mainardi, P. Paradisi, and R. Gorenflo, *Probability distributions generated by fractional diffusion equations*, arXiv preprint arXiv:0704.0320, (2007).

[20] M. M. Meerschaert, *Fractional calculus, anomalous diffusion, and probability*, Fractional Dynamics: Recent Advances, World Scientific, (2012), 265–284.

[21] R. Metzler, W. G. Glöckle, and T. F. Nonnenmacher, *Fractional model equation for anomalous diffusion*, Physica A: Statistical Mechanics and its Applications, *211*(1) (1994), 13–24.

[22] I. A. Mirza and D. Vieru, *Fundamental solutions to advection–diffusion equation with time-fractional Caputo–Fabrizio derivative*, Computers & Mathematics with Applications, *73*(1) (2017), 1–10.

[23] S. T. Mohyud-Din, T. Akram, M. Abbas, A. I. Ismail, and N. H. M. Ali, *A fully implicit finite difference scheme based on extended cubic B-splines for time fractional advection–diffusion equation*, Advances in Difference Equations, *2018* (2018), 1–17.

[24] K. Mustapha, B. Abdallah, and K. M. Furati, *A discontinuous Petrov–Galerkin method for time-fractional diffusion equations*, SIAM Journal on Numerical Analysis, *52*(5) (2014), 2512–2529.

[25] K. Oldham and J. Spanier, *Fractional Calculus*, Academic Press, London/New York, 1973.

[26] M. Onitsuka and I. I. El-Fassi, *Generalized Caputo–Fabrizio fractional differential equation*, Journal of Applied Analysis and Computation, *14*(2) (2024), 964–975.

[27] R. Pakhira, U. Ghosh, and S. Sarkar, *Study of memory effects in an inventory model using fractional calculus*, Applied Mathematical Sciences, *12*(17) (2018), 797–824.

[28] P. Pandey, J. F. Gómez-Aguilar, M. K. A. Kaabar, Z. Siri, and A. M. Abd Allah, *Mathematical modeling of COVID-19 pandemic in India using Caputo–Fabrizio fractional derivative*, Computers in Biology and Medicine, *145* (2022), 105518.
[29] S. Poojitha and A. Awasthi, *Eloquent numerical approach for solving generalized time fractional convection-diffusion-reaction problems*, Physica Scripta, *99*(12) (2024), 125277.
[30] S. Poojitha and A. Awasthi, *Operational matrix based numerical scheme for the solution of time fractional diffusion equations*, Fractional Calculus and Applied Analysis, *27*(2) (2024), 877–895.
[31] I. Podlubny, *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*, Elsevier, Netherlands, 1998.
[32] P. Roul, *A fourth order numerical method based on B-spline functions for pricing Asian options*, Computers & Mathematics with Applications, *80*(3) (2020), 504–521.
[33] P. Roul and V. P. Goura, *A high order numerical scheme for solving a class of non-homogeneous time-fractional reaction diffusion equation*, Numerical Methods for Partial Differential Equations, *37*(2) (2021), 1506–1534.
[34] P. Roul and T. Kumari, *A quartic trigonometric B-spline collocation method for a general class of nonlinear singular boundary value problems*, Journal of Mathematical Chemistry, *0* (2022), 1–17.
[35] P. Roul and V. P. Goura, *A sixth order optimal B-spline collocation method for solving Bratu's problem*, Journal of Mathematical Chemistry, *58* (2020), 967–988.
[36] P. Roul, V. M. K. P. Goura, and R. Agarwal, *A high-order compact finite difference scheme and its analysis for the time-fractional diffusion equation*, Journal of Mathematical Chemistry, *61*(10) (2023), 2146–2175.
[37] P. Roul, V. M. K. P. Goura, and R. Cavoretto, *A numerical technique based on B-spline for a class of time-fractional diffusion equation*, Numerical Methods for Partial Differential Equations, *39*(1) (2023), 45–64.
[38] K. M. Saad, *New fractional derivative with non-singular kernel for deriving Legendre spectral collocation method*, Alexandria Engineering Journal, *59*(4) (2020), 1909–1917.
[39] M. Shafiq, F. A. Abdullah, M. Abbas, A. S. Alzaidi, and M. B. Riaz, *Memory effect analysis using piecewise cubic B-spline of time fractional diffusion equation*, Fractals, *30*(8) (2022), 2240270.
[40] A. Shaikh, A. Tassaddiq, K. S. Nisar, and D. Baleanu, *Analysis of differential equations involving Caputo–Fabrizio fractional operator and its applications to reaction–diffusion equations*, Advances in Difference Equations, *2019*(1) (2019), 1–14.
[41] R. Singh, J. Mishra, and V. K. Gupta, *The dynamical analysis of a tumor growth model under the effect of fractal fractional Caputo–Fabrizio derivative*, International Journal of Mathematics and Computer in Engineering, 2023.
[42] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, SIAM, USA, 2004.
[43] N. H. Sweilam, M. M. Khader, and A. M. S. Mahdy, *Crank-Nicolson finite difference method for solving time-fractional diffusion equation*, Journal of Fractional Calculus and Applications, *2*(2) (2012), 1–9.
[44] H. Sun, W. Chen, and K. Sze, *A semi-discrete finite element method for a class of time-fractional diffusion equations*, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, *371*(1990) (2013), 20120268.
[45] A. Taghavi, A. Babaei, and A. Mohammadpour, *On the stable implicit finite differences approximation of diffusion equation with the time fractional derivative without singular kernel*, Asian-European Journal of Mathematics, *13*(6) (2020), 2050111.
[46] N. H. Tuan, L. V. C. Hoan, and S. Tatar, *An inverse problem for an inhomogeneous time-fractional diffusion equation: a regularization method and error estimate*, Computational and Applied Mathematics, *38* (2019), 1–22.
[47] M. Yaseen, M. Abbas, A. I. Ismail, and T. Nazir, *A cubic trigonometric B-spline collocation approach for the fractional sub-diffusion equations*, Applied Mathematics and Computation, *293* (2017), 311–319.
[48] M. Yaseen, M. Abbas, T. Nazir, and D. Baleanu, *A finite difference scheme based on cubic trigonometric B-splines for a time-fractional diffusion-wave equation*, Advances in Difference Equations, *2017* (2017), 1–18.
[49] M. Yaseen and M. Abbas, *An efficient computational technique based on cubic trigonometric B-splines for time fractional Burgers' equation*, International Journal of Computer Mathematics, *97*(3) (2020), 725–738.

[50] S. M. Zin, A. A. Majid, A. I. M. Ismail, and M. Abbas, *Cubic trigonometric B-spline approach to numerical solution of wave equation*, International Journal of Mathematical, Computational, Physical and Quantum Engineering, *8* (2014), 1212–1216.

[51] P. Zhuang and F. Liu, *Implicit difference approximation for the time fractional diffusion equation*, Journal of Applied Mathematics and Computing, *22* (2006), 87–99.