AIMS Mathematics, 10(7): 17049–17081.
DOI: 10.3934/math.2025765
Received: 23 May 2025
Revised: 19 July 2025
Accepted: 21 July 2025
Published: 30 July 2025

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

Research article

A uniform hyperbolic polynomial B-spline approach for solving the fractional diffusion-wave equations in the Caputo-Fabrizio sense

Muhammad Umar Manzoor[1], Muhammad Yaseen[1,*], Muath Awadalla[2,*] and Hajer Zaway[2]

[1] Department of Mathematics, University of Sargodha, Sargodha, 40100, Pakistan

[2] Department of Mathematics and Statistics, College of Science, King Faisal University, Al Ahsa 31982, Saudi Arabia

* **Correspondence:** Email: yaseen.yaqoob@uos.edu.pk, mawadalla@kfu.edu.sa.

Abstract: Piecewise polynomial functions serve as powerful tools for function approximation and the numerical solution of differential equations. In this study, we presented a robust numerical method for solving the time-fractional diffusion-wave equation involving the Caputo-Fabrizio fractional derivative. The proposed scheme combines the uniform hyperbolic polynomial B-spline basis for spatial discretization with a θ-weighted finite difference approach for temporal integration. The uniform hyperbolic polynomial B-spline, an advanced generalization of B-splines, integrates hyperbolic functions to enhance smoothness and flexibility, making it especially well-suited for problems exhibiting hyperbolic behavior. Rigorous stability and convergence analyses were carried out to ensure the reliability of the method. To demonstrate its effectiveness, the scheme was applied to several benchmark problems. Numerical results reveal that the proposed approach is highly accurate and computationally efficient.

Keywords: time-fractional diffusion wave equation; Caputo-Fabrizio fractional derivative; uniform hyperbolic polynomial B-spline
Mathematics Subject Classification: 26A33, 35R11, 65D07, 65M70

1. Introduction

Fractional differential equations have gained significant attention across scientific and engineering disciplines due to their ability to model complex phenomena with greater accuracy than classical differential equations. In particular, the time-fractional diffusion-wave equation plays a crucial role in describing dynamic processes such as wave propagation in heterogeneous media.

Consider the time-fractional diffusion-wave equation (TFDWE) with damping and reaction terms:

$$\frac{\partial^\lambda u(x,t)}{\partial t^\lambda} + \mu_1 \frac{\partial u(x,t)}{\partial t} + \mu_2 u(x,t) - \frac{\partial^2 u(x,t)}{\partial x^2} = f(x,t), \quad 1 < \lambda < 2, \quad a \le x \le b, \quad t \ge 0, \tag{1.1}$$

subject to the initial conditions (ICs):

$$u(x,0) = \phi_0(x), \quad \frac{\partial u(x,0)}{\partial t} = \phi_1(x), \quad a \le x \le b, \tag{1.2}$$

and the boundary conditions (BCs):

$$u(a,t) = \psi_0(t), \quad u(b,t) = \psi_1(t), \quad t \ge 0. \tag{1.3}$$

Here μ_1 and μ_2 are the damping and reaction coefficients, respectively, and $f(x,t)$ is a source term in Eq (1.1). The Caputo-Fabrizio fractional derivative (CFFD) of order λ ($1 < \lambda < 2$) for $u(x,t)$ with respect to t is defined as [20]:

$$\frac{\partial^\lambda u(x,t)}{\partial t^\lambda} = \frac{M(\lambda)}{2-\lambda} \int_0^t \frac{\partial^2 u(x,s)}{\partial s^2} \exp\Big[-\frac{\lambda(t-s)}{2-\lambda}\Big] ds, \tag{1.4}$$

where $M(\lambda)$ is a normalization function satisfying $M(0) = M(1) = 1$.

Unlike classical fractional derivatives, the CFFD incorporates an exponential kernel $\exp[-\frac{\lambda(t-s)}{2-\lambda}]$ which avoids singularities and provides more realistic modeling of memory effects. Various analytical and numerical methods have been proposed in the literature to solve the TFDWE. However, challenges remain in achieving high accuracy and computational efficiency, particularly for problems involving strong damping or reaction effects. A detailed analysis of fractional calculus is presented in [1]. Podlubny [2] resolved a long-standing problem by providing geometric and physical interpretation of fractional integration and differentiation. Numerical methods for solving non-linear fractional PDEs have been explored in [3]. Sierociuk et al. [4] presented a compelling application of fractional calculus to model heat transfer in heterogeneous media. The time-fractional (TF) telegraph equation [5] describes current and voltage in transmission lines, while the non-linear TF Schrödinger wave equation [6] models wave behavior in dispersive media. The non-linear generalized TF Klein-Gordon equation [7] arises in relativistic quantum field theory.

Shafiq et al. [8] presented a numerical study of the TFDWE incorporating the Caputo-Fabrizio fractional derivative with an exponential kernel using a cubic B-spline collocation method. Yaseen et al. [9] used a finite difference scheme together with trigonometric cubic B-splines to solve the TFDWE. Dehghan et al. [10] presented a meshless numerical method for solving the TFDWE. Solomon et al. [11] investigated anomalous diffusion and Levy flights in rotating flows. Avazzadeh et al. [12] applied radial basis functions with a finite difference scheme to solve the TFDWE. The anomalous diffusion equation was studied in [13] using a fractional dynamics approach, while fractional diffusion in inhomogeneous media was examined in [14]. Shafiq et al. [15] developed a numerical method to solve the time-fractional diffusion equation using B-splines. Wang and Tang [16] analyzed stochastic reaction-diffusion equations on an unbounded domain.

Dwivedi et al. [17] found numerical solutions of fractional order reaction advection diffusion equation using the cubic B-spline collocation method. Hayat et al. [18] explored a time-fractional

Schrödinger equation involving the Caputo fractional derivative. Ali et al. [19] developed a numerical approach for solving the two-dimensional time-fractional diffusion damped model. Caputo and Fabrizio [20] introduced a fractional derivative without a singular kernel. Cruz et al. [21] applied the Gaussian-based Caputo-Fabrizio fractional derivative for signal processing applications. Dokuyucn et al. [22] developed a cancer treatment model with the Caputo-Fabrizio fractional derivative. Mortezaee et al. [23] introduced a fuzzy hyperbolic model for fractional optimal control problems based on the Caputo-Fabrizio fractional derivative. Gahala et al. [24] proposed two linearized finite difference schemes for solving time-fractional non-linear diffusion-wave equations involving a fourth-order spatial derivative.

The uniform hyperbolic polynomial (UHP) B-spline is a piecewise-defined function that retains the fundamental properties of classical B-splines. It is employed for obtaining numerical solutions of partial differential equations and is widely used in computer-aided geometric designs for smooth curve and surface modeling. Lu et al. [25] introduced UHP B-splines as an extension of the classical B-spline. Palav and Pradhan utilized the redefine fourth-order UHP B-spline to solve the advection-diffusion equation [26] and Burger's equation [27]. Shah et al. [28] introduced a hybrid collocation method for approximating solutions to two-dimensional time-fractional diffusion-wave equations. Bhrawy et al. [29] significantly advanced the numerical solution of the fractional diffusion-wave equation by introducing the highly accurate and flexible spectral tau method. Wei and Yang [30] provided a robust and accurate framework for solving the variable order time-fractional diffusion-wave equation using the finite difference scheme for temporal discretization and a local discontinuous Galerkin method in spatial approximation. Zhang et al. [31] developed an innovative local discontinuous Galerkin method with generalized alternating numerical fluxes for solving fourth-order time-fractional sub-diffusion equations. Zhang et al. [32] proposed a high-order local discontinuous Galerkin method to solve the time-fractional mobile/immobile transport equation involving the Caputo-Fabrizio fractional derivative.

Classical schemes for time-fractional diffusion-wave equations lose accuracy and stability because the Caputo kernel is singular and ordinary spline bases are not optimized for wave behavior. By pairing the non-singular Caputo-Fabrizio derivative with the smooth UHP B-spline basis in a θ-weighted finite difference framework, we present a method that remains accurate and unconditionally stable across a broad range of fractional diffusion-wave problems. Stability and convergence demonstrate the robustness and accuracy of our proposed scheme. We conduct numerical experiments and compare our results with those reported in previous studies. Additionally, graphical representations of the results are provided to visually illustrate the accuracy and behavior of the proposed method. These graphs highlight key trends, confirm the stability of the proposed method, and demonstrate its agreement with theoretical predictions.

The key contibutions and strengths of the present study are summarized as follows:

- A novel numerical scheme is proposed by combing the UHP B-spline basis with a θ-weighted finite difference method for solving time-fractional diffusion-wave equations involving the Caputo-Fabrizio derivative.
- The use of a UHP B-spline basis enhances the smoothness, flexibility, and local support properties of the spatial approximation making it suitable for diffusion-dominated problems.
- The Caputo-Fabrizio fractional derivative with an exponential kernel ensures non-singular behavior and realistic memory effects contributing to better numerical stability.

- The proposed scheme is supported by rigorous stability and convergence analyses.
- Extensive numerical experiments demonstrate that our method outperforms existing approaches in terms of accuracy and computational efficiency.

This paper is organized as follows: Section 2 presents the derivation of the proposed numerical scheme based on uniform hyperbolic polynomial B-splines. In Section 3, we establish stability of the proposed scheme to ensure predictable numerical results. Section 4 examines the convergence of the scheme to assess its accuracy and reliability. Section 5 provides numerical experiments and a comparative study of our results with those reported in earlier studies. Finally, Section 6 concludes the paper by highlighting key observations and findings.

2. Materials and methods

In this section, we construct and analyze an efficient finite difference scheme for solving (1.1), employing a θ-weighted approach for temporal discretization and a uniform hyperbolic polynomial B-spline for spatial discretization. The derivation of the numerical scheme is as follows:

For positive integers M and N, the spatial step size h and temporal step size Δt are defined as:

$$h = \frac{b-a}{M}, \quad \Delta t = \frac{T}{N}. \tag{2.1}$$

The spatial domain $[a, b]$ is divided uniformly with grid points $x_j = a + jh$ for $j = 0, 1, 2, .., M$ and the time interval $[0, T]$ is discretized with $t^n = n\Delta t$ for $n = 0, 1, 2, .., N$. The approximate solution at the point (x_j, t^n) is denoted by u_j^n. The spatial domain $[a, b]$ is divided into M sub-intervals $[x_j, x_{j+1}]$ with equally spaced knots x_j, where $a = x_0 < x_1 < ... < x_{M-1} < x_M = b$. The approximate solution $U(x, t)$ to analytical solution $u(x, t)$ is expresses as:

$$U(x,t) = \sum_{j=-3}^{M-1} \delta_j(t) H_{j,4}(x), \tag{2.2}$$

where $H_{j,4}$ are fourth-order uniform hyperbolic polynomial (UHP) B-spline basis functions and $\delta_j(t)$ are time-dependent unknown coefficients determined using the collocation method and boundary conditions. The uniform hyperbolic polynomial (UHP) B-spline basis introduced by Lu et al. [25] generalizes classical B-splines by incorporating hyperbolic functions. This enhances smoothness and flexibility, making it ideal for problems with inherent hyperbolic behavior (e.g., wave-dominated fractional PDEs). The fourth-order UHP B-spline $H_{j,4}(x)$ is constructed as a piecewise function with C^2-continuity, ensuring compatibility with second-order spatial derivatives in (1.1). It is defined as:

$$H_{j,4}(x) = \frac{1}{p_1} \begin{cases} \sinh(x - x_j) - (x - x_j), & x_j \le x \le x_{j+1}, \\ -2\sinh(x - x_{j+1}) - \sinh(x - x_{j+2}) \\ +(2\cosh(h) + 1)(x - x_{j+1}) - h, & x_{j+1} \le x \le x_{j+2}, \\ \sinh(x - x_{j+2}) + 2\sinh(x - x_{j+3}) \\ -(2\cosh(h) + 1)(x - x_{j+2}) + 2h\cosh(h), & x_{j+2} \le x \le x_{j+3}, \\ -\sinh(x - x_{j+4}) + (x - x_{j+4}), & x_{j+3} \le x \le x_{j+4}, \end{cases} \tag{2.3}$$

where $p_1 = 2h(\cosh(h) - 1)$.

The set $\{H_{-3,4}, H_{-2,4}, \ldots, H_{M-2,4}, H_{M-1,4}\}$ forms a basis over $[a, b]$. Due to the local support property of UHP B-splines, only three non-zero basis functions $H_{j-3,4}(x)$, $H_{j-2,4}(x)$, and $H_{j-1,4}(x)$ contribute at the grid point (x_j, t^n). Thus the approximate solution at the n-th time level is given by:

$$u(x_j, t^n) = u_j^n = \sum_{i=j-1}^{j+1} \delta_i^n H_{i-2,4}(x_j), \tag{2.4}$$

where δ_j^n are time-dependent unknowns calculated using initial and boundary conditions. From Eqs (2.3) and (2.4), the approximate solutions and their derivatives are expressed in terms of δ_j^n as:

$$\begin{cases} u_j^n \approx c_1\delta_{j-1}^n + c_2\delta_j^n + c_1\delta_{j+1}^n, \\ (u_j^n)_x \approx -c_3\delta_{j-1}^n + c_3\delta_{j+1}^n, \\ (u_j^n)_{xx} \approx c_4\delta_{j-1}^n + c_5\delta_j^n + c_4\delta_{j+1}^n, \end{cases} \tag{2.5}$$

where the coefficients c_i are given by:

$$\begin{cases} c_1 &= \frac{\sinh(h)-h}{p_1}, \\ c_2 &= \frac{-2(\sinh(h)-h\cosh(h))}{p_1}, \\ c_3 &= \frac{1}{2h}, \\ c_4 &= \frac{\sinh(h)}{p_1}, \\ c_5 &= \frac{-2\sinh(h)}{p_1}. \end{cases}$$

The coefficients c_1 to c_5 in Eq (2.5) are derived by evaluating $H_{j,4}(x)$ and its derivatives at the grid point x_j. For instance:

$$c_1 = \frac{\sinh(h) - h}{p_1}$$

results from $H_{j,4}(x_j)$ and $H_{j,4}(x_{j+1})$ due to local symmetry.

$$c_3 = \frac{1}{2h}$$

is obtained from the first derivative $H'_{j,4}(x_j)$, exploiting the central difference property. Crucially, each UHP B-spline $H_{j,4}(x)$ has local support over $[x_{j-3}, x_{j+1}]$. Thus, at any grid point x_j, only three adjacent basis functions — $H_{j-3,4}$, $H_{j-2,4}$, and $H_{j-1,4}$ — are non-zero. This locality reduces computational cost and ensures sparse linear systems in (2.12). The Caputo-Fabrizio fractional derivative (CFFD) in (1.1) is discretized using a θ-weighted finite difference method, while the spatial derivatives are approximated using UHP B-splines. Following Shafiq et al. [8], the CFFD is discretized at $t = t^{n+1}$,

$$\frac{\partial^\lambda u(x, t^{n+1})}{\partial t^\lambda} = \frac{M(\lambda)}{2-\lambda} \int_0^{t^{n+1}} \frac{\partial^2 u(x, s)}{\partial s^2} \exp\left[\frac{-\lambda}{2-\lambda}(t^{n+1} - s)\right] ds. \tag{2.6}$$

Using a central difference approach, Eq (2.6) becomes:

$$\frac{\partial^\lambda u(x, t^{n+1})}{\partial t^\lambda} = \frac{\eta M(\lambda)}{\lambda(\Delta t)^2} \sum_{k=0}^{n} w_k[u(x, t^{n-k+1}) - 2u(x, t^{n-k}) + u(x, t^{n-k-1})] + r_{\Delta t}^{n+1},$$

$$\frac{\partial^\lambda u^{n+1}}{\partial t^\lambda} = \frac{\eta M(\lambda)}{\lambda(\Delta t)^2}\sum_{k=0}^{n} w_k\left[u^{n-k+1} - 2u^{n-k} + u^{n-k-1}\right] + r_{\Delta t}^{n+1}, \tag{2.7}$$

where $u^n = u(x, t^n)$, $\eta = 1 - \exp\left(\frac{-\lambda}{2-\lambda}\Delta t\right)$, $w_k = \exp\left(\frac{-\lambda}{2-\lambda}k\Delta t\right)$, and $r_{\Delta t}^{n+1}$ is the truncation error given by:

$$r_{\Delta t}^{n+1} = C_u(\Delta t)^{3-\lambda}, \tag{2.8}$$

where C_u is a constant independent of Δt. The weights w_k satisfy the following properties:

- $w_k > 0, \quad k = 0, 1, \ldots, n,$
- $1 = w_0 > w_1 > w_2 > \cdots > w_n \quad$ and $\quad w_n \to 0$ as $n \to \infty,$
- $\sum_{k=0}^{n-1}(w_k - w_{k+1}) + w_n = 1.$

Using Eq (2.7) and the θ-weighted scheme, Eq (1.1) becomes:

$$\begin{aligned}&\frac{\eta M(\lambda)}{\lambda(\Delta t)^2}\sum_{k=0}^{n} w_k\left[u^{n-k+1} - 2u^{n-k} + u^{n-k-1}\right] + \mu_1\frac{(u^{n+1} - u^n)}{\Delta t}\\ &+ \theta(\mu_2 u^{n+1} - (u^{n+1})_{xx}) + (1-\theta)(\mu_2 u^n - (u^n)_{xx}) = f^{n+1}.\end{aligned} \tag{2.9}$$

For $\theta = 1$, the discretization of scheme (2.9) is given by:

$$\frac{\eta M(\lambda)}{\lambda(\Delta t)^2}\sum_{k=0}^{n} w_k\left[u^{n-k+1} - 2u^{n-k} + u^{n-k-1}\right] + \frac{\mu_1}{\Delta t}(u^{n+1} - u^n) + \mu_2 u^{n+1} - (u^{n+1})_{xx} = f^{n+1}. \tag{2.10}$$

This can be rewritten as:

$$(\alpha + \beta + \mu_2)u_j^{n+1} - (u_{xx})_j^{n+1} = (2\alpha + \beta)u_j^n - \alpha u_j^{n-1} - \alpha\sum_{k=1}^{n} w_k\left[u_j^{n-k+1} - 2u_j^{n-k} + u_j^{n-k-1}\right] + f_j^{n+1}, \tag{2.11}$$

where $\alpha = \frac{\eta M(\lambda)}{\lambda(\Delta t)^2}$ and $\beta = \frac{\mu_1}{\Delta t}$.

When we set $k = n$ or $n = 0$ in Eq (2.11), the term u^{-1} appeared. To handle this, we use the initial condition to compute u^{-1}:

$$u_t^0 = \frac{u^1 - u^{-1}}{2\Delta t} \implies u^{-1} = u^1 - 2\Delta t u_t^0 \implies u^{-1} = u^1 - 2\Delta t\phi_1(x).$$

Substituting the approximation (2.5) into (2.11) , we obtain the linear system:

$$\begin{aligned}&((\alpha + \beta + \mu_2)c_1 - c_4)\delta_{j-1}^{n+1} + ((\alpha + \beta + \mu_2)c_2 - c_5)\delta_j^{n+1} + ((\alpha + \beta + \mu_2)c_1 - c_4)\delta_{j+1}^{n+1}\\ &= (2\alpha + \beta)(c_1\delta_{j-1}^n + c_2\delta_j^n + c_1\delta_{j+1}^n) - \alpha(c_1\delta_{j-1}^{n-1} + c_2\delta_j^{n-1} + c_1\delta_{j+1}^{n-1})\\ &\quad - \alpha\sum_{k=1}^{n} w_k[(c_1\delta_{j-1}^{n-k+1} + c_2\delta_j^{n-k+1} + c_1\delta_{j+1}^{n-k+1}) - 2(c_1\delta_{j-1}^{n-k} + c_2\delta_j^{n-k} + c_1\delta_{j+1}^{n-k})\\ &\quad + (c_1\delta_{j-1}^{n-k-1} + c_2\delta_j^{n-k-1} + c_1\delta_{j+1}^{n-k-1})] + f_j^{n+1}.\end{aligned} \tag{2.12}$$

For $j = 0, 1, \cdots, M$, this generates $(M + 1)$ equations with $(M + 3)$ unknowns. To obtain a unique solution, we incorporate the boundary conditions (1.3):

$$\begin{cases} c_1\delta_{-3}^{n+1} + c_2\delta_{-2}^{n+1} + c_1\delta_{-1}^{n+1} = \psi_0(t^{n+1}), \\ c_1\delta_{M-3}^{n+1} + c_2\delta_{M-2}^{n+1} + c_1\delta_{M-1}^{n+1} = \psi_1(t^{n+1}). \end{cases} \tag{2.13}$$

The resulting $(M + 3) \times (M + 3)$ system is solved using numerical techniques such as Gaussian elimination.

The incorporation of the boundary conditions (2.13) closes the system, yielding a well-posed square linear system of dimension $(M + 3) \times (M + 3)$. Since the UHP B-spline basis functions are linearly independent and the scheme maintains local support, the resulting system matrix is sparse and structured. Under typical parameter settings, it is also diagonally dominant, ensuring solvability at each time step using standard numerical solvers.

2.1. Initial vector

The initial vector $D^0 = [\delta_{-3}^0, \delta_{-2}^0, \ldots, \ldots, \delta_{M-2}^0, \delta_{M-1}^0]^T$ is determined using the initial conditions:

$$\begin{cases} u_x(x_0, 0) = \phi_0'(x_0), \\ u(x_i, 0) = \phi_0(x_i), & i = 0, 1, 2, \ldots, n. \\ u_x(x_n, 0) = \phi_0'(x_n), \end{cases} \tag{2.14}$$

This leads to a $(M + 3) \times (M + 3)$ linear system:

$$\begin{bmatrix} -c_3 & 0 & c_3 & & & \\ c_1 & c_2 & c_1 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ & & \ddots & \vdots & \vdots & \vdots \\ & & & c_1 & c_2 & c_1 \\ & & & -c_3 & 0 & c_3 \end{bmatrix} \begin{bmatrix} \delta_{-3}^0 \\ \delta_{-2}^0 \\ \vdots \\ \vdots \\ \delta_{M-2}^0 \\ \delta_{M-1}^0 \end{bmatrix} = \begin{bmatrix} \phi_0'(x_0) \\ \phi_0(x_0) \\ \vdots \\ \vdots \\ \phi_0(x_n) \\ \phi_0'(x_n) \end{bmatrix}. \tag{2.15}$$

By solving Eq (2.15), we obtain D^0 which is used as the initial vector for solving Eq (2.12).

2.2. Computational algorithm

The computational steps of the proposed method are summarized in Figure 1. The implementation begins by specifying the domain parameters a, b, T, grid sizes M, N, fractional order λ, and coefficients μ_1, μ_2. The spatial and temporal domains are uniformly discretized with step sizes h and Δt, respectively. Using the initial conditions, the initial coefficient vector D^0 is computed by solving the system in Eq (2.15). At each time level, the linear system in Eq (2.12) is solved to obtain the coefficients δ_j^n, incorporating both the Caputo–Fabrizio derivative and UHP B-spline-based spatial terms. Finally, the approximate solution $U(x, t)$ is reconstructed using the B-spline expansion given in Eq (2.2). This process is repeated until the final time step.

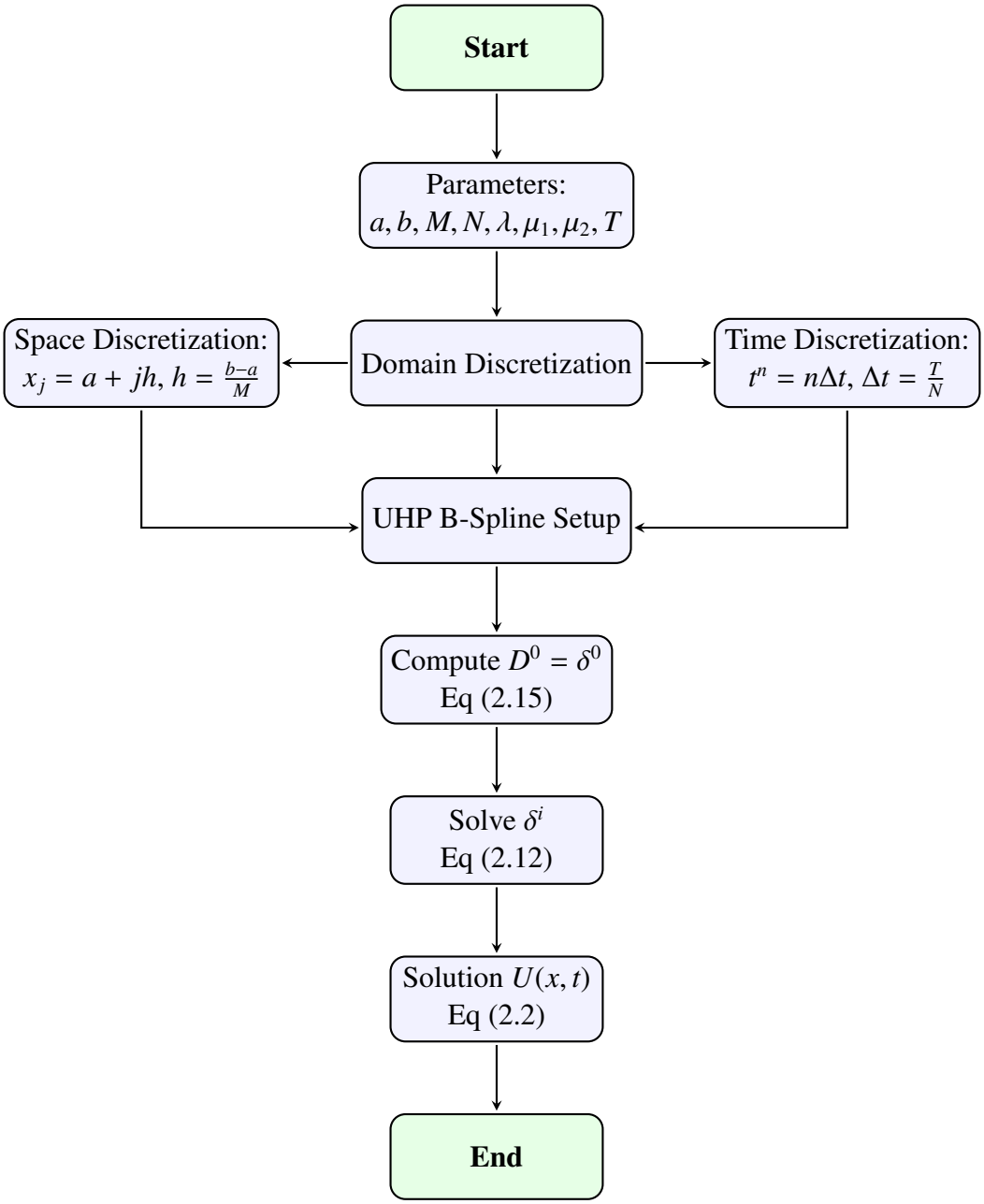


Figure 1. Flowchart of the uniform hyperbolic polynomial B-spline collocation method for solving fractional diffusion-wave equations.

3. Stability analysis

In this section, we present a detailed stability analysis of the proposed numerical scheme for solving Eq (2.12) using the Fourier (or von Neumann) method. The analysis employs Duhamel's principle, where the superposition of homogeneous solutions justifies neglecting the source term (i.e., $f = 0$).

Then the discrete scheme then reduces to:

$$\begin{aligned}
&((\alpha+\beta+\mu_2)c_1-c_4)\delta_{j-1}^{n+1}+((\alpha+\beta+\mu_2)c_2-c_5)\delta_j^{n+1}+((\alpha+\beta+\mu_2)c_1-c_4)\delta_{j+1}^{n+1}\\
&=(2\alpha+\beta)(c_1\delta_{j-1}^n+c_2\delta_j^n+c_1\delta_{j+1}^n)-\alpha(c_1\delta_{j-1}^{n-1}+c_2\delta_j^{n-1}+c_1\delta_{j+1}^{n-1})\\
&\quad-\alpha\sum_{k=1}^{n}w_k[(c_1\delta_{j-1}^{n-k+1}+c_2\delta_j^{n-k+1}+c_1\delta_{j+1}^{n-k+1})-2(c_1\delta_{j-1}^{n-k}+c_2\delta_j^{n-k}+c_1\delta_{j+1}^{n-k})\\
&\quad+(c_1\delta_{j-1}^{n-k-1}+c_2\delta_j^{n-k-1}+c_1\delta_{j+1}^{n-k-1})].
\end{aligned}\tag{3.1}$$

Let ϱ_j^n denote the analytical solution and $\tilde{\varrho}_j^n$ its numerical approximation. The error $E_j^n=\varrho_j^n-\tilde{\varrho}_j^n$ satisfies:

$$\begin{aligned}
&((\alpha+\beta+\mu_2)c_1-c_4)E_{j-1}^{n+1}+((\alpha+\beta+\mu_2)c_2-c_5)E_j^{n+1}+((\alpha+\beta+\mu_2)c_1-c_4)E_{j+1}^{n+1}\\
&=(2\alpha+\beta)(c_1E_{j-1}^n+c_2E_j^n+c_1E_{j+1}^n)-\alpha(c_1E_{j-1}^{n-1}+c_2E_j^{n-1}+c_1E_{j+1}^{n-1})\\
&\quad-\alpha\sum_{k=1}^{n}w_k[(c_1E_{j-1}^{n-k+1}+c_2E_j^{n-k+1}+c_1E_{j+1}^{n-k+1})-2(c_1E_{j-1}^{n-k}+c_2E_j^{n-k}+c_1E_{j+1}^{n-k})\\
&\quad+(c_1E_{j-1}^{n-k-1}+c_2E_j^{n-k-1}+c_1E_{j+1}^{n-k-1})].
\end{aligned}\tag{3.2}$$

The boundary and initial conditions for the error are given by:

$$E_0^n=\psi_0(t^n),\quad E_M^n=\psi_1(t^n),\qquad n=0,1,\ldots,N.\tag{3.3}$$

$$E_j^0=\phi_0(x_j),\quad (E_t)_j^0=\phi_1(x_j),\qquad j=1,2,\ldots,M.\tag{3.4}$$

The error $E^n(x)$ can be expanded in a Fourier series as:

$$E^n(x)=\sum_{m=-\infty}^{\infty}\xi^n(m)e^{i\frac{2\pi mx}{b-a}},\quad n=0,1,\ldots,N,$$

where $\xi^n(m)=\frac{1}{b-a}\int_a^b E^n(x)e^{-i\frac{2\pi mx}{b-a}}\,dx$. Let $E^n=\begin{bmatrix}E_1^n & E_2^n & \ldots & E_{M-1}^n\end{bmatrix}^T$. The L_2-norm of the error E^n is:

$$\|E^n\|_2=\left(\sum_{j=1}^{M-1}h|E_j^n|^2\right)^{\frac{1}{2}}=\left[\int_a^b|E^n(x)|^2\,dx\right]^{\frac{1}{2}}.$$

From the Parseval identity:

$$\int_a^b|E^n(x)|^2\,dx=\sum_{m=-\infty}^{\infty}|\xi^n(m)|^2\implies\|E^n\|_2^2=\sum_{m=-\infty}^{\infty}|\xi^n(m)|^2.\tag{3.5}$$

Assume a solution of the form $E_j^n=\eta_n e^{i\omega js}$, where $i=\sqrt{-1}$ and $\omega\in[-\pi,\pi]$. Substituting into (3.2) and dividing by $e^{i\omega js}$ yields:

$$\begin{aligned}
&\eta_{n+1}\Big[((\alpha+\beta+\mu_2)c_1-c_4)(e^{-i\omega s}+e^{i\omega s})+((\alpha+\beta+\mu_2)c_2-c_5)\Big]\\
&=\eta_n(2\alpha+\beta)(c_1(e^{-i\omega s}+e^{i\omega s})+c_2)-\eta_{n-1}\alpha(c_1(e^{-i\omega s}+e^{i\beta s})+c_2)\\
&\quad-\alpha\sum_{k=1}^{n}w_k[\eta_{n-k+1}(c_1(e^{-i\omega s}+e^{i\omega s})+c_2)\\
&\quad-2\eta_{n-k}(c_1(e^{-i\omega s}+e^{i\beta s})+c_2)+\eta_{n-k-1}(c_1(e^{-i\omega s}+e^{i\omega s})+c_2)].
\end{aligned}\tag{3.6}$$

Using the Euler identity $e^{-i\omega s} + e^{i\omega s} = 2\cos(\omega s)$ and simplifying for η_{n+1}, we obtain:

$$\left(\frac{2((\alpha+\beta+\mu_2)c_1 - c_4)\cos(\omega s) + (\alpha+\beta+\mu_2)c_2 - c_5}{2c_1\cos(\omega s) + c_2}\right)\eta_{n+1}$$
$$= (2\alpha+\beta)\,\eta_n - \eta_{n-1} - \alpha\sum_{k=1}^{n} w_k(\eta_{n-k+1} - 2\eta_{n-k} + \eta_{n-k-1}).$$

Assuming $\omega = 0$, the equation simplifies to:

$$\left(\alpha+\beta+\mu_2 - \frac{2c_4+c_5}{2c_1+c_2}\right)\eta_{n+1} = (2\alpha+\beta)\eta_n - \eta_{n-1} - \alpha\sum_{k=1}^{n} w_k(\eta_{n-k+1} - 2\eta_{n-k} + \eta_{n-k-1}).$$

This leads to the recurrence relation:

$$\eta_{n+1} = \frac{d}{\zeta}\eta_n - \frac{1}{\zeta}\eta_{n-1} - \frac{\alpha}{\zeta}\sum_{k=1}^{n} w_k(\eta_{n-k+1} - 2\eta_{n-k} + \eta_{n-k-1}), \tag{3.7}$$

where $d = 2\alpha + \beta$ and $\zeta = \left(\alpha+\beta+\mu_2 - \frac{2c_4+c_5}{2c_1+c_2}\right)$. Note that $2c_4 + c_5 = 0$, which shows that $\zeta > 1$.

Remark 3.1. Parseval's identity is used to express the square of the L^2-norm of the error in terms of the inner product of the error function with itself. This allows for energy-like estimates in the Fourier space, which are particularly useful in proving stability and convergence of numerical methods.

Proposition 3.1. *If η_n represents the solution of Eq* (3.7) *for $n = 0, 1, 2, \ldots, N$, then $|\eta_n| \le D|\eta_0|$, where $D = |d|^2$.*

Proof. We proceed by mathematical induction. For the base case ($n = 0$), Eq (3.7) yields:

$$|\eta_1| = |\frac{d\eta_0}{\zeta}| \le |d\eta_0| \le |d||\eta_0| \le |d|^2|\eta_0| = D|\eta_0|, \quad \because \zeta > 1.$$

Assume the inequality holds for all $k = 1, 2, \cdots, n$ i.e. $|\eta_k| \le |d||\eta_0| \le D|\eta_0|$. Then from (3.7):

$$\begin{aligned}
|\eta_{n+1}| &\le \frac{|d||\eta_n|}{\zeta} - \frac{1}{\zeta}|\eta_{n-1}| - \frac{|\alpha|}{\zeta}\sum_{k=1}^{n} w_k(|\eta_{n-k+1}| - 2|\eta_{n-k}| + |\eta_{n-k-1}|) \\
&\le |d||\eta_n| - |\eta_{n-1}| - \alpha\sum_{k=1}^{n} w_k(|\eta_{n-k+1}| - 2|\eta_{n-k}| + |\eta_{n-k-1}|) \\
&\le |d|^2|\eta_0| - |d||\eta_0| - \alpha|d|\sum_{k=1}^{n} w_k(|\eta_0| - 2|\eta_0| + |\eta_0|) \\
&\le |d|^2|\eta_0| = D|\eta_0|,
\end{aligned}$$

where we have used the inequality $||a| - |b|| \le |a - b|$ for $a, b \in \mathbb{R}$. Hence,

$$|\eta_n| \le D|\eta_0| \quad \text{for } n = 0, 1, \ldots, N. \tag{3.8}$$

This completes the proof. □

Remark on norms: Two types of norms are used in the stability and convergence analysis.

- **Discrete norm:** In the stability analysis (Theorem 3.1), the norm $\|E^n\|_2$ refers to the discrete approximation of the L_2-norm over the spatial domain, defined as:

$$\|E^n\|_2 = \sum_{j=1}^{M-1} h\,|E_j^n|^2.$$

- **Continuous norm:** In the convergence analysis (Theorem 4.1), the norm $\|e^n\|^2$ is defined using the standard inner product over the continuous domain $[a, b]$:

$$\langle u, v\rangle = \int_a^b u(x)\,v(x)\,dx, \quad \text{so that} \quad \|e^n\| = \langle e^n, e^n\rangle = \int_a^b |e^n(x)|^2 dx.$$

In both cases, the norms are used to measure the magnitude of the error and ensure the scheme's stability and convergence.

Theorem 3.1. *The proposed scheme (2.12) is unconditionally stable.*

Proof. From Proposition 3.1 and Parseval's identity, we have:

$$\|E_n\|_2^2 = \sum_{m=-\infty}^{\infty} |\xi_n(m)|^2 \le D \sum_{m=-\infty}^{\infty} |\xi_0(m)|^2 = D\|E_0\|_2^2,$$

which confirms the unconditional stability of the scheme. □

4. Convergence analysis

In this section, we establish convergence estimates for the scheme described in Eq (2.12), demonstrating that our method provides high accuracy compared to traditional discretization approaches.

4.1. Temporal convergence analysis

Beginning with ($f = 0$) in Eq (2.12), we convert it to homogeneous form and express the summation term as:

$$\begin{aligned} &\sum_{k=0}^{n} w_k \left(u^{n-k+1} - 2u^{n-k} + u^{n-k-1}\right) \\ &= u^{n+1} - u^n + w_n u^1 - w_n u^0 + \sum_{k=0}^{n-1} (w_{k+1} - w_k)(u^{n-k} - u^{n-k-1}) - 2w_n \Delta t \phi_1(x). \end{aligned} \tag{4.1}$$

Substituting (4.1) into the homogeneous form of (2.10) yields:

$$\begin{aligned} &(\alpha + \beta + \mu_2)u^{n+1} - (u_{xx})^{n+1} \\ &= (\alpha + \beta)u^n - \alpha w_n u^1 + \alpha w_n u^0 - \alpha \sum_{k=0}^{n-1} (w_{k+1} - w_k)(u^{n-k} - u^{n-k-1}) - 2\alpha w_n \Delta t \phi_1(x). \end{aligned} \tag{4.2}$$

Theorem 4.1. *Let* $\{u(x,t^n)\}_{n=0}^{N-1}$ *be the analytical solution of* (1.1) *with initial and boundary conditions* (1.2) *and* (1.3)*, and let* $\{u^n\}_{n=0}^{N-1}$ *be the numerical solution of* (4.2)*. Then the error* $e^{n+1} = u(x,t^{n+1}) - u^{n+1}$ *satisfies:*

$$\|e^{n+1}\| \leq C\Delta t^{3-\lambda},$$

where C is a constant independent of Δt*.*

Proof. The exact solution u satisfies the semi-discrete scheme (4.2) up to the truncation error $r_{\Delta t}^{n+1}$:

$$\begin{aligned}
&(\alpha+\beta+\mu_2)u(x,t^{n+1}) - (u(x,t^{n+1}))_{xx} \\
=\ &(\alpha+\beta)u(x,t^n) - \alpha w_n u(x,t^1) + \alpha w_n u(x,t^0) - \alpha\sum_{k=0}^{n-1}(w_{k+1}-w_k)(u(x,t^{n-k}) - u(x,t^{n-k-1})) \\
&- 2\alpha w_n \Delta t\phi_1(x) + r_{\Delta t}^{n+1}.
\end{aligned} \tag{4.3}$$

Subtracting (4.2) from (4.3) gives the error equation:

$$\begin{aligned}
&(\alpha+\beta+\mu_2)e^{n+1} - (e_{xx})^{n+1} \\
=\ &(\alpha+\beta)e^n - \alpha w_n e^1 + \alpha w_n e^0 - \alpha\sum_{k=0}^{n-1}(w_{k+1}-w_k)(e^{n-k} - e^{n-k-1}) + r_{\Delta t}^{n+1}.
\end{aligned} \tag{4.4}$$

With $e^0 = 0$, taking the inner product with e^{n+1} yields:

$$\begin{aligned}
(\alpha+\beta+\mu_2)\|e^{n+1}\|^2 =\ &\langle (e^{n+1})_{xx}, e^{n+1}\rangle + (\alpha+\beta)\langle e^n, e^{n+1}\rangle - \alpha w_n\langle e^1, e^{n+1}\rangle \\
&- \alpha\sum_{k=1}^{n}(w_{k+1}-w_k)\big[\langle e^{n-k}, e^{n+1}\rangle - \langle e^{n-k-1}, e^{n+1}\rangle\big] + \langle r_{\Delta t}^{n+1}, e^{n+1}\rangle \\
=\ &-\|(e^{n+1})_x\|^2 + (\alpha+\beta)\langle e^n, e^{n+1}\rangle - \alpha w_n\langle e^1, e^{n+1}\rangle \\
&- \alpha\sum_{k=1}^{n}(w_{k+1}-w_k)\big[\langle e^{n-k}, e^{n+1}\rangle - \langle e^{n-k-1}, e^{n+1}\rangle\big] + \langle r_{\Delta t}^{n+1}, e^{n+1}\rangle \\
\leq\ &(\alpha+\beta)\langle e^n, e^{n+1}\rangle - \alpha w_n\langle e^1, e^{n+1}\rangle - \alpha\sum_{k=0}^{n-1}(w_{k+1}-w_k)\big[\langle e^{n-k}, e^{n+1}\rangle \\
&- \langle e^{n-k-1}, e^{n+1}\rangle\big] + \langle r_{\Delta t}^{n+1}, e^{n+1}\rangle,
\end{aligned}$$

where we have used $\langle u_{xx}, u\rangle = -\langle u_x, u_x\rangle$ from [28] with the property $-\|(e^{n+1})_x\|^2 \leq 0$. Applying the Cauchy–Schwarz inequality $\langle x, y\rangle \leq \|x\|\|y\|$, we obtain:

$$\begin{aligned}
(\alpha+\beta+\mu_2)\|e^{n+1}\|^2 \leq\ &(\alpha+\beta)\|e^n\|\|e^{n+1}\| - \alpha w_n\|e^1\|\|e^{n+1}\| - \alpha\sum_{k=0}^{n-1}(w_{k+1}-w_k)\big[\|e^{n-k}\|\|e^{n+1}\| \\
&- \|e^{n-k-1}\|\|e^{n+1}\|\big] + \|r_{\Delta t}^{n+1}\|\|e^{n+1}\|.
\end{aligned}$$

Dividing the above equation by $\|e^{n+1}\|$ gives:

$$(\alpha+\beta+\mu_2)\|e^{n+1}\| \leq (\alpha+\beta)\|e^n\| - \alpha w_n\|e^1\| - \alpha\sum_{k=0}^{n-1}(w_{k+1}-w_k)(\|e^{n-k}\| - \|e^{n-k-1}\|) + \|r_{\Delta t}^{n+1}\|. \tag{4.5}$$

We prove this using mathematical induction. For the base case ($n = 0$):

$$\begin{aligned}(2\alpha+\beta+\mu_2)\|e^1\| &\le \|r_{\Delta t}^1\| \\ &\Rightarrow \|e^1\| \le \frac{1}{(2\alpha+\beta+\mu_2)}\|r_{\Delta t}^1\| \\ &\Rightarrow \|e^1\| \le C_0\Delta t^{3-\lambda}, \quad \text{where } C_0 \text{ is a constant independent of } \Delta t.\end{aligned}$$

Now assume the induction hypothesis holds for $n = q-1$ and then $\|e^q\| \le C\Delta t^{3-\lambda}$, where C is a constant independent of Δt.

For $n = q$, we have:

$$(\alpha+\beta+\mu_2)\|e^{q+1}\| \le (\alpha+\beta)\|e^q\| - \alpha w_q\|e^1\| - \alpha\sum_{k=0}^{q-1}(w_{k+1}-w_k)(\|e^{q-k}\| - \|e^{q-k-1}\|) + \|r_{\Delta t}^{q+1}\|. \tag{4.6}$$

Let $\|e^1\| \le C_1\Delta t^{3-\lambda}$, $F = \max_{0\le i\le q}\|e^i\| \le C_2\Delta t^{3-\lambda}$, and $G = \max_{0\le k\le q}\{\|e^{q-k}\| - \|e^{q-k-1}\|\} \le C_3\Delta t^{3-\lambda}$, where C_1, C_2, and C_3 are constants independent of Δt.

$$\begin{aligned}(\alpha+\beta+\mu_2)\|e^{q+1}\| &\le (\alpha+\beta)C_2\Delta t^{3-\lambda} - \alpha w_q C_1\Delta t^{3-\lambda} + \alpha(1-w_n)C_3\Delta t^{3-\lambda} + C_u\Delta t^{3-\lambda} \\ &\le \Big[\underbrace{(\alpha+\beta)C_2 - \alpha w_q C_1 + \alpha(1-w_n)C_3 + C_u}_{constant}\Big]\Delta t^{3-\lambda}.\end{aligned}$$

Note that the constant term is independent of Δt. Hence:

$$\|e^{q+1}\| \le C\Delta t^{3-\lambda}.$$

This completes the proof. □

4.2. Spatial convergence analysis

The methodology given in [8] is used to investigate the convergence of the proposed numerical scheme. We begin with the following fundamental convergence theorem:

Lemma 4.1. *The set $\{H_{-3,4}, H_{-2,4}, \ldots, H_{M-1,4}\}$ of uniform hyperbolic polynomial B-spline basis functions of order 4 satisfies the inequality:*

$$\sum_{j=-3}^{M-1}\left|H_{j,4}(x)\right| \le K, \quad \forall x \in [a,b], \tag{4.7}$$

where K is a positive constant independent of h and x, and the coefficients satisfy $2c_1 + c_2 = 1$.

Proof. Consider the local support property where at most four basis functions are non-zero for any $x \in [a,b]$.

Case 1: At knots $x = x_j$:

$$\sum_{k=-3}^{M-1}|H_{k,4}(x_j)| = |H_{j-3,4}(x_j)| + |H_{j-2,4}(x_j)| + |H_{j-1,4}(x_j)|$$

$$= c_1 + c_2 + c_1 = 2c_1 + c_2 = 1.$$

Case 2: For $x \in (x_j, x_{j+1})$:

$$\begin{aligned}\sum_{k=-3}^{M-1} |H_{k,4}(x)| &\le |H_{j-3,4}(x)| + |H_{j-2,4}(x)| + |H_{j-1,4}(x)| + |H_{j,4}(x)| \\ &\le c_1 + c_2 + c_2 + c_1 = 2(c_1 + c_2) \\ &= (2c_1 + c_2) + c_2 = 1 + c_2.\end{aligned}$$

Boundary Cases: At $x = a$ or $x = b$, fewer than four basis functions are non-zero, so:

$$\sum_{j=-3}^{M-1} |H_{j,4}(x)| < \max(1, 1 + c_2).$$

Taking $K = \max(1, 1 + |c_2|)$ completes the proof. □

Theorem 4.2. *Let $u(x,t) \in C^4[a,b] \times C^2[0,\infty)$. For a uniform partition $\chi = \{a = x_0 < \cdots < x_M = b\}$ with $x_j = a + jh$, $h = (b-a)/M$, let $\widetilde{U}(x,t)$ be the hyperbolic B-spline interpolant of $u(x,t)$ at the knots x_j. Then there exist constants $\rho_j > 0$ independent of h such that:*

$$\|D^j(u - \widetilde{U})\|_\infty \le \rho_j h^{4-j}, \quad j = 0, 1, 2. \tag{4.8}$$

Theorem 4.3. *The numerical solution $U(x,t)$ converges to the exact solution $u(x,t)$ of the TFDWE. Moreover, if $q \in C^2[a,b]$, then there exists a constant $\widetilde{\rho} > 0$ independent of h such that for all $t \ge 0$,*

$$\|u(x,t) - U(x,t)\|_\infty \le \widetilde{\rho} h^2, \tag{4.9}$$

provided h is sufficiently small.

Proof. Decompose the error using the interpolant $\widetilde{U}(x,t) = \sum_{j=-3}^{M-1} d_j(t) H_{j,4}(x)$:

$$\|u - U\|_\infty \le \underbrace{\|u - \widetilde{U}\|_\infty}_{\text{Interpolation error}} + \underbrace{\|\widetilde{U} - U\|_\infty}_{\text{Discretization error}}. \tag{4.10}$$

From Theorem 4.2 with $j = 0$:

$$\|u - \widetilde{U}\|_\infty \le \rho_0 h^4. \tag{4.11}$$

For the discretization error $e(x,t) = \widetilde{U} - U$, consider the collocation equations:

$$\begin{aligned}\mathcal{L}u(x_j,t) &= f(x_j,t), \\ \mathcal{L}\widetilde{U}(x_j,t) &= f(x_j,t) + \tau_j(t),\end{aligned}$$

where τ_j is the spatial truncation error satisfying $\|\tau_j\|_\infty \le C_\tau h^2$ due to the approximation properties of the B-splines. The error equation at the spatial nodes is:

$$\begin{aligned}&[(\alpha + \beta + \mu_2)(2c_1 + c_2) - (2c_4 + c_5)]e_j^{n+1} \\ =\ & (2\alpha + \beta)(c_1 e_{j-1}^n + c_2 e_j^n + c_1 e_{j+1}^n) - \alpha(c_1 e_{j-1}^{n-1} + c_2 e_j^{n-1} + c_1 e_{j+1}^{n-1}) - \alpha \sum_{k=1}^{n} w_k \Delta^2 e_j^{n-k} + \tau_j^{n+1},\end{aligned} \tag{4.12}$$

where $\Delta^2 e_j^m = e_j^{m+1} - 2e_j^m + e_j^{m-1}$ and $\|\tau_j^{n+1}\|_\infty \leq C_\tau h^2$. Using $2c_1 + c_2 = 1$ and $2c_4 + c_5 = 0$, and defining $\Gamma = \alpha + \beta + \mu_2 > 0$:

$$\Gamma\|e^{n+1}\|_\infty \leq (2\alpha + \beta)\|e^n\|_\infty + \alpha\|e^{n-1}\|_\infty + \alpha \sum_{k=1}^{n} w_k \|\Delta^2 e^{n-k}\|_\infty + C_\tau h^2. \tag{4.13}$$

Base Case ($n = 0$)**:**

$$\Gamma\|e^1\|_\infty \leq C_\tau h^2 \implies \|e^1\|_\infty \leq \frac{C_\tau}{\Gamma} h^2.$$

Inductive step: Assume $\|e^k\|_\infty \leq \rho h^2$ for $k \leq n$. Then:

$$\Gamma\|e^{n+1}\|_\infty \leq \left[(2\alpha + \beta) + \alpha + 4\alpha \sum_{k=1}^{n} w_k\right] \rho h^2 + C_\tau h^2.$$

Since $\sum_{k=1}^{n} w_k \leq W$ by the properties of w_k, we obtain:

$$\|e^{n+1}\|_\infty \leq \left[\frac{(3\alpha + \beta + 4\alpha W)\rho + C_\tau}{\Gamma}\right] h^2.$$

Choose ρ large enough so that $\frac{(3\alpha+\beta+4\alpha W)}{\Gamma}\rho + \frac{C_\tau}{\Gamma} \leq \rho$ yields uniform boundedness.

By Lemma 4.1 and the boundedness of the basis:

$$\|\widetilde{U} - U\|_\infty \leq K\rho h^2. \tag{4.14}$$

Combining (4.11) and (4.14):

$$\|u - U\|_\infty \leq \rho_0 h^4 + K\rho h^2 \leq \widetilde{\rho} h^2, \tag{4.15}$$

where $\widetilde{\rho} = \rho_0 h_0^2 + K\rho$ for $h < h_0$, which completes the proof. □

5. Numerical experiments

This section presents numerical experiments to validate the accuracy and efficiency of our proposed scheme (2.12). Comprehensive comparisons are conducted with three established methods: the cubic B-spline approach [8], meshless Galerkin method [10], and the spectral tau algorithm based on the Jacobi operational matrix [29]. For quantitative comparison, we employ the following error metrics:

- Absolute error;
- L_∞ error norm;
- L_2 error norm.

Let $u^{\text{ext}}(x_i)$ and $U^{\text{app}}(x_i)$ denote the exact and approximate solutions at spatial grid point x_i and fixed time t, respectively. The error measures are defined as:

$$\begin{aligned} \text{Absolute error} &= \left|u^{\text{ext}}(x_i) - U^{\text{app}}(x_i)\right|, && \forall x_i \in [a, b], \\ L_\infty &= \max_{0 \leq i \leq M} \left|u^{\text{ext}}(x_i) - U^{\text{app}}(x_i)\right|, && \forall x_i \in [a, b], \end{aligned}$$

$$L_2 = \sqrt{h \sum_{i=0}^{M} (u^{\text{ext}}(x_i) - U^{\text{app}}(x_i))^2}, \qquad \forall x_i \in [a, b],$$

where h is the spatial step size and M the number of grid points. The convergence rates for spatial and temporal refinements are, respectively given by:

$$p_h = \frac{\log(\mathcal{E}(h_2)/\mathcal{E}(h_1))}{\log(h_2/h_1)}, \qquad h_2 > h_1,$$
$$p_{\Delta t} = \frac{\log(\mathcal{E}(\Delta t_2)/\mathcal{E}(\Delta t_1))}{\log(\Delta t_2/\Delta t_1)}, \qquad \Delta t_2 > \Delta t_1,$$

where $\mathcal{E}$ denotes the L_∞ error norm.

Example 5.1. Consider the TFDW Eq (1.1) with damping and reaction coefficients $\mu_1 = 0$ and $\mu_2 = 0$:

$$\frac{\partial^\lambda u(x,t)}{\partial t^\lambda} - \frac{\partial^2 u(x,t)}{\partial x^2} = f(x,t), \quad a \le x \le b, \quad t \ge 0.$$

The initial and boundary conditions are:

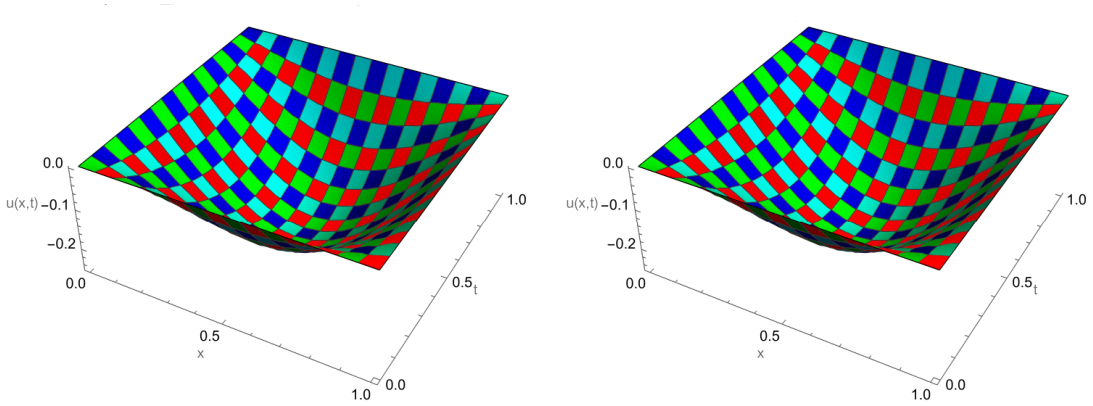
$$\text{Initial Conditions (ICs)}: \quad \phi_0(x) = 0, \quad \phi_1(x) = -\sin(\pi x),$$
$$\text{Boundary Conditions (BCs)}: \quad \psi_0(t) = 0, \quad \psi_1(t) = 0.$$

The source term $f(x,t)$ is given as:

$$f(x,t) = 2\left(\frac{M(\lambda)}{\lambda}\right)\left(1 - \exp\left[\frac{-\lambda t}{2-\lambda}\right]\right)\sin(\pi x) + \pi^2(t^2 - t)\sin(\pi x).$$

The analytical solution of the problem is given by $u(x,t) = (t^2 - t)\sin(\pi x)$.

The numerical solution is computed using our proposed scheme (2.12). All numerical results and figures were obtained for the case where $a = 0$ and $b = 1$. Figure 2 displays a 3D plot of the approximate and analytical solutions for the parameter $\lambda = 1.5$, while Figure 3 illustrates the 3D error between them. Figure 4 compares the approximate and exact solutions at different time levels with temporal size $\Delta t = 0.01$. Figure 5 shows a 2D graph of the absolute error at three different times with temporal step size $\Delta t = 0.001$, illustrating the variation in error magnitudes across the domain. The absolute errors in Table 1 and the error norms (L_2 and L_∞) in Table 2 are compared with the numerical scheme reported in [8]. The obtained convergence order in Table 3 closely matches the theoretically calculated value. In Table 4, the error norms (L_2 and L_∞) are reported for various values of λ and time t. The absolute errors at different points $(x_j, t^n) \in [0,1] \times [0,1]$ for $1 < \lambda < 2$ with $\Delta t = 0.01$ and $M = 40$ are shown in Table 5. Equations (5.1) and (5.2) represent piecewise numerical solutions to this problem obtained with parameters $\Delta t = 0.01$, $M = 10$, and $\lambda = 1.5$.



(a) Space-time graph of approximate solution at $t = 1$. (b) Space-time graph of exact solution at $t = 1$.

Figure 2. 3D comparison of numerical approximation (a) and exact (b) solutions for Example 5.1 with temporal step size $\Delta t = 0.001$ and $M = 180$.

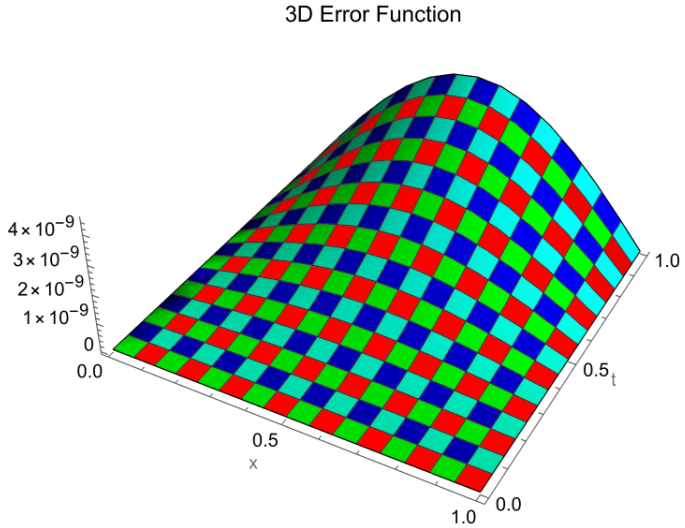


Figure 3. 3D error distribution for Example 5.1 with the parameters $M = 180$, $\Delta t = 0.001$, and $\lambda = 1.5$ at time $t = 1$.

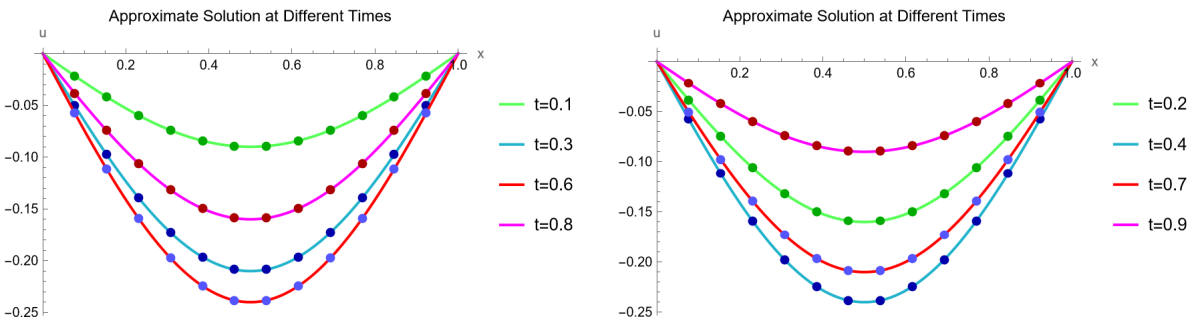


Figure 4. Comparison of numerical solution (solid line) with exact solution (dots) at various time levels for Example 5.1 when $M = 250$ and $\lambda = 1.5$.

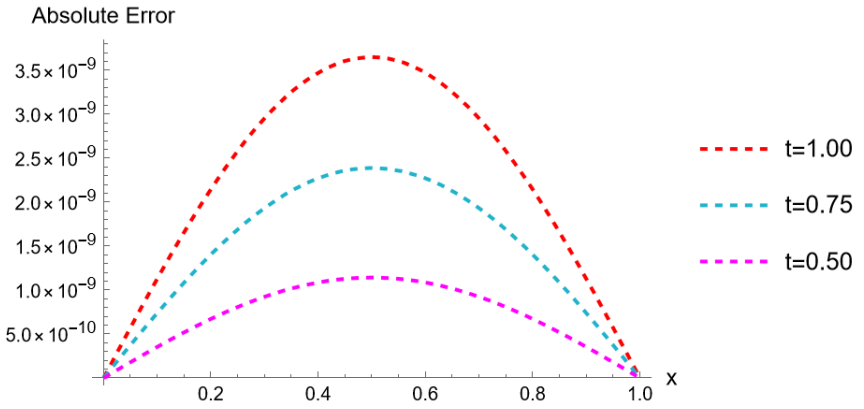


Figure 5. 2D absolute error function plot at different times when parameters are $\lambda = 1.5$ and $M = 200$ for Example 5.1.

Table 1. The absolute error comparison for Example 5.1 at $t = 0.2$, with $\Delta t = 0.02$, $M = 500$, and $\lambda = 1.8$.

	Present method			Method [8]
x	Exact result	Approximate result	Absolute error	Absolute error
0.1	−0.04944271909999	−0.04944271917095	7.09607×10^{-11}	4.81652×10^{-8}
0.2	−0.09404564036679	−0.09404564034156	2.52353×10^{-11}	9.15580×10^{-8}
0.3	−0.12944271909999	−0.12944271906514	3.48481×10^{-11}	1.25991×10^{-7}
0.4	−0.15216904260723	−0.15216904256618	4.10496×10^{-11}	1.48096×10^{-7}
0.5	−0.16000000000000	−0.15999999995677	4.32329×10^{-11}	1.55709×10^{-7}
0.6	−0.15216904260723	−0.15216904256604	4.11843×10^{-11}	1.48085×10^{-7}
0.7	−0.12944271909999	−0.12944271906488	3.51043×10^{-11}	1.25967×10^{-7}
0.8	−0.09404564036680	−0.09404564034121	2.55880×10^{-11}	9.15200×10^{-8}
0.9	−0.04944271909999	−0.04944271908643	1.35670×10^{-11}	4.81148×10^{-8}

Table 2. Comparison of error norms for Example 5.1 at time $t = 1$, $\Delta t = \frac{1}{120}$, and $\lambda = 1.5$.

	Present method			Method [8]		
M	L_∞ Error	L_2 Error	p_h	L_∞ Error	L_2 Error	p_h
10	1.45731×10^{-6}	1.03047×10^{-6}	− − −−	1.11825×10^{-3}	7.90720×10^{-4}	− − −−
20	3.63490×10^{-7}	2.57026×10^{-7}	2.003320	2.82301×10^{-4}	1.99617×10^{-4}	1.985932
40	9.08195×10^{-8}	6.42191×10^{-8}	2.000842	7.07469×10^{-5}	5.00256×10^{-5}	1.996495
80	2.27015×10^{-8}	1.60524×10^{-8}	2.000215	1.76975×10^{-5}	1.25140×10^{-5}	1.999124
160	5.67518×10^{-9}	4.01296×10^{-9}	2.000050	4.42504×10^{-6}	3.12898×10^{-6}	1.999781

Table 3. Temporal convergence order for Example 5.1 at time $t = 1$ and $M = 20$.

	$\lambda = 1.5$			$\lambda = 1.7$		
Δt	L_∞ Error	L_2 Error	$p_{\Delta t}$	L_∞ Error	L_2 Error	$p_{\Delta t}$
$\frac{1}{10}$	3.38792×10^{-7}	2.39562×10^{-7}	$----$	2.07164×10^{-7}	1.46487×10^{-7}	$----$
$\frac{1}{20}$	1.50145×10^{-7}	1.06169×10^{-7}	1.17404	1.08439×10^{-7}	7.66777×10^{-8}	0.93389
$\frac{1}{40}$	6.17627×10^{-8}	4.36728×10^{-8}	1.28155	5.13504×10^{-8}	3.63102×10^{-8}	1.07843
$\frac{1}{80}$	2.41769×10^{-8}	1.70956×10^{-8}	1.35311	2.27819×10^{-8}	1.61092×10^{-8}	1.17249
$\frac{1}{160}$	9.10207×10^{-9}	6.43616×10^{-9}	1.40936	9.47697×10^{-9}	6.70129×10^{-9}	1.26539

Table 4. Error norms at distinct time levels and values of λ for Example 5.1, with $\Delta t = 0.01$ and $M = 200$.

T	L_2	L_∞	L_2	L_∞
	$\lambda = 1.3$		$\lambda = 1.5$	
0.2	3.25674×10^{-10}	4.60572×10^{-10}	1.30069×10^{-10}	1.83944×10^{-10}
0.4	1.23863×10^{-9}	1.75168×10^{-9}	5.22603×10^{-10}	7.39078×10^{-10}
0.6	2.56841×10^{-9}	3.63227×10^{-9}	1.13641×10^{-9}	1.60713×10^{-9}
0.8	4.04544×10^{-9}	5.72111×10^{-9}	1.8708×10^{-9}	2.64572×10^{-9}
1.0	5.30106×10^{-9}	7.49683×10^{-9}	2.5656×10^{-9}	3.62831×10^{-9}
	$\lambda = 1.7$		$\lambda = 1.9$	
0.2	5.72117×10^{-11}	8.0909×10^{-11}	2.25948×10^{-11}	3.19534×10^{-11}
0.4	2.52765×10^{-10}	3.57463×10^{-10}	1.22634×10^{-10}	1.73429×10^{-10}
0.6	5.90575×10^{-10}	8.35208×10^{-10}	3.23615×10^{-10}	4.57661×10^{-10}
0.8	1.03206×10^{-9}	1.45956×10^{-9}	6.15979×10^{-10}	8.71128×10^{-10}
1.0	1.49613×10^{-9}	2.11586×10^{-9}	9.5712×10^{-10}	1.35357×10^{-9}

Table 5. Absolute errors at points (x_j, t^n) for $1 < \lambda < 2$ with $\Delta t = 0.01$ and $M = 40$ for Example 5.1.

(x, t)	$\lambda = 1.1$	$\lambda = 1.3$	$\lambda = 1.5$	$\lambda = 1.7$	$\lambda = 1.9$
$(0.1, 0.1)$	4.019756×10^{-9}	9.030930×10^{-10}	3.493754×10^{-10}	1.443957×10^{-10}	4.794180×10^{-11}
$(0.2, 0.2)$	2.954954×10^{-8}	6.769167×10^{-9}	2.703469×10^{-9}	1.189118×10^{-9}	4.695940×10^{-10}
$(0.3, 0.3)$	8.791703×10^{-8}	2.051442×10^{-8}	8.431334×10^{-9}	3.903956×10^{-9}	1.736451×10^{-9}
$(0.4, 0.4)$	1.753975×10^{-7}	4.165595×10^{-8}	1.757528×10^{-8}	8.500263×10^{-9}	4.123779×10^{-9}
$(0.5, 0.5)$	2.730976×10^{-7}	6.598349×10^{-8}	2.852846×10^{-8}	1.433259×10^{-8}	7.437915×10^{-9}
$(0.6, 0.6)$	3.515097×10^{-7}	8.637889×10^{-8}	3.821915×10^{-8}	1.986202×10^{-8}	1.088378×10^{-8}
$(0.7, 0.7)$	3.787686×10^{-7}	9.467218×10^{-8}	4.283130×10^{-8}	2.295641×10^{-8}	1.316756×10^{-8}
$(0.8, 0.8)$	3.305998×10^{-7}	8.408470×10^{-8}	3.888441×10^{-8}	2.145093×10^{-8}	1.280264×10^{-8}
$(0.9, 0.9)$	1.994235×10^{-7}	5.166315×10^{-8}	2.442816×10^{-8}	1.385602×10^{-8}	8.571595×10^{-9}

Solution at time $t = 1$ (from (2.2)):

$$U(x,1) = \begin{cases} -2.9073 \times 10^{-13} + 0.000049x + 2.9073 \times 10^{-13}\cosh(x) - 0.00005\sinh(x), & 0 \le x \le \frac{1}{10} \\ 4.82602 \times 10^{-7} + 0.0000445x - 4.83407 \times 10^{-7}\cosh(x) - 0.00004\sinh(x), & \frac{1}{10} \le x \le \frac{2}{10} \\ 2.31851 \times 10^{-6} + 0.0000353x - 2.33157 \times 10^{-6}\cosh(x) - 0.00003\sinh(x), & \frac{2}{10} \le x \le \frac{3}{10} \\ 6.10886 \times 10^{-6} + 0.0000227x - 6.17904 \times 10^{-6}\cosh(x) - 0.00002\sinh(x), & \frac{3}{10} \le x \le \frac{4}{10} \\ 0.0000121 + 7.8085 \times 10^{-6}x - 0.0000123\cosh(x) - 1.2499 \times 10^{-6}\sinh(x), & \frac{4}{10} \le x \le \frac{5}{10} \\ 0.0000199 - 7.8086 \times 10^{-6}x - 0.0000204183\cosh(x) + 0.00001636\sinh(x), & \frac{5}{10} \le x \le \frac{6}{10} \\ 0.000028770 - 0.000022662x - 0.000029874\cosh(x) + 0.000033968\sinh(x), & \frac{6}{10} \le x \le \frac{7}{10} \\ 0.000037615 - 0.000035296x - 0.000039458\cosh(x) + 0.000049826\sinh(x), & \frac{7}{10} \le x \le \frac{8}{10} \\ 0.000044958 - 0.000044476x - 0.000047611\cosh(x) + 0.000062104\sinh(x), & \frac{8}{10} \le x \le \frac{9}{10} \\ 0.000049302 - 0.000049302x - 0.000052565\cosh(x) + 0.000069019\sinh(x), & \frac{9}{10} \le x \le 1 \end{cases}. \tag{5.1}$$

Solution at time $t = 0.5$ (from (2.2)):

$$U(x,0.5) = \begin{cases} -1.827 \times 10^{-12} - 8.4665x + 1.8278 \times 10^{-12}\cosh(x) + 7.6811\sinh(x), & 0 \le x \le \frac{1}{10} \\ -0.08288 - 7.63773x + 0.08301\cosh(x) + 6.84824\sinh(x), & \frac{1}{10} \le x \le \frac{2}{10} \\ -0.39816 - 6.06134x + 0.40040\cosh(x) + 5.24021\sinh(x), & \frac{2}{10} \le x \le \frac{3}{10} \\ -1.04907 - 3.89162x + 1.06112\cosh(x) + 2.97212\sinh(x), & \frac{3}{10} \le x \le \frac{4}{10} \\ -2.06933 - 1.34096x + 2.10881\cosh(x) + 0.21467\sinh(x), & \frac{4}{10} \le x \le \frac{5}{10} \\ -3.41030 + 1.34096x + 3.50635\cosh(x) - 2.80953\sinh(x), & \frac{5}{10} \le x \le \frac{6}{10} \\ -4.94069 + 3.89162x + 5.13023\cosh(x) - 5.83325\sinh(x), & \frac{6}{10} \le x \le \frac{7}{10} \\ -6.45949 + 6.06134x + 6.77615\cosh(x) - 8.55662\sinh(x), & \frac{7}{10} \le x \le \frac{8}{10} \\ -7.72061 + 7.63773x + 8.17615\cosh(x) - 10.6649\sinh(x), & \frac{8}{10} \le x \le \frac{9}{10} \\ -8.46649 + 8.46649x + 9.02689\cosh(x) - 11.8526\sinh(x), & \frac{9}{10} \le x \le 1 \end{cases}. \tag{5.2}$$

Example 5.2. Consider the TFDW Eq (1.1) with damping and reaction coefficients $\mu_1 = 1$ and $\mu_2 = 0$:

$$\frac{\partial^\lambda u(x,t)}{\partial t^\lambda} + \frac{\partial u(x,t)}{\partial t} - \frac{\partial^2 u(x,t)}{\partial x^2} = f(x,t), \quad a \le x \le b, \quad t \ge 0.$$

The initial and boundary conditions are given by:

$$\begin{aligned} &\text{Initial Conditions (ICs)}: \quad \phi_0(x) = 0, \quad \phi_1(x) = 0, \\ &\text{Boundary Conditions (BCs)}: \quad \psi_0(t) = 0, \quad \psi_1(t) = (b-1)t^2. \end{aligned}$$

The analytical solution to the given problem is $u(x,t) = x(b-x)t^2$ and the source term is

$$f(x,t) = \frac{2M(\lambda)}{\lambda}x(b-x)\left(1 - \exp\left[\frac{-\lambda t}{2-\lambda}\right]\right) + 2x(b-x)t + 2t^2.$$

We employ the proposed scheme (2.12) to approximate the solution of the problem. All numerical results and figures were obtained for the case where $a = 0$ and $b = 1$ except for Table 9. Figure 6 illustrates a 3D plot of numerical and analytical solutions with temporal step size $\Delta t = 0.001$, while Figure 7 displays the 3D plot of absolute error. In Figure 8, the approximate and exact solutions are plotted at multiple time levels with a time step of $\Delta t = 0.01$. Figure 9 shows the spatial distribution of the absolute error at various time levels for $\Delta t = 0.01$. Table 6 reports the absolute error and Table 7

presents the L_2 and L_∞ error norms, comparing them with the numerical scheme reported in [8]. The L_2 and L_∞ error norms are calculated in Table 8 for $1 < \lambda < 2$ at different time levels. In Table 9, the maximum absolute error (L_∞) is compared with the spectral tau scheme based on the Jacobi operational matrix presented in [29], for the case where $a = 0$ and $b = 2$. The absolute errors at different points $(x_j, t^n) \in [0, 1] \times [0, 1]$ for $1 < \lambda < 2$ with $\Delta t = 0.01$ and $M = 40$ are tabulated in Table 10.

Table 6. Comparison of absolute errors for Example 5.2 at $t = 0.1$ with $\Delta t = 0.0001$, $M = 10$, and $\lambda = 1.9$.

	Present method			Method [8]
x	Exact result	Approximate result	Absolute error	Absolute error
0.1	0.000900000000000	0.000900000000906	9.0648×10^{-13}	1.01445×10^{-7}
0.2	0.001600000000000	0.001600000005360	5.36001×10^{-12}	1.87058×10^{-7}
0.3	0.002100000000000	0.002100000007592	7.59235×10^{-12}	2.49576×10^{-7}
0.4	0.002400000000000	0.002400000009160	9.16001×10^{-12}	2.87195×10^{-7}
0.5	0.002500000000000	0.002500000009635	9.63489×10^{-12}	2.99728×10^{-7}
0.6	0.002400000000000	0.002400000009157	9.15666×10^{-12}	2.87195×10^{-7}
0.7	0.002100000000000	0.002100000007599	7.59867×10^{-12}	2.49576×10^{-7}
0.8	0.001600000000000	0.001600000005364	5.36411×10^{-12}	1.87058×10^{-7}
0.9	0.000900000000000	0.000900000000912	9.1182×10^{-13}	1.01445×10^{-7}

Table 7. Comparison of error norms at time $t = 1$, $M = 10$, and $\lambda = 1.3$ for Example 5.2.

	Present method			Method [8]	
Δt	L_∞ Error	L_2 Error	$p_{\Delta t}$	L_∞ Error	L_2 Error
$\frac{1}{10}$	3.54840×10^{-6}	2.57227×10^{-6}	—	2.53389×10^{-3}	1.79828×10^{-3}
$\frac{1}{20}$	1.76043×10^{-6}	1.26836×10^{-6}	1.0112408	1.27178×10^{-3}	9.02525×10^{-4}
$\frac{1}{40}$	8.44685×10^{-7}	6.00879×10^{-7}	1.0594425	6.36685×10^{-4}	4.51809×10^{-4}
$\frac{1}{80}$	3.81215×10^{-7}	2.63993×10^{-7}	1.1478085	3.18486×10^{-4}	2.25999×10^{-4}
$\frac{1}{160}$	1.48061×10^{-7}	9.76673×10^{-8}	1.3644132	1.59272×10^{-4}	1.13018×10^{-4}
$\frac{1}{320}$	4.61699×10^{-8}	2.36153×10^{-8}	1.6811672	—	—

Table 8. Error norms at different time levels and values of λ for Example 5.2 with $\Delta t = 0.001$ and $M = 10$.

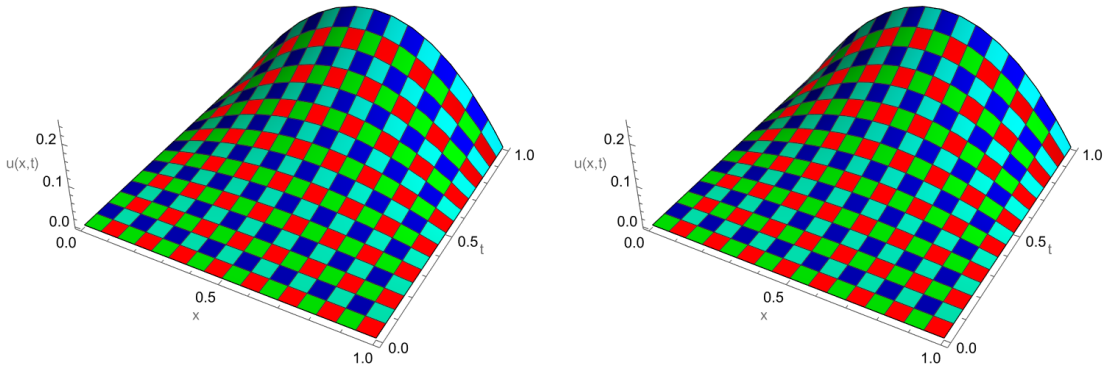
T	L_2	L_∞	L_2	L_∞
	$\lambda = 1.3$		$\lambda = 1.5$	
0.2	2.17439×10^{-9}	3.09837×10^{-9}	9.61493×10^{-10}	1.36443×10^{-9}
0.4	3.12469×10^{-9}	4.71338×10^{-9}	1.63317×10^{-9}	2.48565×10^{-9}
0.6	6.86286×10^{-9}	1.44831×10^{-8}	2.67518×10^{-9}	5.69645×10^{-9}
0.8	2.46895×10^{-8}	4.25772×10^{-8}	9.67498×10^{-9}	1.78209×10^{-8}
1.0	6.14106×10^{-8}	9.46834×10^{-8}	2.55783×10^{-8}	4.12062×10^{-8}
	$\lambda = 1.7$		$\lambda = 1.9$	
0.2	4.99349×10^{-10}	7.04312×10^{-10}	2.73400×10^{-10}	3.81813×10^{-10}
0.4	1.05395×10^{-9}	1.58821×10^{-9}	7.63556×10^{-10}	1.12025×10^{-9}
0.6	1.05395×10^{-9}	1.58821×10^{-9}	1.06573×10^{-9}	1.59826×10^{-9}
0.8	4.20294×10^{-9}	8.46752×10^{-9}	1.85195×10^{-9}	3.95986×10^{-9}
1.0	1.21428×10^{-8}	2.07993×10^{-8}	5.76734×10^{-9}	1.08522×10^{-8}

Table 9. Comparison of maximum absolute errors for Example 5.2 at $t = 1$ and $b = 2$ at various values of λ.

	Present method			Method [29]		
	$\Delta t = 0.002$			$M = N$ and $\alpha = \beta = 0$		
M	$\lambda = 1.1$	$\lambda = 1.5$	$\lambda = 1.9$	$\lambda = 1.1$	$\lambda = 1.5$	$\lambda = 1.9$
3	5.96×10^{-7}	6.39×10^{-8}	1.77×10^{-8}	1.74×10^{-4}	6.17×10^{-4}	6.44×10^{-5}
5	2.16×10^{-7}	2.27×10^{-8}	6.05×10^{-9}	3.76×10^{-5}	1.44×10^{-4}	8.57×10^{-6}
7	1.05×10^{-7}	1.08×10^{-8}	3.89×10^{-9}	1.30×10^{-5}	5.84×10^{-5}	1.92×10^{-6}
9	6.00×10^{-8}	6.02×10^{-9}	2.03×10^{-9}	5.73×10^{-6}	2.76×10^{-5}	2.96×10^{-7}
11	3.78×10^{-8}	3.68×10^{-9}	1.07×10^{-9}	2.90×10^{-6}	1.51×10^{-5}	2.29×10^{-7}
13	2.54×10^{-8}	2.38×10^{-9}	8.97×10^{-10}	1.63×10^{-6}	9.03×10^{-6}	2.73×10^{-7}
15	1.77×10^{-8}	1.87×10^{-9}	7.55×10^{-10}	9.94×10^{-7}	5.77×10^{-6}	2.66×10^{-7}
17	1.09×10^{-8}	1.46×10^{-9}	6.03×10^{-10}	6.26×10^{-7}	3.88×10^{-6}	2.26×10^{-7}
19	9.60×10^{-9}	1.07×10^{-9}	4.78×10^{-10}	4.30×10^{-7}	2.72×10^{-6}	1.91×10^{-7}
21	8.48×10^{-9}	9.00×10^{-10}	3.98×10^{-10}	3.00×10^{-7}	1.97×10^{-6}	1.59×10^{-7}
23	7.17×10^{-9}	8.67×10^{-10}	2.56×10^{-10}	2.16×10^{-7}	1.46×10^{-6}	1.32×10^{-7}

Table 10. Absolute errors at points (x_j, t^n) for $1 < \lambda < 2$ with $\Delta t = 0.01$ and $M = 40$ for Example 5.2.

(x,t)	$\lambda = 1.1$	$\lambda = 1.3$	$\lambda = 1.5$	$\lambda = 1.7$	$\lambda = 1.9$
$(0.1, 0.1)$	2.412004×10^{-8}	5.583236×10^{-9}	2.265765×10^{-9}	1.025996×10^{-9}	4.344524×10^{-10}
$(0.2, 0.2)$	9.506514×10^{-8}	2.296804×10^{-8}	9.930419×10^{-9}	4.989049×10^{-9}	2.589111×10^{-9}
$(0.3, 0.3)$	2.054161×10^{-7}	5.134001×10^{-8}	2.323453×10^{-8}	1.245922×10^{-8}	7.151665×10^{-9}
$(0.4, 0.4)$	3.406909×10^{-7}	8.752845×10^{-8}	4.100478×10^{-8}	2.299664×10^{-8}	1.402434×10^{-8}
$(0.5, 0.5)$	4.793448×10^{-7}	1.260059×10^{-7}	6.065326×10^{-8}	3.515079×10^{-8}	2.232315×10^{-8}
$(0.6, 0.6)$	5.927721×10^{-7}	1.588893×10^{-7}	7.817890×10^{-8}	4.646190×10^{-8}	3.037928×10^{-8}
$(0.7, 0.7)$	6.453061×10^{-7}	1.759391×10^{-7}	8.816611×10^{-8}	5.345963×10^{-8}	3.573776×10^{-8}
$(0.8, 0.8)$	5.942208×10^{-7}	1.645607×10^{-7}	8.378630×10^{-8}	5.166452×10^{-8}	3.515896×10^{-8}
$(0.9, 0.9)$	3.897312×10^{-7}	1.098002×10^{-7}	5.679715×10^{-8}	3.558685×10^{-8}	2.461641×10^{-8}



(a) Space-time graph of approximate solution at $t = 1$.

(b) Space-time graph of exact solution at $t = 1$.

Figure 6. 3D approximate (a) and exact (b) solutions for Example 5.2 with $\lambda = 1.5$ and $M = 200$.

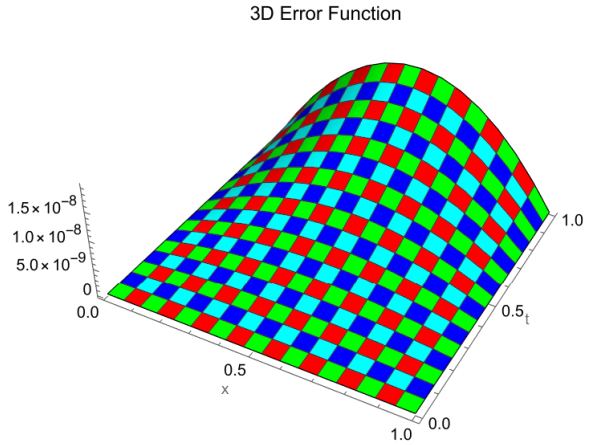


Figure 7. 3D error distribution for Example 5.2 with $M = 200$, $\Delta t = 0.001$, and $\lambda = 1.5$ at time $t = 1$.

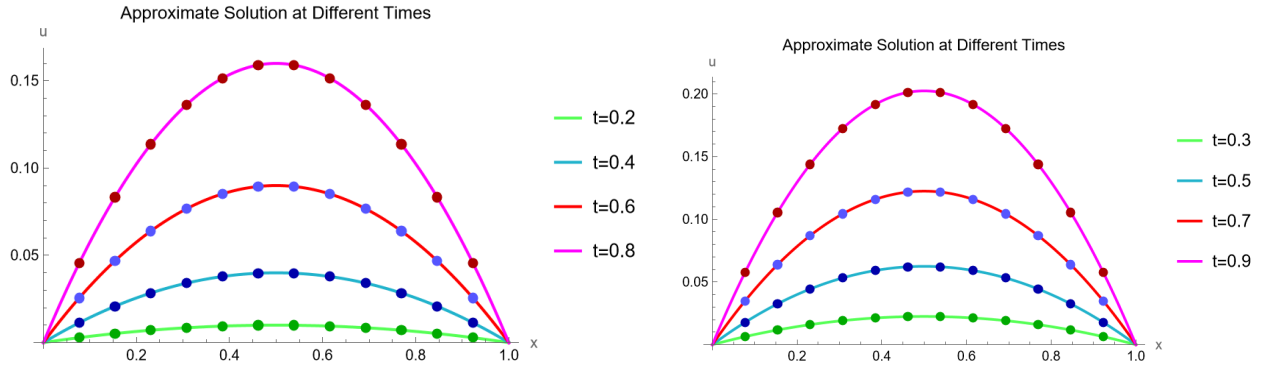


Figure 8. Comparison of numerical and exact solutions at different times for Example 5.2 with parameters $M = 200$ and $\lambda = 1.7$.

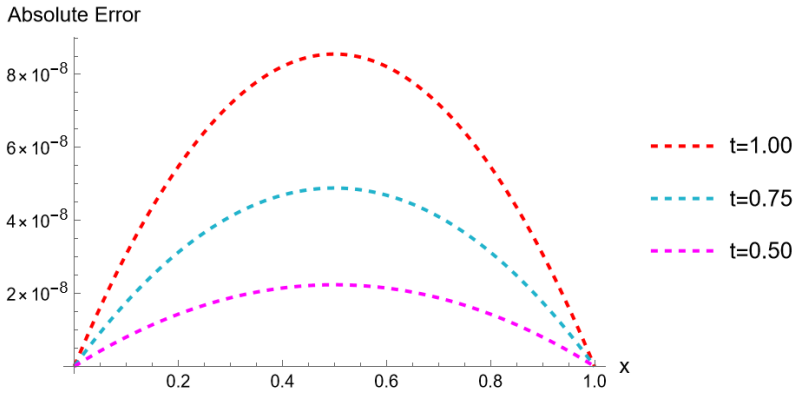


Figure 9. 2D absolute error plot at different times when $\lambda = 1.9$ and $M = 250$ for Example 5.2.

Equations (5.3) and (5.4) represent the piecewise numerical solution of the problem obtained using $\Delta t = 0.01$, $M = 10$, and $\lambda = 1.5$.

Solution at time $t = 1$:

$$U(x,1) = \begin{cases} 2.00000 + 0.921039x - 2.00000\cosh(x) + 0.07891\sinh(x), & 0 \le x \le \frac{1}{10} \\ 2.02265 + 0.694502x - 2.02269\cosh(x) + 0.30658\sinh(x), & \frac{1}{10} \le x \le \frac{2}{10} \\ 2.06127 + 0.501439x - 2.06156\cosh(x) + 0.50352\sinh(x), & \frac{2}{10} \le x \le \frac{3}{10} \\ 2.12181 + 0.299618x - 2.12302\cosh(x) + 0.71449\sinh(x), & \frac{3}{10} \le x \le \frac{4}{10} \\ 2.20161 + 0.100124x - 2.20496\cosh(x) + 0.93016\sinh(x), & \frac{4}{10} \le x \le \frac{5}{10} \\ 2.30173 - 0.100124x - 2.30931\cosh(x) + 1.15596\sinh(x), & \frac{5}{10} \le x \le \frac{6}{10} \\ 2.42143 - 0.299618x - 2.43632\cosh(x) + 1.39246\sinh(x), & \frac{6}{10} \le x \le \frac{7}{10} \\ 2.56271 - 0.501439x - 2.58942\cosh(x) + 1.64578\sinh(x), & \frac{7}{10} \le x \le \frac{8}{10} \\ 2.71716 - 0.694502x - 2.76088\cosh(x) + 1.90399\sinh(x), & \frac{8}{10} \le x \le \frac{9}{10} \\ 2.92104 - 0.921039x - 2.99342\cosh(x) + 2.22863\sinh(x), & \frac{9}{10} \le x \le 1 \end{cases} \tag{5.3}$$

Solution at time $t = 0.5$:

$$U(x,0.5) = \begin{cases} 0.500000 + 0.230277x - 0.500000\cosh(x) + 0.019711\sinh(x), & 0 \le x \le \frac{1}{10} \\ 0.505668 + 0.173601x - 0.505677\cosh(x) + 0.076671\sinh(x), & \frac{1}{10} \le x \le \frac{2}{10} \\ 0.515313 + 0.125372x - 0.515387\cosh(x) + 0.125868\sinh(x), & \frac{2}{10} \le x \le \frac{3}{10} \\ 0.530455 + 0.074899x - 0.530757\cosh(x) + 0.178628\sinh(x), & \frac{3}{10} \le x \le \frac{4}{10} \\ 0.550402 + 0.025033x - 0.551240\cosh(x) + 0.232538\sinh(x), & \frac{4}{10} \le x \le \frac{5}{10} \\ 0.575435 - 0.025033x - 0.577329\cosh(x) + 0.288993\sinh(x), & \frac{5}{10} \le x \le \frac{6}{10} \\ 0.605355 - 0.074899x - 0.609077\cosh(x) + 0.348109\sinh(x), & \frac{6}{10} \le x \le \frac{7}{10} \\ 0.640685 - 0.125372x - 0.647364\cosh(x) + 0.411460\sinh(x), & \frac{7}{10} \le x \le \frac{8}{10} \\ 0.679268 - 0.173601x - 0.690196\cosh(x) + 0.475962\sinh(x), & \frac{8}{10} \le x \le \frac{9}{10} \\ 0.730277 - 0.230277x - 0.748376\cosh(x) + 0.557185\sinh(x), & \frac{9}{10} \le x \le 1 \end{cases}. \tag{5.4}$$

Example 5.3. Consider the TFDW Eq (1.1) with damping and reaction coefficients $\mu_1 = 0$ and $\mu_2 = 0$:

$$\frac{\partial^\lambda u(x,t)}{\partial t^\lambda} - \frac{\partial^2 u(x,t)}{\partial x^2} = f(x,t), \quad a \le x \le b, \quad t \ge 0.$$

The initial and boundary conditions are given by:

$$\begin{aligned} &\text{Initial Conditions (ICs)}: \quad \phi_0(x) = 0, \quad \phi_1(x) = 0, \\ &\text{Boundary Conditions (BCs)}: \quad \psi_0(t) = 0, \quad \psi_1(t) = 0. \end{aligned}$$

The forcing term $f(x,t)$ is defined as:

$$f(x,t) = 2x(x-1)\left(\frac{M(\lambda)}{\lambda}\right)\left(1 - \exp\left[\frac{-\lambda t}{2-\lambda}\right]\right) - 2t^2.$$

The analytical solution of the problem is $u(x,t) = x(x-1)t^2$.

We employed the proposed scheme (2.12) to compute the approximate solution of the problem. All numerical results and figures were obtained for the case where $a = 0$ and $b = 1$. A detailed comparison between the analytical and approximate solutions is presented. Figure 10 provides a 3D illustration of the exact and approximate solutions with parameters $M = 180$ and $\lambda = 1.5$, while Figure 11 shows the 3D error distribution between them. Figure 12 compares the approximate and exact solutions at several time levels. Figure 13 presents a 2D plot of the absolute error for $\Delta t = 0.01$ at various time levels. The L_∞ error norms and corresponding convergence orders are presented in Table 11 and compared with the scheme reported in [10]. Table 12 presents the absolute errors for various spatial step sizes, highlighting the accuracy of the proposed method. The L_2 and L_∞ error norms at times $t = 1$ and $t = 0.5$ are reported in Table 13. The obtained convergence order in Table 14 closely matches the theoretically calculated value. Table 15 presents the absolute error at various points $(x_j, t^n) \in [0,1] \times [0,1]$ for $1 < \lambda < 2$.

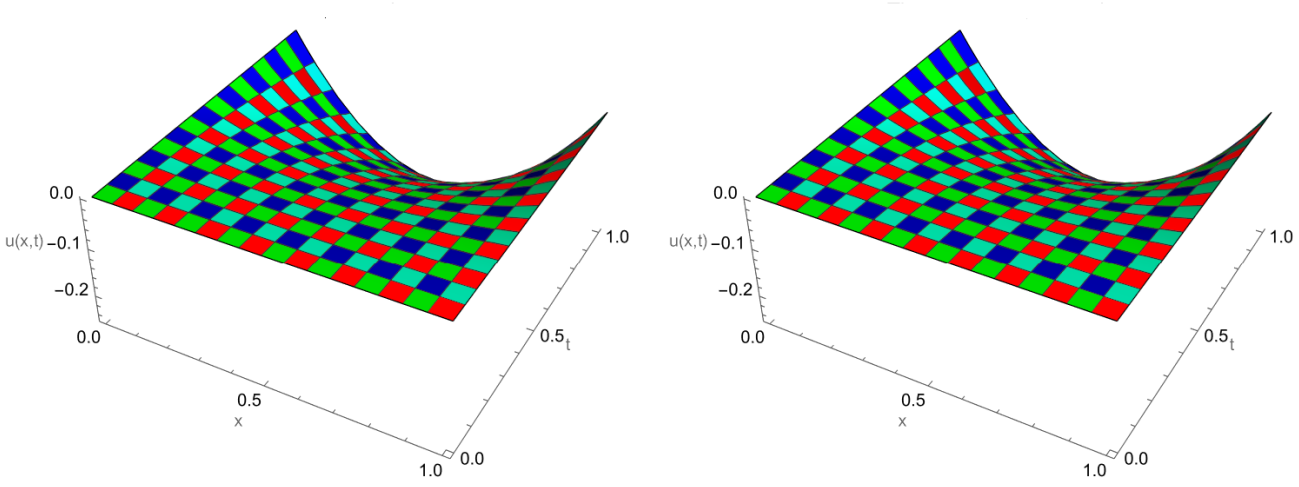
Equations (5.5) and (5.6) represent the piecewise numerical solution of the problem obtained using $\Delta t = 0.01$, $M = 10$, and $\lambda = 1.5$.

Solution at time $t = 1$:

$$U(x,1) = \begin{cases} -2.00000 - 0.909096x + 2.00000\cosh(x) - 0.09071\sinh(x), & 0 \le x \le \frac{1}{10} \\ -2.02044 - 0.704735x + 2.02047\cosh(x) - 0.29610\sinh(x), & \frac{1}{10} \le x \le \frac{2}{10} \\ -2.06099 - 0.501946x + 2.06130\cosh(x) - 0.50295\sinh(x), & \frac{2}{10} \le x \le \frac{3}{10} \\ -2.12140 - 0.300590x + 2.12262\cosh(x) - 0.71344\sinh(x), & \frac{3}{10} \le x \le \frac{4}{10} \\ -2.20159 - 0.100106x + 2.20497\cosh(x) - 0.93018\sinh(x), & \frac{4}{10} \le x \le \frac{5}{10} \\ -2.30170 + 0.100106x + 2.30929\cosh(x) - 1.15594\sinh(x), & \frac{5}{10} \le x \le \frac{6}{10} \\ -2.42199 + 0.300590x + 2.43693\cosh(x) - 1.39361\sinh(x), & \frac{6}{10} \le x \le \frac{7}{10} \\ -2.56294 + 0.501946x + 2.58968\cosh(x) - 1.64634\sinh(x), & \frac{7}{10} \le x \le \frac{8}{10} \\ -2.72517 + 0.704735x + 2.76978\cosh(x) - 1.91756\sinh(x), & \frac{8}{10} \le x \le \frac{9}{10} \\ -2.90910 + 0.909096x + 2.97956\cosh(x) - 2.21043\sinh(x), & \frac{9}{10} \le x \le 1 \end{cases}. \tag{5.5}$$

Solution at time $t = 0.5$:

$$U(x,0.5) = \begin{cases} -0.500000 - 0.228195x + 0.500000\cosh(x) - 0.021774\sinh(x), & 0 \le x \le \frac{1}{10} \\ -0.505234 - 0.175856x + 0.505243\cosh(x) - 0.074375\sinh(x), & \frac{1}{10} \le x \le \frac{2}{10} \\ -0.515384 - 0.125107x + 0.515460\cosh(x) - 0.126143\sinh(x), & \frac{2}{10} \le x \le \frac{3}{10} \\ -0.530418 - 0.074993x + 0.530721\cosh(x) - 0.178528\sinh(x), & \frac{3}{10} \le x \le \frac{4}{10} \\ -0.550416 - 0.024997x + 0.551257\cosh(x) - 0.232577\sinh(x), & \frac{4}{10} \le x \le \frac{5}{10} \\ -0.575413 + 0.024997x + 0.577309\cosh(x) - 0.288952\sinh(x), & \frac{5}{10} \le x \le \frac{6}{10} \\ -0.605411 + 0.074993x + 0.609139\cosh(x) - 0.348221\sinh(x), & \frac{6}{10} \le x \le \frac{7}{10} \\ -0.640490 + 0.125107x + 0.647154\cosh(x) - 0.411121\sinh(x), & \frac{7}{10} \le x \le \frac{8}{10} \\ -0.681090 + 0.175856x + 0.692224\cosh(x) - 0.478995\sinh(x), & \frac{8}{10} \le x \le \frac{9}{10} \\ -0.728195 + 0.228195x + 0.745952\cosh(x) - 0.554002\sinh(x), & \frac{9}{10} \le x \le 1 \end{cases}. \tag{5.6}$$



(a) Space-time graph of approximate solution at $t = 1$.

(b) Space-time graph of exact solution at $t = 1$.

Figure 10. 3D comparison between numerical (a) and exact (b) solutions for Example 5.3 at $t = 1$ and temporal step size $\Delta t = 0.001$.

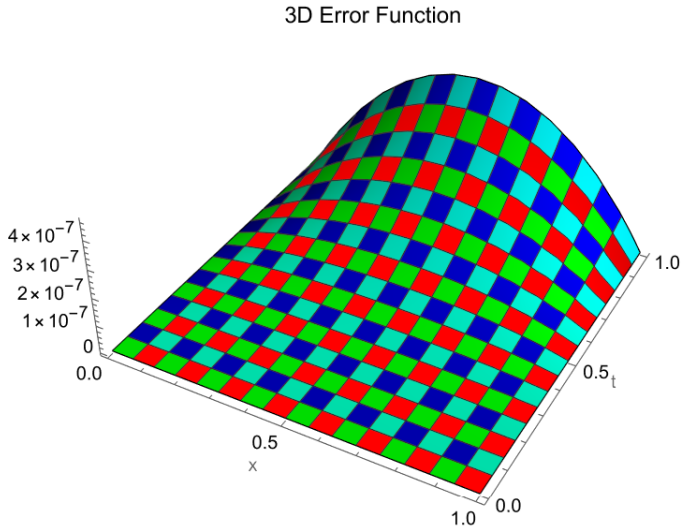


Figure 11. 3D absolute error distribution for Example 5.3 at time $t = 1$ and temporal step size $\Delta t = 0.001$.

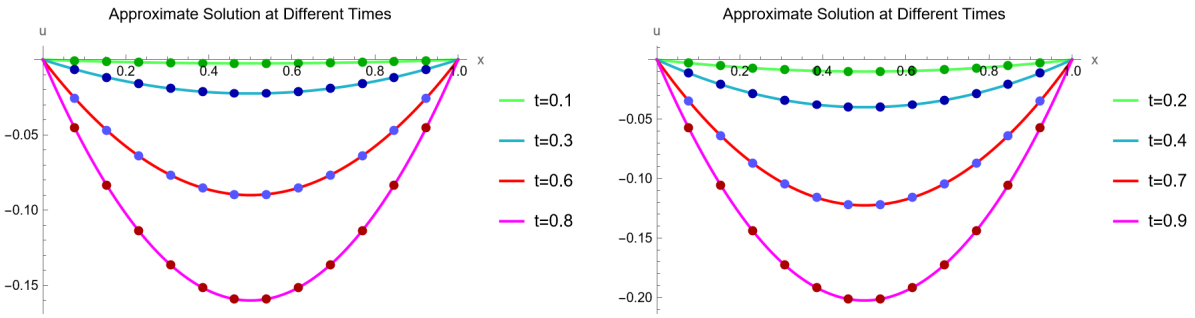


Figure 12. Comparison between the numerical solution (solid lines) and exact solution (dots) at various time levels for Example 5.3, with parameters $\Delta t = 0.01$, $M = 180$, and $\lambda = 1.5$.

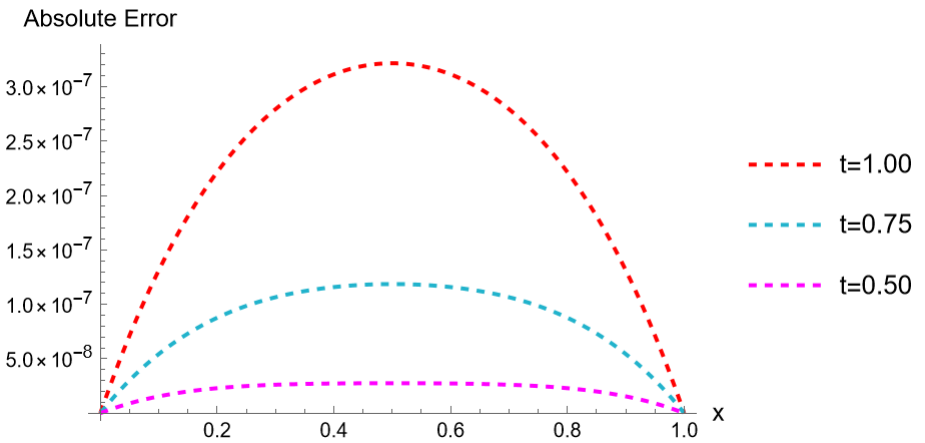


Figure 13. 2D absolute error plots at various time levels with parameters $\lambda = 1.7$ and $M = 180$ for Example 5.3.

Table 11. Comparison of error norms at time $t = 1$ and $\Delta t = 0.002$ for Example 5.3.

	Present method		Method [10]	
M	L_∞	p_h	L_∞	p_h
		$\lambda = 1.15$		
4	4.96614×10^{-4}	——	5.0666×10^{-3}	——
8	1.20943×10^{-4}	2.0377977	1.8240×10^{-3}	1.4739
16	3.00497×10^{-5}	2.0089048	9.1195×10^{-4}	1.0000
32	7.50100×10^{-6}	2.0021957		1.0000
		$\lambda = 1.25$		
4	3.06721×10^{-4}	——	5.0393×10^{-3}	——
8	7.44019×10^{-5}	2.0435156	1.8124×10^{-3}	1.4753
16	1.84787×10^{-5}	2.0094762	9.0865×10^{-4}	0.9961
32	4.61235×10^{-6}	2.0022894		1.0000
		$\lambda = 1.5$		
4	1.17658×10^{-4}	——	4.9545×10^{-3}	——
8	2.79915×10^{-5}	2.0715387	1.7783×10^{-3}	1.4782
16	6.98823×10^{-6}	2.0019898	8.9931×10^{-4}	0.9836
32	1.74637×10^{-6}	2.0005678		1.0000

Table 12. Absolute errors at $t = 0.1$ with parameters $\Delta t = 0.0001$ and $\lambda = 1.5$ for Example 5.3.

	Absolute error			
x	$M = 10$	$M = 20$	$M = 40$	$M = 80$
0.1	1.26811×10^{-8}	2.76525×10^{-9}	6.82990×10^{-10}	1.70320×10^{-10}
0.2	1.07304×10^{-8}	2.74317×10^{-9}	6.85937×10^{-10}	1.71476×10^{-10}
0.3	1.10023×10^{-8}	2.74365×10^{-9}	6.85944×10^{-10}	1.71499×10^{-10}
0.4	1.09711×10^{-8}	2.74364×10^{-9}	6.85922×10^{-10}	1.71434×10^{-10}
0.5	1.09728×10^{-8}	2.74363×10^{-9}	6.85938×10^{-10}	1.71427×10^{-10}
0.6	1.09712×10^{-8}	2.74364×10^{-9}	6.85939×10^{-10}	1.71566×10^{-10}
0.7	1.10023×10^{-8}	2.74364×10^{-9}	6.85933×10^{-10}	1.71557×10^{-10}
0.8	1.07304×10^{-8}	2.74318×10^{-9}	6.85926×10^{-10}	1.71443×10^{-10}
0.9	1.26811×10^{-8}	2.76526×10^{-9}	6.82983×10^{-10}	1.70304×10^{-10}

Table 13. Error norms with varying spatial and temporal step sizes for Example 5.3 when $\lambda = 1.7$.

		$t = 1$		$t = 0.5$	
Δt	M	L_∞ Error	L_2 Error	L_∞ Error	L_2 Error
0.05	10	1.03539×10^{-5}	9.40793×10^{-6}	9.90496×10^{-7}	9.30867×10^{-7}
0.01	20	2.40364×10^{-6}	2.18683×10^{-6}	2.15852×10^{-7}	2.04223×10^{-7}
0.005	40	5.95549×10^{-7}	5.41114×10^{-7}	5.30478×10^{-8}	5.00979×10^{-8}
0.001	60	2.62798×10^{-7}	2.38832×10^{-7}	2.32589×10^{-8}	2.19674×10^{-8}
0.0005	80	1.47692×10^{-7}	1.34214×10^{-7}	1.3061×10^{-8}	1.2334×10^{-8}

Table 14. Temporal convergence order for Example 5.3 at time $t = 1$ and $M = 20$.

	$\lambda = 1.5$			$\lambda = 1.7$		
Δt	L_∞ Error	L_2 Error	$p_{\Delta t}$	L_∞ Error	L_2 Error	$p_{\Delta t}$
$\frac{1}{10}$	5.23015×10^{-6}	4.45331×10^{-6}	$----$	2.82869×10^{-6}	2.53429×10^{-6}	$----$
$\frac{1}{20}$	2.30830×10^{-6}	2.09033×10^{-6}	1.18002	1.45048×10^{-6}	1.34919×10^{-6}	0.96360
$\frac{1}{40}$	9.37512×10^{-7}	8.84772×10^{-7}	1.29992	6.78149×10^{-7}	6.46550×10^{-7}	1.09686
$\frac{1}{80}$	3.62497×10^{-7}	3.47642×10^{-7}	1.37087	2.96695×10^{-7}	2.85003×10^{-7}	1.19262
$\frac{1}{160}$	1.31935×10^{-7}	1.16901×10^{-7}	1.45814	1.22631×10^{-7}	1.06887×10^{-7}	1.27466

Table 15. Absolute errors at points (x_j, t^n) for $1 < \lambda < 2$ with $\Delta t = 0.01$ and $M = 40$ for Example 5.3.

(x, t)	$\lambda = 1.1$	$\lambda = 1.3$	$\lambda = 1.5$	$\lambda = 1.7$	$\lambda = 1.9$
(0.1, 0.1)	5.43140×10^{-9}	1.83562×10^{-9}	7.75716×10^{-10}	3.20655×10^{-10}	9.82815×10^{-11}
(0.2, 0.2)	4.97213×10^{-8}	1.52845×10^{-8}	6.30480×10^{-9}	2.70767×10^{-9}	9.70150×10^{-10}
(0.3, 0.3)	1.79954×10^{-7}	5.32828×10^{-8}	2.21776×10^{-8}	9.91401×10^{-9}	3.96738×10^{-9}
(0.4, 0.4)	4.43404×10^{-7}	1.30461×10^{-7}	5.51385×10^{-8}	2.55228×10^{-8}	1.10920×10^{-8}
(0.5, 0.5)	8.71133×10^{-6}	2.63009×10^{-7}	1.13082×10^{-7}	5.39481×10^{-8}	2.49970×10^{-8}
(0.6, 0.6)	1.44395×10^{-6}	4.65414×10^{-7}	2.04986×10^{-7}	1.00434×10^{-7}	4.89851×10^{-8}
(0.7, 0.7)	2.04149×10^{-6}	7.32211×10^{-7}	3.38989×10^{-7}	1.71030×10^{-7}	8.70103×10^{-8}
(0.8, 0.8)	2.38162×10^{-6}	9.81287×10^{-7}	5.01218×10^{-7}	2.69851×10^{-7}	1.43681×10^{-7}
(0.9, 0.9)	1.96005×10^{-6}	9.41658×10^{-7}	5.54387×10^{-7}	3.44502×10^{-7}	2.11942×10^{-7}

6. Conclusions

In this research, we developed a novel numerical approach using uniform hyperbolic polynomial B-splines for solving the time-fractional diffusion-wave equation with damping and reaction terms. The proposed algorithm employs a fourth-order uniform hyperbolic B-spline discretization for spatial derivatives, combined with a θ-weighted scheme for temporal approximation. The time derivative is discretized using a finite difference approach and incorporates the Caputo-Fabrizio fractional derivative. Through a detailed Fourier stability analysis, we demonstrated that the proposed scheme is unconditionally stable. Additionally, we verified that the discretized solutions converge to the exact solutions as the mesh parameters are refined, confirming numerical convergence. Numerical experiments were conducted to validate the efficiency and accuracy of the proposed scheme. The results demonstrate superior performance compared to existing techniques in the literature. While the results are encouraging, this study has some limitations. It is limited to one-dimensional problems; extending to higher dimensions would need significant changes. The method uses uniform grids, which may not capture sharp solution features. Well-adaptive meshes could help. Also, fixed time steps may be inefficient for long simulations, suggesting the need for adaptive time-stepping. Finally, the analysis assumes smooth solutions; cases with discontinuities need further study. These areas offer directions for future work.

Author contributions

Muhammad Umar Manzoor: Conceptualization, validation, investigation, writing–original draft preparation, visualization; Muhammad Yaseen: Conceptualization, software, formal analysis, investigation, writing–original draft preparation, data curation, visualization; Muath Awadalla: Methodology, validation, investigation, data curation, funding acquisition; Hajer Zaway: Methodology, formal analysis, data curation. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

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

Funding

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Grant No. KFU252656.]

Conflict of interest

The authors declare no conflicts of interest.

References

1. D. E. Koning, *Fractional calculus*, Bachelor's Thesis, University of Groninge, 2015.
2. I. Podlubny, Geometric and physical interpretation of fractional integration and fractional differentiation, *Fract. Calc. Appl. Anal.*, **5** (2002), 367–386.
3. Z. Odibat, S. Momani, Numerical methods for nonlinear partial differential equations of fractional order, *Appl. Math. Model.*, **32** (2008), 28–39. https://doi.org/10.1016/j.apm.2006.10.025
4. D. Sierociuk, A. Dzieliński, G. Sarwas, I. Petras, I. Podlubny, T. Skovranek, Modelling heat transfer in heterogeneous media using fractional calculus, *Phil. Trans. R. Soc. A.*, **371** (2013), 20120146. https://doi.org/10.1098/rsta.2012.0146
5. M. Yaseen, M. Abbas, An efficient cubic trigonometric B-spline collocation scheme for the time-fractional telegraph equation, *Appl. Math. J. Chin. Univ.*, **35** (2020), 359–378. https://doi.org/10.1007/s11766-020-3883-y
6. M. Li, X. Ding, Q. Xu, Non-polynomial spline method for the time-fractional nonlinear Schrödinger equation, *Adv. Differ. Equ.*, **2018** (2018), 318. https://doi.org/10.1186/s13662-018-1743-3
7. M. Yaseen, M. Abbas, B. Ahmad, Numerical simulation of the nonlinear generalized time-fractional Klein–Gordon equation using cubic trigonometric B-spline functions, *Math. Methods Appl. Sci.*, **44** (2021), 901–916. https://doi.org/10.1002/mma.6798
8. M. Shafiq, M. Abbas, H. Emadifar, A. S. M. Alzaidi, T. Nazir, F. A. Abdullah, Numerical investigation of the fractional diffusion wave equation with exponential kernel via cubic B-Spline approach, *PLoS ONE*, **18** (2023), e0295525. https://doi.org/10.1371/journal.pone.0295525
9. M. Yaseen, M. Abbas, T. Nazir, D. Baleanu, A finite difference scheme based on cubic trigonometric B-splines for a time fractional diffusion-wave equation, *Adv. Differ. Equ.*, **2017** (2017), 274. https://doi.org/10.1186/s13662-017-1330-z
10. M. Dehghan, M. Abbaszadeh, A. Mohebbi, Analysis of a meshless method for the time fractional diffusion-wave equation, *Numer. Algor.*, **73** (2016), 445–476. https://doi.org/10.1007/s11075-016-0103-1
11. T. H. Solomon, E. R. Weeks, H. L. Swinney, Observation of anomalous diffusion and Lévy flights in a two-dimensional rotating flow, *Phys. Rev. Lett.*, **71** (1993), 3975–3978. https://doi.org/10.1103/PhysRevLett.71.3975
12. Z. Avazzadeh, V. R. Hosseini, W. Chen, Radial basis functions and FDM for solving fractional diffusion-wave equation, *Iran. J. Sci. Technol. Trans. A Sci.*, **38** (2014), 205–212.
13. R. Metzler, J. Klafter, The random walk's guide to anomalous diffusion: A fractional dynamics approach, *Phys. Rep.*, **339** (2000), 1–77. https://doi.org/10.1016/S0370-1573(00)00070-3
14. A. V. Chechkin, R. Gorenflo, I. M. Sokolov, Fractional diffusion in inhomogeneous media, *J. Phys. A: Math. Gen.*, **38** (2005), L679–L684. https://doi.org/10.1088/0305-4470/38/42/L03
15. M. Shafiq, F. A. Abdullah, M. Abbas, A. S. Alzaidi, M. B. Riaz, Memory effect analysis using piecewise cubic B-spline of time fractional diffusion equation, *Fractals*, **30** (2022), 2240270. https://doi.org/10.1142/S0218348X22402708

16. G. Wang, Y. Tang, (L^2, H^1)-Random attractors for stochastic reaction-diffusion equation on unbounded domains, *Abstr. Appl. Anal.*, **2013** (2013), 279509. https://doi.org/10.1155/2013/279509

17. K. D. Dwivedi, S. Das, Rajeev, D. Baleanu, Numerical solution of highly non-linear fractional order reaction advection diffusion equation using the cubic B-spline collocation method, *Int. J. Nonlinear Sci. Numer. Simul.*, **23** (2022), 1157–1172. https://doi.org/10.1515/ijnsns-2020-0112

18. A. M. Hayat, M. Abbas, H. Emadifar, A. S. M. Alzaidi, T. Nazir, F. A. Abdullah, An efficient computational scheme for solving coupled time-fractional Schrödinger equation via cubic B-spline functions, *PLoS ONE*, **19** (2024), e0296909. https://doi.org/10.1371/journal.pone.0296909

19. A. Ali, T. Akram, A. Iqbal, P. Kumam, T. Sutthibutpong, A numerical approach for 2D time-fractional diffusion damped wave model, *AIMS Mathematics*, **8** (2023), 8249–8273. https://doi.org/10.3934/math.2023416

20. M. Caputo, M. Fabrizio, A new definition of fractional derivative without singular kernel, *Progr. Fract. Differ. Appl.*, **1** (2015), 73–85.

21. J. M. Cruz-Duarte, J. Rosales-Garcia, C. R. Correa-Cely, A. Garcia–Perez, J. G. Avina-Cervantes, A closed form expression for the Gaussian-based Caputo-Fabrizio fractional derivative for signal processing applications, *Commun. Nonlinear Sci. Numer. Simul.*, **61** (2018), 138–148. https://doi.org/10.1016/j.cnsns.2018.01.020

22. M. A. Dokuyucu, E. Celik, H. Bulut, H. M. Baskonus, Cancer treatment model with the Caputo-Fabrizio fractional derivative, *Eur. Phys. J. Plus*, **133** (2018), 92. https://doi.org/10.1140/epjp/i2018-11950-y

23. M. Mortezaee, M. Ghovatmand, A. Nazemi, An application of generalized fuzzy hyperbolic model for solving fractional optimal control problems with Caputo-Fabrizio derivative, *Neural Process. Lett.*, **52** (2020), 1997–2020. https://doi.org/10.1007/s11063-020-10334-4

24. E. G. M. Elmahdi, J. Huang, Two linearized finite difference schemes for time fractional nonlinear diffusion-wave equations with fourth order derivative, *AIMS Mathematics*, **6** (2021), 6356–6376. https://doi.org/10.3934/math.2021373

25. Y. Lü, G. Wang, X. Yang, Uniform hyperbolic polynomial B-spline curves, *Comput. Aided Geom. Des.*, **19** (2002), 379–393. https://doi.org/10.1016/S0167-8396(02)00092-4

26. M. S. Palav, V. H. Pradhan, Redefined fourth order uniform hyperbolic polynomial B-splines based collocation method for solving advection-diffusion equation, *Appl. Math. Comput.*, **484** (2025), 128992. https://doi.org/10.1016/j.amc.2024.128992

27. M. Palav, V. Pradhan, Efficient numerical solution of Burgers' equation using collocation method based on re-defined uniform hyperbolic polynomial B-splines, *J. Appl. Math. Comput.*, **71** (2025), 4429–4474. https://doi.org/10.1007/s12190-025-02390-7

28. F. A. Shah, Kamran, Z. A. Khan, F. Azmi, N. Mlaiki, A hybrid collocation method for the approximation of 2D time fractional diffusion-wave equation, *AIMS Mathematics*, **9** (2024), 27122–27149. https://doi.org/10.3934/math.20241319

29. A. H. Bhrawy, E. H. Doha, D. Baleanu, S. S. Ezz-Eldien, A spectral tau algorithm based on Jacobi operational matrix for numerical solution of time fractional diffusion-wave equations, *J. Comput. Phys.*, **293** (2014), 142–156. http://dx.doi.org/10.1016/j.jcp.2014.03.039

30. L. Wei, Y. Yang, Optimal order finite difference/local discontinuous Galerkin method for variable-order time-fractional diffusion equation, *J. Comput. Appl. Math.*, **383** (2021), 113129. https://doi.org/10.1016/j.cam.2020.113129

31. X. Zhang, L. Wei, J. Liu, Application of the LDG method using generalized alternating numerical flux to the fourth-order time-fractional sub-diffusion model, *Appl. Math. Lett.*, **168** (2025), 109580. https://doi.org/10.1016/j.aml.2025.109580

32. M. Zhang, Y. Liu, H. Li, High-order local discontinuous Galerkin method for a fractal mobile/immobile transport equation with the Caputo–Fabrizio fractional derivative, *Numer. Methods Partial Differ. Equ.*, **35** (2019), 1588–1612. https://doi.org/10.1002/num.22366

AIMS Press